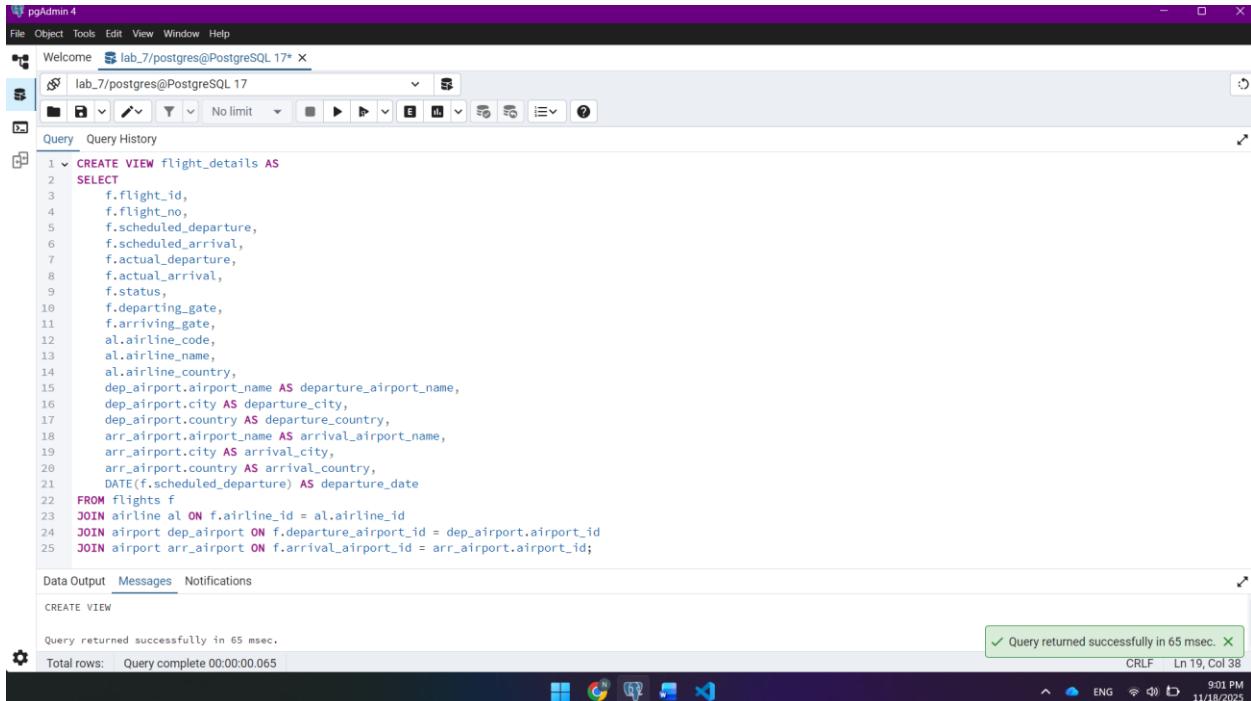


Lab-8

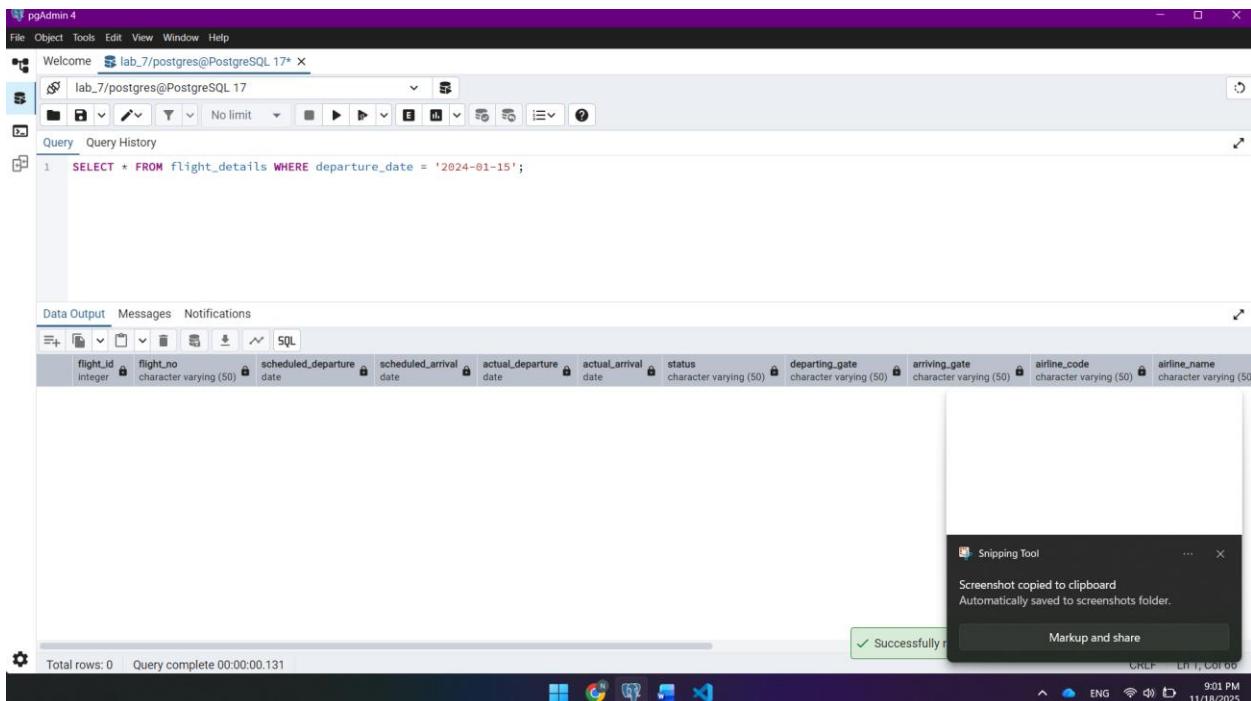
1



The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

```
1 ✓ CREATE VIEW flight_details AS
2   SELECT
3     f.flight_id,
4     f.flight_no,
5     f.scheduled_departure,
6     f.scheduled_arrival,
7     f.actual_departure,
8     f.actual_arrival,
9     f.status,
10    f.departing_gate,
11    f.arriving_gate,
12    al.airline_code,
13    al.airline_name,
14    al.airline_country,
15    dep_airport.airport_name AS departure_airport_name,
16    dep_airport.city AS departure_city,
17    dep_airport.country AS departure_country,
18    arr_airport.airport_name AS arrival_airport_name,
19    arr_airport.city AS arrival_city,
20    arr_airport.country AS arrival_country,
21    DATE(f.scheduled_departure) AS departure_date
22  FROM flights f
23  JOIN airline al ON f.airline_id = al.airline_id
24  JOIN airport dep_airport ON f.departure_airport_id = dep_airport.airport_id
25  JOIN airport arr_airport ON f.arrival_airport_id = arr_airport.airport_id;
```

The status bar at the bottom right indicates "Query returned successfully in 65 msec. CRLF Ln 19, Col 38".



The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

```
1 SELECT * FROM flight_details WHERE departure_date = '2024-01-15';
```

The status bar at the bottom right indicates "Query returned successfully in 65 msec. CRLF Ln 19, Col 38".

A Snipping Tool window is visible in the foreground, stating "Screenshot copied to clipboard" and "Automatically saved to screenshots folder".

CREATE VIEW flight_details AS

SELECT

f.flight_id,
f.flight_no,
f.scheduled_departure,
f.scheduled_arrival,
f.actual_departure,
f.actual_arrival,
f.status,
f.departing_gate,
f.arriving_gate,
al.airline_code,
al.airline_name,
al.airline_country,
dep_airport.airport_name AS
departure_airport_name,
dep_airport.city AS departure_city,
dep_airport.country AS departure_country,
arr_airport.airport_name AS
arrival_airport_name,

```
arr_airport.city AS arrival_city,
arr_airport.country AS arrival_country,
DATE(f.scheduled_departure) AS
departure_date
FROM flights f
JOIN airline al ON f.airline_id = al.airline_id
JOIN airport dep_airport ON
f.departure_airport_id = dep_airport.airport_id
JOIN airport arr_airport ON f.arrival_airport_id =
arr_airport.airport_id;
```

```
SELECT * FROM flight_details WHERE
departure_date = '2024-01-15';
```

2

```

pgAdmin 4
File Object Tools Edit View Window Help
Welcome lab_7/postgres@PostgreSQL 17* x
lab_7/postgres@PostgreSQL 17* PSQL Tool Workspace
Query Query History
41
42 FROM booking b
43 JOIN passengers p ON b.passenger_id = p.passenger_id
44 JOIN booking_flight bf ON b.booking_id = bf.booking_id
45 JOIN flights f ON bf.flight_id = f.flight_id
46 JOIN airline al ON f.airline_id = al.airline_id
47 JOIN airport dep_airport ON f.departure_airport_id = dep_airport.airport_id
48 JOIN airport arr_airport ON f.arrival_airport_id = arr_airport.airport_id
49 LEFT JOIN boarding_pass bp ON b.booking_id = bp.booking_id
50 LEFT JOIN baggage bg ON b.booking_id = bg.booking_id
51 WHERE f.scheduled_departure >= CURRENT_DATE
52 AND f.scheduled_departure < CURRENT_DATE + INTERVAL '7 days'
53 AND b.status != 'cancelled'; -- Exclude cancelled bookings
Data Output Messages Notifications
CREATE VIEW
Query returned successfully in 32 msec.

Total rows: Query complete 00:00:00.032
CRLF Ln 53, Col 62

```



```

pgAdmin 4
File Object Tools Edit View Window Help
Welcome lab_7/postgres@PostgreSQL 17* x
lab_7/postgres@PostgreSQL 17* PSQL Tool Workspace
Query Query History
1 -- Get all upcoming week bookings
2 SELECT * FROM upcoming_week_bookings;
3
4 -- Get bookings for a specific passenger
5 v SELECT * FROM upcoming_week_bookings
6 WHERE first_name = 'John' AND last_name = 'Smith';
7
8 -- Get bookings by airline
9 v SELECT * FROM upcoming_week_bookings
10 WHERE airline_name = 'Delta Airlines'
11 ORDER BY scheduled_departure;
Data Output Messages Notifications
booking_id integer
booking_platform character varying(50)
booking_status character varying(50)
price numeric(7,2)
booking_date date
passenger_id integer
first_name character varying(50)
last_name character varying(50)
date_of_birth date
gender character varying(50)
passport_number character varying(50)
flight_id integer

```



```

Total rows: 0 Query complete 00:00:00.046
CRLF Ln 11, Col 30

```

CREATE VIEW upcoming_week_bookings AS

SELECT

b.booking_id,

b.booking_platform,

b.status AS booking_status,
b.price,
b.created_at AS booking_date,

p.passenger_id,
p.first_name,
p.last_name,
p.date_of_birth,
p.gender,
p.passport_number,

f.flight_id,
f.flight_no,
f.scheduled_departure,
f.scheduled_arrival,
f.actual_departure,
f.actual_arrival,
f.status AS flight_status,
f.departing_gate,

f.arriving_gate,

al.airline_code,

al.airline_name,

**dep_airport.airport_name AS
departure_airport_name,**

dep_airport.city AS departure_city,

dep_airport.country AS departure_country,

**arr_airport.airport_name AS
arrival_airport_name,**

arr_airport.city AS arrival_city,

arr_airport.country AS arrival_country,

bp.seat,

bp.boarding_time,

bg.weight_in_kg AS baggage_weight

FROM booking b

**JOIN passengers p ON b.passenger_id =
p.passenger_id**

**JOIN booking_flight bf ON b.booking_id =
bf.booking_id**

JOIN flights f ON bf.flight_id = f.flight_id

JOIN airline al ON f.airline_id = al.airline_id

**JOIN airport dep_airport ON
f.departure_airport_id = dep_airport.airport_id**

**JOIN airport arr_airport ON f.arrival_airport_id =
arr_airport.airport_id**

**LEFT JOIN boarding_pass bp ON b.booking_id =
bp.booking_id**

**LEFT JOIN baggage bg ON b.booking_id =
bg.booking_id**

**WHERE f.scheduled_departure >=
CURRENT_DATE**

**AND f.scheduled_departure < CURRENT_DATE
+ INTERVAL '7 days'**

**AND b.status != 'cancelled'; -- Exclude
cancelled bookings**

SELECT * FROM upcoming_week_bookings;

**SELECT * FROM upcoming_week_bookings
WHERE first_name = 'John' AND last_name =
'Smith';**

**SELECT * FROM upcoming_week_bookings
WHERE airline_name = 'Delta Airlines'
ORDER BY scheduled_departure;**

3

```
1 SELECT * FROM top_5_popular_routes;
2
3
4
5 v SELECT
6   rank,
7   departure_city || ' to ' || arrival_city AS route,
8   total_bookings,
9   average_booking_price
10  FROM top_5_popular_routes;
11
12 v SELECT * FROM top_5_popular_routes
13 WHERE departure_city = 'New York' AND arrival_city = 'Los Angeles';
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
```

```
1 v CREATE VIEW top_5_popular_routes AS
2 SELECT
3   ROW_NUMBER() OVER (ORDER BY COUNT(b.booking_id) DESC) AS rank,
4   COUNT(b.booking_id) AS total_bookings,
5   dep_airport.city AS departure_city,
6   dep_airport.airport_name AS departure_airport,
7   arr_airport.city AS arrival_city,
8   arr_airport.airport_name AS arrival_airport,
9   dep_airport.country AS departure_country,
10  arr_airport.country AS arrival_country,
11  al.airline_name,
12  COUNT(DISTINCT f.flight_id) AS number_of_flights,
13  AVG(b.price) AS average_booking_price
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
129
130
131
132
133
134
135
136
137
138
139
139
140
141
142
143
144
145
146
147
148
149
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
178
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
218
219
219
220
221
222
223
224
225
226
227
228
229
229
230
231
232
233
234
235
236
237
238
239
239
240
241
242
243
244
245
246
247
248
249
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
278
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
```

CREATE VIEW top_5_popular_routes AS
SELECT

```
ROW_NUMBER() OVER (ORDER BY
COUNT(b.booking_id) DESC) AS rank,
COUNT(b.booking_id) AS total_bookings,
dep_airport.city AS departure_city,
dep_airport.airport_name AS
departure_airport,
arr_airport.city AS arrival_city,
arr_airport.airport_name AS arrival_airport,
dep_airport.country AS departure_country,
arr_airport.country AS arrival_country,
al.airline_name,
COUNT(DISTINCT f.flight_id) AS
number_of_flights,
AVG(b.price) AS average_booking_price
FROM booking b
JOIN booking_flight bf ON b.booking_id =
bf.booking_id
JOIN flights f ON bf.flight_id = f.flight_id
JOIN airline al ON f.airline_id = al.airline_id
```

```
JOIN airport dep_airport ON
f.departure_airport_id = dep_airport.airport_id

JOIN airport arr_airport ON f.arrival_airport_id =
arr_airport.airport_id

WHERE b.status != 'cancelled' -- Exclude
cancelled bookings

GROUP BY

dep_airport.city,
dep_airport.airport_name,
arr_airport.city,
arr_airport.airport_name,
dep_airport.country,
arr_airport.country,
al.airline_name

ORDER BY total_bookings DESC

LIMIT 5;

SELECT * FROM top_5_popular_routes;
```

```
SELECT
    rank,
    departure_city || ' to ' || arrival_city AS route,
    total_bookings,
    average_booking_price
FROM top_5_popular_routes;
```

```
SELECT * FROM top_5_popular_routes
WHERE departure_city = 'New York' AND
arrival_city = 'Los Angeles';
```

4

The screenshot shows the pgAdmin 4 interface. In the top navigation bar, 'File', 'Object', 'Tools', 'Edit', 'View', 'Window', and 'Help' are listed. Below the bar, a 'Welcome' message and a connection status 'lab_7/postgres@PostgreSQL 17*' are displayed. The main area is a query editor with a toolbar above it containing icons for file operations, search, and help. The toolbar also includes dropdown menus for 'No limit', 'Data', 'Text', 'Binary', 'SQL', 'DDL', 'DML', 'Script', 'Copy', 'Paste', 'Find', 'Replace', 'Format', 'Reset', and 'Help'. The query history section shows a single entry:

```
1 ✓ CREATE VIEW airline_flights AS
2   SELECT
3     f.flight_id,
4     f.flight_no,
5     f.scheduled_departure,
6     f.scheduled_arrival,
7     f.actual_departure,
8     f.actual_arrival,
9     f.status,
10    f.departing_gate,
11    f.arriving_gate,
12    al.airline_name,
13    al.airline_code.
```

Below the query, 'Data Output', 'Messages', and 'Notifications' tabs are visible. A message at the bottom of the screen states 'Query returned successfully in 60 msec.' with a timestamp 'Query complete 00:00:00.060' and a note 'CRLF Ln 22, Col 30'.

CREATE VIEW airline_flights AS

SELECT

f.flight_id,

f.flight_no,

f.scheduled_departure,

f.scheduled_arrival,

f.actual_departure,

f.actual_arrival,

f.status,

f.departing_gate,

f.arriving_gate,

```
al.airline_name,
al.airline_code,
dep_airport.airport_name AS
departure_airport,
dep_airport.city AS departure_city,
arr_airport.airport_name AS arrival_airport,
arr_airport.city AS arrival_city

FROM flights f

JOIN airline al ON f.airline_id = al.airline_id
JOIN airport dep_airport ON
f.departure_airport_id = dep_airport.airport_id
JOIN airport arr_airport ON f.arrival_airport_id =
arr_airport.airport_id

WHERE al.airline_code = 'AA';
```

5

The screenshot shows the pgAdmin 4 interface with a purple header bar. The main window displays a SQL query in the 'Query' tab:

```
12     al.airline_name,
13     al.airline_code,
14     dep_airport.airport_name AS departure_airport,
15     dep_airport.city AS departure_city,
16     arr_airport.airport_name AS arrival_airport,
17     arr_airport.city AS arrival_city
18
19 FROM flights f
20 JOIN airline al ON f.airline_id = al.airline_id
21 JOIN airport dep_airport ON f.departure_airport_id = dep_airport.airport_id
22 JOIN airport arr_airport ON f.arrival_airport_id = arr_airport.airport_id
23 WHERE al.airline_code = 'AA'
24 AND f.scheduled_departure >= CURRENT_DATE
25 AND f.scheduled_departure < CURRENT_DATE + INTERVAL '7 days';
```

Below the query, the status bar shows "CREATE VIEW" and "Query returned successfully in 29 msec.". The system tray at the bottom right indicates the date and time as "9/7 PM".

CREATE OR REPLACE VIEW airline_flights AS

SELECT

f.flight_id,
f.flight_no,
f.scheduled_departure,
f.scheduled_arrival,
f.actual_departure,
f.actual_arrival,
f.status,
f.departing_gate,
f.arriving_gate,

```
al.airline_name,  
al.airline_code,  
dep_airport.airport_name AS  
departure_airport,  
dep_airport.city AS departure_city,  
arr_airport.airport_name AS arrival_airport,  
arr_airport.city AS arrival_city  
FROM flights f  
JOIN airline al ON f.airline_id = al.airline_id  
JOIN airport dep_airport ON  
f.departure_airport_id = dep_airport.airport_id  
JOIN airport arr_airport ON f.arrival_airport_id =  
arr_airport.airport_id  
WHERE al.airline_code = 'AA'  
AND f.scheduled_departure >=  
CURRENT_DATE  
AND f.scheduled_departure < CURRENT_DATE  
+ INTERVAL '7 days';
```

The screenshot shows the pgAdmin 4 interface with a purple header bar. The main window displays a SQL query in a query editor tab. The query creates a view named 'flights_delayed_over_24_hours' with various flight and airport details, including departure and arrival times and their differences. A message at the bottom indicates the query was executed successfully in 27 msec. The status bar at the bottom shows 'Total rows: 0' and 'Query complete 00:00:00.027'.

```
CREATE VIEW flights_delayed_over_24_hours AS
SELECT
    f.flight_id,
    f.flight_no,
    f.scheduled_departure,
    f.actual_departure,
    f.scheduled_arrival,
    f.actual_arrival,
    (f.actual_departure - f.scheduled_departure) AS departure_delay,
    al.airline_name,
    dep_airport.airport_name AS departure_airport,
    arr_airport.airport_name AS arrival_airport
FROM flights f
JOIN airline al ON f.airline_id = al.airline_id
JOIN airport dep_airport ON f.departure_airport_id = dep_airport.airport_id
JOIN airport arr_airport ON f.arrival_airport_id = arr_airport.airport_id
```

**CREATE VIEW flights_delayed_over_24_hours
AS**

SELECT

f.flight_id,

f.flight_no,

f.scheduled_departure,

f.actual_departure,

f.scheduled_arrival,

f.actual_arrival,

(f.actual_departure - f.scheduled_departure)

AS departure_delay,

```
al.airline_name,  
    dep_airport.airport_name AS  
departure_airport,  
    arr_airport.airport_name AS arrival_airport  
FROM flights f  
JOIN airline al ON f.airline_id = al.airline_id  
JOIN airport dep_airport ON  
f.departure_airport_id = dep_airport.airport_id  
JOIN airport arr_airport ON f.arrival_airport_id =  
arr_airport.airport_id  
WHERE f.actual_departure IS NOT NULL  
    AND f.actual_departure >  
f.scheduled_departure + INTERVAL '24 hours';
```

The screenshot shows the pgAdmin 4 interface. In the top navigation bar, 'File', 'Object', 'Tools', 'Edit', 'View', 'Window', and 'Help' are visible. The title bar says 'Welcome lab_7/postgres@PostgreSQL 17'. The main area has tabs for 'Query' and 'Query History'. A code editor window displays the following SQL query:

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

Below the code editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is selected, showing the message 'CREATE VIEW'. Below that, it says 'Query returned successfully in 31 msec.'.

In the bottom status bar, it shows 'Total rows: Query complete 00:00:00.031' and a green success icon with the message 'Query returned successfully in 31 msec.' followed by 'CRLF Ln 8, Col 47'. On the far right, there are system icons for battery, signal, and time (9:09 PM, 11/18/2025).

**CREATE VIEW leffler_thompson_passengers
AS
SELECT
 p.first_name,
 p.last_name,
 p.country_of_citizenship AS country_of_origin
FROM passengers p
JOIN booking b ON p.passenger_id =
 b.passenger_id
WHERE b.booking_platform = 'Leffler-
Thompson';**

8

The screenshot shows the pgAdmin 4 interface. The title bar says "pgAdmin 4". The main window has a toolbar at the top with various icons. Below the toolbar is a "Query History" section containing a single query. The query is:

```
1 v CREATE VIEW top_10_most_visited_countries AS
2   SELECT
3     a.country,
4     COUNT(b.booking_id) AS total_visitors
5   FROM booking b
6   JOIN passengers p ON b.passenger_id = p.passenger_id
7   JOIN booking_flight bf ON b.booking_id = bf.booking_id
8   JOIN flights f ON bf.flight_id = f.flight_id
9   JOIN airport a ON f.arrival_airport_id = a.airport_id
10  WHERE b.status != 'cancelled'
11  GROUP BY a.country
12  ORDER BY total_visitors DESC
13  LIMIT 10;
```

Below the query history is a "Data Output" tab which contains the results of the query:

```
CREATE VIEW
```

Query returned successfully in 30 msec.

At the bottom of the pgAdmin window, there is a status bar with the message "Query returned successfully in 30 msec. CRLF Ln 13, Col 10". To the right of the status bar, there is a system tray with icons for battery, signal, and date/time (9:09 PM, 11/18/2025).

**CREATE VIEW top_10_most_visited_countries
AS**

SELECT

a.country,

COUNT(b.booking_id) AS total_visitors

FROM booking b

**JOIN passengers p ON b.passenger_id =
p.passenger_id**

```
JOIN booking_flight bf ON b.booking_id =
bf.booking_id
JOIN flights f ON bf.flight_id = f.flight_id
JOIN airport a ON f.arrival_airport_id =
a.airport_id
WHERE b.status != 'cancelled'
GROUP BY a.country
ORDER BY total_visitors DESC
LIMIT 10;
```

The screenshot shows the pgAdmin 4 interface with a purple header bar. The main window displays a SQL query in a query editor tab titled 'lab_7/postgres@PostgreSQL 17*'. The query creates a view named 'top_10_most_visited_countries' with 16 numbered lines of code. Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. A message in the 'Messages' tab indicates that the query was executed successfully in 43 msec. At the bottom of the screen, a status bar shows 'Total rows: 0' and 'Query complete 00:00:00.043'. A green success message in the status bar says 'Query returned successfully in 43 msec.' with a timestamp 'CRLF - Ln 16, Col 10' and a date '9/10/2023'.

```
1 ✓ CREATE OR REPLACE VIEW top_10_most_visited_countries AS
2   SELECT
3     a.country,
4     COUNT(b.booking_id) AS total_visitors,
5     COUNT(DISTINCT p.passenger_id) AS unique_visitors,
6     AVG(b.price) AS average_booking_price,
7     STRING_AGG(DISTINCT a.city, ', ') AS popular_cities
8   FROM booking b
9   JOIN passengers p ON b.passenger_id = p.passenger_id
10  JOIN booking_flight bf ON b.booking_id = bf.booking_id
11  JOIN flights f ON bf.flight_id = f.flight_id
12  JOIN airport a ON f.arrival_airport_id = a.airport_id
13 WHERE b.status != 'cancelled'
14 GROUP BY a.country
15 ORDER BY total_visitors DESC
16 LIMIT 10;
```

**CREATE OR REPLACE VIEW
top_10_most_visited_countries AS
SELECT**

a.country,
COUNT(b.booking_id) AS total_visitors,
**COUNT(DISTINCT p.passenger_id) AS
unique_visitors,**
AVG(b.price) AS average_booking_price,
**STRING_AGG(DISTINCT a.city, ', ') AS
popular_cities**
FROM booking b

**JOIN passengers p ON b.passenger_id =
p.passenger_id**

**JOIN booking_flight bf ON b.booking_id =
bf.booking_id**

JOIN flights f ON bf.flight_id = f.flight_id

**JOIN airport a ON f.arrival_airport_id =
a.airport_id**

WHERE b.status != 'cancelled'

GROUP BY a.country

ORDER BY total_visitors DESC

LIMIT 10;

10

The screenshot shows the pgAdmin 4 interface with a query window containing the following SQL code:

```
1 DROP VIEW IF EXISTS flights_departing_on_date;
2 DROP VIEW IF EXISTS upcoming_week_bookings;
3 DROP VIEW IF EXISTS top_5_popular_routes;
4 DROP VIEW IF EXISTS airline_flights;
5 DROP VIEW IF EXISTS flights_delayed_over_24_hours;
6 DROP VIEW IF EXISTS leffler_thompson_passengers;
7 DROP VIEW IF EXISTS top_10_most_visited_countries;
```

The "Messages" tab displays the following log entries:

```
NOTICE: view "flights_departing_on_date" does not exist, skipping
NOTICE: view "upcoming_week_bookings" does not exist, skipping
NOTICE: view "top_5_popular_routes" does not exist, skipping
NOTICE: view "airline_flights" does not exist, skipping
NOTICE: view "flights_delayed_over_24_hours" does not exist, skipping
NOTICE: view "leffler_thompson_passengers" does not exist, skipping
NOTICE: view "top_10_most_visited_countries" does not exist, skipping
DROP VIEW
```

Below the log, it says "Query returned successfully in 29 msec." and "Total rows: Query complete 00:00:00.029". The status bar at the bottom right shows "Query returned successfully in 29 CRLF - Ln 7, Col 51" and the system time "9:11 PM 11/18/2025".

**DROP VIEW IF EXISTS
flights_departing_on_date;**

**DROP VIEW IF EXISTS
upcoming_week_bookings;**

DROP VIEW IF EXISTS top_5_popular_routes;

DROP VIEW IF EXISTS airline_flights;

**DROP VIEW IF EXISTS
flights_delayed_over_24_hours;**

**DROP VIEW IF EXISTS
leffler_thompson_passengers;**

**DROP VIEW IF EXISTS
top_10_most_visited_countries;**

