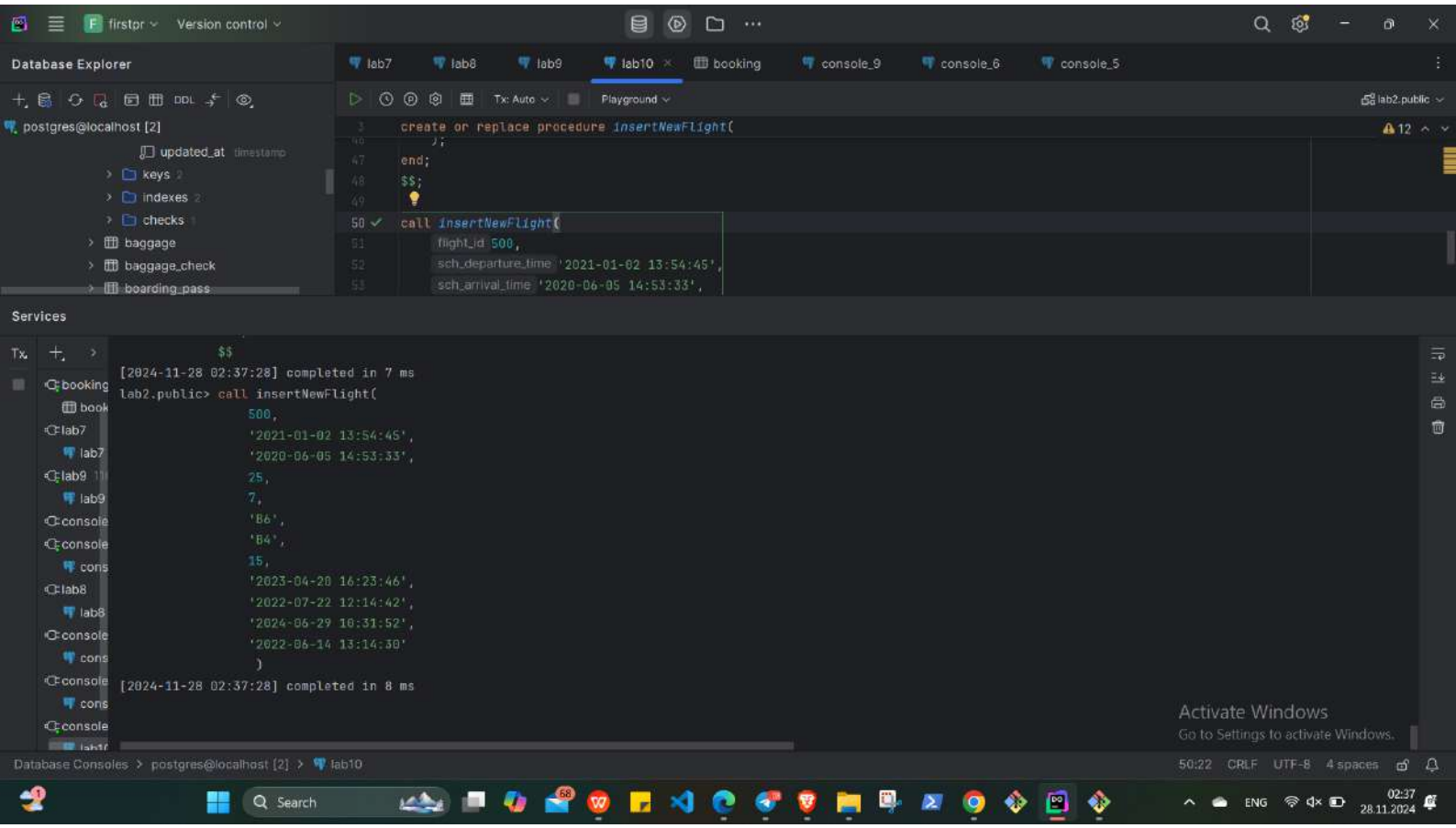
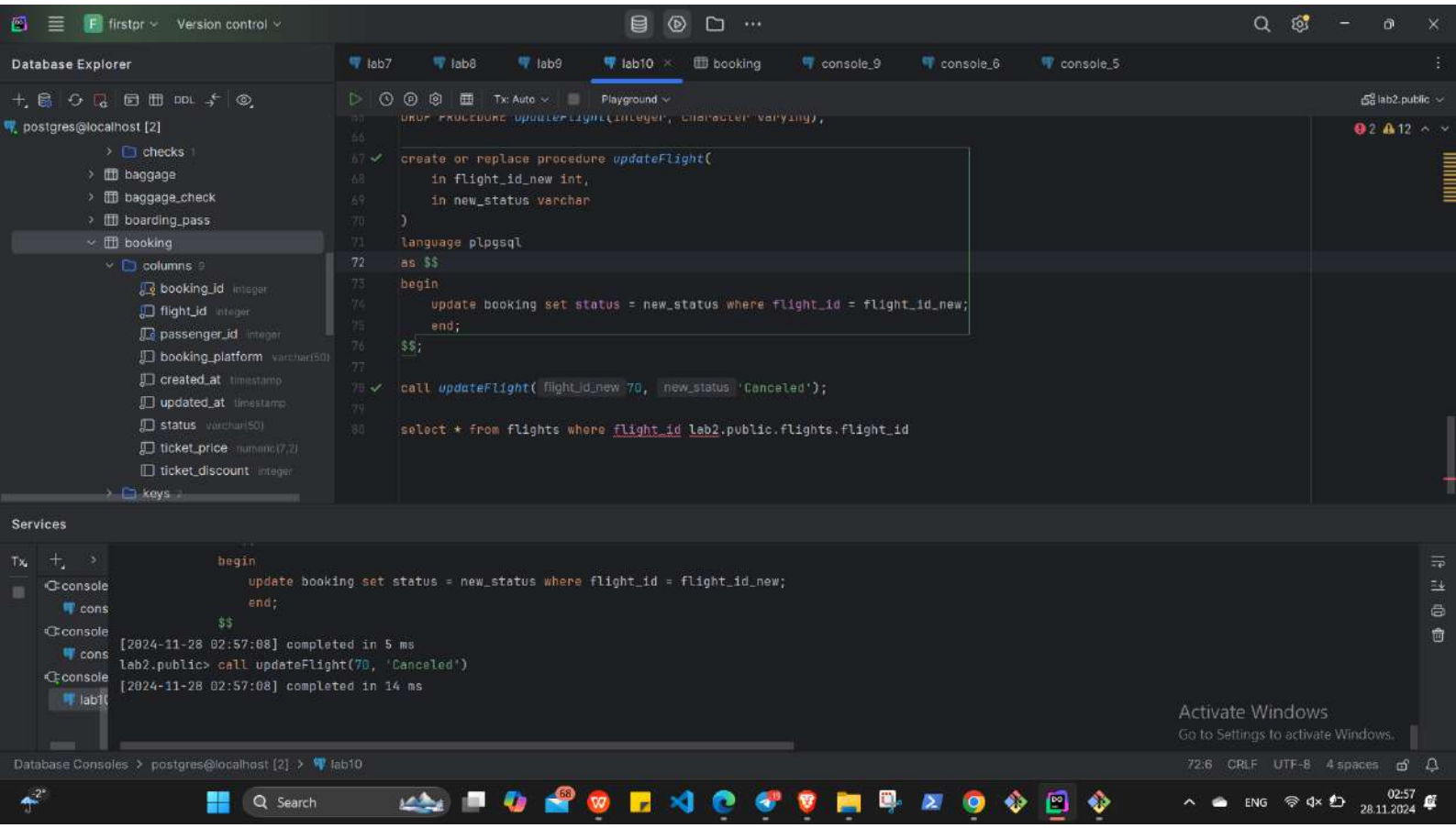
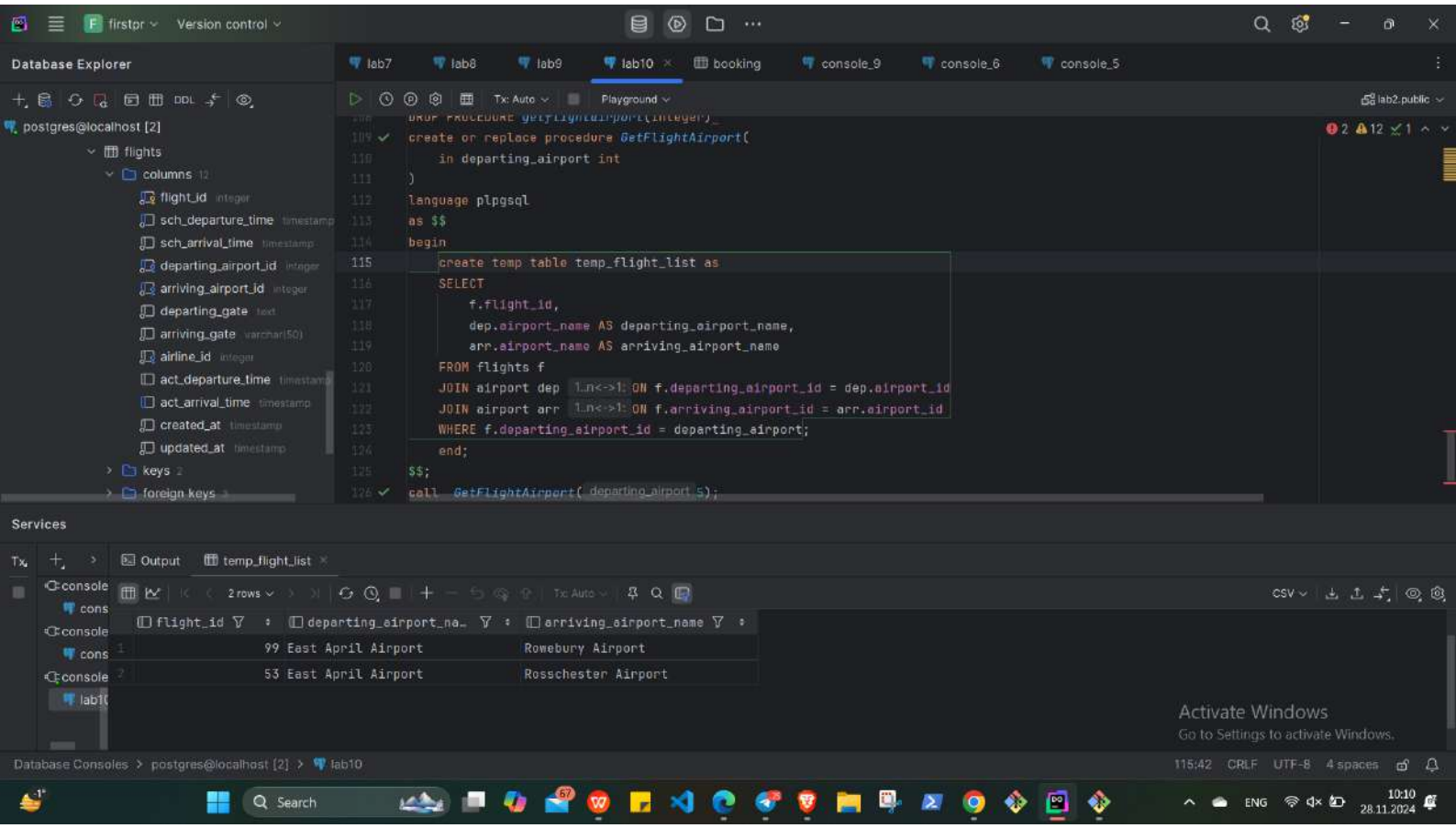
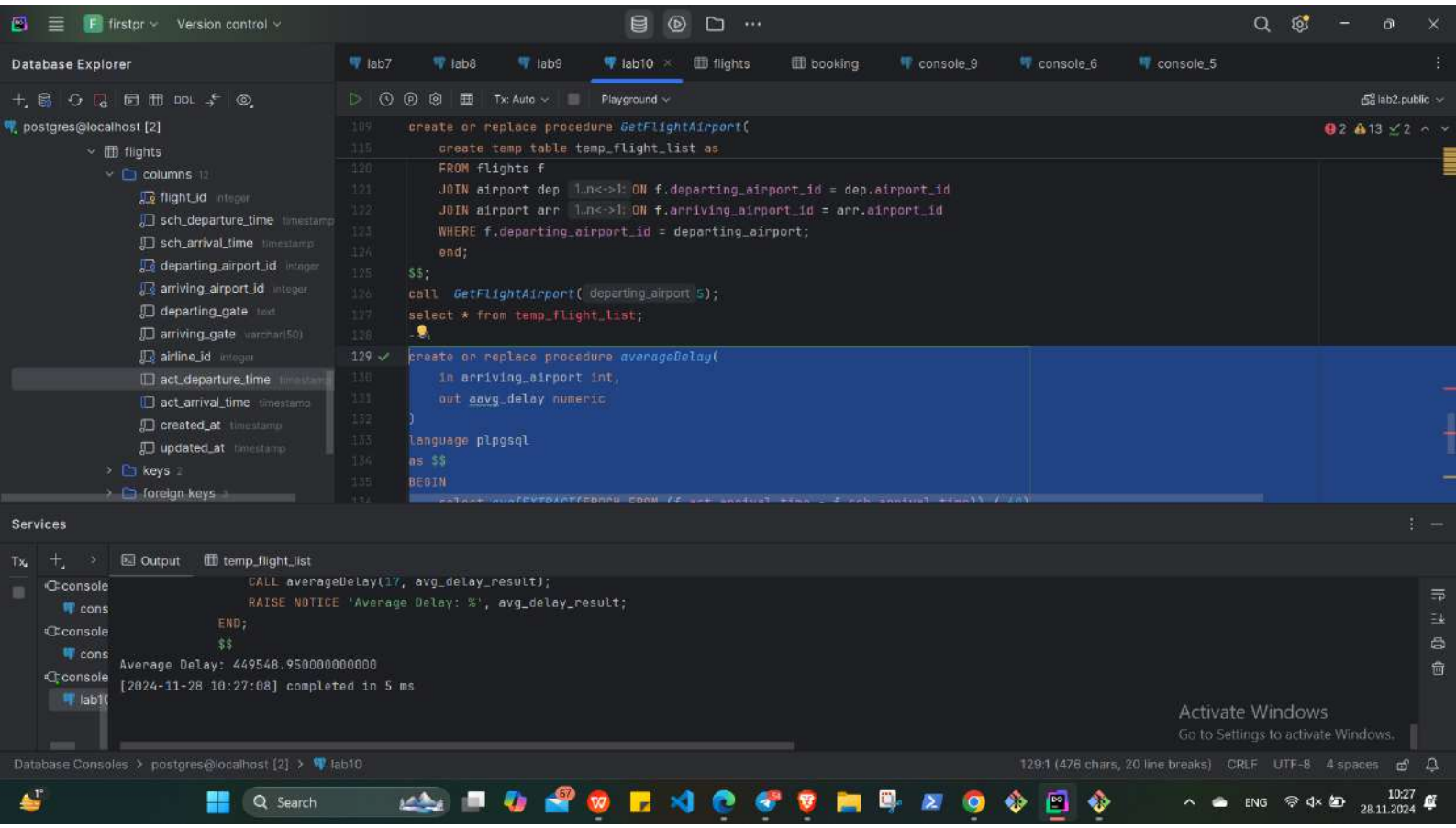
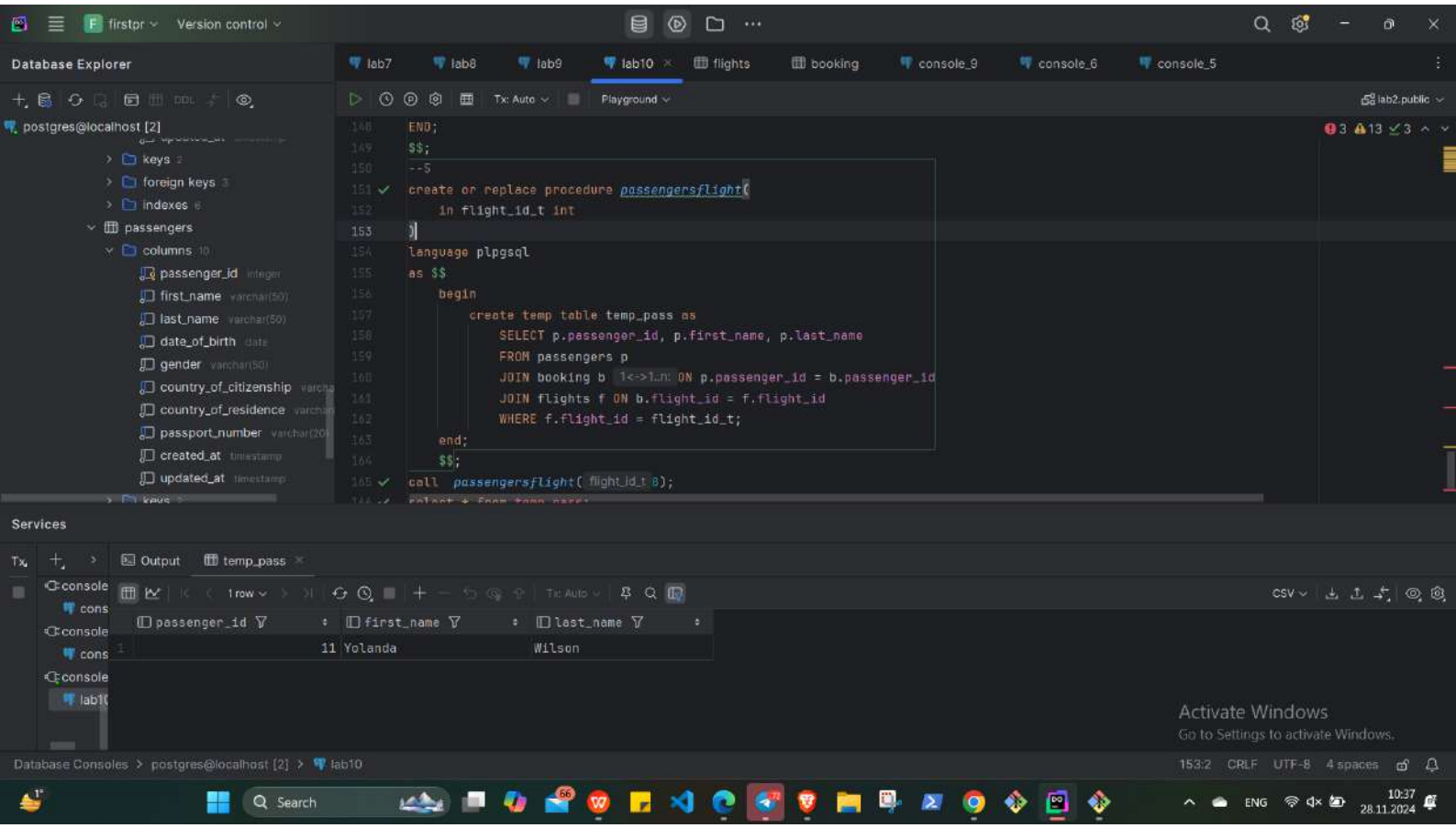

Database Explorer
lab7
lab8
lab9
lab10
booking
console_9
console_6
console_5
postgres@localhost [2]
updated_at timestamp
keys 2
indexes 2
checks 1
baggage
baggage_check
boarding_pass
Tx: Auto
Playground
lab2.public
create or replace procedure insertNewFlight(
end;
$$;
call insertNewFlight(
flight_id 500,
sch_departure_time '2021-01-02 13:54:45',
sch_arrival_time '2020-06-05 14:53:33',
Services
$$
[2024-11-28 02:37:28] completed in 7 ms
lab2.public> call insertNewFlight(
500,
'2021-01-02 13:54:45',
'2020-06-05 14:53:33',
25,
7,
'B6',
'B4',
15,
'2023-04-20 16:23:46',
'2022-07-22 12:14:42',
'2024-06-29 10:31:52',
'2022-06-14 13:14:30'
)
[2024-11-28 02:37:28] completed in 8 ms
Activate Windows
Go to Settings to activate Windows.
Database Consoles > postgres@localhost [2] > lab10
50:22 CRLF UTF-8 4 spaces
Search
ENG
02:37
28.11.2024

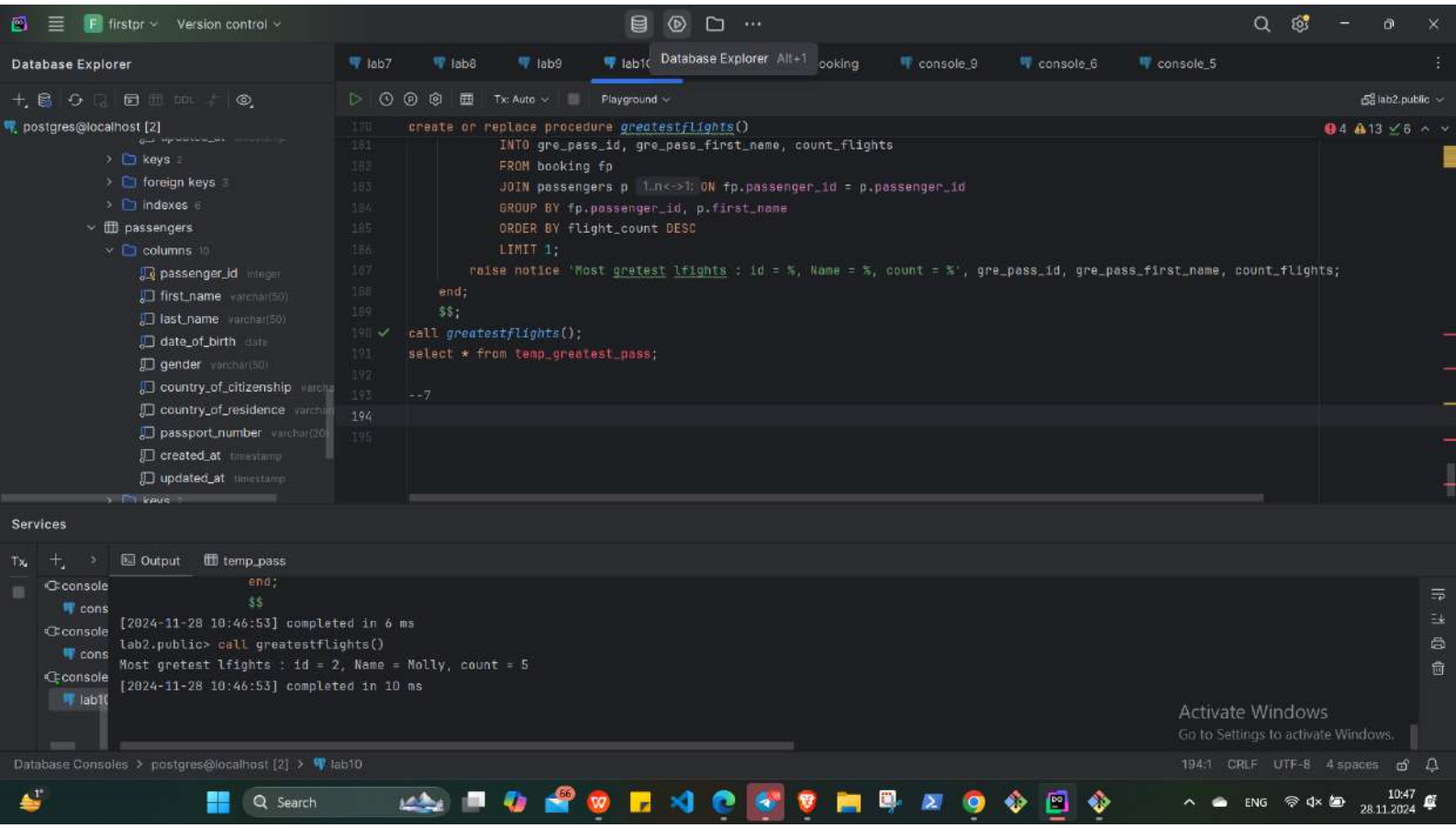

Database Explorer
lab7
lab8
lab9
lab10
booking
console_9
console_6
console_5
postgres@localhost [2]
checks
baggage
baggage_check
boarding_pass
booking
columns
booking_id integer
flight_id integer
passenger_id integer
booking_platform varchar(50)
created_at timestamp
updated_at timestamp
status varchar(50)
ticket_price numeric(7,2)
ticket_discount integer
keys
Tx: Auto
Playground
lab2.public
create or replace procedure updateFlight(
in flight_id_new int,
in new_status varchar
)
language plpgsql
as $$
begin
update booking set status = new_status where flight_id = flight_id_new;
end;
$$;
call updateFlight(flight_id_new 70, new_status 'Canceled');
select * from flights where flight_id lab2.public.flights.flight_id

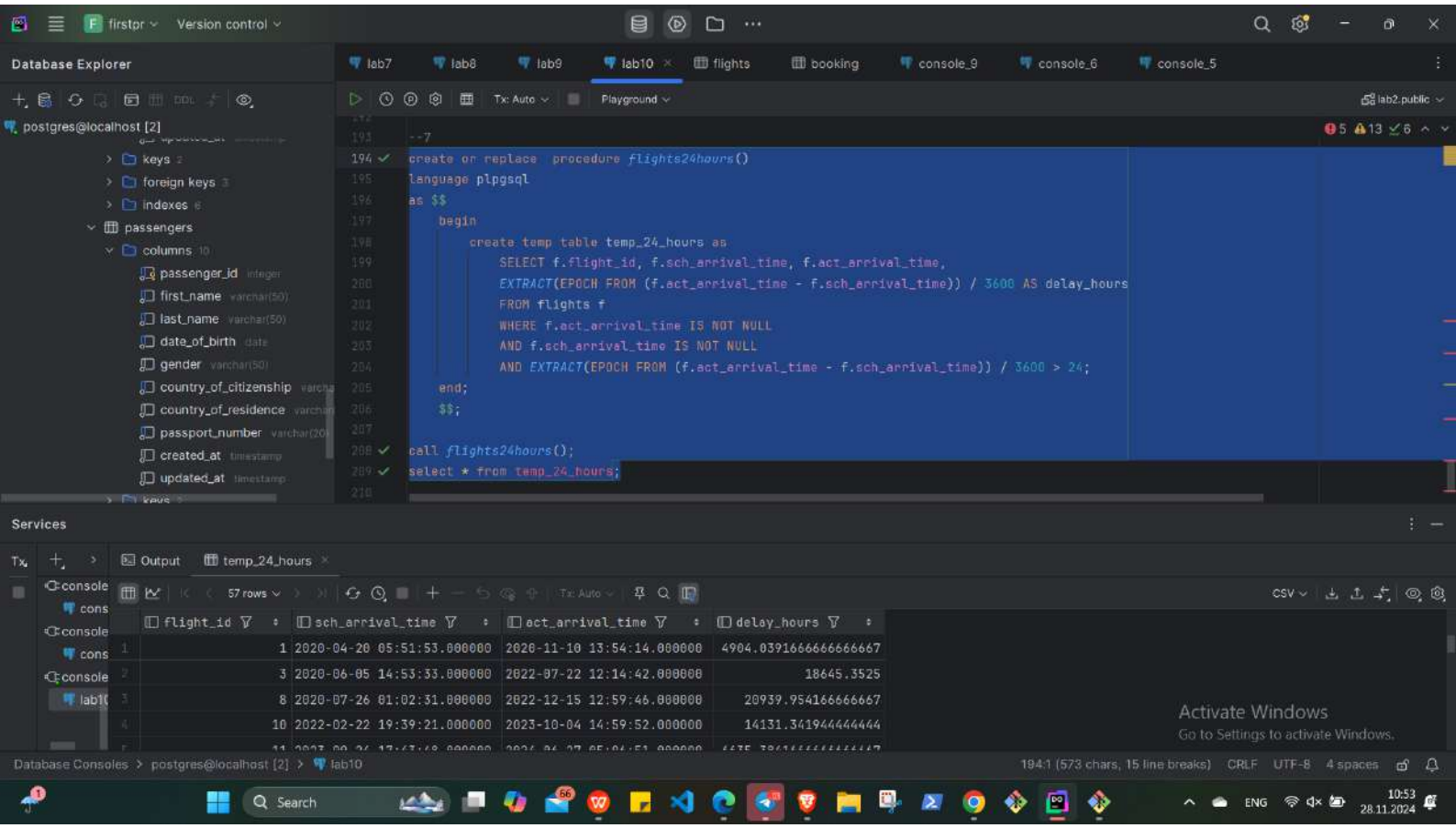
Services
begin
update booking set status = new_status where flight_id = flight_id_new;
end;
$$
[2024-11-28 02:57:08] completed in 5 ms
lab2.public> call updateFlight(70, 'Canceled')
[2024-11-28 02:57:08] completed in 14 ms
Activate Windows
Go to Settings to activate Windows.
Database Consoles > postgres@localhost [2] > lab10
72:6 CRLF UTF-8 4 spaces
Search
ENG
02:57
28.11.2024

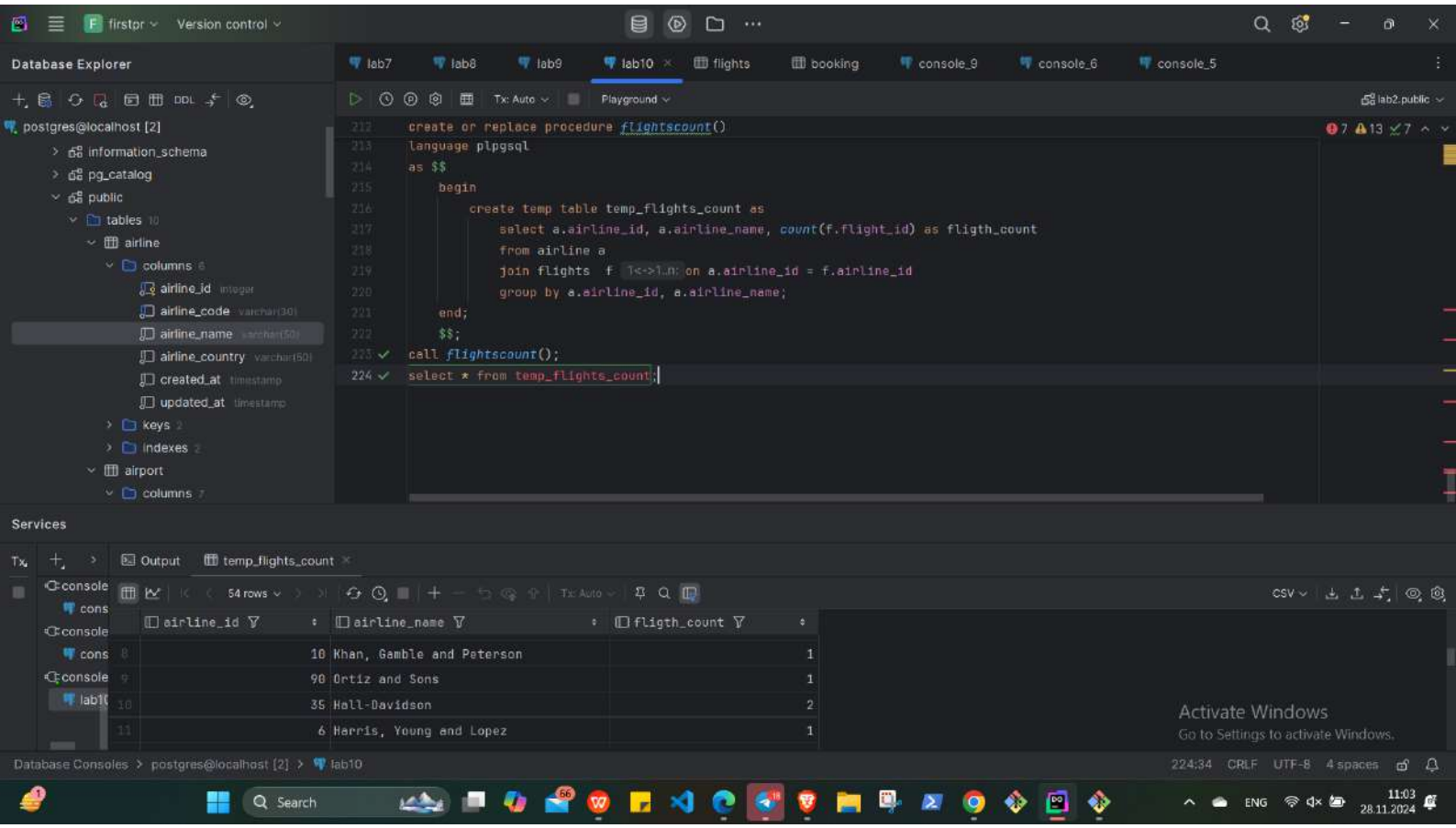

Database Explorer
lab7
lab8
lab9
lab10
booking
console_9
console_6
console_5
postgres@localhost [2]
flights
columns 12
flight_id integer
sch_departure_time timestamp
sch_arrival_time timestamp
departing_airport_id integer
arriving_airport_id integer
departing_gate text
arriving_gate varchar(50)
airline_id integer
act_departure_time timestamp
act_arrival_time timestamp
created_at timestamp
updated_at timestamp
keys 2
foreign keys 3
Playground
lab2.public
create or replace procedure GetFlightAirport(
in departing_airport int
)
language plpgsql
as $$
begin
create temp table temp_flight_list as
SELECT
f.flight_id,
dep.airport_name AS departing_airport_name,
arr.airport_name AS arriving_airport_name
FROM flights f
JOIN airport dep ON f.departing_airport_id = dep.airport_id
JOIN airport arr ON f.arriving_airport_id = arr.airport_id
WHERE f.departing_airport_id = departing_airport;
end;
$$;
call GetFlightAirport(departing_airport 5);
Services
Output
temp_flight_list
2 rows
CSV
flight_id
departing_airport_na...
arriving_airport_name
1 99 East April Airport Rowebury Airport
2 53 East April Airport Rosschester Airport
Activate Windows
Go to Settings to activate Windows.
Database Consoles > postgres@localhost [2] > lab10
115:42 CRLF UTF-8 4 spaces
Search
ENG
10:10
28.11.2024

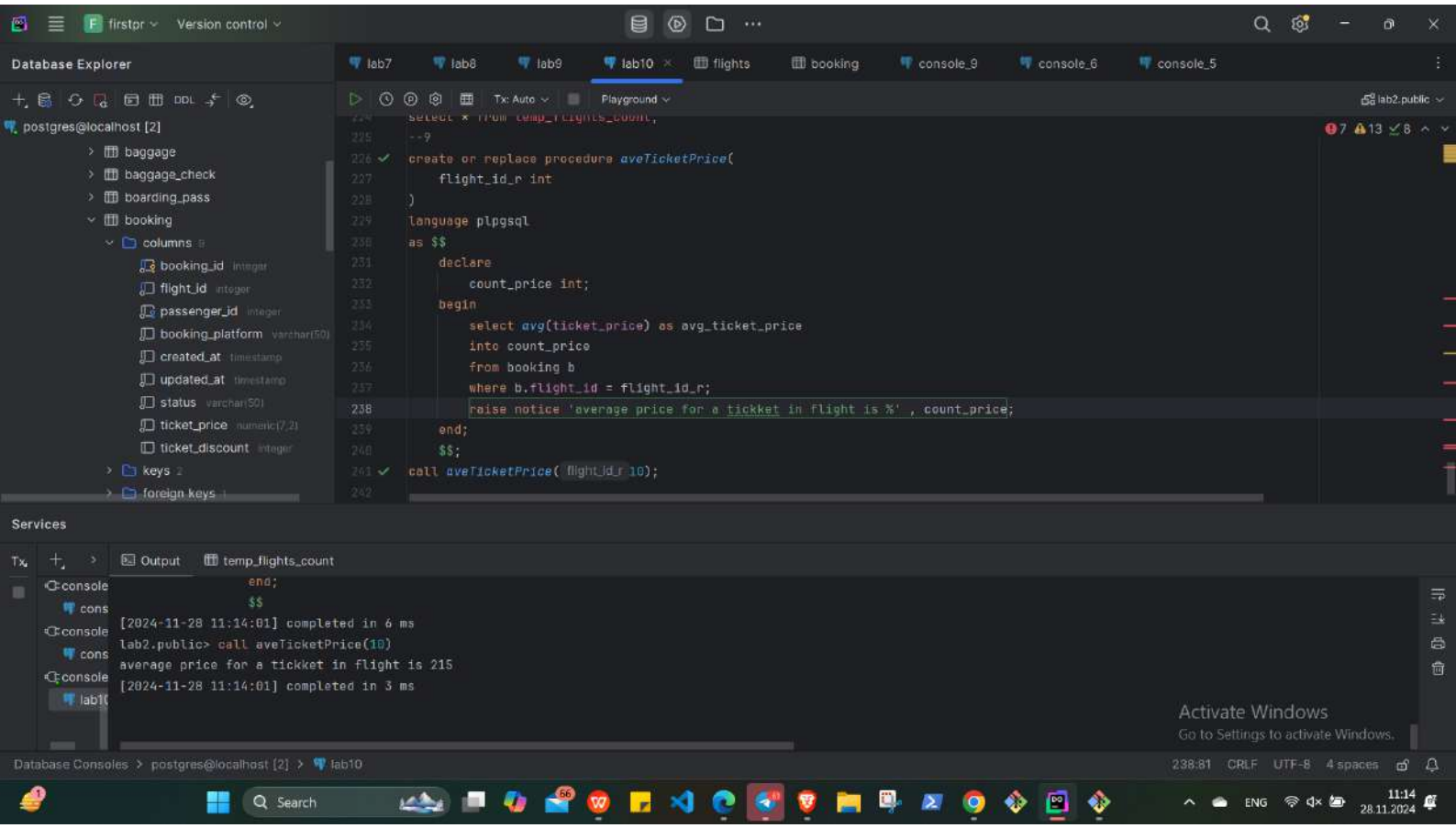

Database Explorer
lab7
lab8
lab9
lab10
flights
booking
console_9
console_6
console_5
postgres@localhost [2]
flights
columns 12
flight_id integer
sch_departure_time timestamp
sch_arrival_time timestamp
departing_airport_id integer
arriving_airport_id integer
departing_gate text
arriving_gate varchar(50)
airline_id integer
act_departure_time timestamp
act_arrival_time timestamp
created_at timestamp
updated_at timestamp
keys 2
foreign keys 3
Tx: Auto
Playground
lab2.public
create or replace procedure GetFlightAirport(
create temp table temp_flight_list as
FROM flights f
JOIN airport dep 1..n<->1: ON f.departing_airport_id = dep.airport_id
JOIN airport arr 1..n<->1: ON f.arriving_airport_id = arr.airport_id
WHERE f.departing_airport_id = departing_airport;
end;
$$;
call GetFlightAirport(departing_airport 5);
select * from temp_flight_list;
create or replace procedure averageDelay(
in arriving_airport int,
out avg_delay numeric
)
language plpgsql
as $$
BEGIN
Services
Output
temp_flight_list
CALL averageDelay(17, avg_delay_result);
RAISE NOTICE 'Average Delay: %', avg_delay_result;
END;
$$
Average Delay: 449548.950000000000
[2024-11-28 10:27:08] completed in 5 ms
Activate Windows
Go to Settings to activate Windows.
Database Consoles > postgres@localhost [2] > lab10
129:1 (476 chars, 20 line breaks) CRLF UTF-8 4 spaces
Search
ENG
10:27
28.11.2024

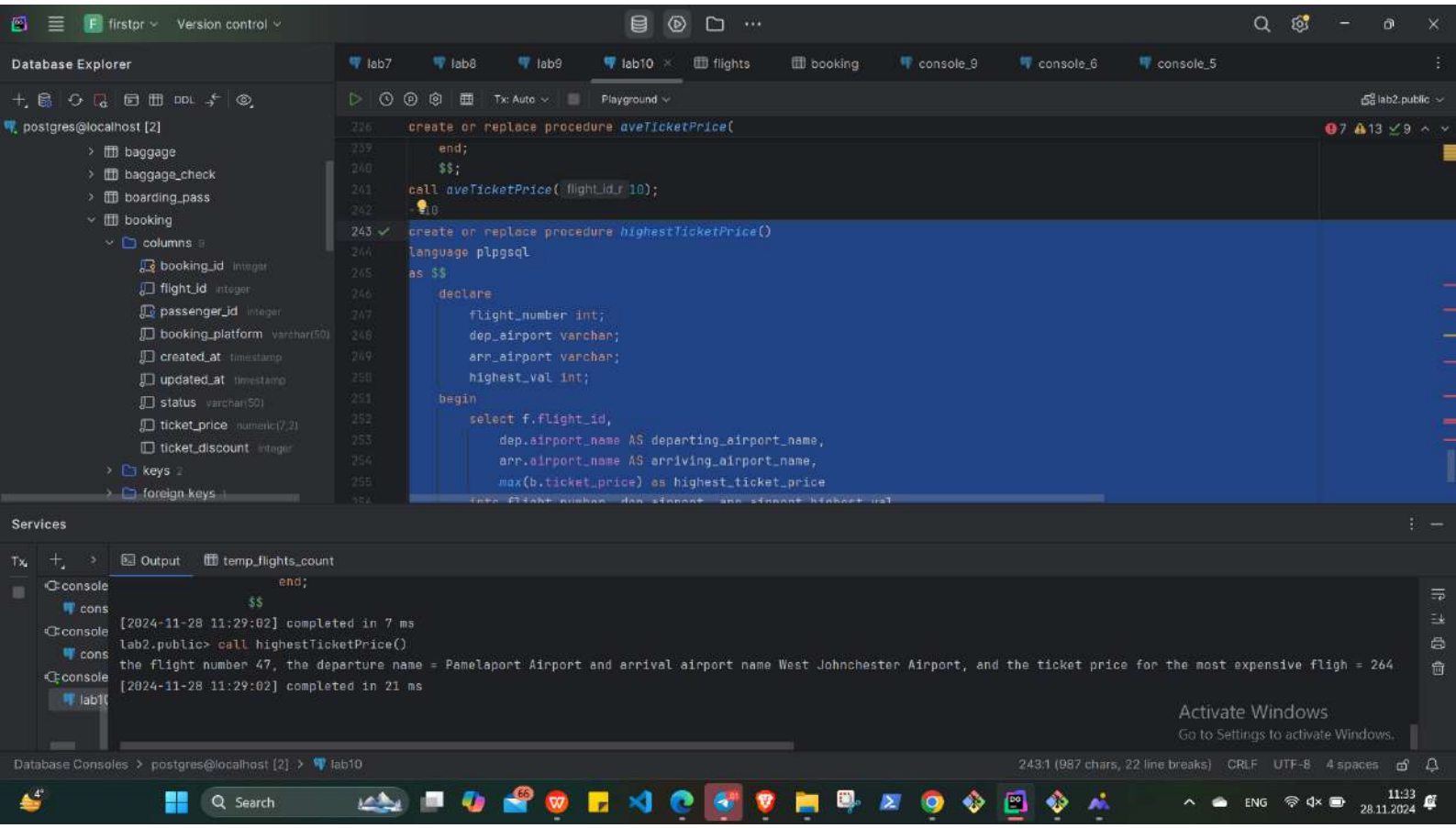

firstpr
Version control
Database Explorer
lab7
lab8
lab9
lab10
flights
booking
console_9
console_6
console_5
Playground
postgres@localhost [2]
keys
foreign keys
indexes
passengers
columns
passenger_id
first_name
last_name
date_of_birth
gender
country_of_citizenship
country_of_residence
passport_number
created_at
updated_at
END;
$$;
--5
create or replace procedure passengersflight(
in flight_id_t int
)
language plpgsql
as $$
begin
create temp table temp_pass as
SELECT p.passenger_id, p.first_name, p.last_name
FROM passengers p
JOIN booking b ON p.passenger_id = b.passenger_id
JOIN flights f ON b.flight_id = f.flight_id
WHERE f.flight_id = flight_id_t;
end;
$$;
call passengersflight(flight_id_t 8);
Services
Output
temp_pass
1 row
CSV
passenger_id
first_name
last_name
11
Yolanda
Wilson
Activate Windows
Go to Settings to activate Windows.
Database Consoles > postgres@localhost [2] > lab10
153:2
CRLF
UTF-8
4 spaces
Search
ENG
10:37
28.11.2024


Database Explorer
lab7
lab8
lab9
Database Explorer Alt+1
console_9
console_6
console_5
postgres@localhost [2]
keys 2
foreign keys 3
indexes 6
passengers
columns 10
passenger_id integer
first_name varchar(50)
last_name varchar(50)
date_of_birth date
gender varchar(50)
country_of_citizenship
country_of_residence
passport_number
created_at timestamp
updated_at timestamp
Tx: Auto
Playground
lab2.public
create or replace procedure greatestflights()
INTO gre_pass_id, gre_pass_first_name, count_flights
FROM booking fp
JOIN passengers p 1..n<->1; ON fp.passenger_id = p.passenger_id
GROUP BY fp.passenger_id, p.first_name
ORDER BY flight_count DESC
LIMIT 1;
raise notice 'Most gretest lfights : id = %, Name = %, count = %', gre_pass_id, gre_pass_first_name, count_flights;
end;
$$;
call greatestflights();
select * from temp_greatest_pass;
--7
Services
Output
temp_pass
end;
$$
[2024-11-28 10:46:53] completed in 6 ms
lab2.public> call greatestflights()
Most gretest lfights : id = 2, Name = Molly, count = 5
[2024-11-28 10:46:53] completed in 10 ms
Activate Windows
Go to Settings to activate Windows.
Database Consoles > postgres@localhost [2] > lab10
194:1 CRLF UTF-8 4 spaces
Search
ENG
10:47
28.11.2024


firstpr
Version control
Database Explorer
lab7
lab8
lab9
lab10
flights
booking
console_9
console_6
console_5
Playground
lab2.public
postgres@localhost [2]
keys 2
foreign keys 3
indexes 6
passengers
columns 10
passenger_id integer
first_name varchar(50)
last_name varchar(50)
date_of_birth date
gender varchar(50)
country_of_citizenship
country_of_residence
passport_number
created_at timestamp
updated_at timestamp
--7
create or replace procedure flights24hours()
language plpgsql
as $$
begin
create temp table temp_24_hours as
SELECT f.flight_id, f.sch_arrival_time, f.act_arrival_time,
EXTRACT(EPOCH FROM (f.act_arrival_time - f.sch_arrival_time)) / 3600 AS delay_hours
FROM flights f
WHERE f.act_arrival_time IS NOT NULL
AND f.sch_arrival_time IS NOT NULL
AND EXTRACT(EPOCH FROM (f.act_arrival_time - f.sch_arrival_time)) / 3600 > 24;
end;
$$;
call flights24hours();
select * from temp_24_hours;
Services
Output
temp_24_hours
57 rows
CSV
flight_id
sch_arrival_time
act_arrival_time
delay_hours
1 2020-04-20 05:51:53.000000 2020-11-10 13:54:14.000000 4904.0391666666666667
3 2020-06-05 14:53:33.000000 2022-07-22 12:14:42.000000 18645.3525
8 2020-07-26 01:02:31.000000 2022-12-15 12:59:46.000000 20939.954166666667
10 2022-02-22 19:39:21.000000 2023-10-04 14:59:52.000000 14131.341944444444
Activate Windows
Go to Settings to activate Windows.
Database Consoles > postgres@localhost [2] > lab10
194:1 (573 chars, 15 line breaks)
CRLF
UTF-8
4 spaces
Search
ENG
10:53
28.11.2024


Database Explorer
lab7
lab8
lab9
lab10
flights
booking
console_9
console_6
console_5
postgres@localhost [2]
information_schema
pg_catalog
public
tables
airline
columns
airline_id
airline_code
airline_name
airline_country
created_at
updated_at
keys
indexes
airport
create or replace procedure flightscount()
language plpgsql
as $$
begin
create temp table temp_flights_count as
select a.airline_id, a.airline_name, count(f.flight_id) as fligth_count
from airline a
join flights f on a.airline_id = f.airline_id
group by a.airline_id, a.airline_name;
end;
$$;
call flightscount();
select * from temp_flights_count;
Services
Output
temp_flights_count
54 rows
airline_id
airline_name
fligth_count
10 Khan, Gamble and Peterson 1
90 Ortiz and Sons 1
35 Hall-Davidson 2
6 Harris, Young and Lopez 1
Activate Windows
Go to Settings to activate Windows.
Database Consoles > postgres@localhost [2] > lab10
224:34 CRLF UTF-8 4 spaces
Search
ENG
11:03
28.11.2024


firstpr
Version control
Database Explorer
lab7
lab8
lab9
lab10
flights
booking
console_9
console_6
console_5
Tx: Auto
Playground
lab2.public
postgres@localhost [2]
baggage
baggage_check
boarding_pass
booking
columns
booking_id integer
flight_id integer
passenger_id integer
booking_platform varchar(50)
created_at timestamp
updated_at timestamp
status varchar(50)
ticket_price numeric(7,2)
ticket_discount integer
keys
foreign keys
--9
create or replace procedure aveTicketPrice(
flight_id_r int
)
language plpgsql
as $$
declare
count_price int;
begin
select avg(ticket_price) as avg_ticket_price
into count_price
from booking b
where b.flight_id = flight_id_r;
raise notice 'average price for a tickket in flight is %' , count_price;
end;
$$;
call aveTicketPrice(flight_id_r 10);
Services
Output
temp_flights_count
end;
$$
[2024-11-28 11:14:01] completed in 6 ms
lab2.public> call aveTicketPrice(10)
average price for a tickket in flight is 215
[2024-11-28 11:14:01] completed in 3 ms
Activate Windows
Go to Settings to activate Windows.
Database Consoles > postgres@localhost [2] > lab10
238:81
CRLF
UTF-8
4 spaces
Search
ENG
11:14
28.11.2024


Database Explorer
lab7
lab8
lab9
lab10
flights
booking
console_9
console_6
console_5
postgres@localhost [2]
baggage
baggage_check
boarding_pass
booking
columns
booking_id integer
flight_id integer
passenger_id integer
booking_platform varchar(50)
created_at timestamp
updated_at timestamp
status varchar(50)
ticket_price numeric(7,2)
ticket_discount integer
keys 2
foreign keys 1
Playground
lab2.public
create or replace procedure aveTicketPrice(
end;
$$;
call aveTicketPrice(flight_id_r 10);
create or replace procedure highestTicketPrice()
language plpgsql
as $$
declare
flight_number int;
dep_airport varchar;
arr_airport varchar;
highest_val int;
begin
select f.flight_id,
dep.airport_name AS departing_airport_name,
arr.airport_name AS arriving_airport_name,
max(b.ticket_price) as highest_ticket_price
Services
Output
temp_flights_count
end;
$$
[2024-11-28 11:29:02] completed in 7 ms
lab2.public> call highestTicketPrice()
the flight number 47, the departure name = Pamelaport Airport and arrival airport name West Johnchester Airport, and the ticket price for the most expensive fligh = 264
[2024-11-28 11:29:02] completed in 21 ms
Activate Windows
Go to Settings to activate Windows.
Database Consoles > postgres@localhost [2] > lab10
243:1 (987 chars, 22 line breaks) CRLF UTF-8 4 spaces
Search
ENG
11:33
28.11.2024