```
CREATE OR REPLACE PROCEDURE my_procedure()
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, s
    VALUES ( flight_id 1001, flight_no 'hui2342', scheduled_departure '2025-12-2', scheduled_arrival '2025-12-2', departure_airport_id 3, arrival_airport_id 4, departing_gate 'a', arr

    RAISE NOTICE 'new flight insertes';
END;
$$;

CALL my_procedure();
```

```
CREATE OR REPLACE PROCEDURE update_flight_status(
    id INTEGER
)
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE flights SET status = 'fine'
    WHERE flights.flight_id = id;
END;
$$;

CALL update_flight_status( id 1);
```

```
CREATE OR REPLACE PROCEDURE get_flights_from_airport(
    p_airport_id INTEGER,
    ref OUT REFCURSOR
)
LANGUAGE plpgsql
AS $$
BEGIN
    OPEN ref FOR
        SELECT *
        FROM flights
        WHERE departure_airport_id = p_airport_id;
END;
$$;


DO $$
DECLARE
    res REFCURSOR;
BEGIN
    get_flights_from_airport(1, res);
    RAISE NOTICE 'Result: %', res;
END $$;
```

```sql
CREATE OR REPLACE FUNCTION get_avg_arrival_delay(airport_id INTEGER)
RETURNS INTERVAL
LANGUAGE plpgsql
AS $$
DECLARE avg_delay INTERVAL;
BEGIN
    SELECT AVG(actual_arrival - scheduled_arrival)
    INTO avg_delay
    FROM flights
    WHERE arrival_airport_id = airport_id
    AND actual_arrival IS NOT NULL AND scheduled_arrival IS NOT NULL;

    RETURN avg_delay;
END;
$$;

SELECT get_avg_arrival_delay( airport_id 3);
```

Output | get_avg_arrival_delay(3):interval

get_avg_arrival_delay

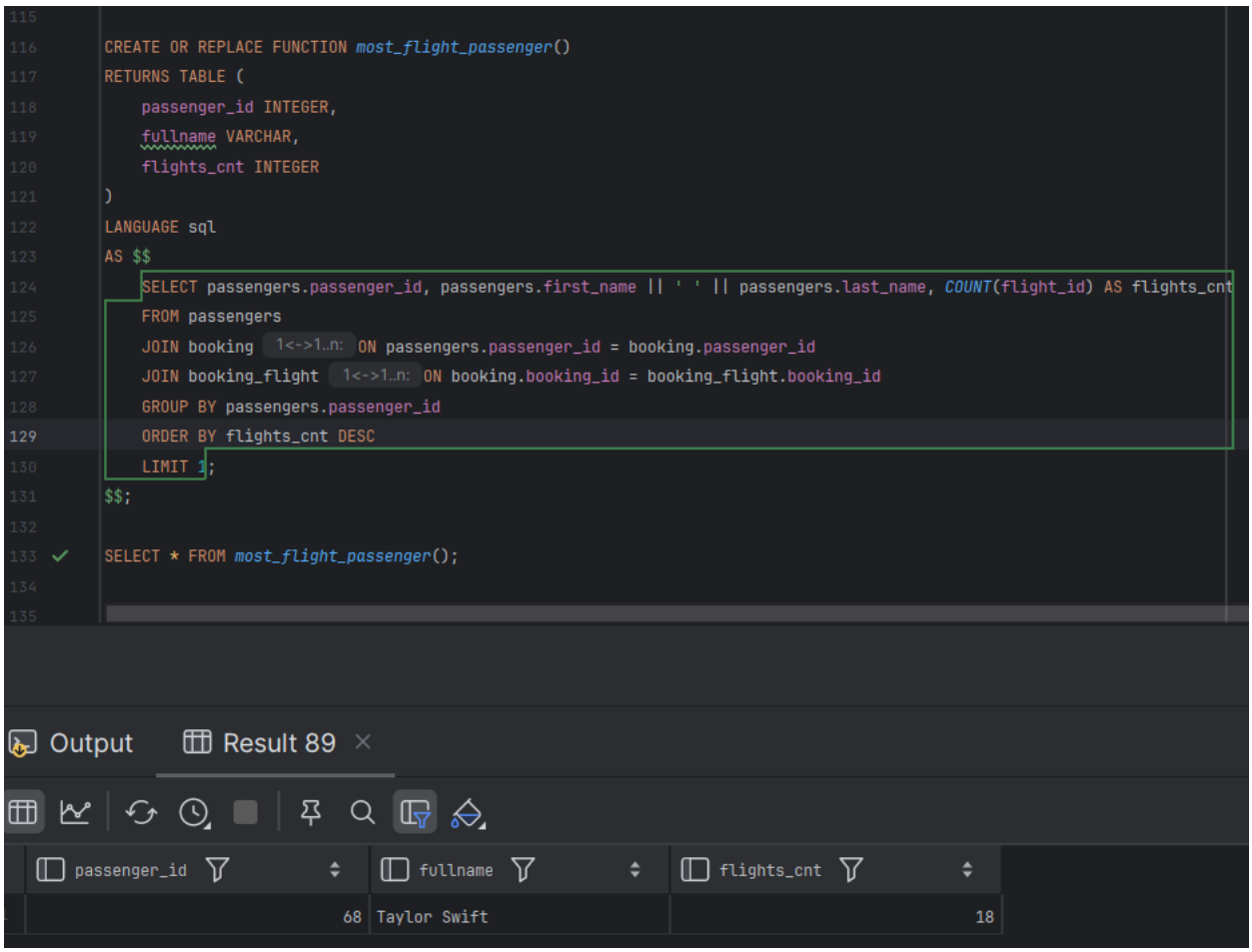
0 years 0 mons 0 days 0 hours 0 mins -15.62 se

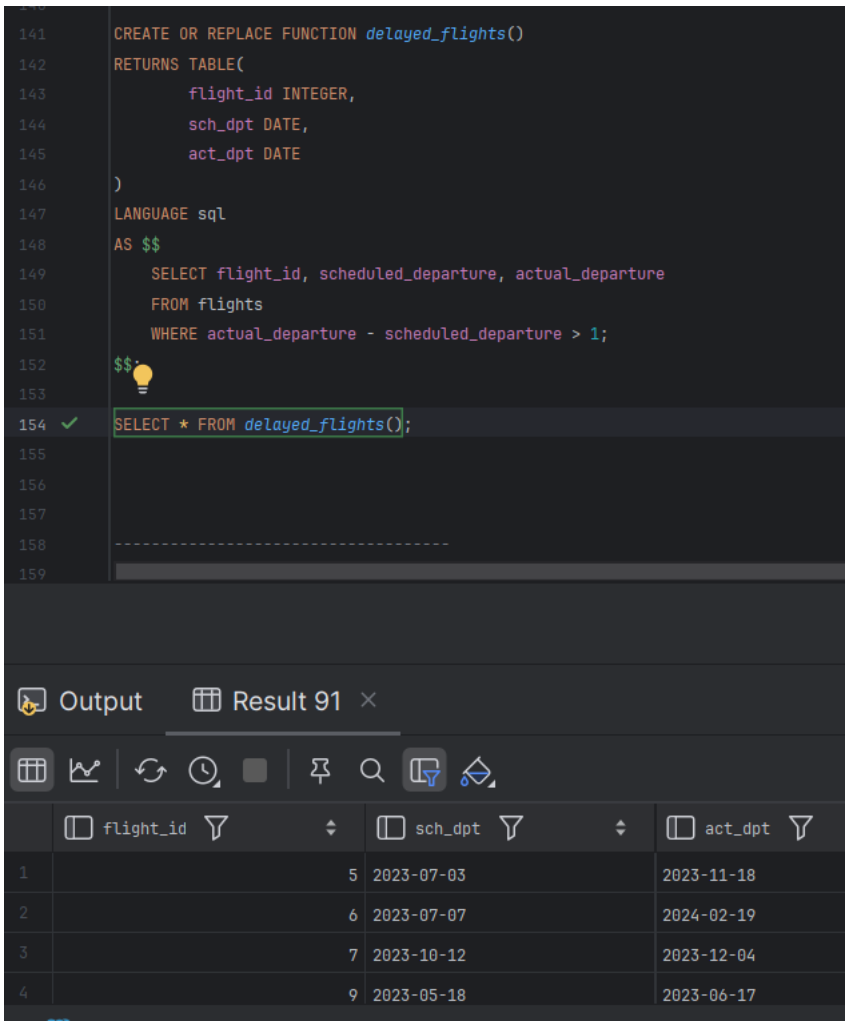
```sql
CREATE OR REPLACE FUNCTION getPassengersOfFlight(no VARCHAR)
RETURNS TABLE (
    passport_number VARCHAR,
    created_at DATE,
    update_at DATE
)
LANGUAGE sql
AS $$
    SELECT p.*
    FROM passengers p
    JOIN booking b 1<->1..n: ON p.passenger_id = b.passenger_id
    JOIN booking_flight bf 1<->1..n: ON b.booking_id = bf.booking_id
    JOIN flights f 1..n<->1: ON bf.flight_id = f.flight_id
    WHERE f.flight_no = no;
$$;

SELECT * FROM getPassengersOfFlight( no 'US-CT');
```

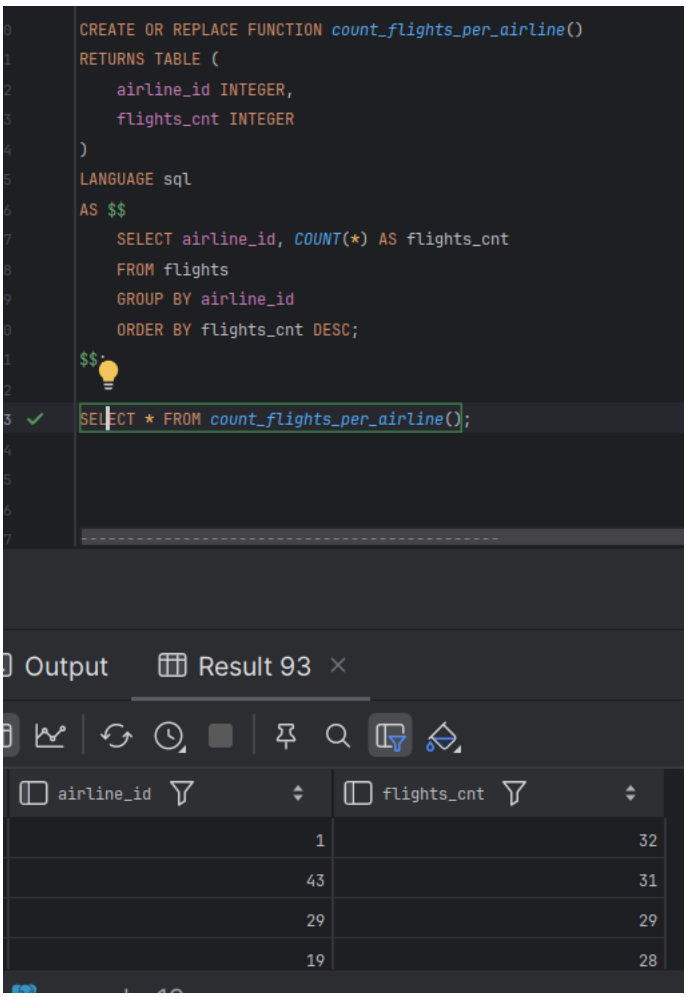
Output | Result 87

CSV

passenger_id	first_name	last_name	date_of_birth	gender	country_of_citizenship	countr
9	Alphonso	Philippou	1975-08-30	Male	Tunisia	Brazil
32	Elfrida	Schukert	1991-03-18	Female	Peru	Russia


CREATE OR REPLACE FUNCTION most_flight_passenger()
RETURNS TABLE (
passenger_id INTEGER,
fullname VARCHAR,
flights_cnt INTEGER
)
LANGUAGE sql
AS $$
SELECT passengers.passenger_id, passengers.first_name || ' ' || passengers.last_name, COUNT(flight_id) AS flights_cnt
FROM passengers
JOIN booking 1<->1..n: ON passengers.passenger_id = booking.passenger_id
JOIN booking_flight 1<->1..n: ON booking.booking_id = booking_flight.booking_id
GROUP BY passengers.passenger_id
ORDER BY flights_cnt DESC
LIMIT 1;
$$;
SELECT * FROM most_flight_passenger();
Output
Result 89
passenger_id
fullname
flights_cnt
68
Taylor Swift
18


CREATE OR REPLACE FUNCTION delayed_flights()
RETURNS TABLE(
flight_id INTEGER,
sch_dpt DATE,
act_dpt DATE
)
LANGUAGE sql
AS $$
SELECT flight_id, scheduled_departure, actual_departure
FROM flights
WHERE actual_departure - scheduled_departure > 1;
$$;
SELECT * FROM delayed_flights();
Output
Result 91
flight_id
sch_dpt
act_dpt
1 5 2023-07-03 2023-11-18
2 6 2023-07-07 2024-02-19
3 7 2023-10-12 2023-12-04
4 9 2023-05-18 2023-06-17


CREATE OR REPLACE FUNCTION count_flights_per_airline()
RETURNS TABLE (
airline_id INTEGER,
flights_cnt INTEGER
)
LANGUAGE sql
AS $$
SELECT airline_id, COUNT(*) AS flights_cnt
FROM flights
GROUP BY airline_id
ORDER BY flights_cnt DESC;
$$;
SELECT * FROM count_flights_per_airline();
Output
Result 93
airline_id
flights_cnt
1
32
43
31
29
29
19
28

```sql
CREATE OR REPLACE FUNCTION avg_ticket_price_for_flight(
    flight_no_input INTEGER
)
RETURNS INTEGER
LANGUAGE plpgsql
AS $$
DECLARE
    avg_price INTEGER;
BEGIN
    SELECT AVG(price)::INTEGER
    INTO avg_price
    FROM booking
    JOIN booking_flight bf 1<->1..n: ON booking.booking_id = bf.booking_id
    JOIN flights f 1..n<->1: ON bf.flight_id = f.flight_id
    WHERE f.flight_id = flight_no_input;

    RETURN avg_price;
END;
$$;

SELECT * FROM avg_ticket_price_for_flight( flight_no_input 1);
```

Output | avg_ticket_price_for_flight

Tx: Auto

	avg_ticket_price_for_flight
1	7369

```sql
CREATE OR REPLACE FUNCTION most_expensive_flight()
RETURNS TABLE (
)
LANGUAGE sql
AS $$
    SELECT
        f.flight_id,
        f.flight_no,
        dept.airport_name AS departure_airport,
        arr.airport_name AS arrival_airport,
        MAX(b.price) AS max_price
    FROM booking b
    JOIN booking_flight bf 1<->1..n: ON b.booking_id = bf.booking_id
    JOIN flights f 1..n<->1: ON bf.flight_id = f.flight_id
    JOIN airport dept 1..n<->1: ON f.departure_airport_id = dept.airport_id
    JOIN airport arr 1..n<->1: ON f.arrival_airport_id = arr.airport_id
    GROUP BY f.flight_id, f.flight_no, dept.airport_name, arr.airport_name
    ORDER BY max_price DESC
    LIMIT 1;
$$;

SELECT * FROM most_expensive_flight();
```

1 2

Output | Result 97

CSV

flight_id	flight_no	departure_airport	arrival_airport	max_price
7	AU-TAS	Ocean Falls Seaplane Base	chula Airport	20000

1 row