

The screenshot shows the DBeaver interface with the following details:

- Servers:** PostgreSQL 17 (selected), PostgreSQL 18
- Databases:** 4 databases listed under PostgreSQL 17
- Login/Group Roles:** Available under PostgreSQL 17
- Tablespaces:** Available under PostgreSQL 17
- Query Editor:** Contains a CREATE VIEW SQL statement named "flight_details". The code is as follows:

```
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
24    ON f.airline_id = al.airline_id
25  JOIN airport dep_airport
26    ON f.departure_airport_id = dep_airport.airport_id
27  JOIN airport arr_airport
28    ON f.arrival_airport_id = arr_airport.airport_id;
```

Data Output: CREATE VIEW

Messages: Query returned successfully in 861 msec.

Notifications: A green notification bar at the bottom right indicates "Query returned successfully in 861 msec." with a checkmark icon.

Bottom Status Bar: Total rows: 0, Query complete 00:00:00.861, LF, Ln 29, Col 1

airport_database/postgres@PostgreSQL 17*

Servers (2) airport_database/postgres@PostgreSQL 17

Databases (4) No limit

Login/Group Roles Tablespaces

Tablespaces

PostgreSQL 18

```

1 SELECT *
2 FROM flight_details
3 WHERE departure_date = '2024-01-15';
4

```

Data Output Messages Notifications

Showing rows: 1 to 6 Page No: 1 of 1

| | flight_id | flight_no | scheduled_departure | scheduled_arrival | actual_departure | actual_arrival | status | departing_gate | arriving_gate | airline |
|---|-----------|-----------|---------------------|-------------------|------------------|----------------|---------|----------------|---------------|---------|
| 1 | 28 | NO-18 | 2024-01-15 | 2024-02-28 | 2023-05-18 | 2023-07-19 | Delayed | 56 | 436 | [null] |
| 2 | 134 | TZ-12 | 2024-01-15 | 2024-01-09 | 2024-02-20 | 2023-10-06 | Delayed | 1400 | 3250 | LTBD |
| 3 | 344 | GB-ENG | 2024-01-15 | 2023-10-07 | 2024-03-07 | 2023-07-09 | Delayed | 202 | 425 | MRM |
| 4 | 465 | PG-GPK | 2024-01-15 | 2023-10-19 | 2023-08-07 | 2024-02-09 | Delayed | 70 | 1027 | MRM |
| 5 | 575 | US-GA | 2024-01-15 | 2023-12-22 | 2023-10-19 | 2023-05-15 | Delayed | 354 | 7 | ZSR |
| 6 | 660 | CM-OU | 2024-01-15 | 2023-10-13 | 2023-10-20 | 2023-11-14 | Delayed | 4593 | 2040 | AYD |

Total rows: 6 Query complete 00:00:00.919 LF Ln 4, Col 1

airport_database/postgres@PostgreSQL 17* X

Servers (2) PostgreSQL 17 Databases (4) Login/Group Roles Tablespaces PostgreSQL 18

airport_database/postgres@PostgreSQL 17

No limit

Query History

```
18    f.scheduled_departure,
19    f.scheduled_arrival,
20    f.actual_departure,
21    f.actual_arrival,
22    f.status      AS flight_status,
23    f.departing_gate,
24    f.arriving_gate,
25
26    al.airline_code,
27    al.airline_name,
28
29    dep_airport.airport_name AS departure_airport_name,
30    dep_airport.city        AS departure_city,
31    dep_airport.country     AS departure_country,
32
33    arr_airport.airport_name AS arrival_airport_name,
34    arr_airport.city        AS arrival_city,
35    arr_airport.country     AS arrival_country,
36
37    bp.seat,
38    bp.boarding_time,
39
40    bg.weight_in_kg AS baggage_weight
41 FROM booking b
42 JOIN passengers p      ON b.passenger_id      = p.passenger_id
43 JOIN booking_flight bf ON b.booking_id       = bf.booking_id
44 JOIN flights f         ON bf.flight_id        = f.flight_id
45 JOIN airline al        ON f.airline_id        = al.airline_id
46 JOIN airport dep_airport
47   ON f.departure_airport_id = dep_airport.airport_id
48 JOIN airport arr_airport
49   ON f.arrival_airport_id  = arr_airport.airport_id
50 LEFT JOIN boarding_pass bp ON b.booking_id = bp.booking_id
51 LEFT JOIN baggage      bg   ON b.booking_id = bg.booking_id
52 WHERE f.scheduled_departure >= CURRENT_DATE
53   AND f.scheduled_departure < CURRENT_DATE + INTERVAL '7 days'
54   AND b.status != 'cancelled';
```

Data Output Messages Notifications

Total rows: 6 | Query complete 00:00:00.919 | LF | Ln 55, Col 1

airport_database/postgres@PostgreSQL 17* X

Servers (2) airport_database/postgres@PostgreSQL 17

Databases (4) No limit

Login/Group Roles Tablespaces

PostgreSQL 18

Query History

```
1 SELECT * FROM upcoming_week_bookings;
2
3 SELECT *
4 FROM upcoming_week_bookings
5 WHERE first_name = 'John'
6   AND last_name  = 'Smith';
7
8 SELECT *
9 FROM upcoming_week_bookings
10 WHERE airline_name = 'Delta Airlines'
11 ORDER BY scheduled_departure;
12
```

Data Output Messages Notifications

SQL

| | booking_id | booking_platform | booking_status | price | booking_date | passenger_id | first_name | last_name | date_of_birth | gender |
|--|------------|------------------------|------------------------|---------------|--------------|--------------|------------------------|------------------------|---------------|------------------------|
| | integer | character varying (50) | character varying (50) | numeric (7,2) | date | integer | character varying (50) | character varying (50) | date | character varying (50) |

Total rows: 0 Query complete 00:00:00.445 LF Ln 12, Col 1

airport_database/postgres@PostgreSQL 17* X

Servers (2) PostgreSQL 17 Databases (4) Login/Group Roles Tablespaces PostgreSQL 18

airport_database/postgres@PostgreSQL 17

No limit

Query History

```
1 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 FROM booking b
15 JOIN booking_flight bf ON b.booking_id = bf.booking_id
16 JOIN flights f ON bf.flight_id = f.flight_id
17 JOIN airline al ON f.airline_id = al.airline_id
18 JOIN airport dep_airport
19     ON f.departure_airport_id = dep_airport.airport_id
20 JOIN airport arr_airport
21     ON f.arrival_airport_id = arr_airport.airport_id
22 WHERE b.status != 'cancelled'
23 GROUP BY
24     dep_airport.city,
25     dep_airport.airport_name,
26     arr_airport.city,
27     arr_airport.airport_name,
28     dep_airport.country,
29     arr_airport.country,
30     al.airline_name
31 ORDER BY total_bookings DESC
32 LIMIT 5;
```

Data Output Messages Notifications

booking_id booking_platform booking_status price booking_date passenger_id first_name last_name date_of_birth gender

Total rows: 0 Query complete 00:00:00.445 LF Ln 33, Col 1

airport_database/postgres@PostgreSQL 17* X

Servers (2) airport_database/postgres@PostgreSQL 17

Databases (4) No limit

Login/Group Roles Tablespaces

PostgreSQL 18

Query History

```
1 SELECT * FROM top_5_popular_routes;
2
3 SELECT
4     rank,
5     departure_city || ' to ' || arrival_city AS route,
6     total_bookings,
7     average_booking_price
8 FROM top_5_popular_routes;
9
10
11
12
13
14
```

Data Output Messages Notifications

rank bigint
total_bookings bigint
departure_city character varying (50)
departure_airport character varying (50)
arrival_city character varying (50)
arrival_airport character varying (50)
departure_country character varying (50)
arrival_country character varying (50)
airline_name character varying (50)

Total rows: 0 Query complete 00:00:01.022 LF Ln 14, Col 1

The screenshot shows the DBeaver IDE interface with the following details:

- Servers:** A tree view on the left shows two servers: "PostgreSQL 17" (selected) and "PostgreSQL 18". Under "PostgreSQL 17", there are four databases: "Databases (4)", "Login/Group Roles", and "Tablespaces".
- Query Editor:** The main area displays a SQL script for creating a view named "airline_flights". The code is as follows:

```
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,
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    FROM flights f
19    JOIN airline al      ON f.airline_id      = al.airline_id
20    JOIN airport dep_airport
21        ON f.departure_airport_id = dep_airport.airport_id
22    JOIN airport arr_airport
23        ON f.arrival_airport_id  = arr_airport.airport_id
24    WHERE al.airline_code = 'AA';
25
```

- Data Output:** Below the code, the "Data Output" tab is selected, showing the results of the query creation:

 - Text: "CREATE VIEW"
 - Text: "Query returned successfully in 1 secs 267 msec."

- Bottom Status Bar:** Shows "Total rows: 0" and "Query complete 00:00:01.267".
- Bottom Right:** Includes status indicators "LF" and "Ln 25, Col 1".

The screenshot shows the DBeaver IDE interface with the following details:

- Servers:** A tree view on the left shows "Servers (2)" with "PostgreSQL 17" expanded, displaying "Databases (4)", "Login/Group Roles", and "Tablespaces".
- Query Editor:** The main area displays a PostgreSQL query for creating a view named "airline_flights". The code is as follows:

```
1 CREATE OR REPLACE 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,
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  FROM flights f
19  JOIN airline al      ON f.airline_id          = al.airline_id
20  JOIN airport dep_airport
21    ON f.departure_airport_id = dep_airport.airport_id
22  JOIN airport arr_airport
23    ON f.arrival_airport_id  = arr_airport.airport_id
24 WHERE al.airline_code = 'AA'
25   AND f.scheduled_departure >= CURRENT_DATE
26   AND f.scheduled_departure <  CURRENT_DATE + INTERVAL '7 days';
27
```

- Data Output:** Below the query, the output shows:
 - "CREATE VIEW"
 - "Query returned successfully in 1 secs 545 msec."
- Status Bar:** At the bottom right, a green message box indicates: "✓ Query returned successfully in 1 secs 545 msec. ✘".
- Bottom Navigation:** The status bar also shows "Total rows: 0" and "Query complete 00:00:01.545".

The screenshot shows the DBeaver IDE interface with the following details:

- Servers:** A tree view on the left shows "Servers (2)" with "PostgreSQL 17" expanded, displaying "Databases (4)", "Login/Group Roles", and "Tablespaces".
- Connections:** The top center shows a connection to "airport_database/postgres@PostgreSQL 17".
- Toolbar:** Standard database toolbar with icons for filter, table, search, etc.
- Query Editor:** The main area contains a SQL script for creating a view. The code is as follows:

```
1 CREATE VIEW flights_delayed_over_24_hours AS
2 SELECT
3     f.flight_id,
4     f.flight_no,
5     f.scheduled_departure,
6     f.actual_departure,
7     f.scheduled_arrival,
8     f.actual_arrival,
9     (f.actual_departure - f.scheduled_departure) AS departure_delay,
10    al.airline_name,
11    dep_airport.airport_name AS departure_airport,
12    arr_airport.airport_name AS arrival_airport
13   FROM flights f
14  JOIN airline al      ON f.airline_id      = al.airline_id
15  JOIN airport dep_airport
16    ON f.departure_airport_id = dep_airport.airport_id
17  JOIN airport arr_airport
18    ON f.arrival_airport_id  = arr_airport.airport_id
19 WHERE f.actual_departure IS NOT NULL
20   AND f.actual_departure > f.scheduled_departure + INTERVAL '24 hours';
21
```

- Data Output:** Below the editor, tabs for "Data Output", "Messages", and "Notifications" are present. The "Data Output" tab shows the executed query and its success message.
- Messages:** The "Messages" tab shows the output: "CREATE VIEW" and "Query returned successfully in 1 secs 721 msec."
- Status Bar:** The bottom status bar displays "Total rows: 0" and "Query complete 00:00:01.721".
- Bottom Right:** A green notification bar indicates "Query returned successfully in 1 secs 721 msec." with a checkmark icon.

The screenshot shows the DBeaver interface with a PostgreSQL connection active. The left sidebar displays server connections, including 'PostgreSQL 17' which is expanded to show 'Databases (4)'. The main area contains a SQL query window with the following code:

```
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
8     ON p.passenger_id = b.passenger_id
9   WHERE b.booking_platform = 'Leffler-Thompson';
10
```

The 'Messages' tab in the bottom panel shows the execution results:

CREATE VIEW

Query returned successfully in 578 msec.

A green notification bar at the bottom right indicates: ✓ Query returned successfully in 578 msec. X

At the very bottom, status bars show 'Total rows:' and 'Query complete 00:00:00.578'.

The screenshot shows the DBeaver database management tool interface. The top navigation bar indicates the connection is to `airport_database/postgres@PostgreSQL 17*`. The left sidebar lists servers and databases, with `PostgreSQL 17` selected. The main area contains a query editor with the following SQL code:

```
1 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;
14
```

The results pane at the bottom shows the output of the query:

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 93 msec.

A green notification bar at the bottom right states: ✓ Query returned successfully in 93 msec. X

Total rows: | Query complete 00:00:00.093 | LF | Ln 14, Col 1

The screenshot shows the DBeaver application interface. On the left, the 'Servers' tree view displays two servers: 'PostgreSQL 17' (selected) and 'PostgreSQL 18'. The 'Databases' node under PostgreSQL 17 has four children. The main panel contains a query editor with the following SQL code:

```
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;
17
```

The 'Messages' tab in the bottom panel shows the execution results:

CREATE VIEW

Query returned successfully in 334 msec.

A green notification bar at the bottom right indicates: ✓ Query returned successfully in 334 msec. X

Bottom status bar: Total rows: | Query complete 00:00:00.334 | LF | Ln 17, Col 1

The screenshot shows the DBeaver interface with the following details:

- Servers:** A tree view on the left shows "Servers (2)" expanded, with "PostgreSQL 17" selected. Under PostgreSQL 17, "Databases (4)", "Login/Group Roles", and "Tablespaces" are listed. There is also a connection to "PostgreSQL 18".
- Query Editor:** The main area displays a series of SQL commands numbered 1 to 8, intended to drop various views if they exist. The queries are:

```
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;
```
- Data Output:** Below the queries, the "Data Output" tab is active, showing the results of the execution:

```
NOTICE:  view "flights_departing_on_date" does not exist, skipping
DROP VIEW

Query returned successfully in 2 secs 347 msec.
```
- Bottom Status:** The status bar at the bottom indicates "Total rows: 0" and "Query complete 00:00:02.456".