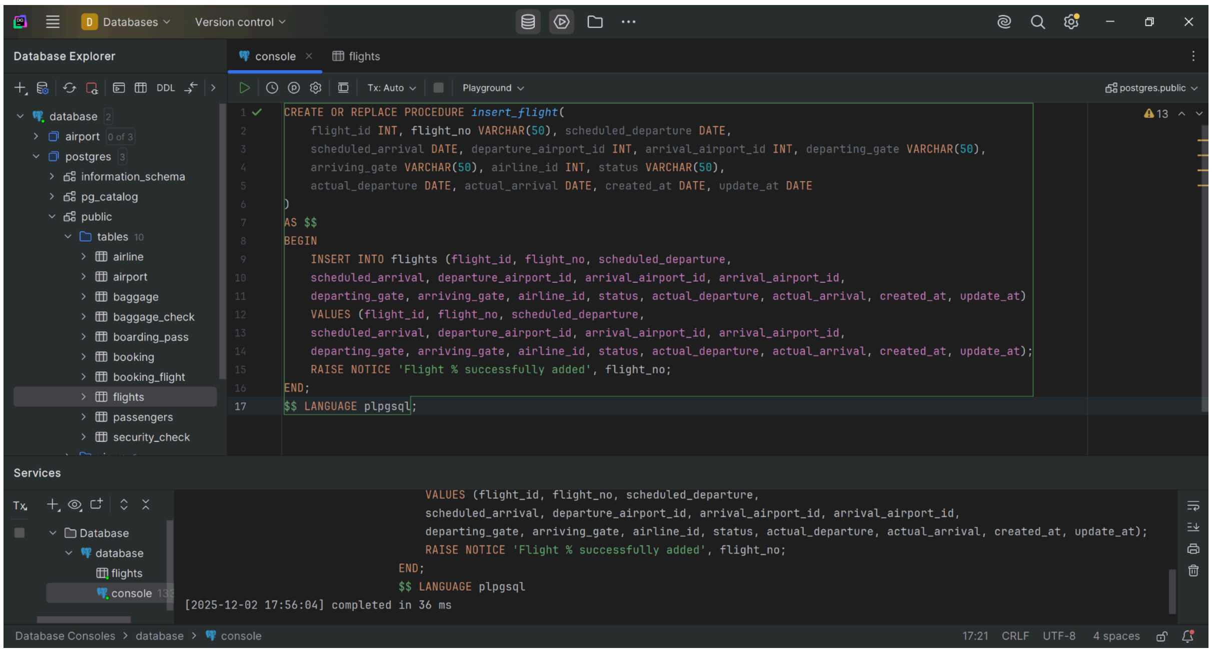
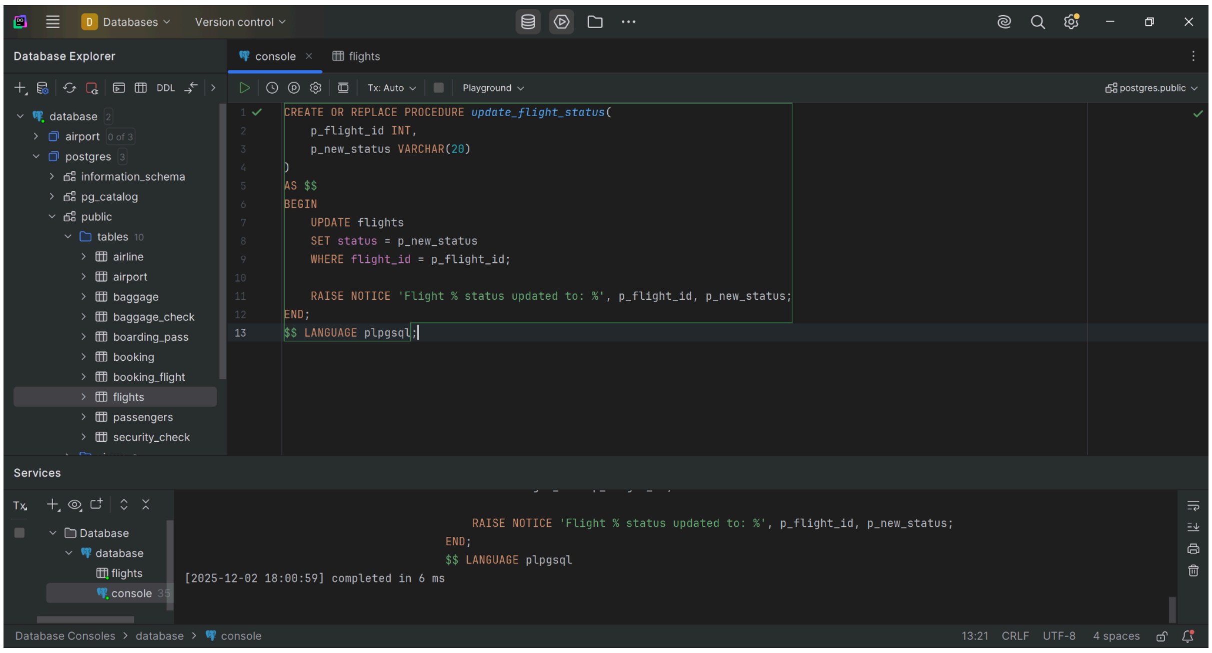
1)



```sql
CREATE OR REPLACE PROCEDURE insert_flight(
    flight_id INT, flight_no VARCHAR(50), scheduled_departure DATE,
    scheduled_arrival DATE, departure_airport_id INT, arrival_airport_id INT, departing_gate VARCHAR(50),
    arriving_gate VARCHAR(50), airline_id INT, status VARCHAR(50),
    actual_departure DATE, actual_arrival DATE, created_at DATE, update_at DATE
)
AS $$
BEGIN
    INSERT INTO flights (flight_id, flight_no, scheduled_departure,
    scheduled_arrival, departure_airport_id, arrival_airport_id, arrival_airport_id,
    departing_gate, arriving_gate, airline_id, status, actual_departure, actual_arrival, created_at, update_at)
    VALUES (flight_id, flight_no, scheduled_departure,
    scheduled_arrival, departure_airport_id, arrival_airport_id, arrival_airport_id,
    departing_gate, arriving_gate, airline_id, status, actual_departure, actual_arrival, created_at, update_at);
    RAISE NOTICE 'Flight % successfully added', flight_no;
END;
$$ LANGUAGE plpgsql;
```

```
                    VALUES (flight_id, flight_no, scheduled_departure,
                    scheduled_arrival, departure_airport_id, arrival_airport_id, arrival_airport_id,
                    departing_gate, arriving_gate, airline_id, status, actual_departure, actual_arrival, created_at, update_at);
                    RAISE NOTICE 'Flight % successfully added', flight_no;
                END;
                $$ LANGUAGE plpgsql
[2025-12-02 17:56:04] completed in 36 ms
```

2)

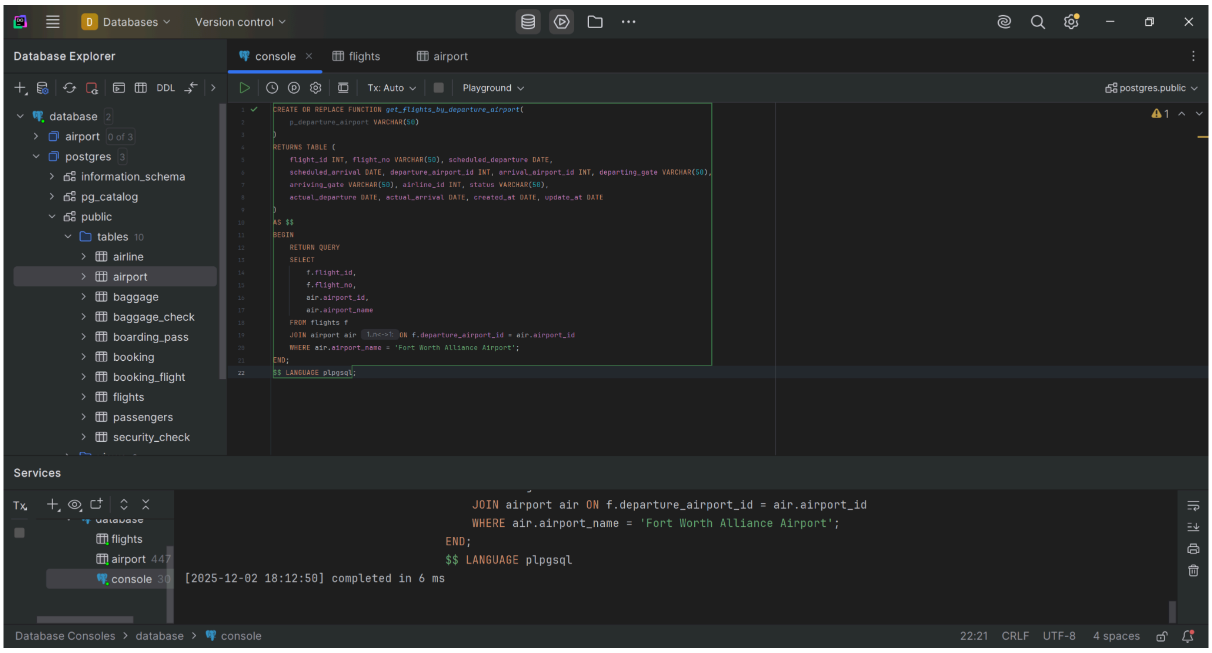


```sql
CREATE OR REPLACE PROCEDURE update_flight_status(
    p_flight_id INT,
    p_new_status VARCHAR(20)
)
AS $$
BEGIN
    UPDATE flights
    SET status = p_new_status
    WHERE flight_id = p_flight_id;

    RAISE NOTICE 'Flight % status updated to: %', p_flight_id, p_new_status;
END;
$$ LANGUAGE plpgsql;
```

```
                    RAISE NOTICE 'Flight % status updated to: %', p_flight_id, p_new_status;
                END;
                $$ LANGUAGE plpgsql
[2025-12-02 18:00:59] completed in 6 ms
```

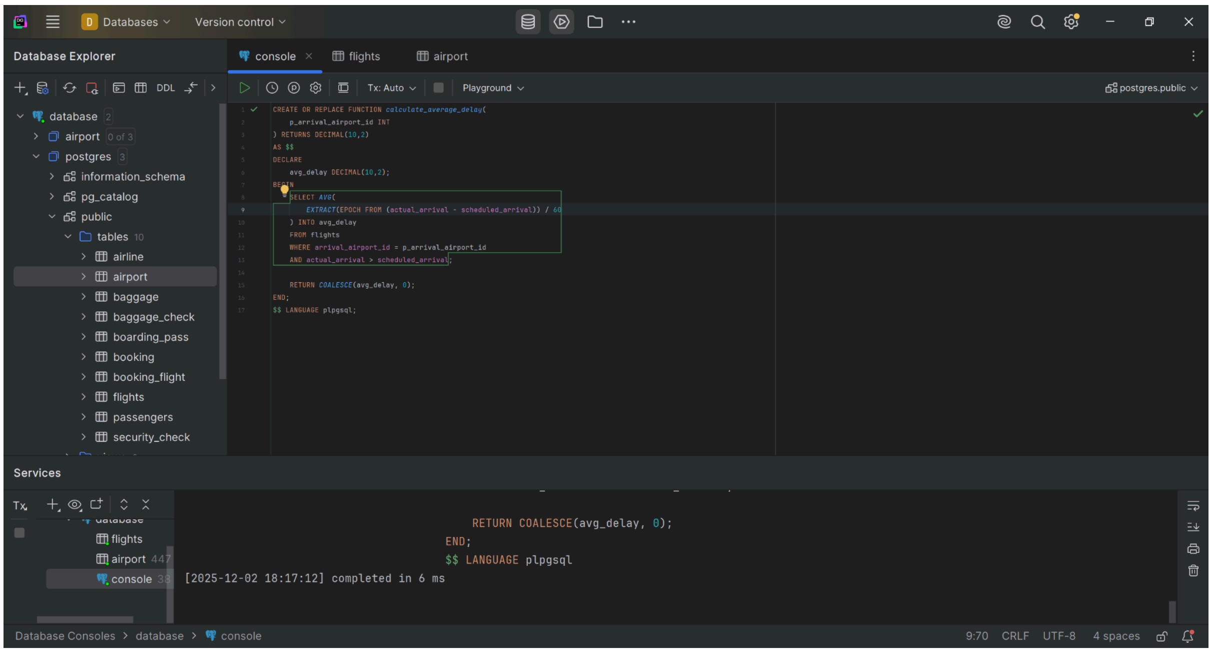
3)



```sql
CREATE OR REPLACE FUNCTION get_flights_by_departure_airport(
    p_departure_airport VARCHAR(50)
)
RETURNS TABLE (
    flight_id INT, flight_no VARCHAR(50), scheduled_departure DATE,
    scheduled_arrival DATE, departure_airport_id INT, arrival_airport_id INT, departing_gate VARCHAR(50),
    arriving_gate VARCHAR(50), airline_id INT, status VARCHAR(50),
    actual_departure DATE, actual_arrival DATE, created_at DATE, update_at DATE
)
AS $$
BEGIN
    RETURN QUERY
    SELECT
        f.flight_id,
        f.flight_no,
        air.airport_id,
        air.airport_name
    FROM flights f
    JOIN airport air ON f.departure_airport_id = air.airport_id
    WHERE air.airport_name = 'Fort Worth Alliance Airport';
END;
$$ LANGUAGE plpgsql;
```

```
            JOIN airport air ON f.departure_airport_id = air.airport_id
            WHERE air.airport_name = 'Fort Worth Alliance Airport';
        END;
        $$ LANGUAGE plpgsql
[2025-12-02 18:12:50] completed in 6 ms
```

4)

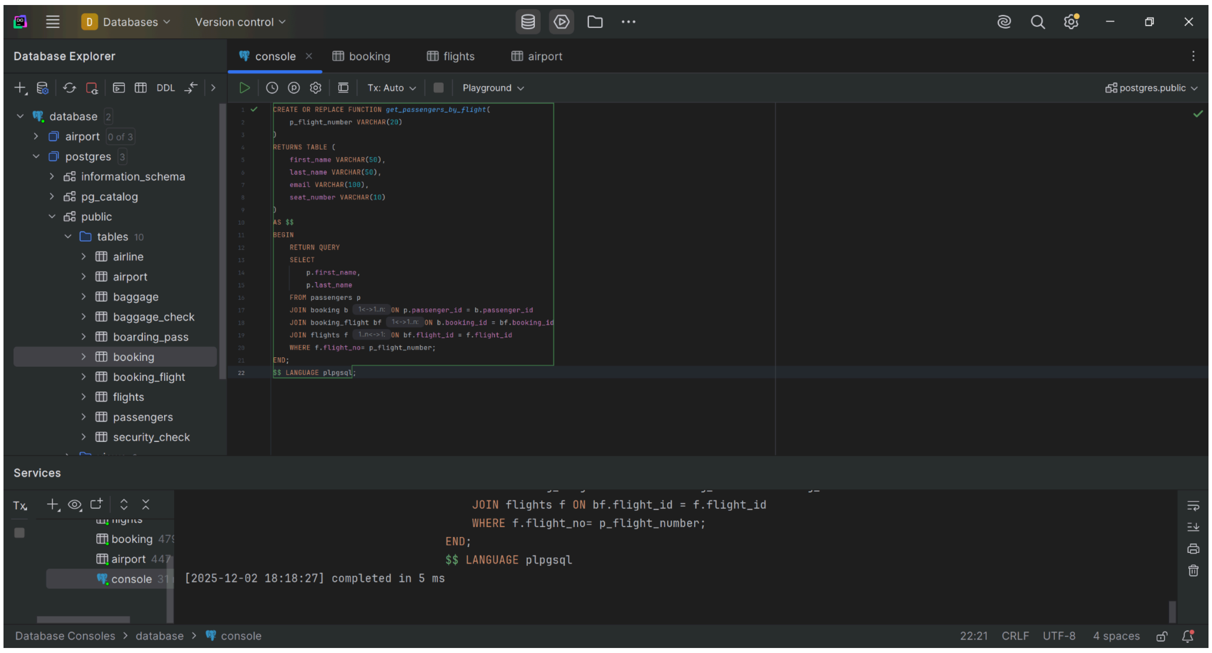


```sql
CREATE OR REPLACE FUNCTION calculate_average_delay(
    p_arrival_airport_id INT
) RETURNS DECIMAL(10,2)
AS $$
DECLARE
    avg_delay DECIMAL(10,2);
BEGIN
    SELECT AVG(
        EXTRACT(EPOCH FROM (actual_arrival - scheduled_arrival)) / 60
    ) INTO avg_delay
    FROM flights
    WHERE arrival_airport_id = p_arrival_airport_id
    AND actual_arrival > scheduled_arrival;

    RETURN COALESCE(avg_delay, 0);
END;
$$ LANGUAGE plpgsql;
```

```
            RETURN COALESCE(avg_delay, 0);
        END;
        $$ LANGUAGE plpgsql
[2025-12-02 18:17:12] completed in 6 ms
```

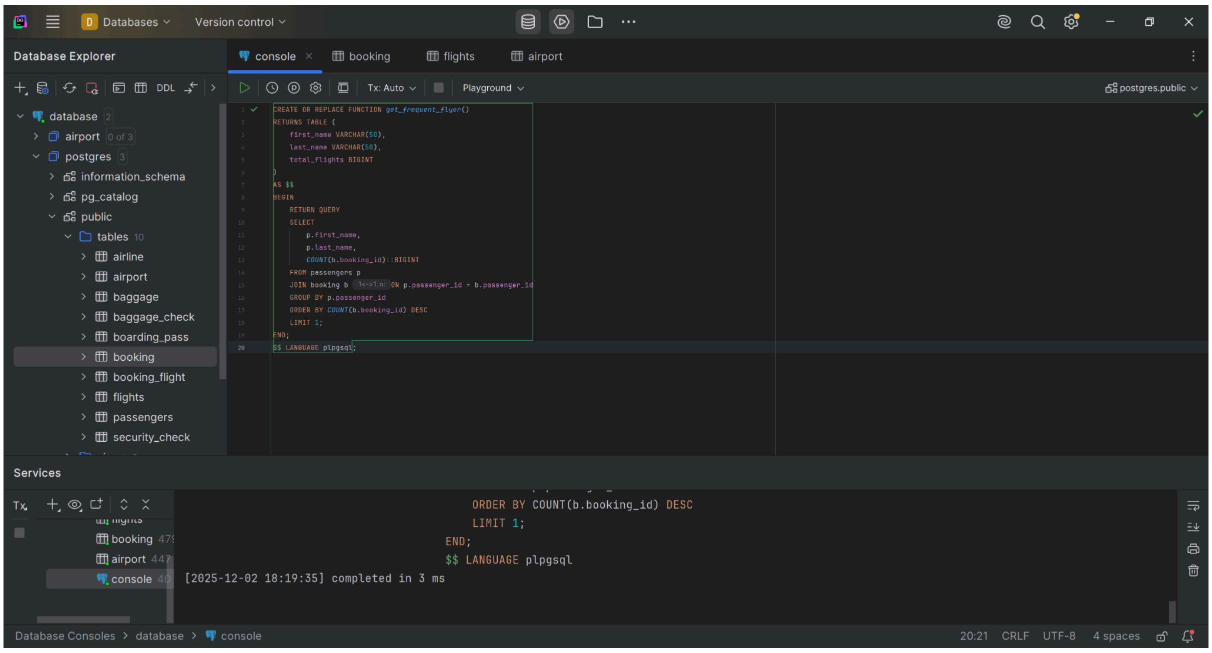
5)



```sql
CREATE OR REPLACE FUNCTION get_passengers_by_flight(
    p_flight_number VARCHAR(20)
)
RETURNS TABLE (
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100),
    seat_number VARCHAR(10)
)
AS $$
BEGIN
    RETURN QUERY
    SELECT
        p.first_name,
        p.last_name
    FROM passengers p
    JOIN booking b ON p.passenger_id = b.passenger_id
    JOIN booking_flight bf ON b.booking_id = bf.booking_id
    JOIN flights f ON bf.flight_id = f.flight_id
    WHERE f.flight_no= p_flight_number;
END;
$$ LANGUAGE plpgsql;
```

```
                                JOIN flights f ON bf.flight_id = f.flight_id
                                WHERE f.flight_no= p_flight_number;
                            END;
                            $$ LANGUAGE plpgsql
[2025-12-02 18:18:27] completed in 5 ms
```

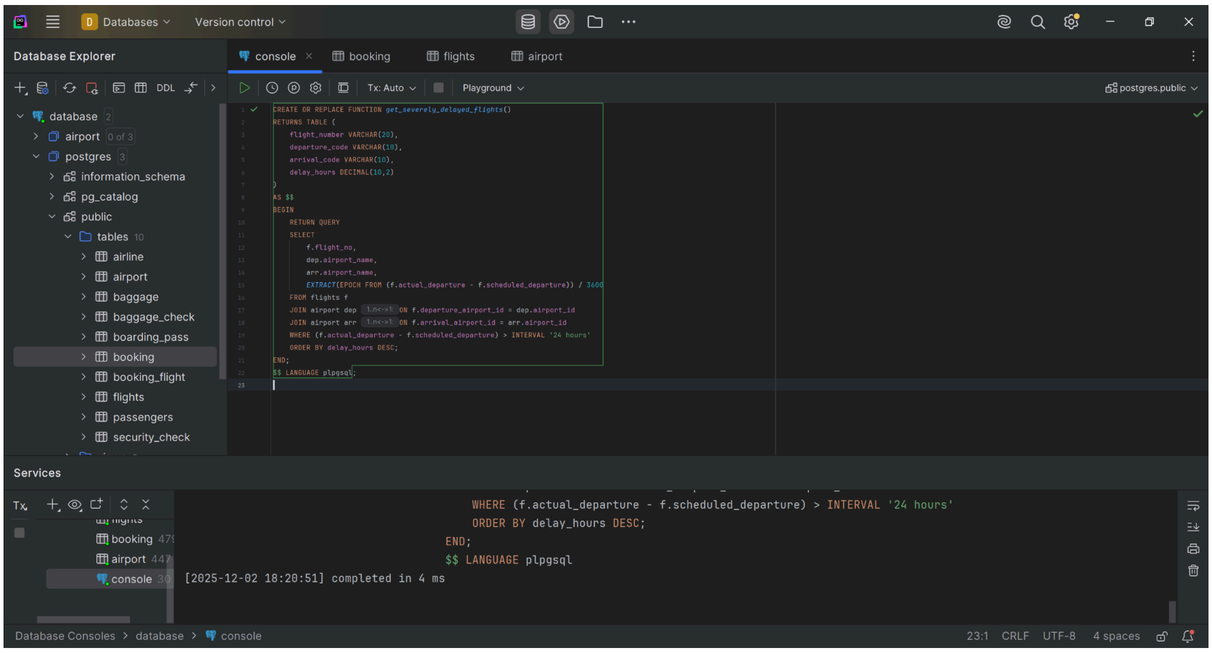
6)



```sql
CREATE OR REPLACE FUNCTION get_frequent_flyer()
RETURNS TABLE (
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    total_flights BIGINT
)
AS $$
BEGIN
    RETURN QUERY
    SELECT
        p.first_name,
        p.last_name,
        COUNT(b.booking_id)::BIGINT
    FROM passengers p
    JOIN booking b ON p.passenger_id = b.passenger_id
    GROUP BY p.passenger_id
    ORDER BY COUNT(b.booking_id) DESC
    LIMIT 1;
END;
$$ LANGUAGE plpgsql;
```

```
                                ORDER BY COUNT(b.booking_id) DESC
                                LIMIT 1;
                            END;
                            $$ LANGUAGE plpgsql
[2025-12-02 18:19:35] completed in 3 ms
```

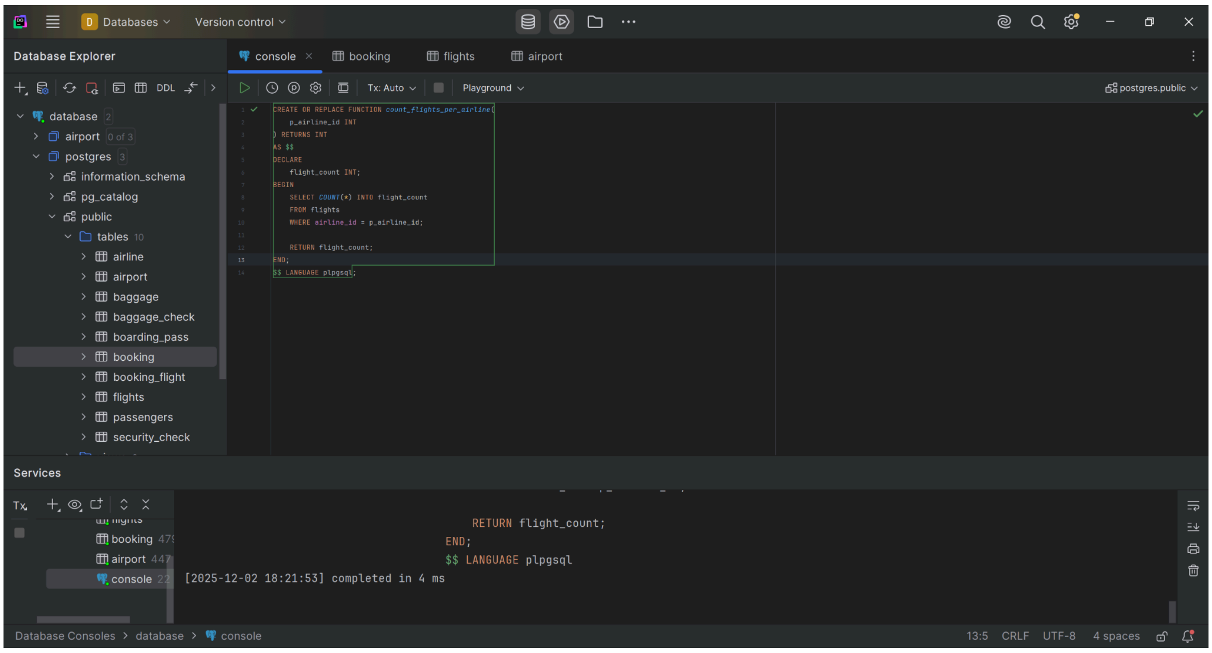
7)



```sql
CREATE OR REPLACE FUNCTION get_severely_delayed_flights()
RETURNS TABLE (
    flight_number VARCHAR(20),
    departure_code VARCHAR(10),
    arrival_code VARCHAR(10),
    delay_hours DECIMAL(10,2)
)
AS $$
BEGIN
    RETURN QUERY
    SELECT
        f.flight_no,
        dep.airport_name,
        arr.airport_name,
        EXTRACT(EPOCH FROM (f.actual_departure - f.scheduled_departure)) / 3600
    FROM flights f
    JOIN airport dep 1..n<->1: ON f.departure_airport_id = dep.airport_id
    JOIN airport arr 1..n<->1: ON f.arrival_airport_id = arr.airport_id
    WHERE (f.actual_departure - f.scheduled_departure) > INTERVAL '24 hours'
    ORDER BY delay_hours DESC;
END;
$$ LANGUAGE plpgsql;
```

```
                                        WHERE (f.actual_departure - f.scheduled_departure) > INTERVAL '24 hours'
                                        ORDER BY delay_hours DESC;
                                    END;
                                    $$ LANGUAGE plpgsql
[2025-12-02 18:20:51] completed in 4 ms
```

8)

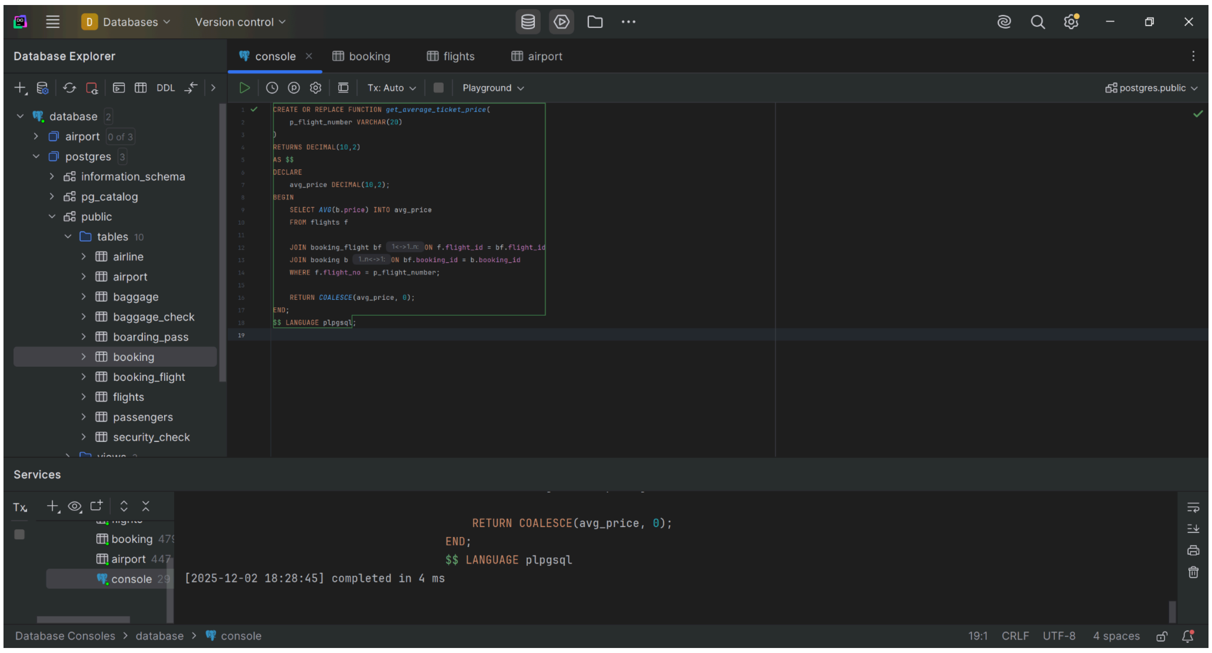


```sql
CREATE OR REPLACE FUNCTION count_flights_per_airline(
    p_airline_id INT
) RETURNS INT
AS $$
DECLARE
    flight_count INT;
BEGIN
    SELECT COUNT(*) INTO flight_count
    FROM flights
    WHERE airline_id = p_airline_id;

    RETURN flight_count;
END;
$$ LANGUAGE plpgsql;
```

```
                                        RETURN flight_count;
                                    END;
                                    $$ LANGUAGE plpgsql
[2025-12-02 18:21:53] completed in 4 ms
```

9)



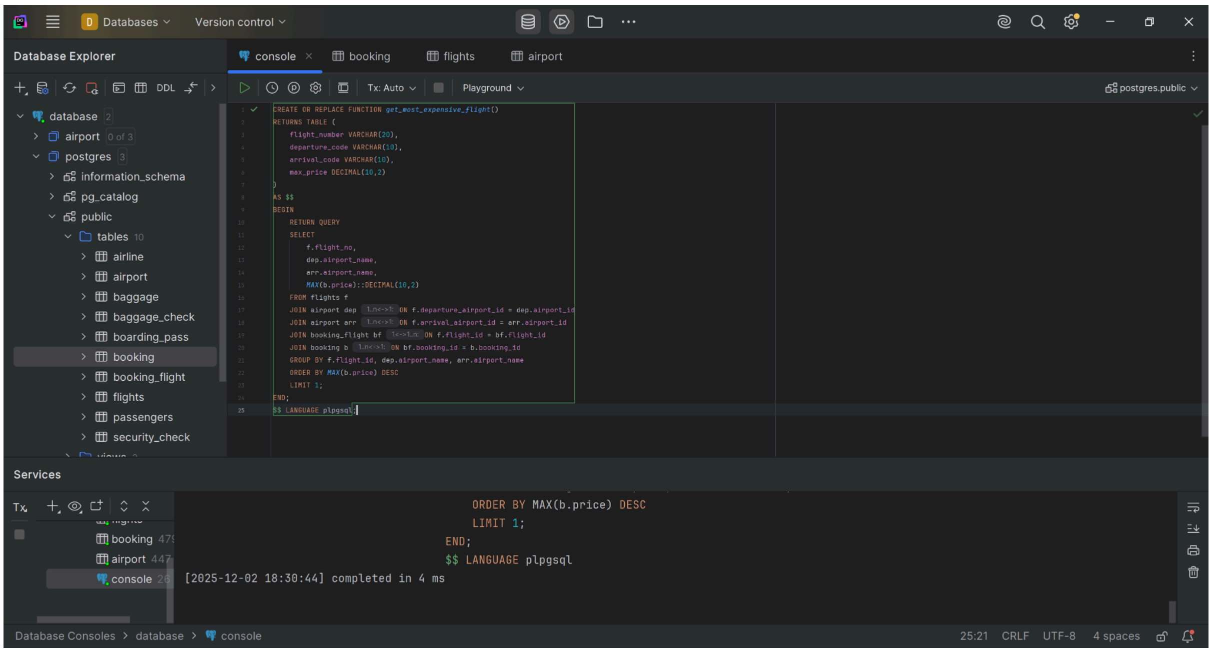
```sql
CREATE OR REPLACE FUNCTION get_average_ticket_price(
    p_flight_number VARCHAR(20)
)
RETURNS DECIMAL(10,2)
AS $$
DECLARE
    avg_price DECIMAL(10,2);
BEGIN
    SELECT AVG(b.price) INTO avg_price
    FROM flights f

    JOIN booking_flight bf ON f.flight_id = bf.flight_id
    JOIN booking b ON bf.booking_id = b.booking_id
    WHERE f.flight_no = p_flight_number;

    RETURN COALESCE(avg_price, 0);
END;
$$ LANGUAGE plpgsql;
```

```
            RETURN COALESCE(avg_price, 0);
        END;
        $$ LANGUAGE plpgsql
[2025-12-02 18:28:45] completed in 4 ms
```

10)



```sql
CREATE OR REPLACE FUNCTION get_most_expensive_flight()
RETURNS TABLE (
    flight_number VARCHAR(20),
    departure_code VARCHAR(10),
    arrival_code VARCHAR(10),
    max_price DECIMAL(10,2)
)
AS $$
BEGIN
    RETURN QUERY
    SELECT
        f.flight_no,
        dep.airport_name,
        arr.airport_name,
        MAX(b.price)::DECIMAL(10,2)
    FROM flights f
    JOIN airport dep ON f.departure_airport_id = dep.airport_id
    JOIN airport arr ON f.arrival_airport_id = arr.airport_id
    JOIN booking_flight bf ON f.flight_id = bf.flight_id
    JOIN booking b ON bf.booking_id = b.booking_id
    GROUP BY f.flight_id, dep.airport_name, arr.airport_name
    ORDER BY MAX(b.price) DESC
    LIMIT 1;
END;
$$ LANGUAGE plpgsql;
```

```
            ORDER BY MAX(b.price) DESC
            LIMIT 1;
        END;
        $$ LANGUAGE plpgsql
[2025-12-02 18:30:44] completed in 4 ms
```