

1)

```
CREATE OR REPLACE PROCEDURE add_flight(
    p_flight_id INT,
    p_flight_no VARCHAR,
    p_scheduled_departure DATE,
    p_scheduled_arrival DATE,
    p_departure_airport_id INT,
    p_arrival_airport_id INT,
    p_departing_gate VARCHAR,
    p_arriving_gate VARCHAR,
    p_airline_id INT,
    p_status VARCHAR
)
LANGUAGE plpgsql
AS $$

BEGIN
    INSERT INTO flights (
        flight_id,
        flight_no,
        scheduled_departure,
        scheduled_arrival,
        departure_airport_id,
        arrival_airport_id,
        departing_gate,
        arriving_gate,
        airline_id,
        status
    )
    VALUES (
        p_flight_id,
        p_flight_no,
        p_scheduled_departure,
        p_scheduled_arrival,
        p_departure_airport_id,
        p_arrival_airport_id,
        p_departing_gate,
        p_arriving_gate,
        p_airline_id,
        p_status
    );
END;
```

scheduled_departure	scheduled_arrival	departure_airport_id	arrival_airport_id	departing_gate	arriving_gate	airline_id	status	actual_departure	actual_arrival	created_at	update_at	
2023-08-07	2023-08-12	15	12	0239	70	49	Boarding	2023-08-07	2023-08-16	2024-02-07	2023-09-17	
971	2023-06-05		9	7 [null]	328	49	Boarding	2023-09-30	2023-12-11	2023-05-14	2024-01-28	
972	2023-11-25		20	6	3292	32	Delayed	2024-03-10	2023-03-27	2023-07-04	2023-05-03	
973	2024-01-23		19	2	241	22	13	Boarding	2023-05-25	2024-01-31	2024-03-16	2023-06-06
974	2023-09-01		7	15	5640	4804	6	Boarding	2023-04-12	2023-10-11	2023-06-18	2023-09-22
975	2023-08-25		17	13	55	676	32	Delayed	2023-07-29	2024-01-15	2024-02-29	2023-10-15

2)

```
CREATE OR REPLACE PROCEDURE status_update(
    p_flight_id INT,
    p_status VARCHAR
)
LANGUAGE plpgsql
AS $$

BEGIN
    UPDATE flights
    SET status = p_status
    WHERE flight_id = p_flight_id;
END;
$$;

CALL status_update(2, 'Delayed');

SELECT * FROM flights
```

CALL

Query returned successfully in 84 msec.

✓ Query returned successfully in 84 msec. X

3)

pgAdmin 4

Welcome lab_10/postgres@PostgreSQL 17*

Query History

```

1 v CREATE OR REPLACE PROCEDURE get_flights_from_airport(
2     IN p_airport_id INT
3 )
4 LANGUAGE plpgsql
5 AS $$*
6 BEGIN
7
8     RAISE NOTICE 'Flights departing from airport_id = %', p_airport_id;
9
10
11    SELECT f.flight_id,
12        f.sch_departure_time,
13        f.sch_arrival_time,
14        a.airport_name AS departing_airport,
15        f.departing_gate,
16        f.arriving_gate,
17        f.airline_id
18    FROM Flights f
19    JOIN Airport a ON f.departing_airport_id = a.airport_id
20    WHERE f.departing_airport_id = p_airport_id;
21 END;
22 $$*

```

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 62 msec.

Total rows: Query complete 00:00:00.062

3°C Mostly clear

Query returned successfully in 62 msec. CRLF Ln 1, Col 1

21:08 ENG 02.12.2025

4)

pgAdmin 4

Welcome lab_10/postgres@PostgreSQL 17*

Query History

```

1 v CREATE OR REPLACE FUNCTION avg_delay(p_airport_id INT)
2 RETURNS INTERVAL
3 LANGUAGE plpgsql
4 AS $$*
5 DECLARE
6     result INTERVAL;
7 BEGIN
8     SELECT AVG(f.actual_arrival - f.scheduled_arrival)
9     INTO result
10    FROM flights f
11   WHERE f.arrival_airport_id = p_airport_id
12     AND f.actual_arrival IS NOT NULL
13     AND f.scheduled_arrival IS NOT NULL;
14
15 RETURN result;
16 END;
17 $$;
18
19 select avg_delay(13)

```

Data Output Messages Notifications

	avg_delay	interval
1	-0:00:14.916667	

Showing rows: 1 to 1 | Page No: 1 | of 1 | < > | << >> | <<< >>> | <<<< >>>> |

Total rows: 1 Query complete 00:00:00.060

4°C Partly cloudy

CRLF Ln 13, Col 43

22:05 02.12.2025

5)

pgAdmin 4

Welcome lab_10/postgres@PostgreSQL 17*

Query History

```

1 v CREATE OR REPLACE PROCEDURE passengers_for_flight(
2     p_flight_id INT
3 )
4 LANGUAGE plpgsql
5 AS $$
6 BEGIN
7     SELECT p.passenger_id, p.first_name, p.last_name
8     FROM passengers p
9     JOIN booking b ON p.passenger_id = b.passenger_id
10    WHERE b.flight_id = p_flight_id;
11 END;
12 $$;
13 |
14 call passengers_for_flight(5);

```

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 56 msec.

Total rows: Query complete 00:00:00.056

AWO1 +0.21% 22:35 02.12.2025

Query returned successfully in 56 msec. CRLF Ln 13, Col 1

6)

pgAdmin 4

Welcome lab_10/postgres@PostgreSQL 17*

Query History

```

1 v CREATE OR REPLACE PROCEDURE top_passenger()
2 LANGUAGE plpgsql
3 AS $$
4 DECLARE
5     top_passenger_id INT;
6 BEGIN
7     SELECT p.passenger_id
8     INTO top_passenger_id
9     FROM passengers p
10    JOIN booking b ON p.passenger_id = b.passenger_id
11    JOIN booking_flight bf ON b.booking_id = bf.booking_id
12    GROUP BY p.passenger_id
13    ORDER BY COUNT(bf.flight_id) DESC
14    LIMIT 1;
15
16    RAISE NOTICE 'Passenger with most flights: %', top_passenger_id;
17 END;
18 $$;
19 call top_passenger();

```

Data Output Messages Notifications

NOTICE: Passenger with most flights: 68

CALL

Query returned successfully in 65 msec.

Total rows: Query complete 00:00:00.065

USD/INR +0.33% 23:22 02.12.2025

7)

pgAdmin 4

Welcome lab_10/postgres@PostgreSQL 17*

Query History

```

1 v CREATE OR REPLACE PROCEDURE flights_delayed_24h()
2 LANGUAGE plpgsql
3 AS $$
4 BEGIN
5   RAISE NOTICE 'Flights delayed more than 24 hours!';
6   PERFORM flight_id, flight_no, actual_arrival - scheduled_arrival AS delay
7   FROM flights
8   WHERE actual_arrival - scheduled_arrival > INTERVAL '24 hours';
9 END;
10 $$;
11 call flights_delayed_24h();

```

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 54 msec.

Total rows: Query complete 00:00:00.054 CRLF Ln 1, Col 1

23:23 ENG 02.12.2025

8)

pgAdmin 4

Welcome lab_10/postgres@PostgreSQL 17*

Query History

```

1 v CREATE OR REPLACE FUNCTION count_flights_per_airline()
2 RETURNS TABLE(airline_id INT, flight_count INT)
3 LANGUAGE plpgsql
4 AS $$
5 BEGIN
6   RETURN QUERY
7   SELECT airline_id, COUNT(*)
8   FROM flights
9   GROUP BY airline_id;
10 END;
11 $$;
12 select * from count_flights_per_airline();
13

```

Data Output Messages Notifications

CREATE FUNCTION

Query returned successfully in 87 msec.

Total rows: Query complete 00:00:00.087 CRLF Ln 1, Col 1

23:24 ENG 02.12.2025

9)

The screenshot shows the pgAdmin 4 interface. A query window is open with the following SQL code:

```
1 v CREATE OR REPLACE PROCEDURE avg_ticket_price(p_flight_id INT)
2 LANGUAGE plpgsql
3 AS $$
4 DECLARE
5     avg_price NUMERIC;
6 v BEGIN
7     SELECT AVG(b.price)
8     INTO avg_price
9     FROM booking b
10    JOIN booking_flight bf ON b.booking_id = bf.booking_id
11   WHERE bf.flight_id = p_flight_id;
12
13  RAISE NOTICE 'Average ticket price for flight %: %', p_flight_id, avg_price;
14 END;
15 $$;
16
17 call avg_ticket_price(500);
```

The Data Output tab shows the result of the query:

```
NOTICE: Average ticket price for flight 500: 6555.2450000000000000
CALL
Query returned successfully in 59 msec.
```

The system tray at the bottom right shows the date and time as 02.12.2025.

10)

The screenshot shows the pgAdmin 4 interface. A query window is open with the following SQL code:

```
1 v CREATE OR REPLACE PROCEDURE most_expensive_flight(
2     OUT o_flight_id INT,
3     OUT o_departing_airport INT,
4     OUT o_arriving_airport INT,
5     OUT o_ticket_price DECIMAL
6 )
7 LANGUAGE plpgsql
8 AS $$
9 BEGIN
10
11     SELECT
12         b.flight_id,
13         f.departing_airport_id,
14         f.arriving_airport_id,
15         b.price
16     INTO
17         o_flight_id,
18         o_departing_airport,
19         o_arriving_airport,
20         o_ticket_price
21     FROM booking b
22     JOIN flights f ON b.flight_id = f.flight_id
23     ORDER BY b.price DESC
24     LIMIT 1;
```

The Data Output tab shows the result of the query:

```
CREATE PROCEDURE
Query returned successfully in 48 msec.
```

The system tray at the bottom right shows the date and time as 02.12.2025.