

The screenshot shows the pgAdmin 4 interface for PostgreSQL version 17. The left sidebar displays the database structure under "Local PostgreSQL / Databases (11)". The main query editor window contains the following SQL code:

```
CREATE OR REPLACE PROCEDURE insert_new_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,
    p_actual_departure DATE,
    p_actual_arrival DATE
)
LANGUAGE plpgsql
AS $$

BEGIN
    INSERT INTO flights(
        flight_id,
        flight_no,
        scheduled_departure,
        scheduled_arrival,

```

The status bar at the bottom indicates "Query returned successfully in 78 msec." and "Total rows: 0". A green message box in the bottom right corner also states "Query returned successfully in 78 msec.".

The screenshot shows the pgAdmin 4 interface with a query editor window open. The left sidebar displays a tree view of database connections and local PostgreSQL databases. The main window has tabs for 'Query' and 'Query History', with 'Query' selected. The query text is a PostgreSQL procedure definition:

```
CREATE OR REPLACE PROCEDURE update_flight_status(
    p_flight_id INT,
    p_new_status VARCHAR
)
LANGUAGE plpgsql
AS $$

BEGIN
    IF NOT EXISTS (SELECT 1 FROM flights WHERE flight_id = p_flight_id) THEN
        RAISE EXCEPTION 'Flight with id % does not exist', p_flight_id;
    END IF;

    UPDATE flights
    SET
        status = p_new_status,
        update_at = CURRENT_DATE
    WHERE flight_id = p_flight_id;

END;
$$;
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab shows the command 'CREATE PROCEDURE' and the message 'Query returned successfully in 78 msec.' The bottom status bar indicates 'Total rows: 0' and 'Query complete 00:00:00.078'. A green notification bar at the bottom right says '✓ Query returned successfully in 78 msec. LF Ln 5, Col 17'.

The screenshot shows the pgAdmin 4 interface for PostgreSQL version 17. The left sidebar displays the database structure under 'Local PostgreSQL' with 11 databases listed. The main window contains a query editor tab titled 'database\_subj/postgres@PostgreSQL 17'. The query being run is:

```
1 CREATE OR REPLACE PROCEDURE list_flights_from_airport(p_airport_id INT)
2 LANGUAGE plpgsql
3 AS $$
4 BEGIN
5     SELECT *
6     FROM flights
7     WHERE departure_airport_id = p_airport_id;
8 END;
9 $$;
```

The results pane at the bottom shows the message 'CREATE PROCEDURE' and 'Query returned successfully in 42 msec.' A status bar at the bottom indicates 'Total rows: 0' and 'Query complete 00:00:00.042'. A green notification bar at the bottom right says '✓ Query returned successfully in 42 msec. LF Ln 9, Col 4'.

The screenshot shows the pgAdmin 4 interface. On the left is the Object Browser tree, which includes a 'PostgreSQL FastAPI (2)' node, a 'Local PostgreSQL' node with 'Databases (11)' expanded, and nodes for 'public.passengers...', 'database\_subj/postgres@PostgreSQL 17\*', 'public.booking/dat...', and 'database\_subj/postgres@PostgreSQL 17\*'. The main area is a query editor with the following SQL code:

```
1 CREATE OR REPLACE FUNCTION average_delay_at_airport(p_airport_id INT)
2 RETURNS NUMERIC
3 LANGUAGE plpgsql
4 AS $$*
5 DECLARE
6     avg_delay NUMERIC;
7 BEGIN
8     SELECT AVG(actual_arrival - scheduled_arrival)
9         INTO avg_delay
10        FROM flights
11       WHERE arrival_airport_id = p_airport_id
12             AND actual_arrival IS NOT NULL;
13
14     RETURN avg_delay;
15
16 $$;
```

Below the query editor, the 'Data Output' tab is selected, showing the command 'CREATE FUNCTION' and the message 'Query returned successfully in 41 msec.'.

Total rows: Query complete 00:00:00.041

✓ Query returned successfully in 41 msec. ×

LF Ln 8, Col 53

The screenshot shows the pgAdmin 4 interface with a database connection titled "database\_subj/postgres@PostgreSQL 17". The left sidebar displays the schema tree for the "public" schema, including tables like passengers, booking, flights, and others. The main query editor window contains the following SQL code:

```
1 CREATE OR REPLACE PROCEDURE list_passengers_by_flight(
2     p_flight_no VARCHAR)
3 LANGUAGE plpgsql
4 AS $$*
5 BEGIN
6     SELECT p.passenger_id,
7            p.first_name,
8            p.last_name,
9            p.date_of_birth,
10           p.gender,
11           p.passport_number
12    FROM passengers p
13   JOIN booking b ON p.passenger_id = b.passenger_id
14   JOIN booking_flight bf ON b.booking_id = bf.booking_id
15   JOIN flights f ON bf.flight_id = f.flight_id
16  WHERE f.flight_no = p_flight_no
17  ORDER BY p.last_name, p.first_name;
18*
19 $$;
```

The "Data Output" tab shows the results of the executed query:

```
CREATE PROCEDURE
```

Query returned successfully in 52 msec.

Total rows: Query complete 00:00:00.052 ✓ Query returned successfully in 52 msec. LF Ln 18, Col 5

The screenshot shows the pgAdmin 4 interface for PostgreSQL 17. The left sidebar displays the database schema, including the 'public' schema which contains various objects like functions, tables, and procedures. The main window shows a query editor with the following SQL code:

```
CREATE OR REPLACE PROCEDURE passengers_top_flights()
LANGUAGE plpgsql
AS $$

BEGIN
    SELECT p.passenger_id,
           p.first_name,
           p.last_name,
           COUNT(bf.flight_id) AS flights_taken
    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 flights_taken DESC
    LIMIT 1;
END;
$$;
```

The 'Execute script' button (F5) is highlighted. Below the query, the output shows:

CREATE PROCEDURE

Query returned successfully in 44 msec.

Total rows: Query complete 00:00:00.044

A green success message at the bottom right corner states: ✓ Query returned successfully in 44 msec. LF Ln 16, Col 4

The screenshot shows the pgAdmin 4 interface with a database connection titled "database\_subj/postgres@PostgreSQL 17". The left sidebar displays the schema tree for the "public" schema, including tables like "airline", "airport", "baggage", "boarder", "bool", "flight", "passenger", "sector", and "trigger". The main pane contains a SQL query editor with the following code:

```
1 CREATE OR REPLACE PROCEDURE flights_delayed_over_24h()
2 LANGUAGE plpgsql
3 AS $$
4 BEGIN
5     SELECT flight_id,
6            flight_no,
7            scheduled_departure,
8            actual_departure,
9            scheduled_arrival,
10           actual_arrival,
11           (actual_arrival - scheduled_arrival) AS delay_days
12      FROM flights
13     WHERE actual_arrival IS NOT NULL
14       AND (actual_arrival - scheduled_arrival) > 1 -- 24 hours
15    ORDER BY delay_days DESC;
16 END;
17 $$;
```

The "Data Output" tab shows the results of the query:

```
CREATE PROCEDURE
```

Query returned successfully in 43 msec.

Total rows: Query complete 00:00:00.043

LF Ln 17, Col 4

The screenshot shows the pgAdmin 4 interface with a database connection to 'database\_subj/postgres@PostgreSQL 17'. The left sidebar displays the schema structure under 'public' with various objects like Aggregates, Collations, Domains, FTS Configurations, FTS Directories, FTS Parameters, FTS Terms, Foreign Tables, Functions, Materials, Operators, Procedures, Sequences, and Tables. The 'Tables' section is currently selected. The main query editor window contains the following PostgreSQL code:

```
1 CREATE OR REPLACE FUNCTION count_flights_per_airline()
2 RETURNS TABLE (
3     airline_id INT,
4     airline_name VARCHAR,
5     flights_count INT
6 )
7 LANGUAGE plpgsql
8 AS $$
9 BEGIN
10    RETURN QUERY
11    SELECT a.airline_id,
12           a.airline_name,
13           COUNT(f.flight_id) AS flights_count
14    FROM airline a
15    LEFT JOIN flights f ON a.airline_id = f.airline_id
16    GROUP BY a.airline_id, a.airline_name
17    ORDER BY flights_count DESC;
18 END;
19 $$;
```

The 'Data Output' tab shows the results of the 'CREATE FUNCTION' command:

```
CREATE FUNCTION
```

Below the results, a message states: "Query returned successfully in 51 msec." A status bar at the bottom indicates "Total rows: 0" and "Query complete 00:00:00.051". A green notification bar at the bottom right also says "Query returned successfully in 51 msec." with a checkmark icon.

The screenshot shows the pgAdmin 4 interface with a database connection named 'database\_subj/postgres@PostgreSQL 17'. The left sidebar displays the schema tree for the 'public' schema, including tables like 'airline', 'airport', 'baggage', 'boarding', 'bool', 'flight', 'passenger', 'sector', and 'trigger'. The main query editor window contains the following SQL code:

```
1 CREATE OR REPLACE PROCEDURE average_ticket_price(p_flight_no VARCHAR)
2 LANGUAGE plpgsql
3 AS $$
4 BEGIN
5     SELECT f.flight_no,
6           AVG(b.price) AS avg_ticket_price
7     FROM flights f
8     JOIN booking_flight bf ON f.flight_id = bf.flight_id
9     JOIN booking b ON bf.booking_id = b.booking_id
10    WHERE f.flight_no = p_flight_no
11    GROUP BY f.flight_no;
12 END;
13 $$;
```

The 'Data Output' tab shows the results of the query execution:

CREATE PROCEDURE

Query returned successfully in 39 msec.

Total rows: Query complete 00:00:00.039

A green status bar at the bottom right indicates: ✓ Query returned successfully in 39 msec. LF Ln 13, Col 4

The screenshot shows the pgAdmin 4 interface with a single database connection named "database\_subj/postgres@PostgreSQL 17". The left sidebar displays the schema browser for the "public" schema, showing various objects like tables, functions, and procedures. The main query editor window contains the following PostgreSQL code:

```
CREATE OR REPLACE PROCEDURE most_expensive_Flight()
LANGUAGE plpgsql
AS $$

BEGIN
    SELECT f.flight_no,
           dep.airport_name AS departure_airport,
           arr.airport_name AS arrival_airport,
           MAX(b.price) AS max_ticket_price
    FROM flights f
    JOIN booking_flight bf ON f.flight_id = bf.flight_id
    JOIN booking b ON bf.booking_id = b.booking_id
    JOIN airport dep ON f.departure_airport_id = dep.airport_id
    JOIN airport arr ON f.arrival_airport_id = arr.airport_id
    GROUP BY f.flight_no, dep.airport_name, arr.airport_name
    ORDER BY max_ticket_price DESC
    LIMIT 1;
END;
$$;
```

The query was executed successfully, as indicated by the message "Query returned successfully in 42 msec." at the bottom of the results pane.