

Servers (2)

PostgreSQL 17

Databases (4)

Login/Group Roles

Tablespaces

PostgreSQL 18

airport\_database/postgres@PostgreSQL 17\*

airport\_database/postgres@PostgreSQL 17

No limit

Query

Query History

31

)

32

VALUES (

33

p\_flight\_no,

34

p\_scheduled\_departure,

35

p\_scheduled\_arrival,

36

p\_departure\_airport\_id,

37

p\_arrival\_airport\_id,

38

p\_departing\_gate,

39

p\_arriving\_gate,

40

p\_airline\_id,

41

p\_status,

42

p\_actual\_departure,

43

p\_actual\_arrival,

44

CURRENT\_DATE,

45

CURRENT\_DATE

46

);

47

END;

48

\$\$;

49

50

CALL insert\_new\_flight(

51

'KC123',

52

'2025-12-01',

53

'2025-12-01',

54

1,

55

2,

56

'A5',

57

'B3',

58

1,

59

'Scheduled',

60

NULL,

61

NULL

62

);

63

Data Output

Messages

Notifications

CALL

Query returned successfully in 2 secs 265 msec.

Total rows:

Query complete 00:00:02.265

LF

Ln 63, Col 1

Servers (2)

PostgreSQL 17

Databases (4)

Login/Group Roles

Tablespaces

PostgreSQL 18

airport\_database/postgres@PostgreSQL 17\*

airport\_database/postgres@PostgreSQL 17

No limit

Query

Query History

1

CREATE OR REPLACE PROCEDURE update\_flight\_status(

2

p\_flight\_id INT,

3

p\_new\_status VARCHAR

4

)

5

LANGUAGE plpgsql

6

AS \$\$

7

BEGIN

8

UPDATE flights

9

SET status = p\_new\_status,

10

update\_at = CURRENT\_DATE

11

WHERE flight\_id = p\_flight\_id;

12

13

IF NOT FOUND THEN

14

RAISE NOTICE 'Flight with id % not found.', p\_flight\_id;

15

END IF;

16

END;

17

\$\$;

18

19

CALL update\_flight\_status(1, 'Delayed');

20

Data Output

Messages

Notifications

CALL

Query returned successfully in 1 secs 676 msec.

Total rows:

Query complete 00:00:01.676

✓ Query returned successfully in 1 secs 676 msec.

✕

LF

Ln 20, Col 1

Servers (2)

PostgreSQL 17

Databases (4)

Login/Group Roles

Tablespaces

PostgreSQL 18

airport\_database/postgres@PostgreSQL 17\* X

airport\_database/postgres@PostgreSQL 17

No limit

Execute script

F5

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

CREATE OR REPLACE FUNCTION

get\_flights\_from\_airport

(

p\_departure\_airport\_id

INT

)

RETURNS TABLE (

flight\_id

INT,

flight\_no

VARCHAR,

scheduled\_departure

DATE,

scheduled\_arrival

DATE,

departure\_airport\_id

INT,

arrival\_airport\_id

INT,

airline\_id

INT,

status

VARCHAR

)

LANGUAGE sql

AS \$\$

SELECT

flight\_id,

flight\_no,

scheduled\_departure,

scheduled\_arrival,

departure\_airport\_id,

arrival\_airport\_id,

airline\_id,

status

FROM flights

WHERE departure\_airport\_id = p\_departure\_airport\_id

ORDER BY scheduled\_departure;

\$\$;

SELECT \* FROM get\_flights\_from\_airport(1);

Data Output

Messages

Notifications

Showing rows: 1 to 50

Page No: 1 of 1

|   | flight_id<br>integer | flight_no<br>character varying | scheduled_departure<br>date | scheduled_arrival<br>date | departure_airport_id<br>integer | arrival_airport_id<br>integer | airline_id<br>integer | status<br>character varying |
|---|----------------------|--------------------------------|-----------------------------|---------------------------|---------------------------------|-------------------------------|-----------------------|-----------------------------|
| 1 | 168                  | IL-D                           | 2023-03-21                  | 2023-05-24                | 1                               | 18                            | 12                    | Delayed                     |

Total rows: 50Query complete 00:00:00.577

LF

Ln 31, Col 1



Servers (2)

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

airport\_database/postgres@PostgreSQL 17\*

airport\_database/postgres@PostgreSQL 17

No limit

E

Query

Query History

```
1 CREATE OR REPLACE FUNCTION avg_arrival_delay(  
2     p_arrival_airport_id INT  
3 )  
4 RETURNS NUMERIC(6,2)  
5 LANGUAGE sql  
6 AS $$  
7     SELECT AVG(actual_arrival - scheduled_arrival)  
8     FROM flights  
9     WHERE arrival_airport_id = p_arrival_airport_id  
10        AND actual_arrival IS NOT NULL;  
11 $$;  
12  
13 SELECT avg_arrival_delay(2) AS average_delay_days;  
14
```

Data Output

Messages

Notifications

Showing rows: 1 to 1

Page No: 1 of 1

|   | average_delay_days numeric |
|---|----------------------------|
| 1 | 22.7500000000000000        |

Total rows: 1 | Query complete 00:00:01.426

Successfully run. Total query runtime: 1 secs 426 msec. 1 rows affected.

LF | Ln 14, Col 1

Servers (2)

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

airport\_database/postgres@PostgreSQL 17\*

airport\_database/postgres@PostgreSQL 17

No limit

Query Query History

```
1 CREATE OR REPLACE FUNCTION get_passengers_by_flight(  
2     p_flight_no VARCHAR  
3 )  
4 RETURNS TABLE (  
5     passenger_id INT,  
6     first_name   VARCHAR,  
7     last_name    VARCHAR,  
8     date_of_birth DATE,  
9     gender       VARCHAR,  
10    passport_number VARCHAR  
11 )  
12 LANGUAGE sql  
13 AS $$  
14     SELECT  
15         p.passenger_id,  
16         p.first_name,  
17         p.last_name,  
18         p.date_of_birth,  
19         p.gender,  
20         p.passport_number  
21     FROM passengers p  
22     JOIN booking b      ON p.passenger_id = b.passenger_id  
23     JOIN booking_flight bf ON b.booking_id  = bf.booking_id  
24     JOIN flights f      ON bf.flight_id   = f.flight_id  
25     WHERE f.flight_no = p_flight_no  
26     ORDER BY p.last_name, p.first_name;  
27 $$;  
28  
29 SELECT * FROM get_passengers_by_flight('KC123');  
30
```

Data Output Messages Notifications

SQL

| passenger_id | first_name        | last_name         | date_of_birth | gender            | passport_number   |
|--------------|-------------------|-------------------|---------------|-------------------|-------------------|
| integer      | character varying | character varying | date          | character varying | character varying |

Total rows: 0

Query complete 00:00:00.802

LF Ln 30, Col 1

Successfully run. Total query runtime: 802 msec. 0 rows affected.



Servers (2)

PostgreSQL 17

Databases (4)

Login/Group Roles

Tablespaces

PostgreSQL 18

airport\_database/postgres@PostgreSQL 17\*

airport\_database/postgres@PostgreSQL 17

No limit

Execute script

F5

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

```
CREATE OR REPLACE PROCEDURE get_top_passenger_no_cursor()
LANGUAGE plpgsql
AS $$
DECLARE
    top_passenger RECORD;
BEGIN
    SELECT
        p.passenger_id,
        p.first_name,
        p.last_name,
        COUNT(bf.flight_id) AS total_flights
    INTO top_passenger
    FROM passengers p
    JOIN booking b      ON p.passenger_id = b.passenger_id
    JOIN booking_flight bf ON b.booking_id  = bf.booking_id
    GROUP BY p.passenger_id, p.first_name, p.last_name
    ORDER BY total_flights DESC
    LIMIT 1;

    RAISE NOTICE
        'Passenger ID: %, Name: % %, Total Flights: %',
        top_passenger.passenger_id,
        top_passenger.first_name,
        top_passenger.last_name,
        top_passenger.total_flights;
END;
$$;

CALL get_top_passenger_no_cursor();
```

Data Output

Messages

Notifications

NOTICE: Passenger ID: 68, Name: Sheela Roux, Total Flights: 6

CALL

Query returned successfully in 704 msec.

Total rows:

Query complete 00:00:00.704

LF

Ln 30, Col 1

✓ Query returned successfully in 704 msec. ✕

Servers (2)

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

airport\_database/postgres@PostgreSQL 17\*

airport\_database/postgres@PostgreSQL 17

No limit

Query Query History

```
2 LANGUAGE plpgsql
3 AS $$
4 BEGIN
5     CREATE TEMP TABLE IF NOT EXISTS temp_delayed_flights AS
6     SELECT
7         flight_id,
8         flight_no,
9         scheduled_departure,
10        actual_departure,
11        scheduled_arrival,
12        actual_arrival,
13        departure_airport_id,
14        arrival_airport_id,
15        airline_id,
16        status,
17        (actual_departure - scheduled_departure) AS delay_days
18    FROM flights
19    WHERE actual_departure IS NOT NULL
20          AND scheduled_departure IS NOT NULL
21          -- задержка больше 1 дня (24 часов)
22          AND (actual_departure - scheduled_departure) > 1
23    ORDER BY delay_days DESC;
24
25    RAISE NOTICE 'Delayed flights over 24h inserted into temp_delayed_flights';
26 END;
27 $$;
28
29 CALL get_flights_delayed_over_24h_no_cursor();
30
31 SELECT * FROM temp_delayed_flights;
```

Data Output Messages Notifications

Showing rows: 1 to 492

Page No: 1 of 1

|   | flight_id<br>integer | flight_no<br>character varying (50) | scheduled_departure<br>date | actual_departure<br>date | scheduled_arrival<br>date | actual_arrival<br>date | departure_airport_id<br>integer | arrival_airport_id<br>integer | airline_id<br>integer | status<br>character varying (50) |
|---|----------------------|-------------------------------------|-----------------------------|--------------------------|---------------------------|------------------------|---------------------------------|-------------------------------|-----------------------|----------------------------------|
| 1 | 952                  | US-AZ                               | 2023-03-26                  | 2024-03-11               | 2023-12-19                | 2023-08-13             | 14                              | 14                            | 43                    | Boarding                         |
| 2 | 711                  | CA-NU                               | 2023-03-23                  | 2024-02-29               | 2023-12-12                | 2023-06-18             | 12                              | 3                             | 33                    | Boarding                         |

Total rows: 492 | Query complete 00:00:00.463 | LF | Ln 32, Col 1



Servers (2)

PostgreSQL 17

Databases (4)

Login/Group Roles

Tablespaces

PostgreSQL 18

airport\_database/postgres@PostgreSQL 17\*

airport\_database/postgres@PostgreSQL 17

No limit

Query

Query History

1

CREATE OR REPLACE FUNCTION count\_flights\_per\_airline()

2

RETURNS TABLE (

3

airline\_id    INT,

4

airline\_name  VARCHAR,

5

total\_flights INT

6

)

7

LANGUAGE sql

8

AS \$\$

9

SELECT

10

a.airline\_id,

11

a.airline\_name,

12

COUNT(f.flight\_id) AS total\_flights

13

FROM airline a

14

LEFT JOIN flights f ON a.airline\_id = f.airline\_id

15

GROUP BY a.airline\_id, a.airline\_name

16

ORDER BY total\_flights DESC;

17

\$\$;

18

19

SELECT \* FROM count\_flights\_per\_airline();

20

Data Output

Messages

Notifications

SQL

Showing rows: 1 to 50

Page No: 1 of 1

|   | airline_id<br>integer | airline_name<br>character varying | total_flights<br>integer |
|---|-----------------------|-----------------------------------|--------------------------|
| 1 | 1                     | IPC                               | 33                       |
| 2 | 43                    | KMA                               | 30                       |
| 3 | 29                    | NHT                               | 29                       |
| 4 | 19                    | CSC                               | 28                       |
| 5 | 47                    | DNU                               | 27                       |
| 6 | 16                    | DUC                               | 25                       |
| 7 | 23                    | SPI                               | 25                       |
| 8 | 26                    | YHB                               | 25                       |
| 9 | 8                     | QIG                               | 25                       |

Total rows: 50

Query complete 00:00:03.215

LF

Ln 20, Col 1



Servers (2)

PostgreSQL 17

Databases (4)

Login/Group Roles

Tablespaces

PostgreSQL 18

airport\_database/postgres@PostgreSQL 17\*

airport\_database/postgres@PostgreSQL 17

No limit

Execute script

F5

Query

Query History

1

CREATE OR REPLACE PROCEDURE avg\_ticket\_price\_for\_flight(  
2     p\_flight\_id INT  
3 )  
4 LANGUAGE plpgsql  
5 AS \$\$  
6 DECLARE  
7     avg\_price NUMERIC(10,2);  
8 BEGIN  
9     SELECT AVG(price) INTO avg\_price  
10    FROM booking b  
11    JOIN booking\_flight bf ON b.booking\_id = bf.booking\_id  
12    WHERE bf.flight\_id = p\_flight\_id;  
13  
14 IF avg\_price IS NULL THEN  
15     RAISE NOTICE 'No bookings found for flight ID %', p\_flight\_id;  
16 ELSE  
17     RAISE NOTICE 'Average ticket price for flight ID % is %',  
18         p\_flight\_id, avg\_price;  
19    END IF;  
20 END;  
21 \$\$;  
22  
23 CALL avg\_ticket\_price\_for\_flight(1);  
24

Data Output

Messages

Notifications

NOTICE: No bookings found for flight ID 1

CALL

Query returned successfully in 1 secs 188 msec.

Total rows:

Query complete 00:00:01.188

LF

Ln 24, Col 1

Query returned successfully in 1 secs 188 msec.

The screenshot displays the DBeaver interface with a dark theme. On the left sidebar, the 'Servers' tree shows two PostgreSQL instances; the first instance is expanded to show its databases, roles, and tablespaces. The main editor window is titled 'airport\_database/postgres@PostgreSQL 17\*' and contains a SQL script. The script defines a procedure named 'get\_most\_expensive\_flight()' which selects flight details ordered by price in descending order, limited to one result. It includes conditional logic to raise a notice if no flights are found or to print the flight details. Below the script editor, there are tabs for 'Query', 'Query History', 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, displaying a notice message from the executed query: 'NOTICE: Flight: NP-SA, Departure: Henri Coandă International Airport, Arrival: Armidale Airport, Ticket Price: 9977.57'. At the bottom status bar, it indicates 'Total rows: Query complete 00:00:00.609' and the current cursor position is 'Ln 35, Col 1'.