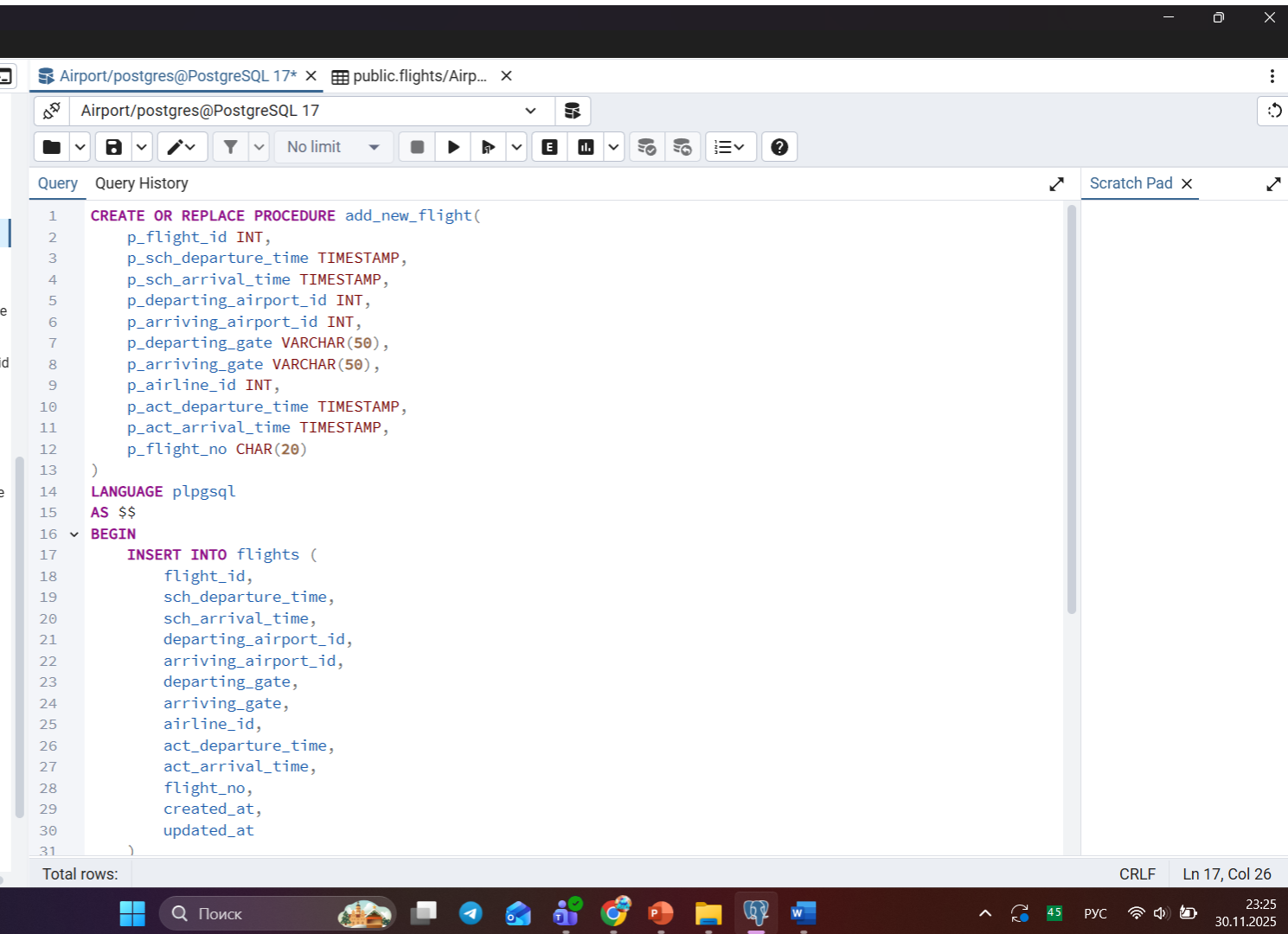


TASK 1



The screenshot shows a PostgreSQL IDE window with the title 'Airport/postgres@PostgreSQL 17*'. The main editor displays a PL/pgSQL procedure named 'add_new_flight'. The procedure has several parameters: 'p_flight_id' (INT), 'p_sch_departure_time' (TIMESTAMP), 'p_sch_arrival_time' (TIMESTAMP), 'p_departing_airport_id' (INT), 'p_arriving_airport_id' (INT), 'p_departing_gate' (VARCHAR(50)), 'p_arriving_gate' (VARCHAR(50)), 'p_airline_id' (INT), 'p_act_departure_time' (TIMESTAMP), 'p_act_arrival_time' (TIMESTAMP), 'p_flight_no' (CHAR(20)). The procedure body starts with an 'INSERT INTO flights' statement listing various columns. The IDE interface includes a toolbar with icons for query execution, a 'Query History' tab, and a 'Scratch Pad' tab. The status bar at the bottom indicates 'Total rows: 0', 'CRLF', and 'Ln 17, Col 26'. The Windows taskbar is visible at the bottom with various application icons and the system clock showing 23:25 on 30.11.2025.

```
1 CREATE OR REPLACE PROCEDURE add_new_flight(  
2     p_flight_id INT,  
3     p_sch_departure_time TIMESTAMP,  
4     p_sch_arrival_time TIMESTAMP,  
5     p_departing_airport_id INT,  
6     p_arriving_airport_id INT,  
7     p_departing_gate VARCHAR(50),  
8     p_arriving_gate VARCHAR(50),  
9     p_airline_id INT,  
10    p_act_departure_time TIMESTAMP,  
11    p_act_arrival_time TIMESTAMP,  
12    p_flight_no CHAR(20)  
13 )  
14 LANGUAGE plpgsql  
15 AS $$  
16 BEGIN  
17     INSERT INTO flights (  
18         flight_id,  
19         sch_departure_time,  
20         sch_arrival_time,  
21         departing_airport_id,  
22         arriving_airport_id,  
23         departing_gate,  
24         arriving_gate,  
25         airline_id,  
26         act_departure_time,  
27         act_arrival_time,  
28         flight_no,  
29         created_at,  
30         updated_at  
31     )
```

```
31     ),  
32     VALUES (  
33         p_flight_id,  
34         p_sch_departure_time,  
35         p_sch_arrival_time,  
36         p_departing_airport_id,  
37         p_arriving_airport_id,  
38         p_departing_gate,  
39         p_arriving_gate,  
40         p_airline_id,  
41         p_act_departure_time,  
42         p_act_arrival_time,  
43         p_flight_no,  
44         CURRENT_TIMESTAMP,  
45         CURRENT_TIMESTAMP  
46     );  
47 END;  
48 $$;
```

TASK 2

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows a tree structure of database objects. The 'flights' table is selected, and its columns are listed: flight_id, sch_departure_time, sch_arrival_time, departing_airport_id, arriving_airport_id, departing_gate, arriving_gate, airline_id, act_departure_time, act_arrival_time, created_at, updated_at, and flight_no. The main pane shows a SQL query being executed in the 'Airport/postgres@PostgreSQL 17*' database. The query is a PL/pgSQL procedure named 'update_flight_arrival' that updates the 'act_arrival_time' and 'updated_at' fields of the 'flights' table for a given 'flight_id'. The query is executed successfully, returning a message: 'Query returned successfully in 51 msec.' The bottom status bar indicates 'Total rows: Query complete 00:00:00.051' and 'Ln 14, Col 5'.

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- > baggage
- > baggage_check
- > boarding_pass
- > booking
- > booking_flight
- > flights
 - Columns (13)
 - flight_id
 - sch_departure_time
 - sch_arrival_time
 - departing_airport_id
 - arriving_airport_id
 - departing_gate
 - arriving_gate
 - airline_id
 - act_departure_time
 - act_arrival_time
 - created_at
 - updated_at
 - flight_no
 - > Constraints
 - > Indexes
 - > RLS Policies
 - > Rules
 - > Triggers
 - > passengers
 - > security_check
 - > Trigger Functions
 - > Types
 - > Views

Airport/postgres@PostgreSQL 17* × public.flights/Airp... ×

Airport/postgres@PostgreSQL 17

No limit

Query Query History

```
1 CREATE OR REPLACE PROCEDURE update_flight_arrival(  
2     p_flight_id INT,  
3     p_act_arrival_time TIMESTAMP  
4 )  
5 LANGUAGE plpgsql  
6 AS $$  
7 BEGIN  
8     UPDATE flights  
9     SET  
10         act_arrival_time = p_act_arrival_time,  
11         updated_at = CURRENT_TIMESTAMP  
12     WHERE flight_id = p_flight_id;  
13 END;  
14 $$;
```

Scratch Pad ×

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 51 msec.

Total rows: Query complete 00:00:00.051 CRLF Ln 14, Col 5

23:35 30.11.2025

TASK 3

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- > baggage
- > baggage_check
- > boarding_pass
- > booking
- > booking_flight
- > flights
 - Columns (13)
 - flight_id
 - sch_departure_time
 - sch_arrival_time
 - departing_airport_id
 - arriving_airport_id
 - departing_gate
 - arriving_gate
 - airline_id
 - act_departure_time
 - act_arrival_time
 - created_at
 - updated_at
 - flight_no
 - > Constraints
 - > Indexes
 - > RLS Policies
 - > Rules
 - > Triggers
- > passengers
- > security_check
- > Trigger Functions
- > Types
- > Views

Airport/postgres@PostgreSQL 17* x public.flights/Airp... x

Airport/postgres@PostgreSQL 17

Query Query History

```
1 CREATE OR REPLACE PROCEDURE get_departing_flights(  
2     p_departure_airport_id INT  
3 )  
4 LANGUAGE plpgsql  
5 AS $$  
6 BEGIN  
7     SELECT  
8         flight_id,  
9         sch_departure_time,  
10        sch_arrival_time,  
11        departing_airport_id,  
12        arriving_airport_id,  
13        departing_gate,  
14        arriving_gate,  
15        airline_id,  
16        act_departure_time,  
17        act_arrival_time,  
18        created_at,  
19        updated_at  
20 FROM flights  
21 WHERE departing_airport_id = p_departure_airport_id;  
22 END;  
23 $$;
```

Scratch Pad x

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 51 msec.

Total rows: Query complete 00:00:00.051

✓ Query returned successfully in 51 ms

CRLF Ln 23, Col 4



Поиск



TASK 4

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- baggage
- baggage_check
- boarding_pass
- booking
- booking_flight
- flights
 - Columns (13)
 - flight_id
 - sch_departure_time
 - sch_arrival_time
 - departing_airport_id
 - arriving_airport_id
 - departing_gate
 - arriving_gate
 - airline_id
 - act_departure_time
 - act_arrival_time
 - created_at
 - updated_at
 - flight_no
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- passengers
- security_check
- Trigger Functions
- Types
- Views

Airport/postgres@PostgreSQL 17* x public.flights/Airp... x

Airport/postgres@PostgreSQL 17

Query Query History

```
1 CREATE OR REPLACE FUNCTION calculate_average_delay(  
2     p_arrival_airport_id INT  
3 )  
4 RETURNS INTERVAL  
5 LANGUAGE plpgsql  
6 AS $$  
7 DECLARE  
8     avg_delay INTERVAL;  
9 BEGIN  
10    SELECT AVG(act_arrival_time - sch_arrival_time)  
11    INTO avg_delay  
12    FROM flights  
13    WHERE arriving_airport_id = p_arrival_airport_id  
14           AND act_arrival_time IS NOT NULL  
15           AND sch_arrival_time IS NOT NULL;  
16  
17    RETURN avg_delay;  
18 END;  
19 $$;
```

Scratch Pad x

Data Output Messages Notifications

CREATE FUNCTION

Query returned successfully in 45 msec.

Total rows: Query complete 00:00:00.045

Ln 19, Col 4

✓ Query returned successfully in 45 msec. x



Поиск



23:43

30.11.2025

TASK 5

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- > baggage
- > baggage_check
- > boarding_pass
- > booking
- > booking_flight
- > flights
 - Columns (13)
 - flight_id
 - sch_departure_time
 - sch_arrival_time
 - departing_airport_id
 - arriving_airport_id
 - departing_gate
 - arriving_gate
 - airline_id
 - act_departure_time
 - act_arrival_time
 - created_at
 - updated_at
 - flight_no
 - > Constraints
 - > Indexes
 - > RLS Policies
 - > Rules
 - > Triggers
- > passengers
- > security_check
- > Trigger Functions
- > Types
- > Views

Airport/postgres@PostgreSQL 17* x public.flights/Airp... x

Airport/postgres@PostgreSQL 17

Query Query History

```
1 CREATE OR REPLACE PROCEDURE get_passengers_by_flight(  
2     p_flight_no CHAR(20)  
3 )  
4 LANGUAGE plpgsql  
5 AS $$  
6 BEGIN  
7     SELECT  
8         p.passenger_id,  
9         p.first_name,  
10        p.last_name,  
11        p.passport_number,  
12        b.booking_id,  
13        b.seat,  
14        b.status  
15 FROM passengers p  
16 JOIN booking b ON p.passenger_id = b.passenger_id  
17 JOIN flights f ON b.flight_id = f.flight_id  
18 WHERE f.flight_no = p_flight_no;  
19 END;  
20 $$;
```

Scratch Pad x

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 55 msec.

Total rows: Query complete 00:00:00.055

✓ Query returned successfully in 55 msec. x

CRLF Ln 20, Col 4



Поиск



CRLF Ln 20, Col 4

23:47
30.11.2025

TASK 6

The screenshot displays the pgAdmin 4 web interface. On the left, the Object Explorer shows a tree structure of database objects, with the 'flights' table selected under the 'public' schema. The main query editor window shows a SQL query to create or replace a procedure named 'find_most_frequent_flyer()'. The query is as follows:

```
1 CREATE OR REPLACE PROCEDURE find_most_frequent_flyer()
2 LANGUAGE plpgsql
3 AS $$
4 BEGIN
5     SELECT
6         p.passenger_id,
7         p.first_name,
8         p.last_name,
9         p.passport_number,
10        COUNT(b.booking_id) AS total_flights
11    FROM passengers p
12   JOIN booking b ON p.passenger_id = b.passenger_id
13  GROUP BY p.passenger_id, p.first_name, p.last_name, p.passport_number
14  ORDER BY total_flights DESC
15  LIMIT 1;
16 END;
17 $$;
```

Below the query editor, the Messages tab shows the execution status: 'Query returned successfully in 54 msec.' A green notification box at the bottom right confirms this: '✓ Query returned successfully in 54 msec. X'. The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.054' and 'Ln 17, Col 4'.

TASK 7

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- baggage
- baggage_check
- boarding_pass
- booking
- booking_flight
- flights
 - Columns (13)
 - flight_id
 - sch_departure_time
 - sch_arrival_time
 - departing_airport_id
 - arriving_airport_id
 - departing_gate
 - arriving_gate
 - airline_id
 - act_departure_time
 - act_arrival_time
 - created_at
 - updated_at
 - flight_no
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- passengers
- security_check
- Trigger Functions
- Types
- Views

Airport/postgres@PostgreSQL 17* x public.flights/Air...

Airport/postgres@PostgreSQL 17

No limit

Query Query History

```
1 CREATE OR REPLACE PROCEDURE find_flights_delayed_over_24h()
2 LANGUAGE plpgsql
3 AS $$
4 DECLARE
5     flight_record RECORD;
6 BEGIN
7     FOR flight_record IN
8     SELECT
9         f.flight_id,
10        f.flight_no,
11        f.sch_departure_time,
12        f.act_departure_time,
13        (f.act_departure_time - f.sch_departure_time) AS delay_interval
14    FROM flights f
15    WHERE (f.act_departure_time - f.sch_departure_time) > INTERVAL '24 hours'
16 LOOP
17     RAISE NOTICE 'Flight ID: %, Number: %, Scheduled: %, Actual: %, Delay: %',
18         flight_record.flight_id,
19         flight_record.flight_no,
20         flight_record.sch_departure_time,
21         flight_record.act_departure_time,
22         flight_record.delay_interval;
23 END LOOP;
24 END;
25 $$;
```

Scratch Pad x

Data Output Messages Notifications

CREATE PROCEDURE

Query returned successfully in 51 msec.

Total rows: Query complete 00:00:00.051

CRLF Ln 25, Col 4



Поиск



TASK 8

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- > baggage
- > baggage_check
- > boarding_pass
- > booking
- > booking_flight
- > flights
 - Columns (13)
 - flight_id
 - sch_departure_time
 - sch_arrival_time
 - departing_airport_id
 - arriving_airport_id
 - departing_gate
 - arriving_gate
 - airline_id
 - act_departure_time
 - act_arrival_time
 - created_at
 - updated_at
 - flight_no
 - > Constraints
 - > Indexes
 - > RLS Policies
 - > Rules
 - > Triggers
 - > passengers
 - > security_check
 - > Trigger Functions
 - > Types
 - > Views

Airport/postgres@PostgreSQL 17* × public.flights/Airp... ×

Airport/postgres@PostgreSQL 17

Query Query History

```
1 CREATE OR REPLACE FUNCTION count_flights_per_airline_simple()
2 RETURNS TABLE(
3     airline_id INT,
4     flight_count BIGINT
5 )
6 LANGUAGE plpgsql
7 AS $$
8 BEGIN
9     RETURN QUERY
10    SELECT
11        f.airline_id,
12        COUNT(f.flight_id) AS flight_count
13    FROM flights f
14    GROUP BY f.airline_id
15    ORDER BY flight_count DESC;
16 END;
17 $$;
```

Scratch Pad ×

Data Output Messages Notifications

CREATE FUNCTION

Query returned successfully in 52 msec.

✓ Query returned successfully in 52 msec. ✕

Total rows: Query complete 00:00:00.052 CRLF Ln 17, Col 4

00:14 01.12.2025

TASK 9

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' shows a database structure with tables like 'baggage', 'boarding_pass', 'booking', and 'flights'. The 'flights' table is selected, showing its columns: flight_id, sch_departure_time, sch_arrival_time, departing_airport_id, arriving_airport_id, departing_gate, arriving_gate, airline_id, act_departure_time, act_arrival_time, created_at, updated_at, and flight_no. The main pane shows a SQL query being executed in the 'Airport/postgres@PostgreSQL 17' database. The query is a PL/pgSQL procedure named 'calculate_avg_ticket_price' that calculates the average ticket price for a given flight. The query is as follows:

```
1 CREATE OR REPLACE PROCEDURE calculate_avg_ticket_price(  
2     p_flight_id INT,  
3     OUT p_avg_price DECIMAL,  
4     OUT p_flight_no CHAR(20)  
5 )  
6 LANGUAGE plpgsql  
7 AS $$  
8 BEGIN  
9     SELECT  
10        AVG(b.ticket_price),  
11        f.flight_no  
12    INTO  
13        p_avg_price,  
14        p_flight_no  
15    FROM booking b  
16    JOIN flights f ON b.flight_id = f.flight_id  
17    WHERE b.flight_id = p_flight_id  
18    GROUP BY f.flight_no;  
19 END;  
20 $$;
```

The 'Data Output' tab at the bottom shows the execution result: 'CREATE PROCEDURE' and 'Query returned successfully in 48 msec.' The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.048' and 'Ln 20, Col 4'.

TASK 10

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' shows a tree structure of database objects. The 'flights' table is selected, and its columns are listed: flight_id, sch_departure_time, sch_arrival_time, departing_airport_id, arriving_airport_id, departing_gate, arriving_gate, airline_id, act_departure_time, act_arrival_time, created_at, updated_at, and flight_no. The main pane shows a SQL query editor with the following code:

```
1 CREATE OR REPLACE PROCEDURE find_most_expensive_flight(  
2     OUT p_flight_no CHAR(20),  
3     OUT p_departure_airport VARCHAR(50),  
4     OUT p_arrival_airport VARCHAR(50),  
5     OUT p_ticket_price DECIMAL  
6 )  
7 LANGUAGE plpgsql  
8 AS $$  
9 BEGIN  
10     SELECT  
11         f.flight_no,  
12         dep.airport_name,  
13         arr.airport_name,  
14         b.ticket_price  
15     INTO  
16         p_flight_no,  
17         p_departure_airport,  
18         p_arrival_airport,  
19         p_ticket_price  
20     FROM booking b  
21     JOIN flights f ON b.flight_id = f.flight_id  
22     JOIN airport dep ON f.departing_airport_id = dep.airport_id  
23     JOIN airport arr ON f.arriving_airport_id = arr.airport_id  
24     ORDER BY b.ticket_price DESC  
25     LIMIT 1;  
26 END;  
27 $$;
```

The bottom status bar indicates 'Total rows: Query complete 00:00:00.048' and 'Ln 28, Col 1'.



Поиск

