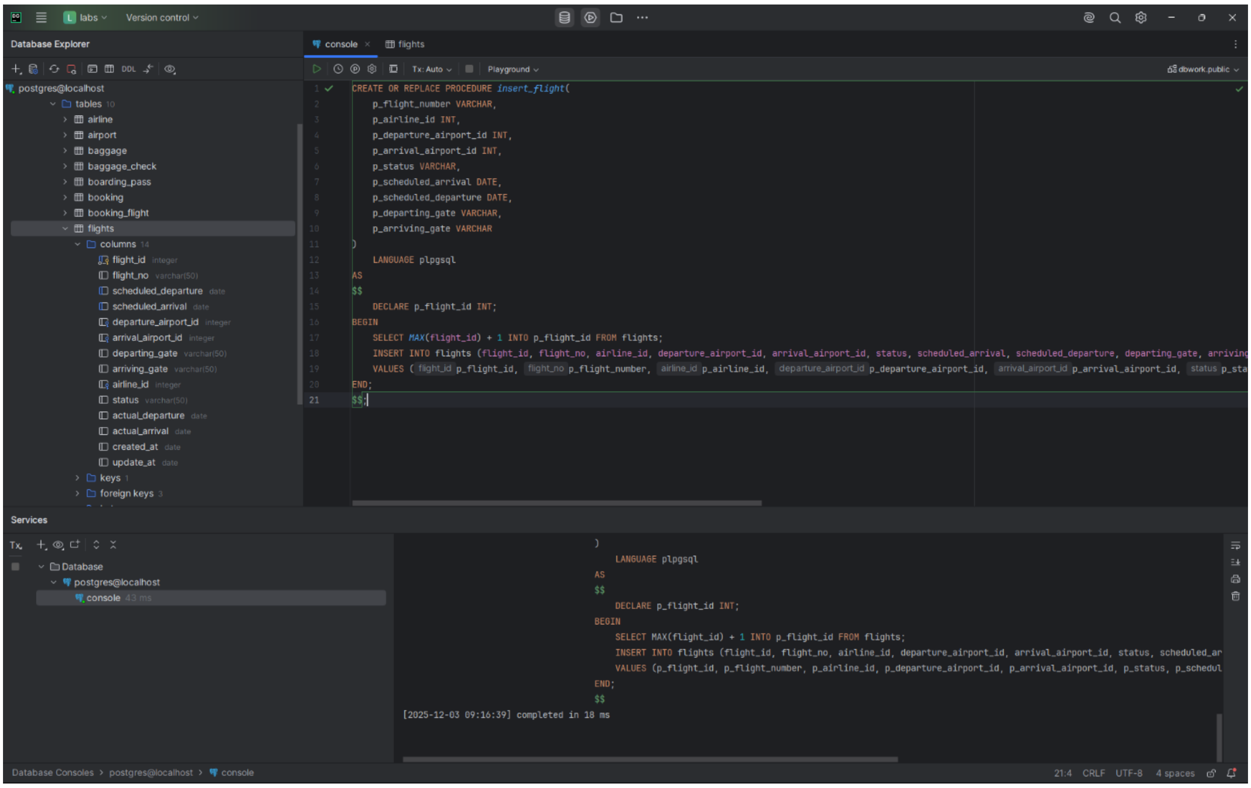
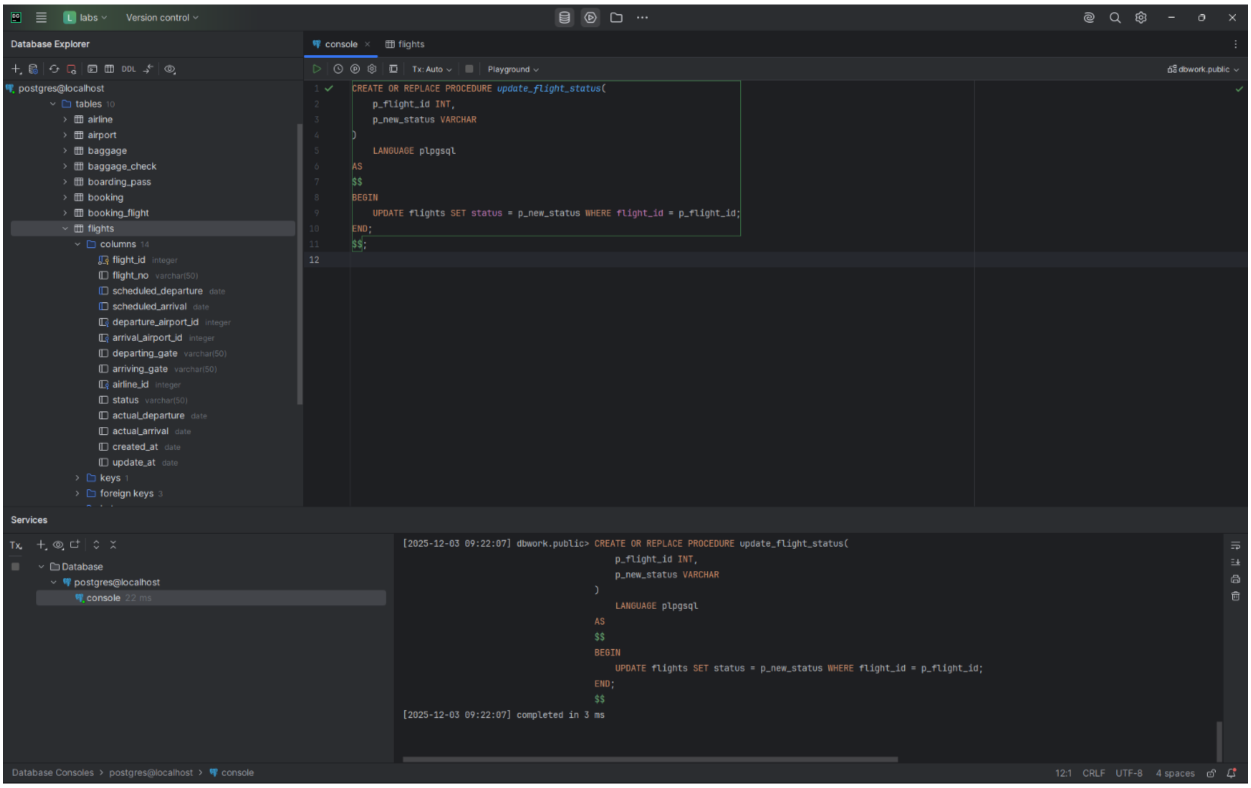
1_



```sql
CREATE OR REPLACE PROCEDURE insert_flight(
    p_flight_number VARCHAR,
    p_airline_id INT,
    p_departure_airport_id INT,
    p_arrival_airport_id INT,
    p_status VARCHAR,
    p_scheduled_arrival DATE,
    p_scheduled_departure DATE,
    p_departing_gate VARCHAR,
    p_arriving_gate VARCHAR
)
    LANGUAGE plpgsql
AS
$$
    DECLARE p_flight_id INT;
BEGIN
    SELECT MAX(flight_id) + 1 INTO p_flight_id FROM flights;
    INSERT INTO flights (flight_id, flight_no, airline_id, departure_airport_id, arrival_airport_id, status, scheduled_arrival, scheduled_departure, departing_gate, arriving
    VALUES ( p_flight_id, p_flight_number, p_airline_id, p_departure_airport_id, p_arrival_airport_id, p_sta
END;
$$;
```

```
                                )
                                    LANGUAGE plpgsql
                                AS
                                $$
                                    DECLARE p_flight_id INT;
                                BEGIN
                                    SELECT MAX(flight_id) + 1 INTO p_flight_id FROM flights;
                                    INSERT INTO flights (flight_id, flight_no, airline_id, departure_airport_id, arrival_airport_id, status, scheduled_ar
                                    VALUES (p_flight_id, p_flight_number, p_airline_id, p_departure_airport_id, p_arrival_airport_id, p_status, p_schedul
                                END;
                                $$
[2025-12-03 09:16:39] completed in 18 ms
```

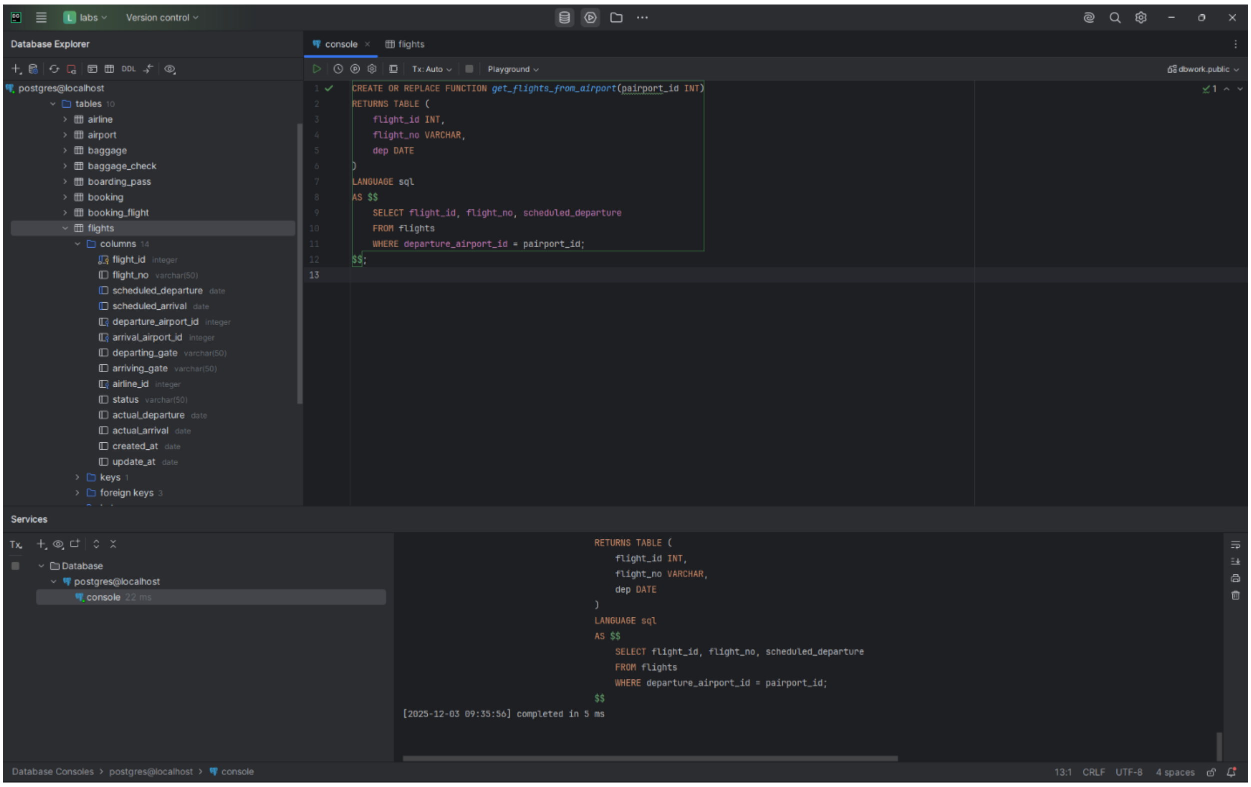
2_



```sql
CREATE OR REPLACE PROCEDURE update_flight_status(
    p_flight_id INT,
    p_new_status VARCHAR
)
    LANGUAGE plpgsql
AS
$$
BEGIN
    UPDATE flights SET status = p_new_status WHERE flight_id = p_flight_id;
END;
$$;
```

```
[2025-12-03 09:22:07] dbwork.public> CREATE OR REPLACE PROCEDURE update_flight_status(
                                         p_flight_id INT,
                                         p_new_status VARCHAR
                                     )
                                         LANGUAGE plpgsql
                                     AS
                                     $$
                                     BEGIN
                                         UPDATE flights SET status = p_new_status WHERE flight_id = p_flight_id;
                                     END;
                                     $$
[2025-12-03 09:22:07] completed in 3 ms
```

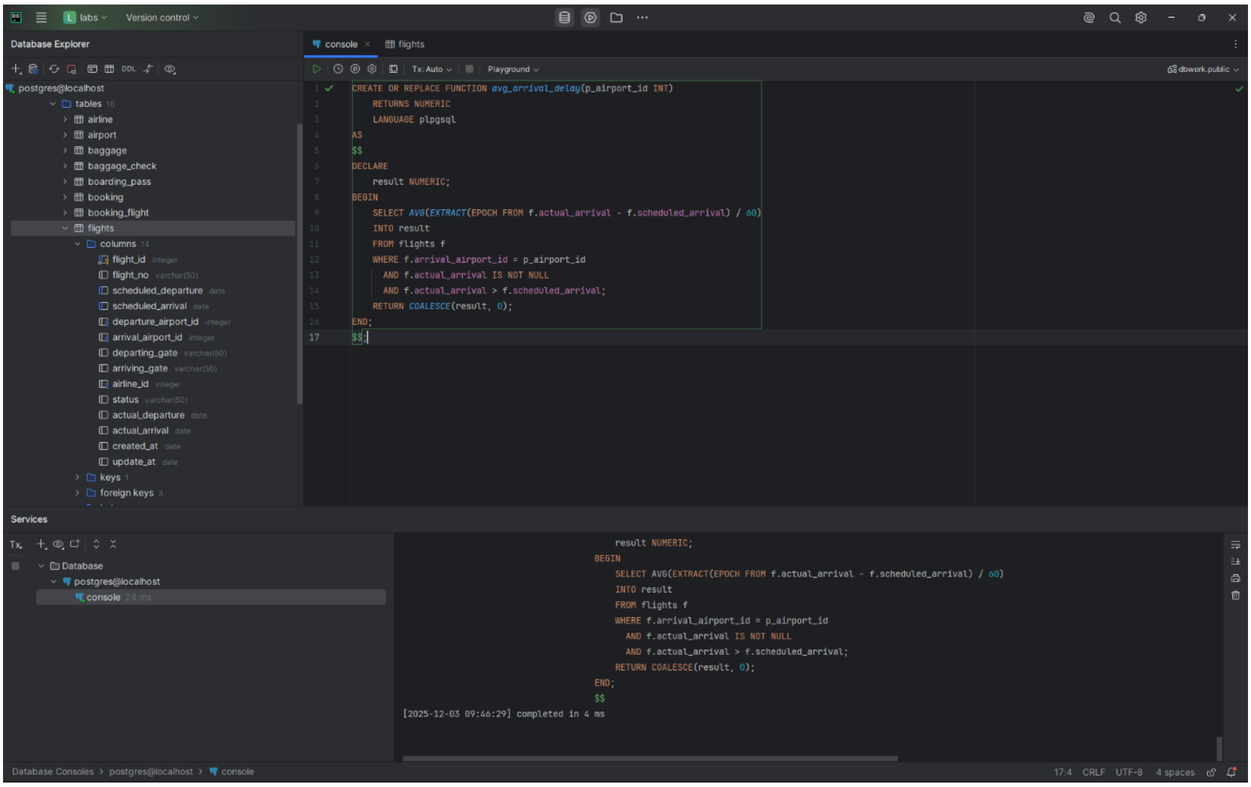
3_



```sql
CREATE OR REPLACE FUNCTION get_flights_from_airport(pairport_id INT)
RETURNS TABLE (
    flight_id INT,
    flight_no VARCHAR,
    dep DATE
)
LANGUAGE sql
AS $$
    SELECT flight_id, flight_no, scheduled_departure
    FROM flights
    WHERE departure_airport_id = pairport_id;
$$;
```

```
RETURNS TABLE (
    flight_id INT,
    flight_no VARCHAR,
    dep DATE
)
LANGUAGE sql
AS $$
    SELECT flight_id, flight_no, scheduled_departure
    FROM flights
    WHERE departure_airport_id = pairport_id;
$$
[2025-12-03 09:35:56] completed in 5 ms
```

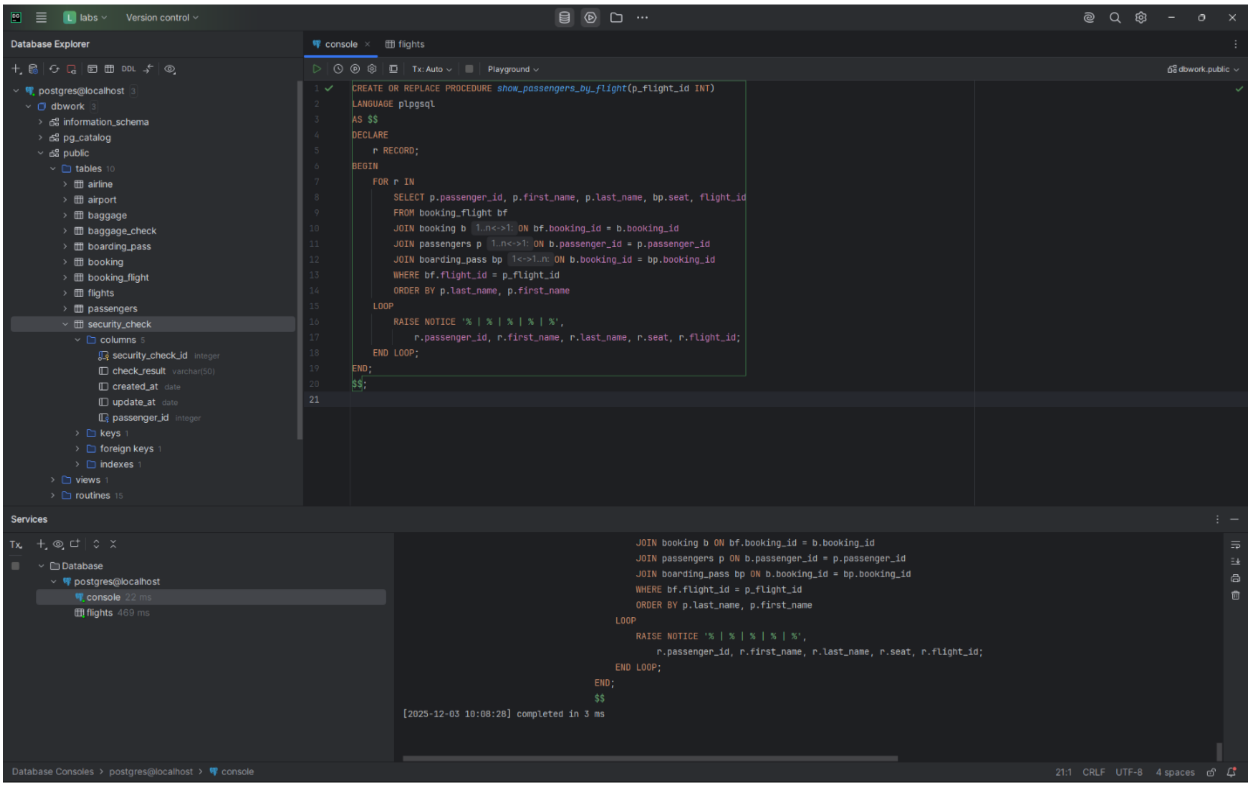
4_



```sql
CREATE OR REPLACE FUNCTION avg_arrival_delay(p_airport_id INT)
    RETURNS NUMERIC
    LANGUAGE plpgsql
AS
$$
DECLARE
    result NUMERIC;
BEGIN
    SELECT AVG(EXTRACT(EPOCH FROM f.actual_arrival - f.scheduled_arrival) / 60)
    INTO result
    FROM flights f
    WHERE f.arrival_airport_id = p_airport_id
      AND f.actual_arrival IS NOT NULL
      AND f.actual_arrival > f.scheduled_arrival;
    RETURN COALESCE(result, 0);
END;
$$;
```

```
    result NUMERIC;
BEGIN
    SELECT AVG(EXTRACT(EPOCH FROM f.actual_arrival - f.scheduled_arrival) / 60)
    INTO result
    FROM flights f
    WHERE f.arrival_airport_id = p_airport_id
      AND f.actual_arrival IS NOT NULL
      AND f.actual_arrival > f.scheduled_arrival;
    RETURN COALESCE(result, 0);
END;
$$
[2025-12-03 09:46:29] completed in 4 ms
```

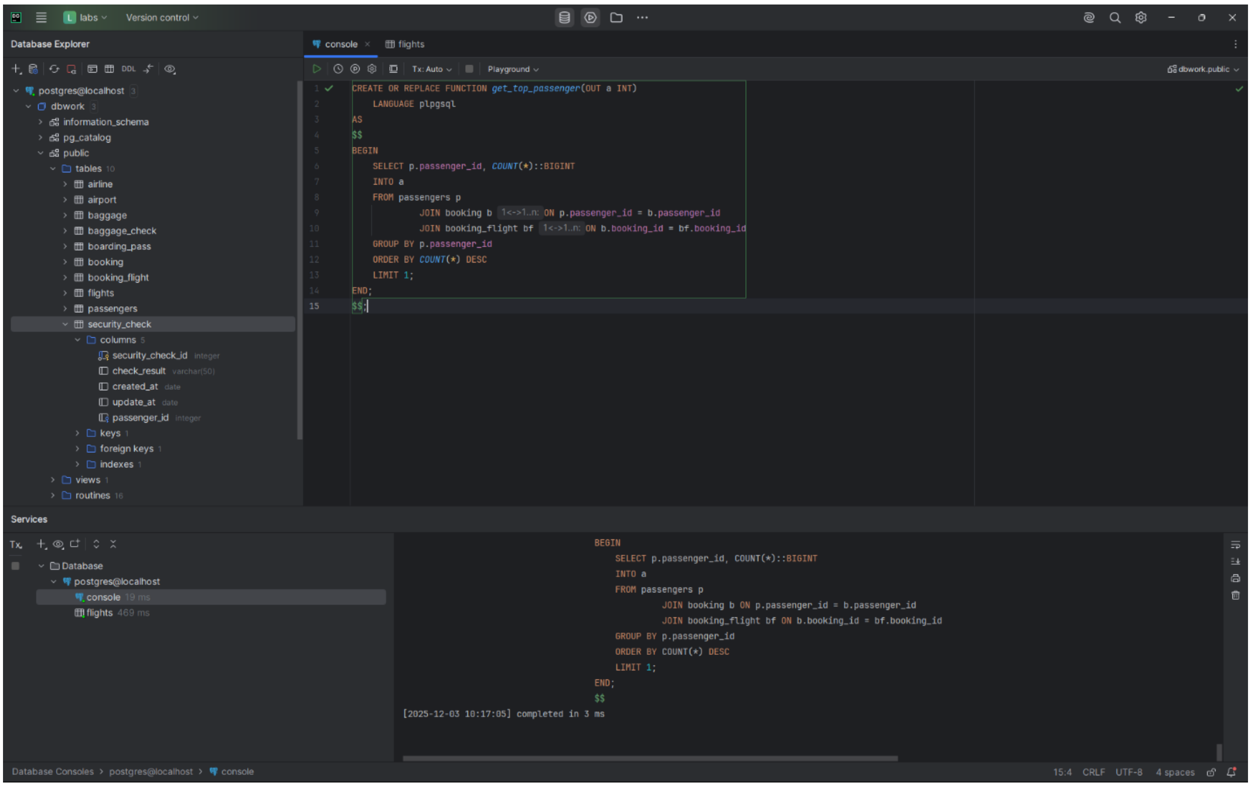
5_



```sql
CREATE OR REPLACE PROCEDURE show_passengers_by_flight(p_flight_id INT)
LANGUAGE plpgsql
AS $$
DECLARE
    r RECORD;
BEGIN
    FOR r IN
        SELECT p.passenger_id, p.first_name, p.last_name, bp.seat, flight_id
        FROM booking_flight bf
        JOIN booking b ON bf.booking_id = b.booking_id
        JOIN passengers p ON b.passenger_id = p.passenger_id
        JOIN boarding_pass bp ON b.booking_id = bp.booking_id
        WHERE bf.flight_id = p_flight_id
        ORDER BY p.last_name, p.first_name
    LOOP
        RAISE NOTICE '% | % | % | % | %',
            r.passenger_id, r.first_name, r.last_name, r.seat, r.flight_id;
    END LOOP;
END;
$$;
```

```
        JOIN booking b ON bf.booking_id = b.booking_id
        JOIN passengers p ON b.passenger_id = p.passenger_id
        JOIN boarding_pass bp ON b.booking_id = bp.booking_id
        WHERE bf.flight_id = p_flight_id
        ORDER BY p.last_name, p.first_name
    LOOP
        RAISE NOTICE '% | % | % | % | %',
            r.passenger_id, r.first_name, r.last_name, r.seat, r.flight_id;
    END LOOP;
END;
$$
[2025-12-03 10:08:28] completed in 3 ms
```

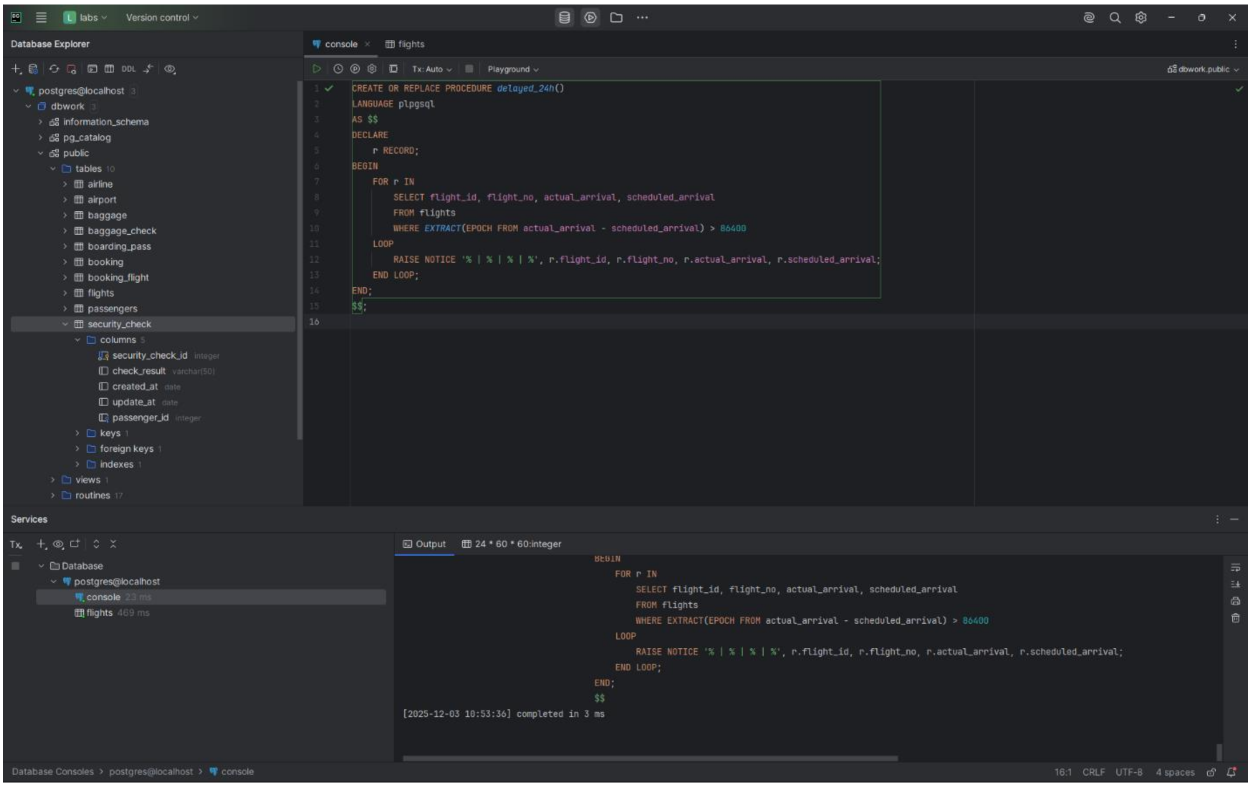
6_



```sql
CREATE OR REPLACE FUNCTION get_top_passenger(OUT a INT)
    LANGUAGE plpgsql
AS
$$
BEGIN
    SELECT p.passenger_id, COUNT(*)::BIGINT
    INTO a
    FROM passengers p
            JOIN booking b ON p.passenger_id = b.passenger_id
            JOIN booking_flight bf ON b.booking_id = bf.booking_id
    GROUP BY p.passenger_id
    ORDER BY COUNT(*) DESC
    LIMIT 1;
END;
$$;
```

```
BEGIN
    SELECT p.passenger_id, COUNT(*)::BIGINT
    INTO a
    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
    ORDER BY COUNT(*) DESC
    LIMIT 1;
END;
$$
[2025-12-03 10:17:05] completed in 3 ms
```

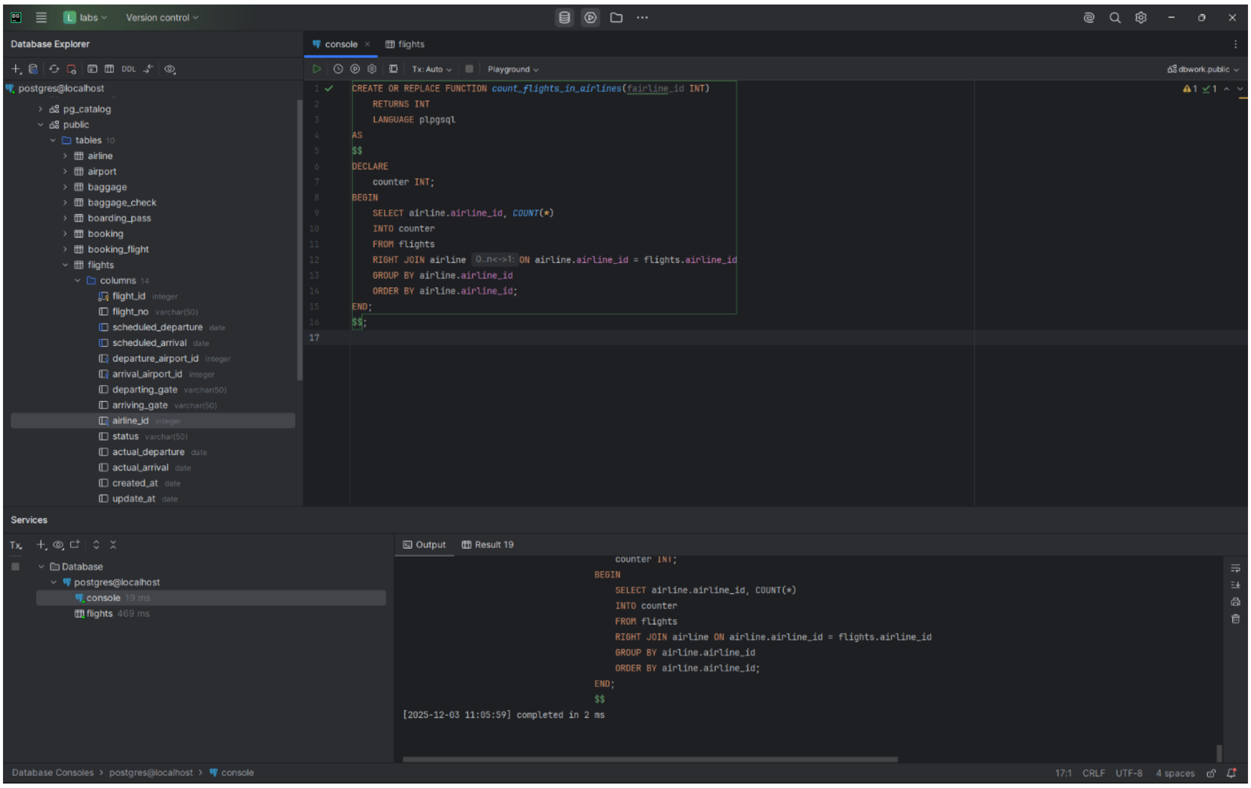
7_



```
CREATE OR REPLACE PROCEDURE delayed_24h()
LANGUAGE plpgsql
AS $$
DECLARE
    r RECORD;
BEGIN
    FOR r IN
        SELECT flight_id, flight_no, actual_arrival, scheduled_arrival
        FROM flights
        WHERE EXTRACT(EPOCH FROM actual_arrival - scheduled_arrival) > 86400
    LOOP
        RAISE NOTICE '% | % | % | %', r.flight_id, r.flight_no, r.actual_arrival, r.scheduled_arrival;
    END LOOP;
END;
$$;
```

```
BEGIN
    FOR r IN
        SELECT flight_id, flight_no, actual_arrival, scheduled_arrival
        FROM flights
        WHERE EXTRACT(EPOCH FROM actual_arrival - scheduled_arrival) > 86400
    LOOP
        RAISE NOTICE '% | % | % | %', r.flight_id, r.flight_no, r.actual_arrival, r.scheduled_arrival;
    END LOOP;
END;
$$
[2025-12-03 10:53:36] completed in 3 ms
```

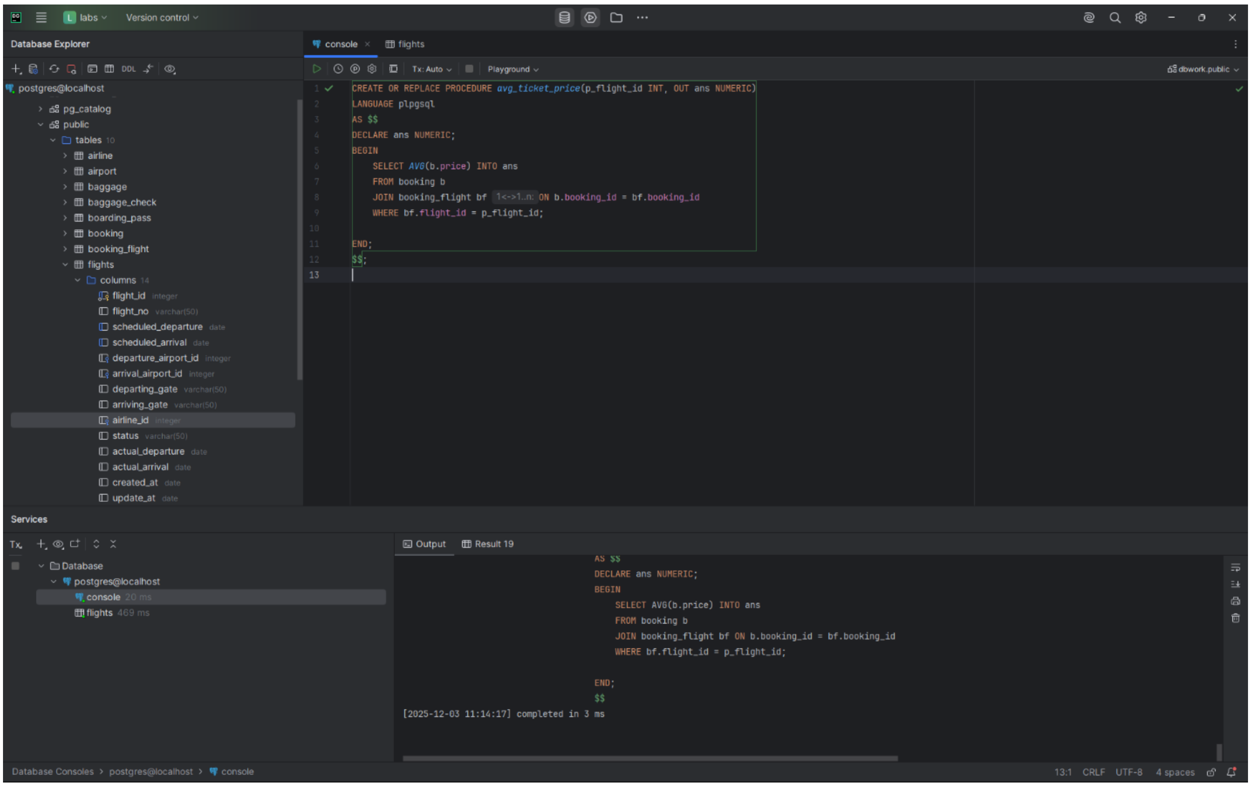
8_



```
CREATE OR REPLACE FUNCTION count_flights_in_airlines(fairline_id INT)
    RETURNS INT
    LANGUAGE plpgsql
AS
$$
DECLARE
    counter INT;
BEGIN
    SELECT airline.airline_id, COUNT(*)
    INTO counter
    FROM flights
    RIGHT JOIN airline 0..n<->1: ON airline.airline_id = flights.airline_id
    GROUP BY airline.airline_id
    ORDER BY airline.airline_id;
END;
$$;
```

```
    counter INT;
BEGIN
    SELECT airline.airline_id, COUNT(*)
    INTO counter
    FROM flights
    RIGHT JOIN airline ON airline.airline_id = flights.airline_id
    GROUP BY airline.airline_id
    ORDER BY airline.airline_id;
END;
$$
[2025-12-03 11:05:59] completed in 2 ms
```

9_



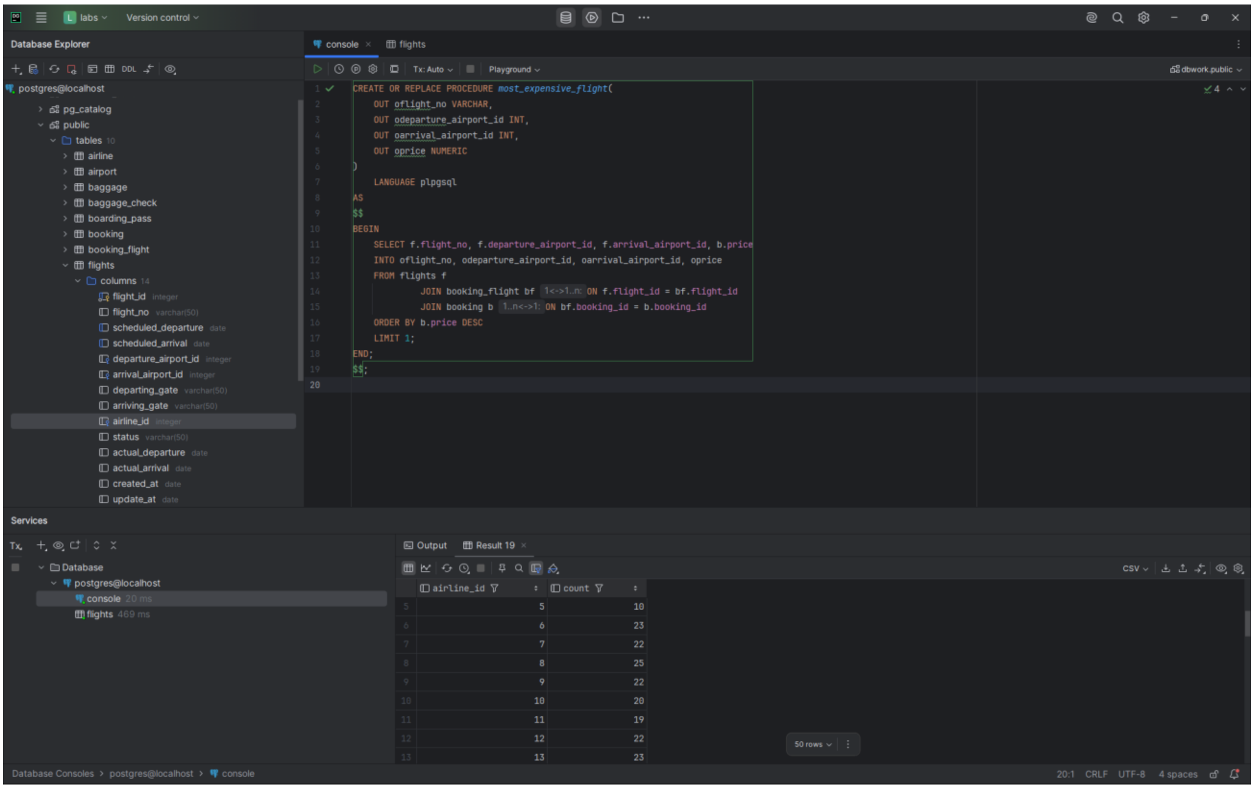
```sql
CREATE OR REPLACE PROCEDURE avg_ticket_price(p_flight_id INT, OUT ans NUMERIC)
LANGUAGE plpgsql
AS $$
DECLARE ans NUMERIC;
BEGIN
    SELECT AVG(b.price) INTO ans
    FROM booking b
    JOIN booking_flight bf ON b.booking_id = bf.booking_id
    WHERE bf.flight_id = p_flight_id;

END;
$$;
```

```
AS $$
DECLARE ans NUMERIC;
BEGIN
    SELECT AVG(b.price) INTO ans
    FROM booking b
    JOIN booking_flight bf ON b.booking_id = bf.booking_id
    WHERE bf.flight_id = p_flight_id;

END;
$$
[2025-12-03 11:14:17] completed in 3 ms
```

10_



```sql
CREATE OR REPLACE PROCEDURE most_expensive_flight(
    OUT oflight_no VARCHAR,
    OUT odeparture_airport_id INT,
    OUT oarrival_airport_id INT,
    OUT oprice NUMERIC
)
    LANGUAGE plpgsql
AS
$$
BEGIN
    SELECT f.flight_no, f.departure_airport_id, f.arrival_airport_id, b.price
    INTO oflight_no, odeparture_airport_id, oarrival_airport_id, oprice
    FROM flights f
            JOIN booking_flight bf ON f.flight_id = bf.flight_id
            JOIN booking b ON bf.booking_id = b.booking_id
    ORDER BY b.price DESC
    LIMIT 1;
END;
$$;
```

	airline_id	count
5	5	10
6	6	23
7	7	22
8	8	25
9	9	22
10	10	20
11	11	19
12	12	22
13	13	23