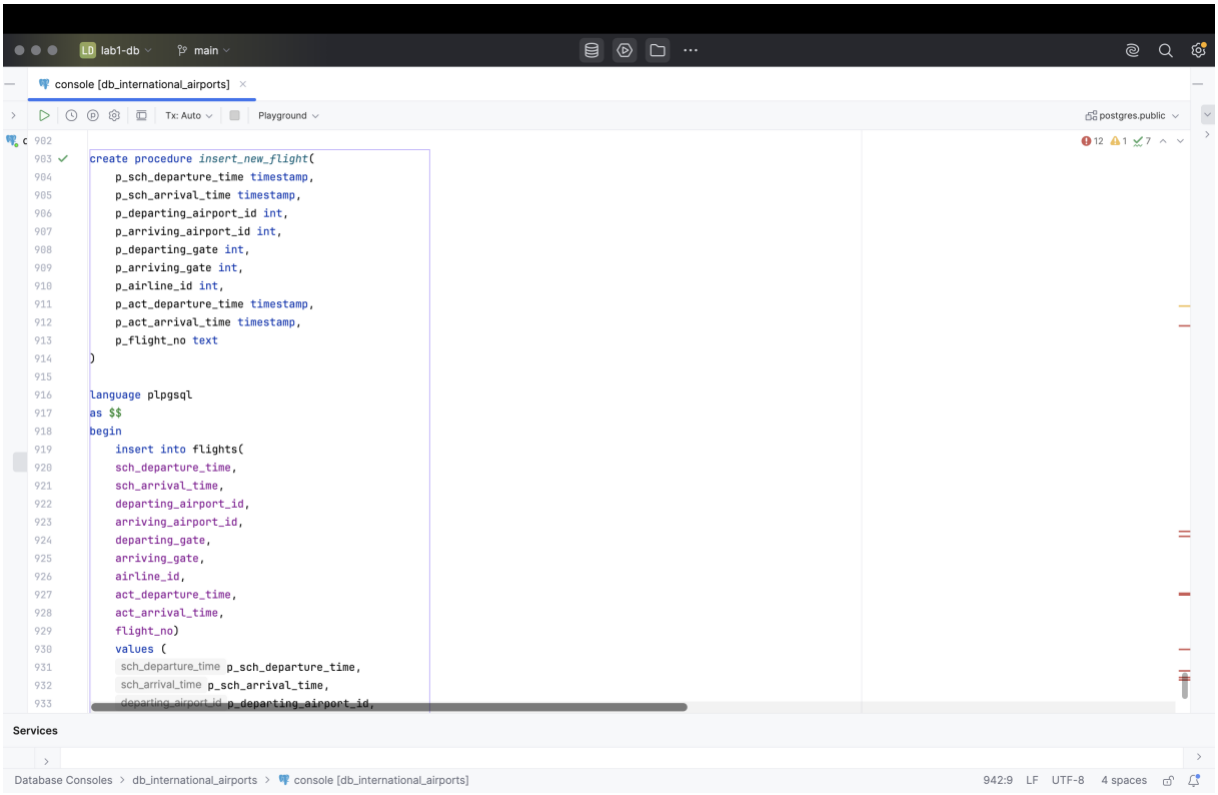
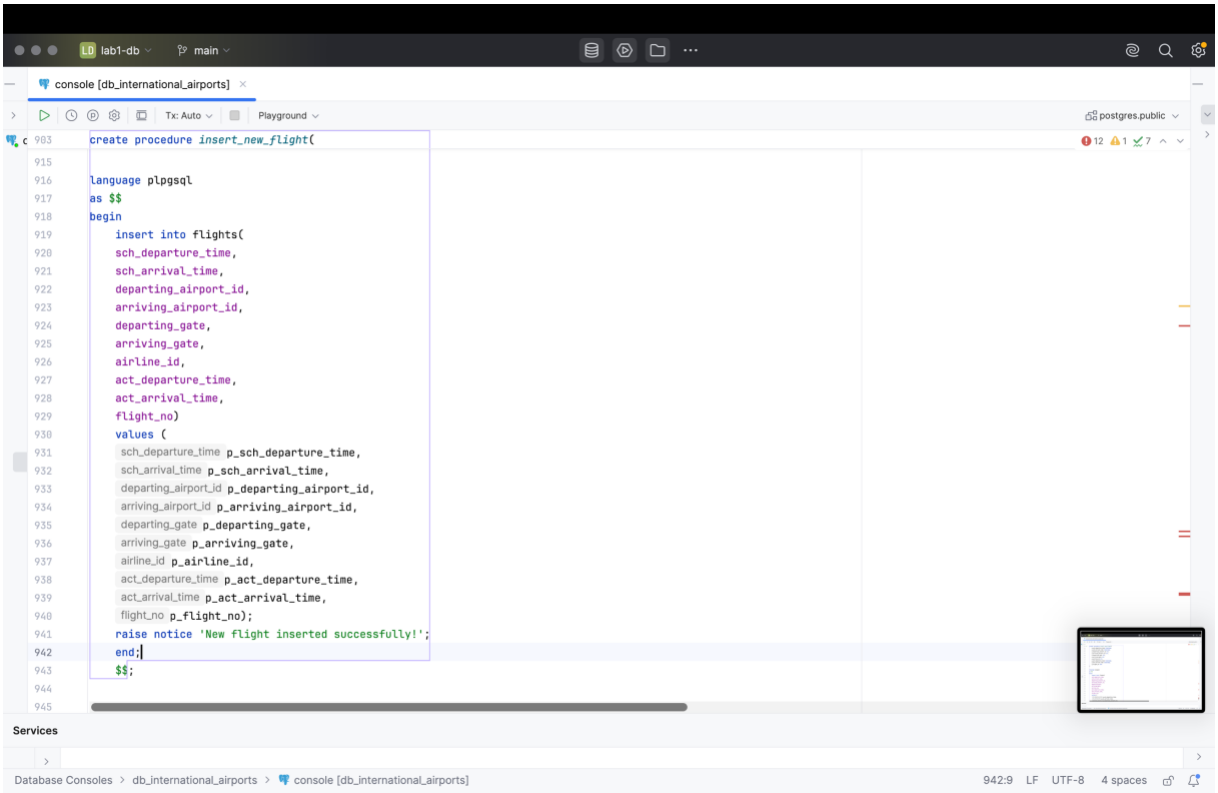
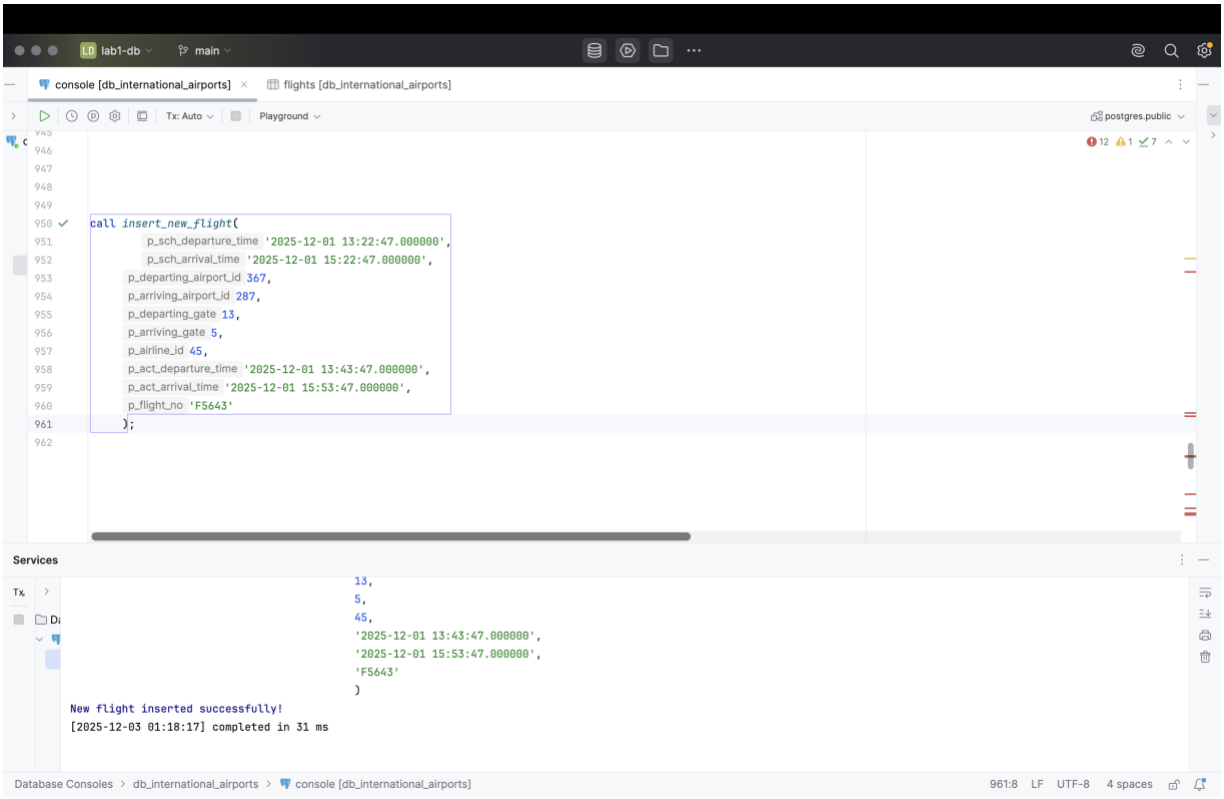
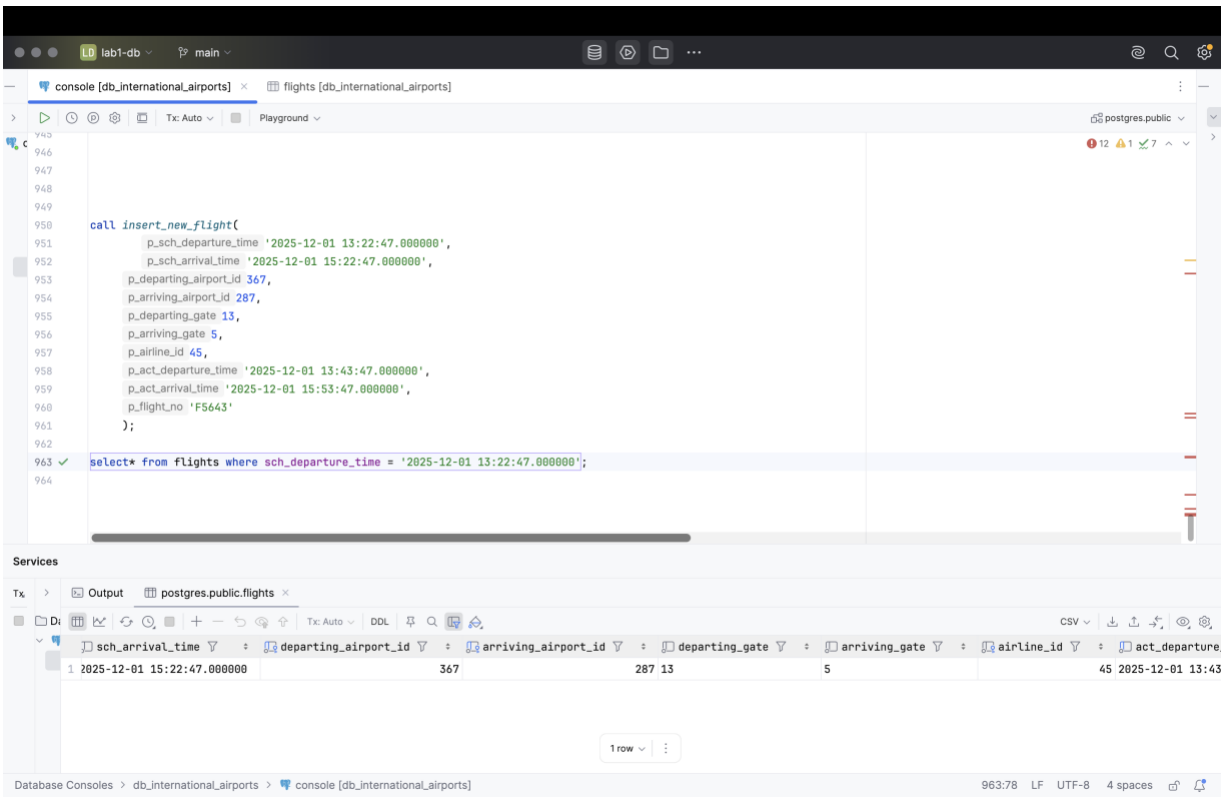
Lab 10

1. Create a stored procedure to insert a new flight into the flights table.

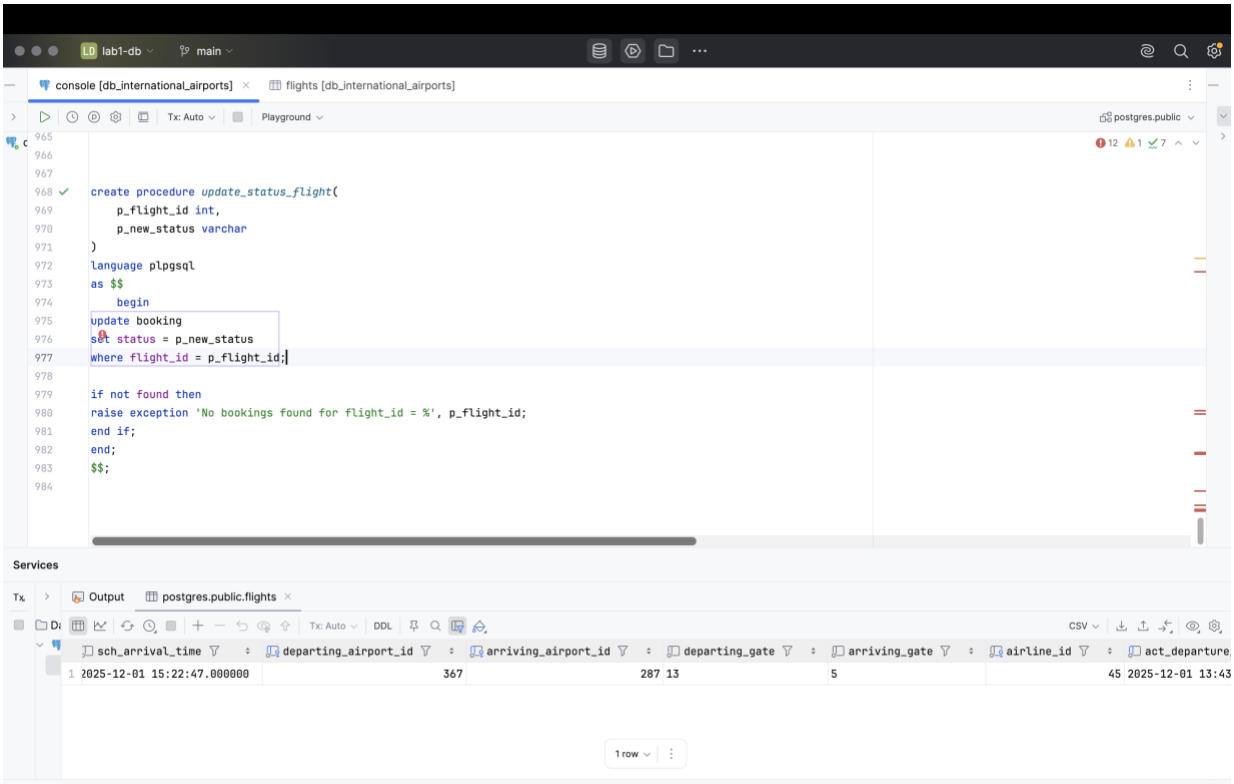


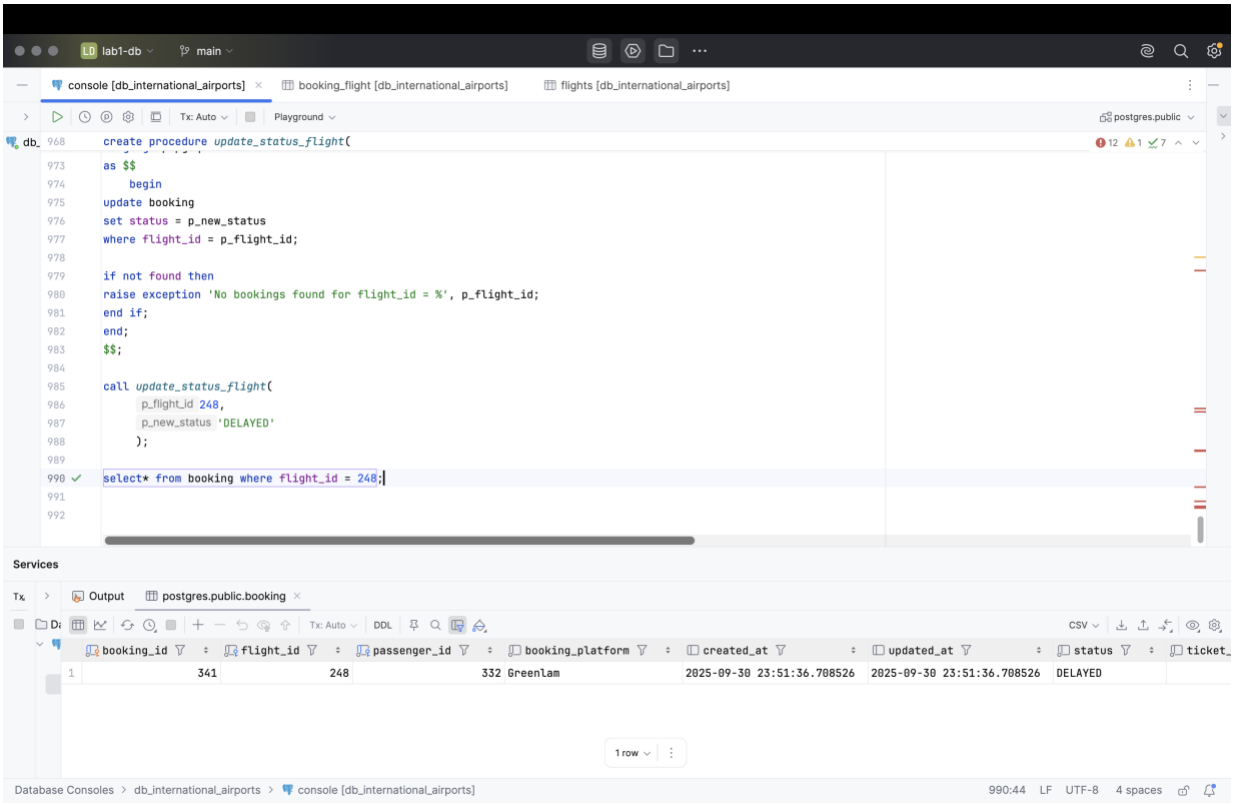



lab1-db
main
console [db_international_airports]
flights [db_international_airports]
Tx: Auto
Playground
postgres.public
call insert_new_flight(
p_sch_departure_time '2025-12-01 13:22:47.000000',
p_sch_arrival_time '2025-12-01 15:22:47.000000',
p_departing_airport_id 367,
p_arriving_airport_id 287,
p_departing_gate 13,
p_arriving_gate 5,
p_airline_id 45,
p_act_departure_time '2025-12-01 13:43:47.000000',
p_act_arrival_time '2025-12-01 15:53:47.000000',
p_flight_no 'F5643'
);
Services
13,
5,
45,
'2025-12-01 13:43:47.000000',
'2025-12-01 15:53:47.000000',
'F5643'
)
New flight inserted successfully!
[2025-12-03 01:18:17] completed in 31 ms
Database Consoles
db_international_airports
console [db_international_airports]
961:8
LF
UTF-8
4 spaces

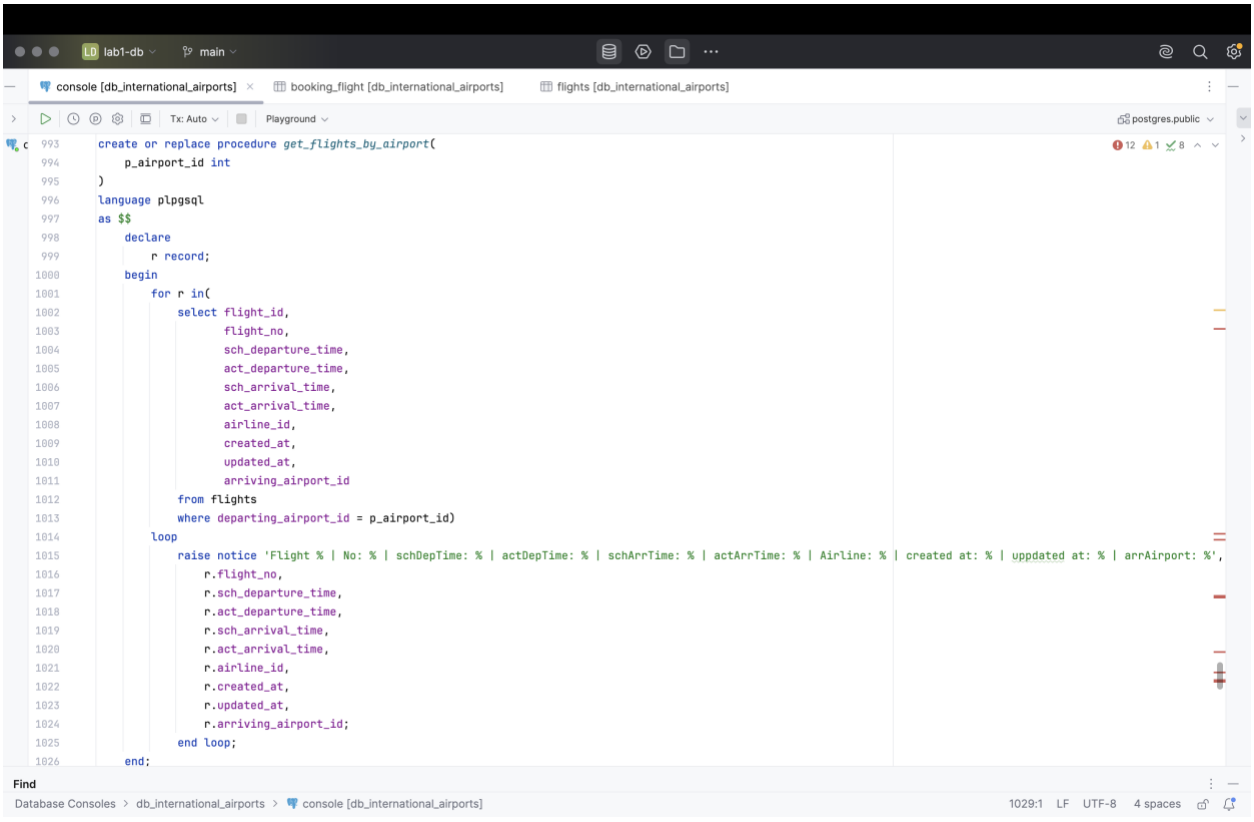

lab1-db
main
console [db_international_airports]
flights [db_international_airports]
Tx: Auto
Playground
postgres.public
call insert_new_flight(
p_sch_departure_time '2025-12-01 13:22:47.000000',
p_sch_arrival_time '2025-12-01 15:22:47.000000',
p_departing_airport_id 367,
p_arriving_airport_id 287,
p_departing_gate 13,
p_arriving_gate 5,
p_airline_id 45,
p_act_departure_time '2025-12-01 13:43:47.000000',
p_act_arrival_time '2025-12-01 15:53:47.000000',
p_flight_no 'F5643'
);
select* from flights where sch_departure_time = '2025-12-01 13:22:47.000000';
Services
Output
postgres.public.flights
DDL
CSV
sch_arrival_time
departing_airport_id
arriving_airport_id
departing_gate
arriving_gate
airline_id
act_departure
2025-12-01 15:22:47.000000
367
287
13
5
45
2025-12-01 13:43
1 row
Database Consoles
db_international_airports
console [db_international_airports]
963:78
LF
UTF-8
4 spaces

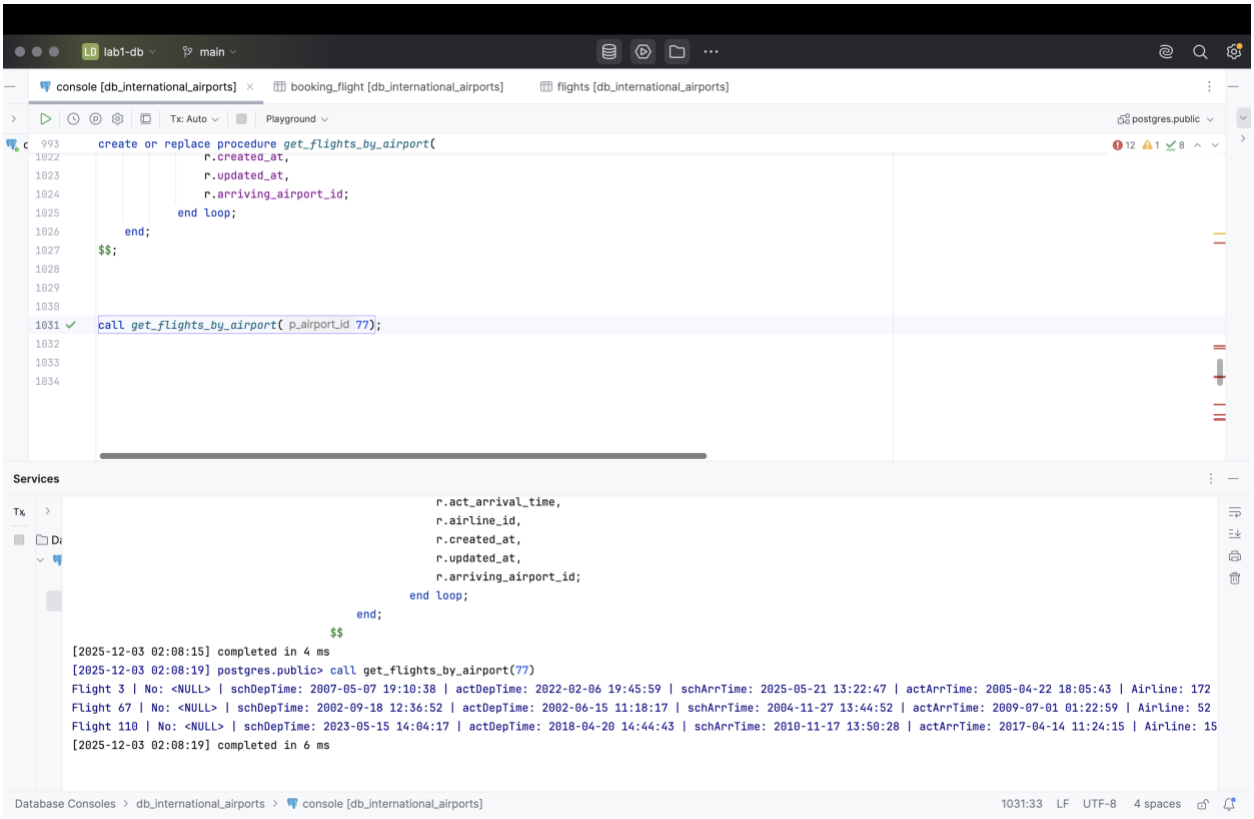
2. Create a stored procedure to update the status of a flight.



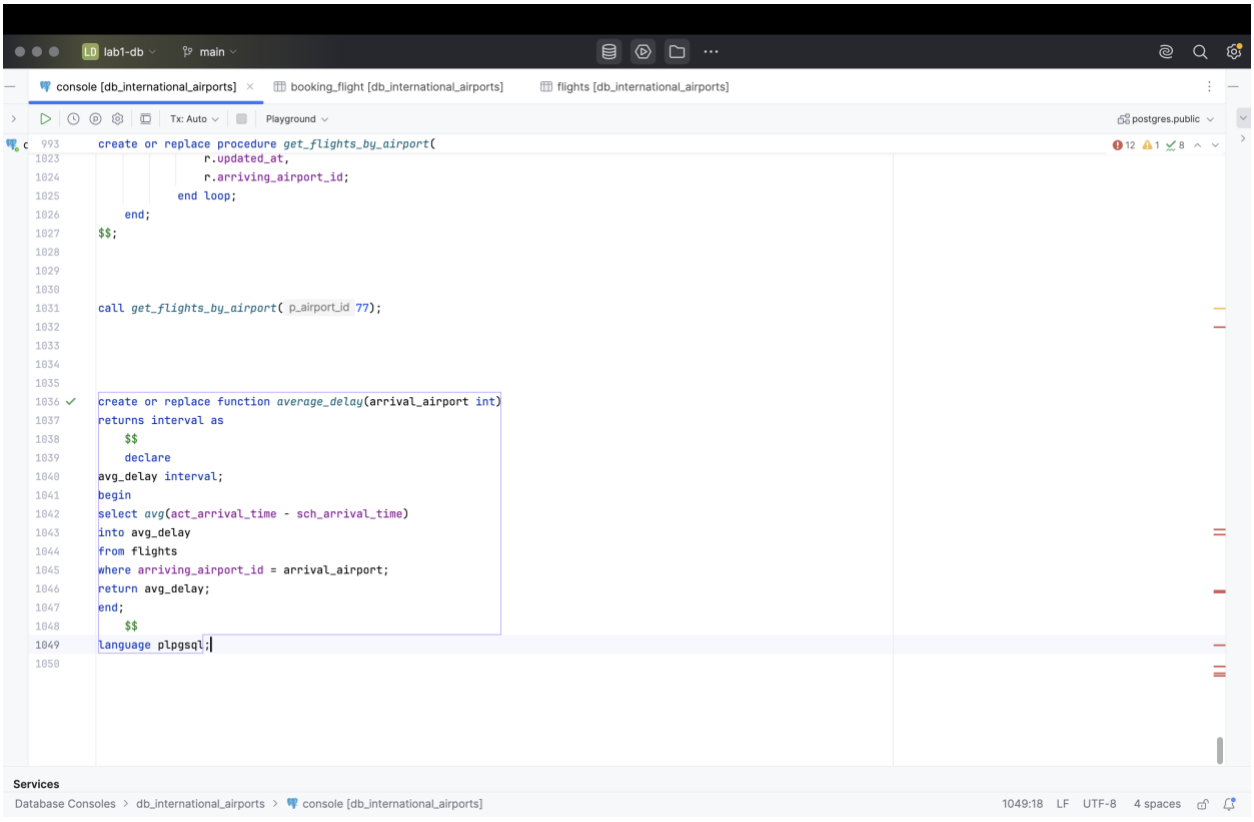


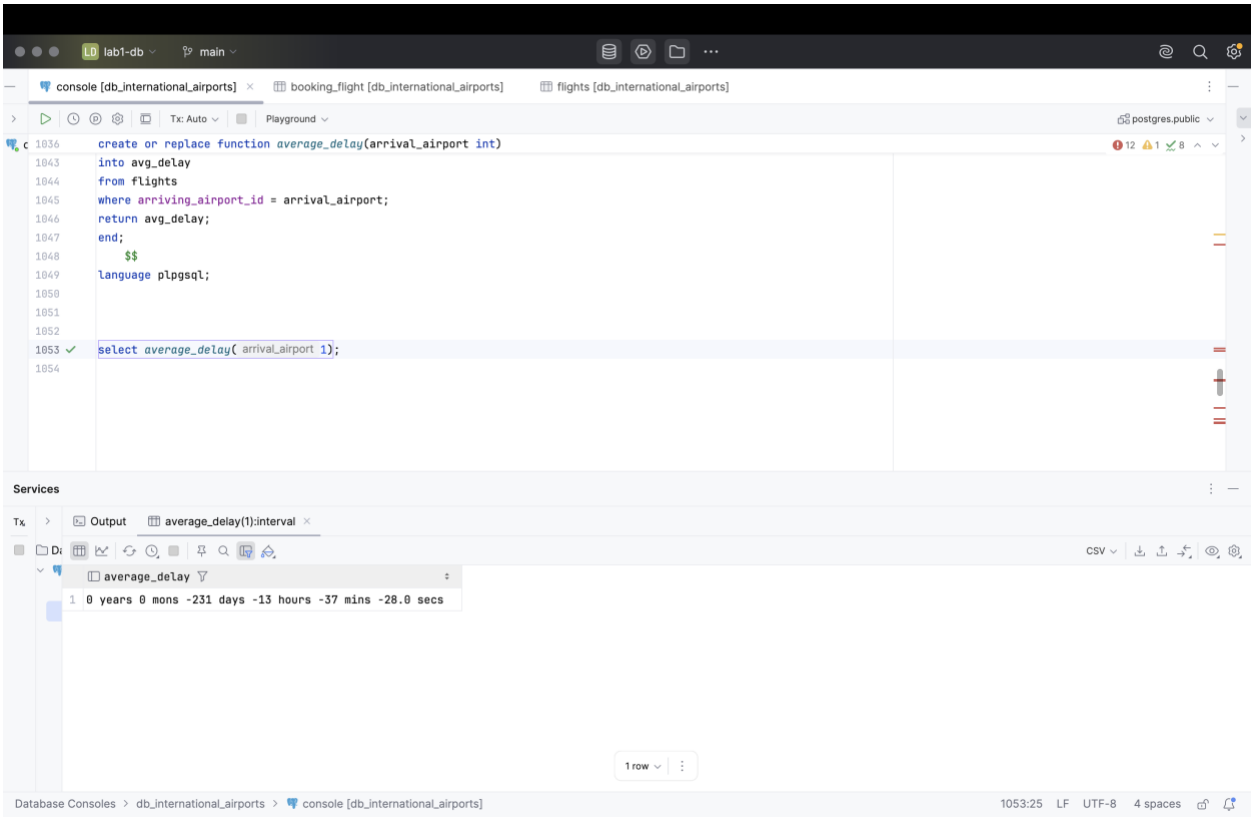
3. Create a stored procedure that returns a list of flights departing from a specific airport.



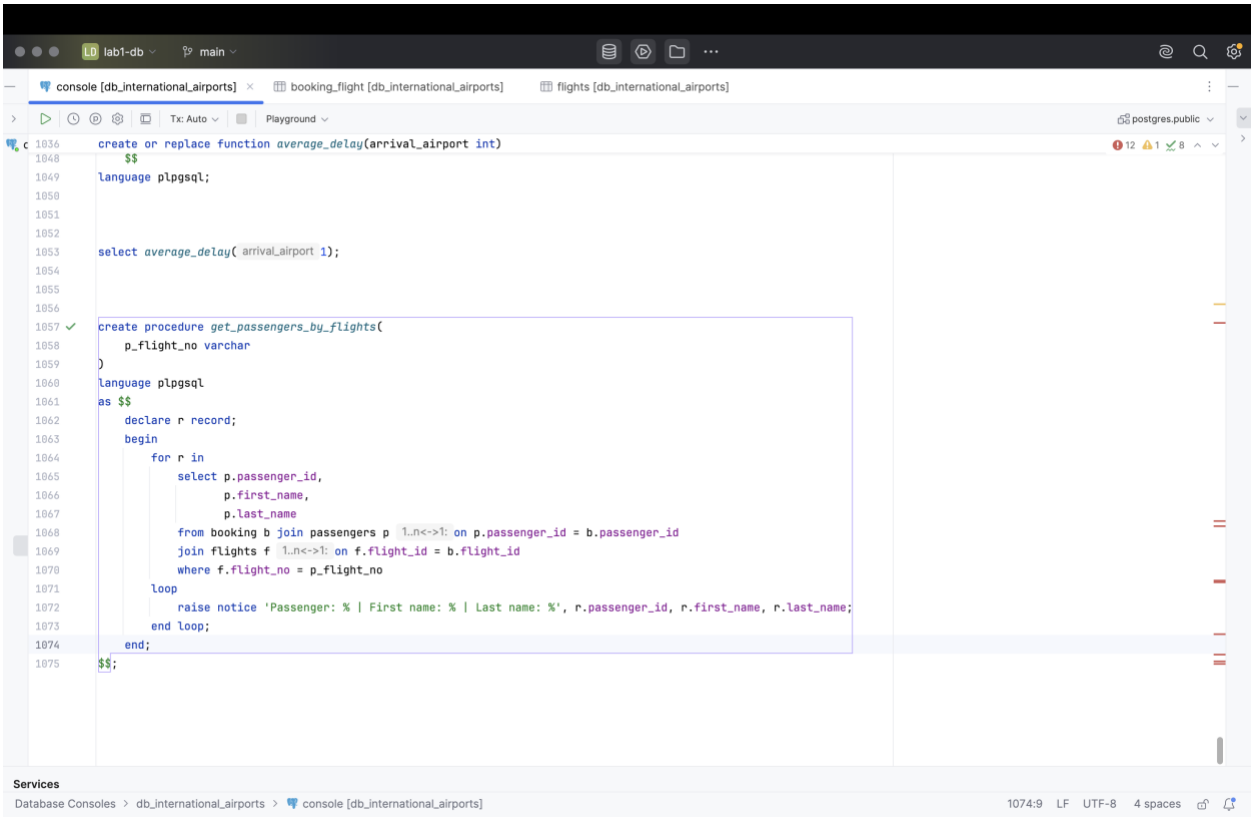


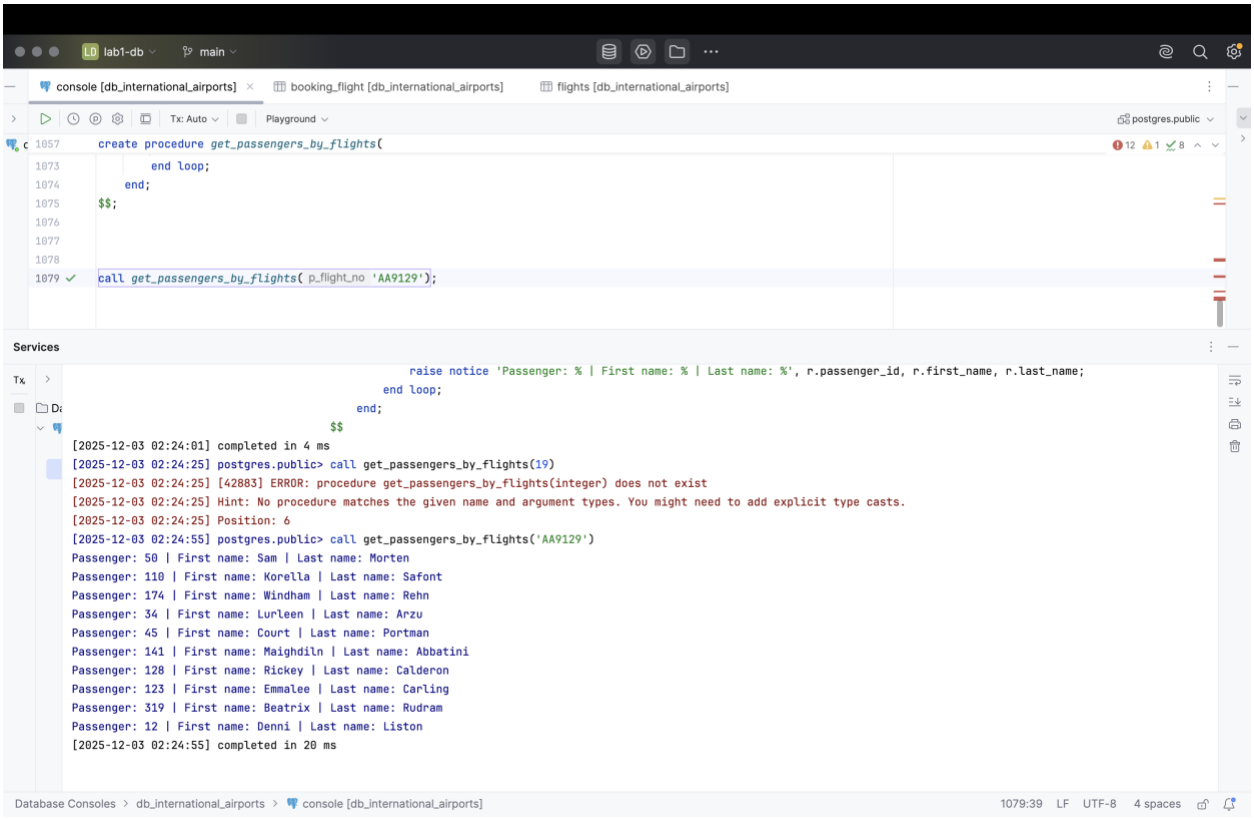
4. Create a function to calculate the average delay time of flights arriving at a specific airport.



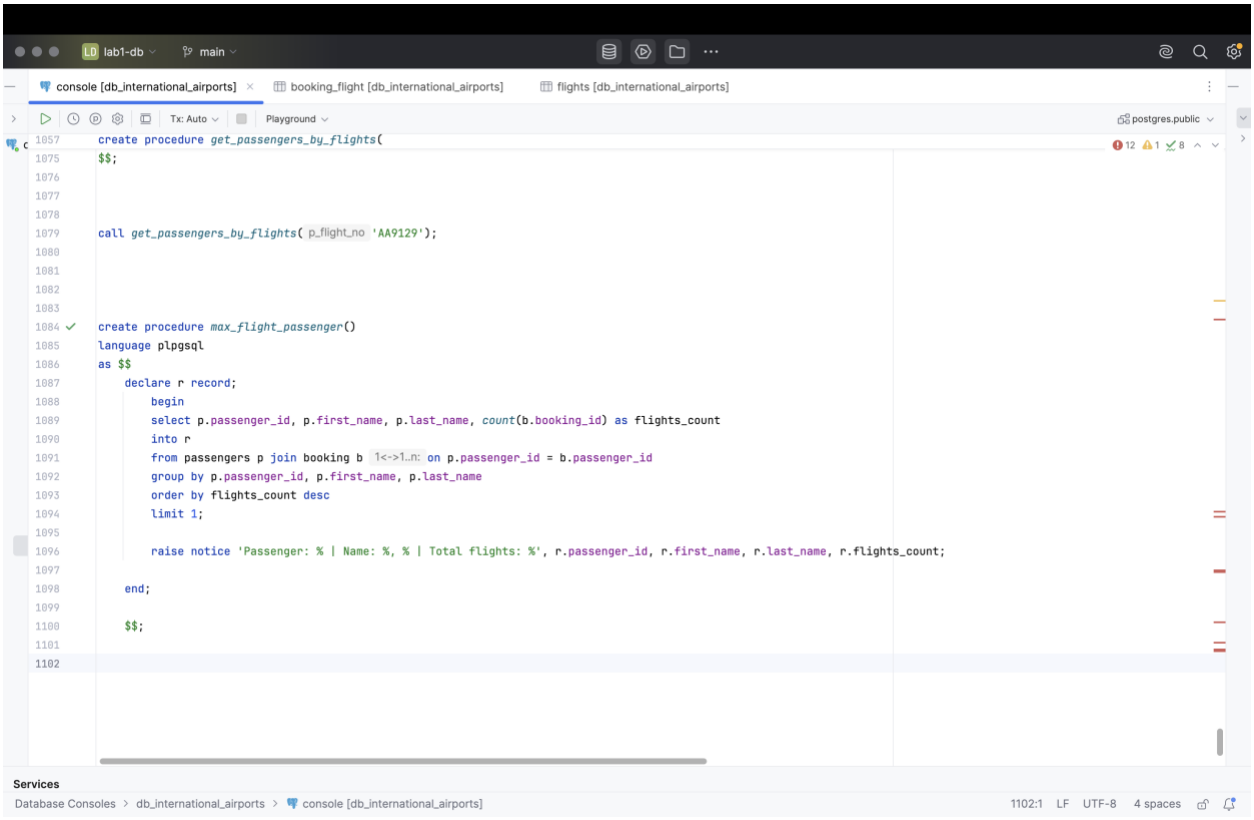


5. Create a stored procedure that lists all passengers for a given flight number.

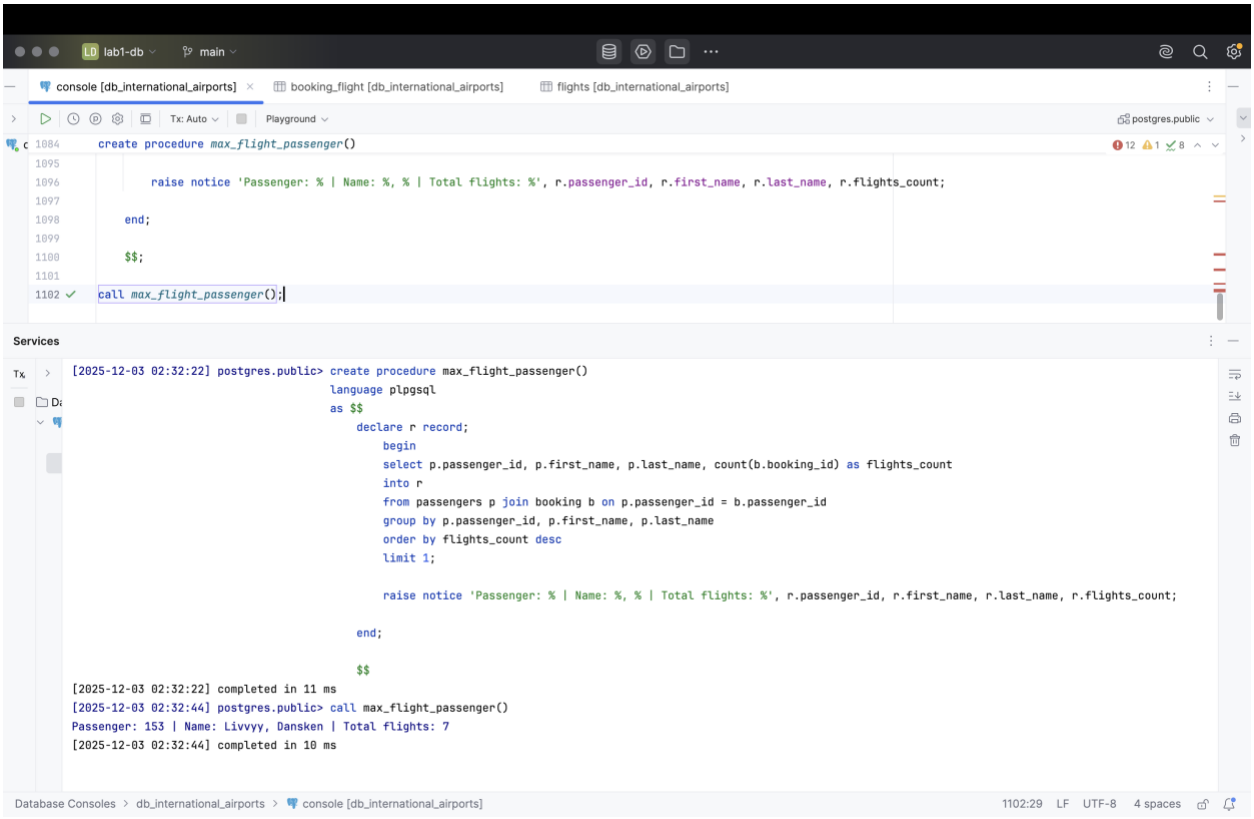




6. Create a stored procedure to find the passenger who has taken the greatest number of flights.



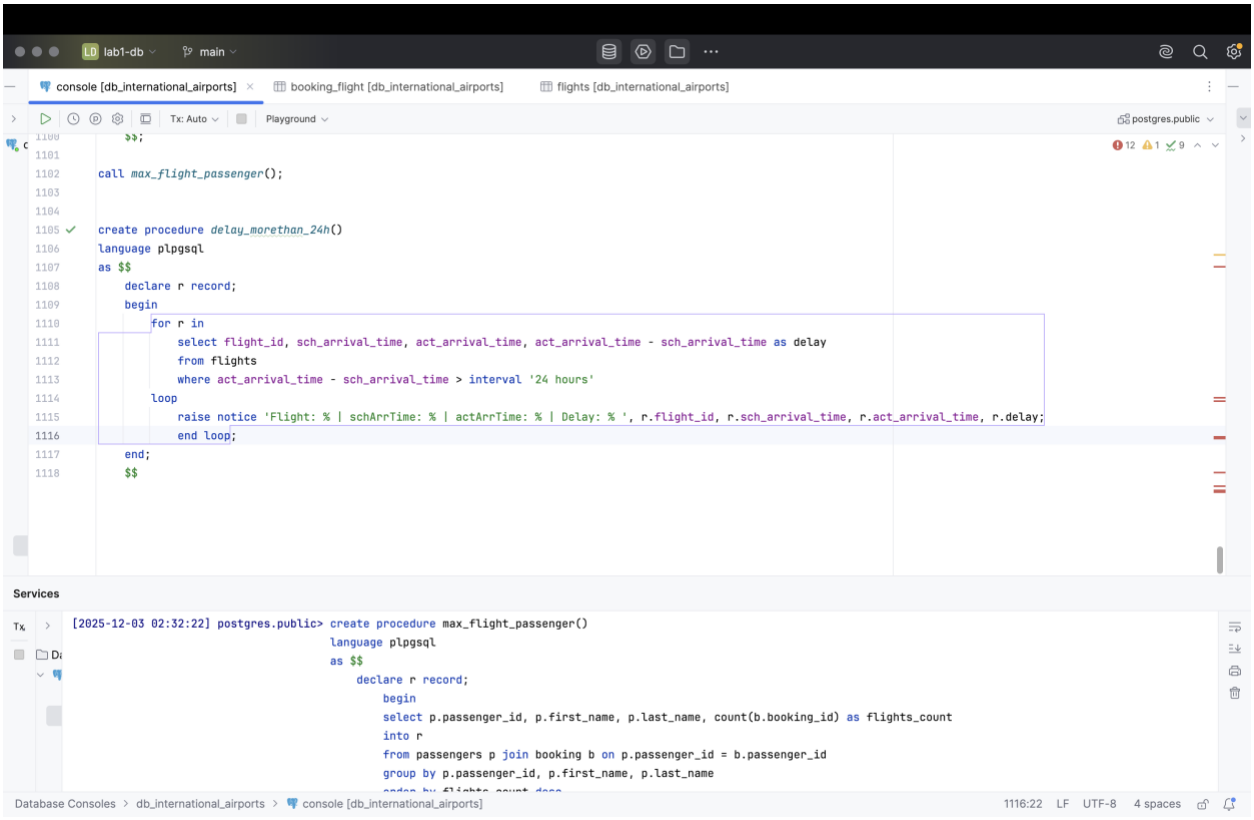
```
create procedure get_passengers_by_flights(
$$;


call get_passengers_by_flights( p_flight_no 'AA9129');



create procedure max_flight_passenger()
language plpgsql
as $$
    declare r record;
        begin
        select p.passenger_id, p.first_name, p.last_name, count(b.booking_id) as flights_count
        into r
        from passengers p join booking b 1<->1..n: on p.passenger_id = b.passenger_id
        group by p.passenger_id, p.first_name, p.last_name
        order by flights_count desc
        limit 1;

        raise notice 'Passenger: % | Name: %, % | Total flights: %', r.passenger_id, r.first_name, r.last_name, r.flights_count;

    end;

    $$;
```

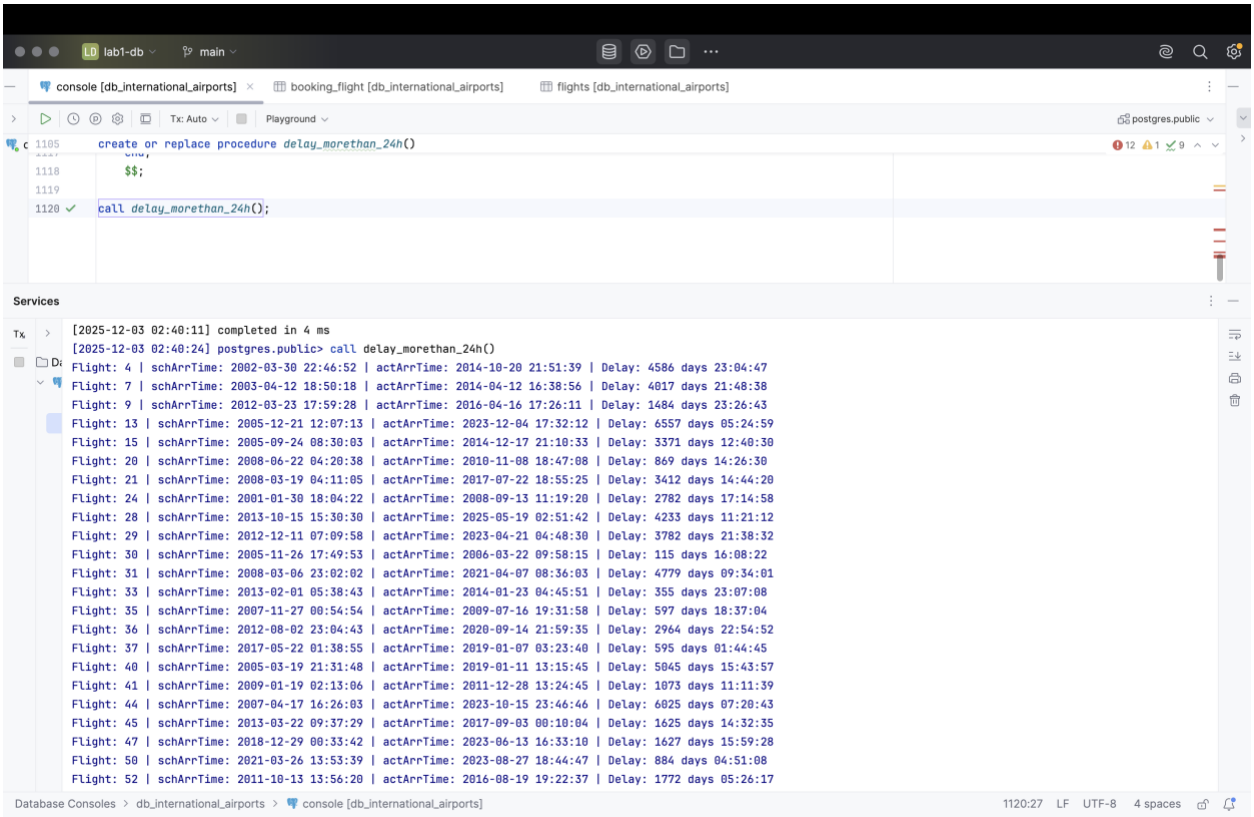


```
create procedure max_flight_passenger()

        raise notice 'Passenger: % | Name: %, % | Total flights: %', r.passenger_id, r.first_name, r.last_name, r.flights_count;

    end;

    $$;

call max_flight_passenger();
```

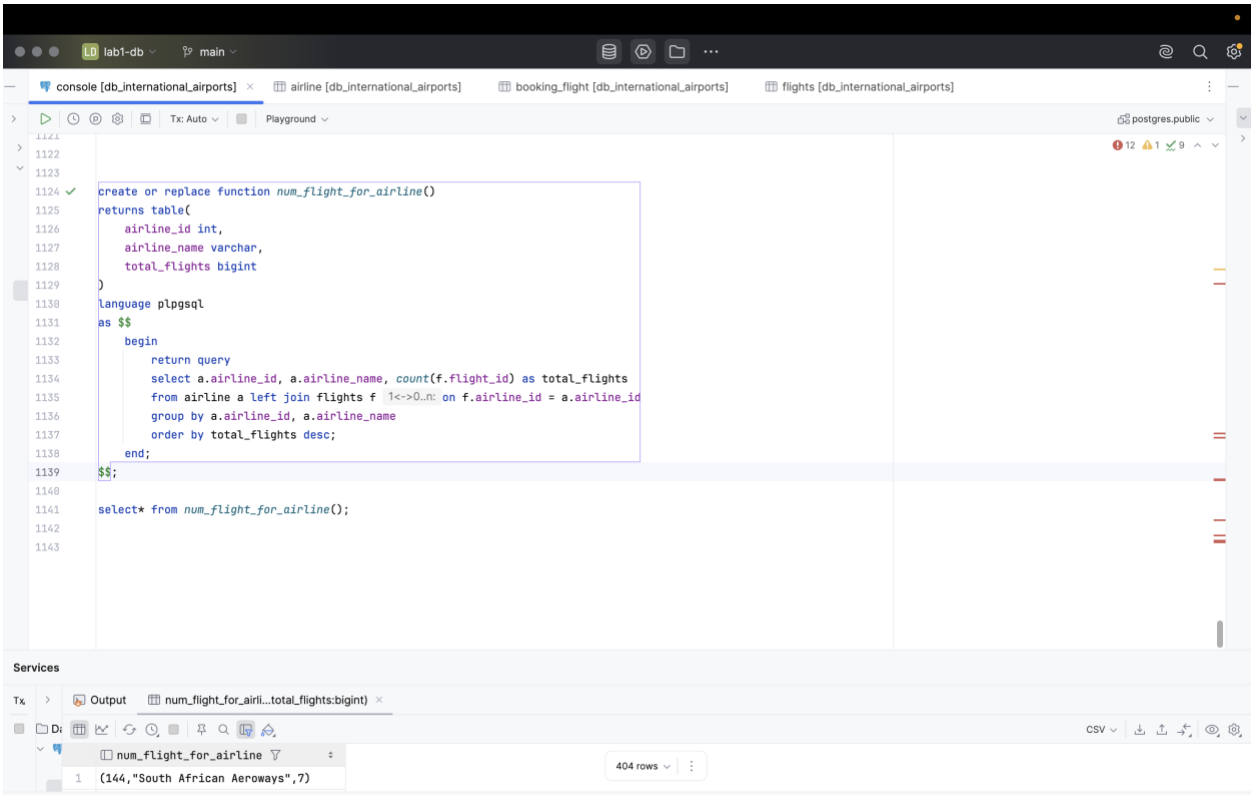
```
[2025-12-03 02:32:22] postgres.public> create procedure max_flight_passenger()
                                       language plpgsql
                                       as $$
                                           declare r record;
                                               begin
                                               select p.passenger_id, p.first_name, p.last_name, count(b.booking_id) as flights_count
                                               into r
                                               from passengers p join booking b on p.passenger_id = b.passenger_id
                                               group by p.passenger_id, p.first_name, p.last_name
                                               order by flights_count desc
                                               limit 1;

                                               raise notice 'Passenger: % | Name: %, % | Total flights: %', r.passenger_id, r.first_name, r.last_name, r.flights_count;

                                           end;

                                           $$
[2025-12-03 02:32:22] completed in 11 ms
[2025-12-03 02:32:44] postgres.public> call max_flight_passenger()
Passenger: 153 | Name: Livvyy, Dansken | Total flights: 7
[2025-12-03 02:32:44] completed in 10 ms
```

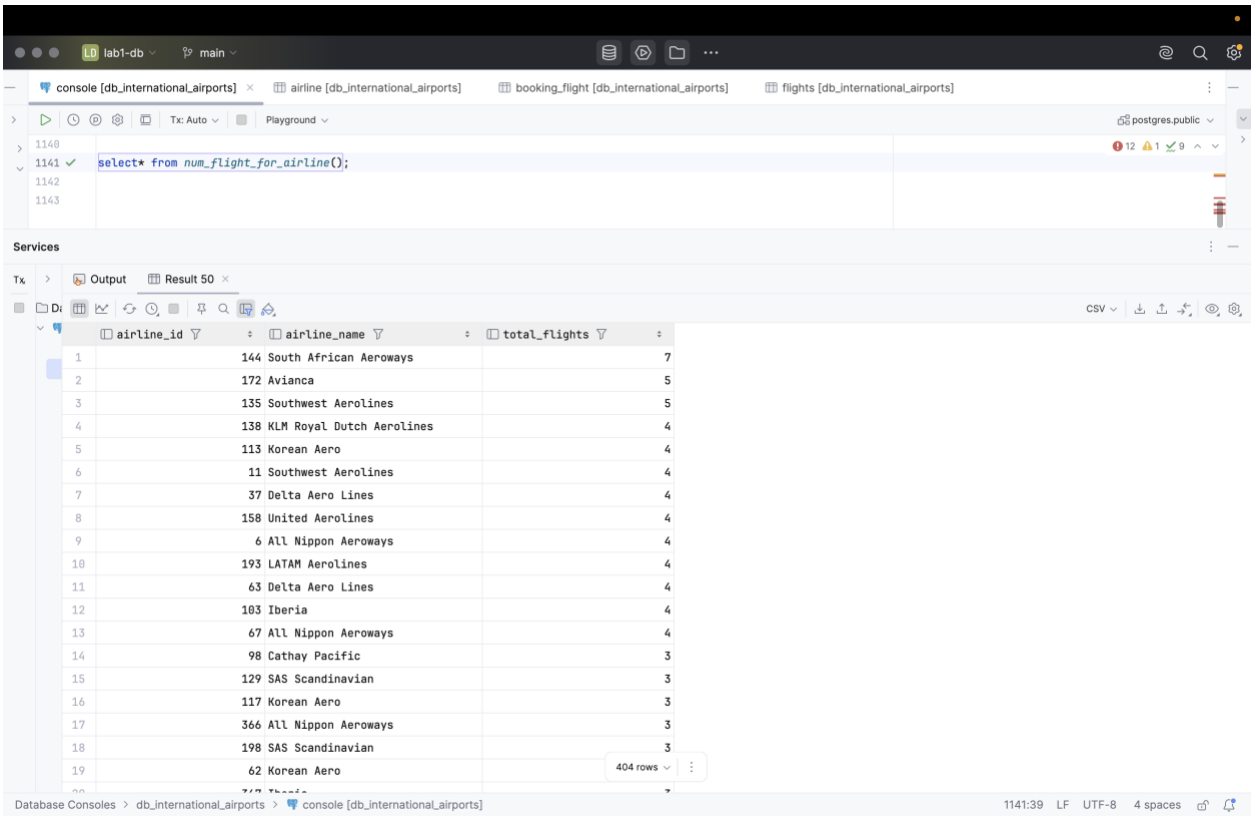
7. Create a stored procedure to find all flights that are delayed by more than 24 hours.



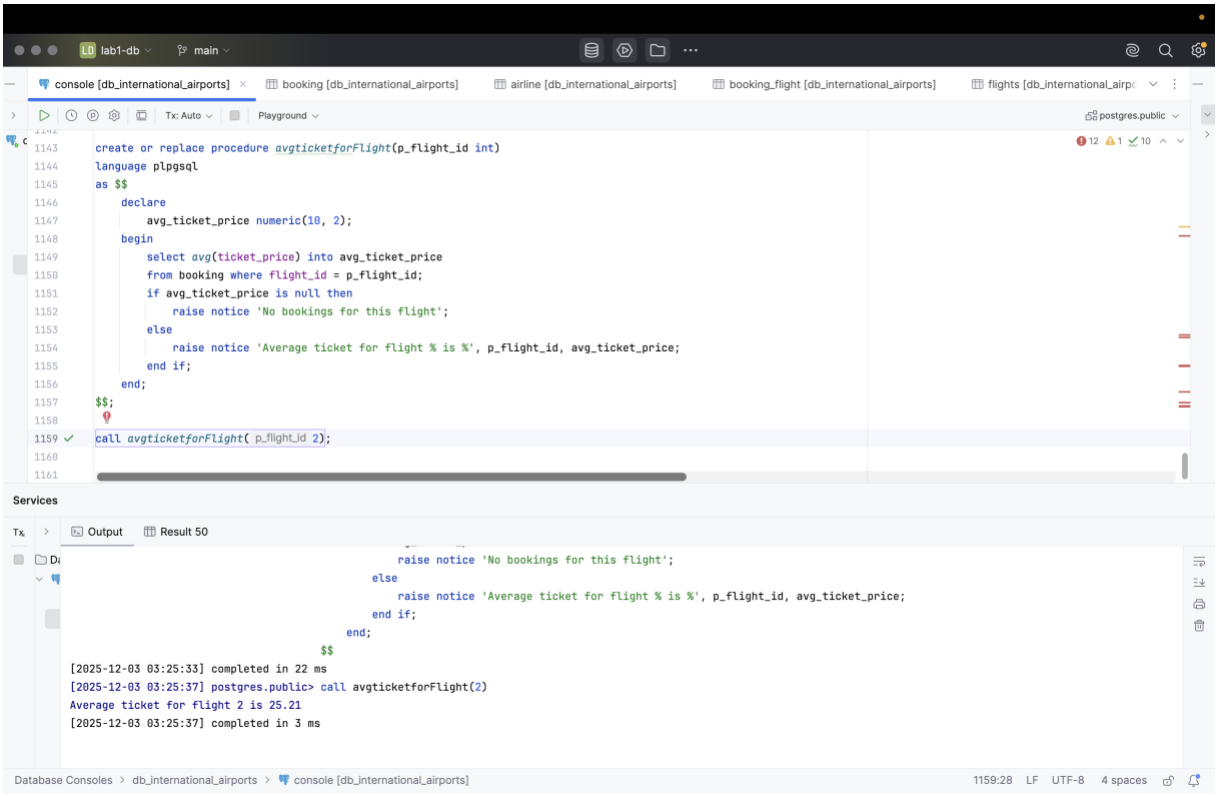


8. Create a function that counts the number of flights for each airline.





9. Create a stored procedure to calculate the average ticket price for a specific flight.



10. Create a stored procedure to find the flight with the highest ticket price. The procedure should return the flight number, the departure and arrival

airports, and the ticket price for the most expensive flight.