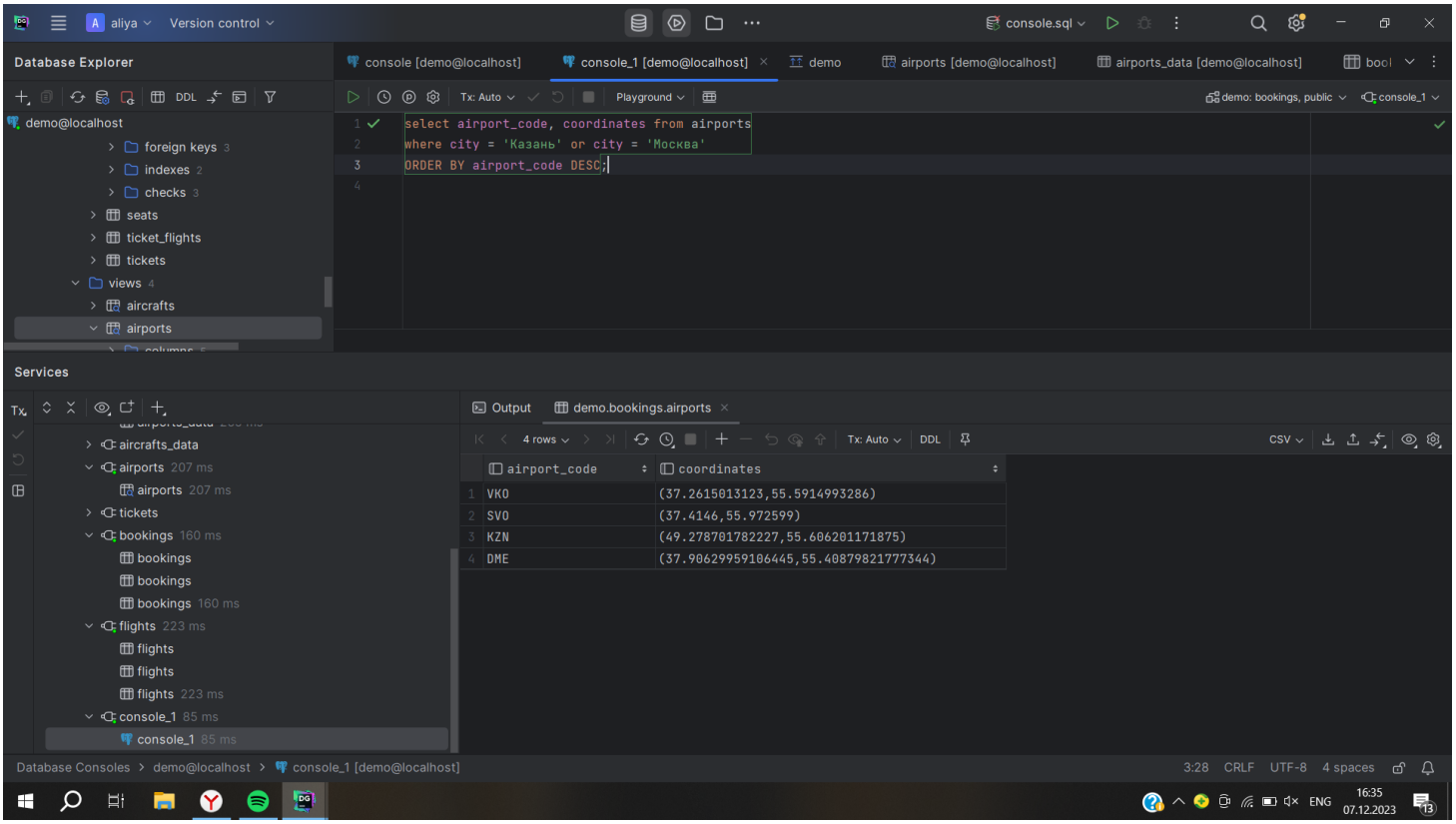
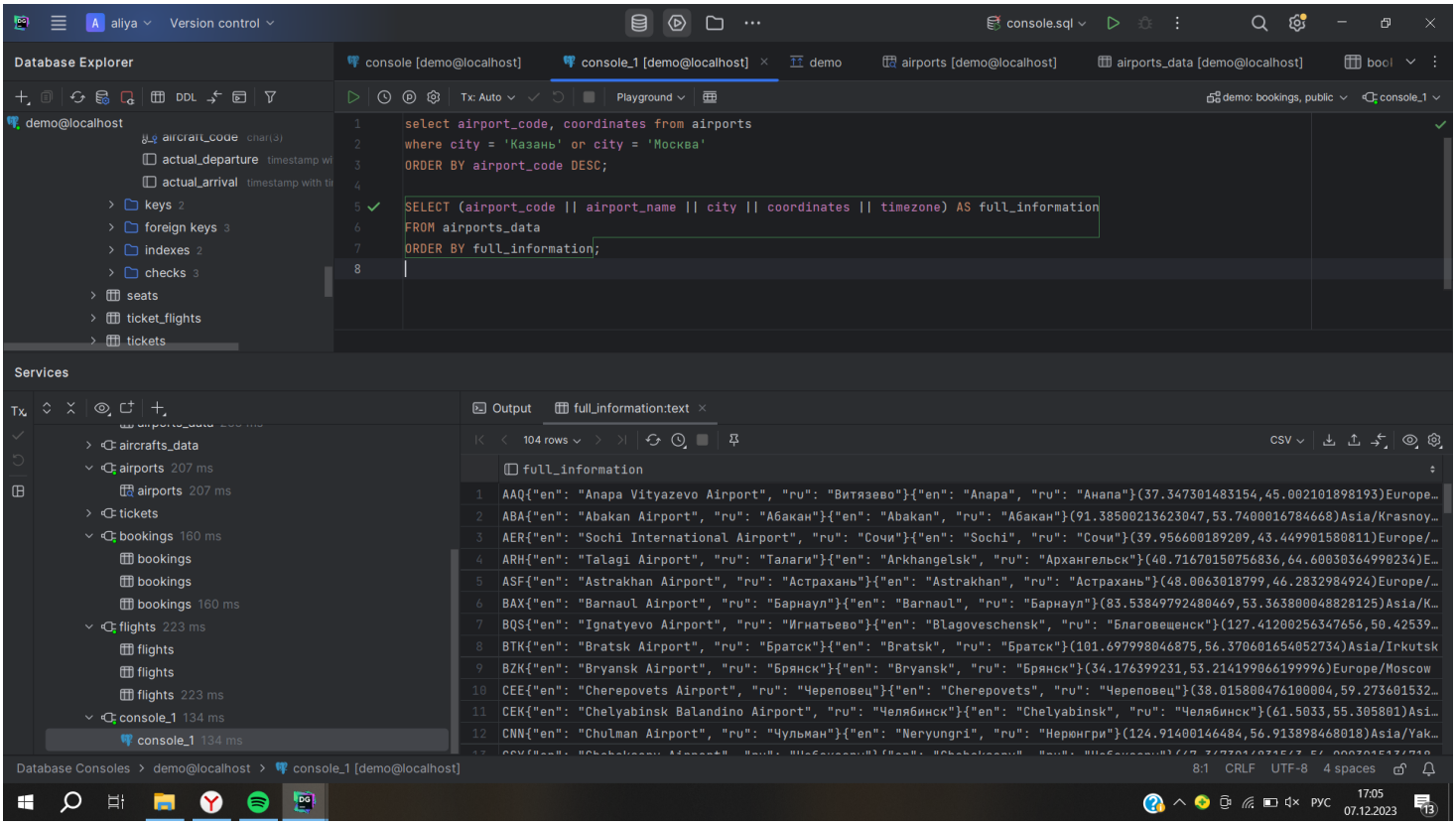
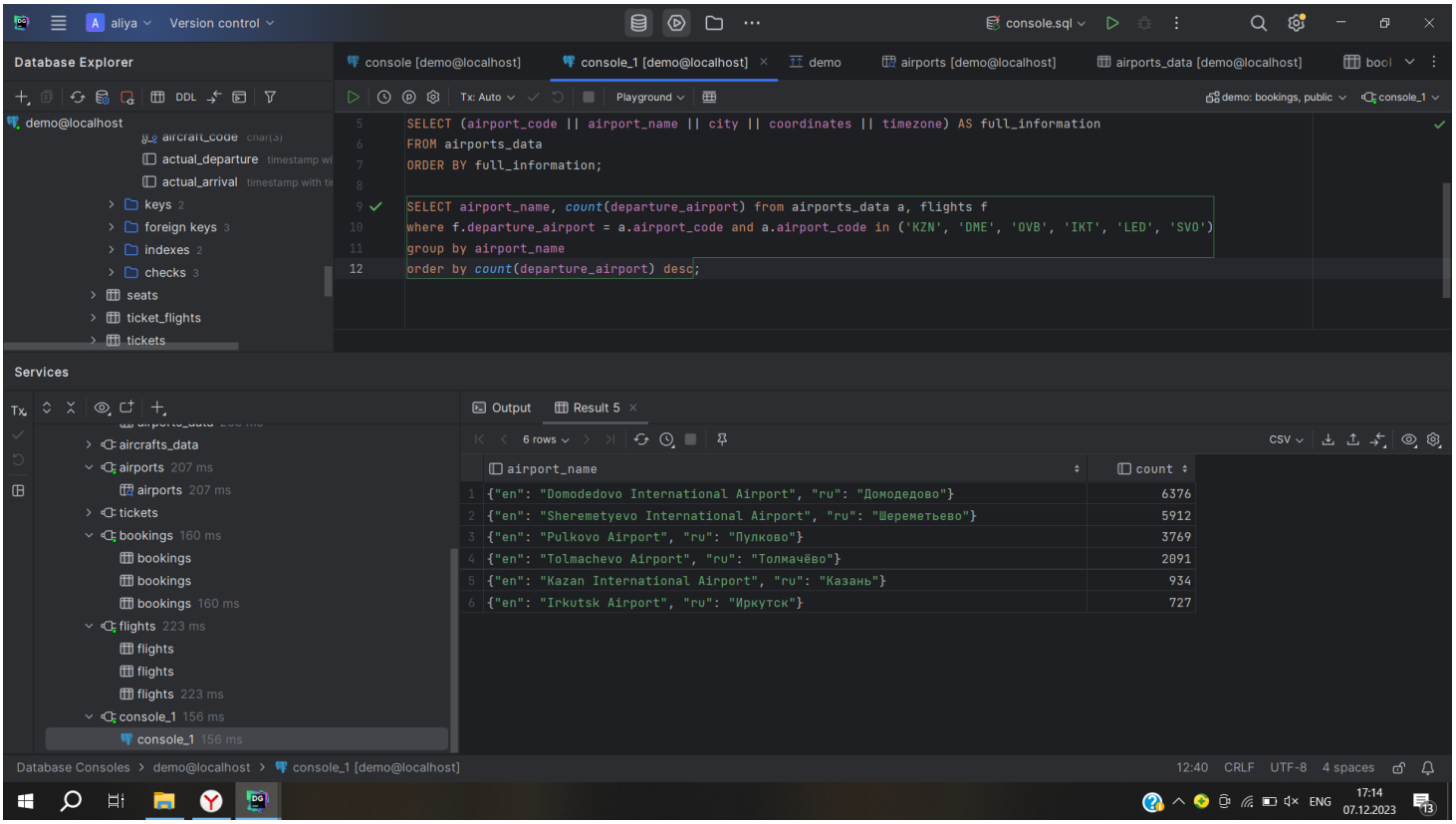
A)


select airport_code, coordinates from airports
where city = 'Казань' or city = 'Москва'
ORDER BY airport_code DESC;
airport_code
coordinates
VKO (37.2615013123,55.5914993286)
SVO (37.4146,55.972599)
KZN (49.278701782227,55.606201171875)
DME (37.90629959106445,55.40879821777344)

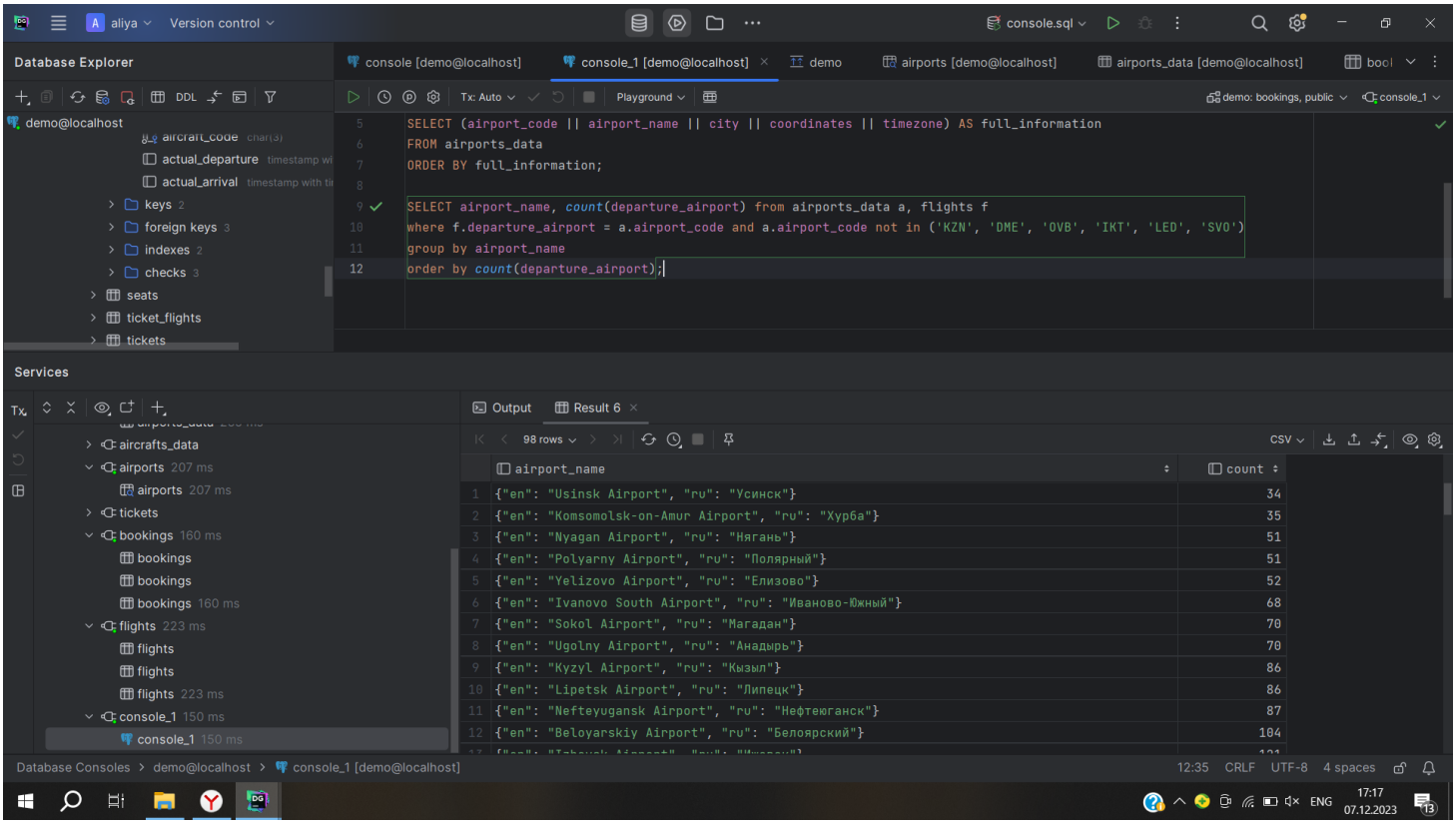
B)


select airport_code, coordinates from airports
where city = 'Казань' or city = 'Москва'
ORDER BY airport_code DESC;
SELECT (airport_code || airport_name || city || coordinates || timezone) AS full_information
FROM airports_data
ORDER BY full_information;
full_information
AAQ{"en": "Anapa Vityazevo Airport", "ru": "Витязево"}{"en": "Anapa", "ru": "Анапа"}(37.347301483154,45.002101898193)Europe…
ABA{"en": "Abakan Airport", "ru": "Абакан"}{"en": "Abakan", "ru": "Абакан"}(91.38500213623047,53.7400016784668)Asia/Krasnoy…
AER{"en": "Sochi International Airport", "ru": "Сочи"}{"en": "Sochi", "ru": "Сочи"}(39.956600189209,43.449901580811)Europe/…
ARH{"en": "Talagi Airport", "ru": "Талаги"}{"en": "Arkhangelsk", "ru": "Архангельск"}(40.71670150756836,64.60030364990234)E…
ASF{"en": "Astrakhan Airport", "ru": "Астрахань"}{"en": "Astrakhan", "ru": "Астрахань"}(48.0063018799,46.2832984924)Europe/…
BAX{"en": "Barnaul Airport", "ru": "Барнаул"}{"en": "Barnaul", "ru": "Барнаул"}(83.53849792480469,53.363800048828125)Asia/K…
BQS{"en": "Ignatyevo Airport", "ru": "Игнатьево"}{"en": "Blagoveschensk", "ru": "Благовещенск"}(127.41200256347656,50.42539…
BTK{"en": "Bratsk Airport", "ru": "Братск"}{"en": "Bratsk", "ru": "Братск"}(101.697998046875,56.370601654052734)Asia/Irkutsk
BZK{"en": "Bryansk Airport", "ru": "Брянск"}{"en": "Bryansk", "ru": "Брянск"}(34.176399231,53.214199066199996)Europe/Moscow
CEE{"en": "Cherepovets Airport", "ru": "Череповец"}{"en": "Cherepovets", "ru": "Череповец"}(38.015800476100004,59.273601532…
CEK{"en": "Chelyabinsk Balandino Airport", "ru": "Челябинск"}{"en": "Chelyabinsk", "ru": "Челябинск"}(61.5033,55.305801)Asi…
CNN{"en": "Chulman Airport", "ru": "Чульман"}{"en": "Neryungri", "ru": "Нерюнгри"}(124.91400146484,56.913898468018)Asia/Yak…

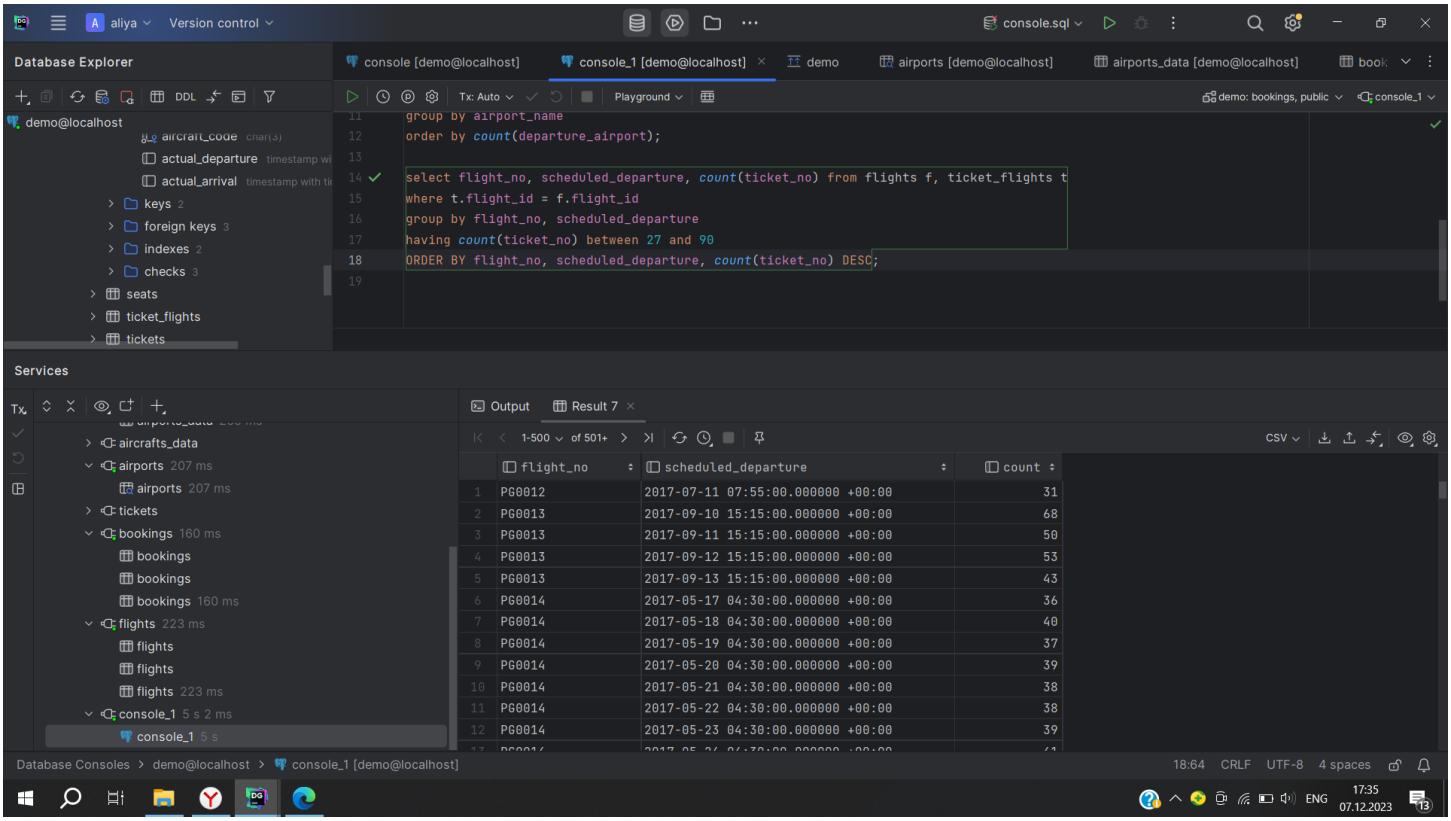
C)


SELECT (airport_code || airport_name || city || coordinates || timezone) AS full_information
FROM airports_data
ORDER BY full_information;
SELECT airport_name, count(departure_airport) from airports_data a, flights f
where f.departure_airport = a.airport_code and a.airport_code in ('KZN', 'DME', 'OVB', 'IKT', 'LED', 'SVO')
group by airport_name
order by count(departure_airport) desc;
airport_name
count
{"en": "Domodedovo International Airport", "ru": "Домодедово"} 6376
{"en": "Sheremetyevo International Airport", "ru": "Шереметьево"} 5912
{"en": "Pulkovo Airport", "ru": "Пулково"} 3769
{"en": "Tolmachevo Airport", "ru": "Толмачёво"} 2091
{"en": "Kazan International Airport", "ru": "Казань"} 934
{"en": "Irkutsk Airport", "ru": "Иркутск"} 727

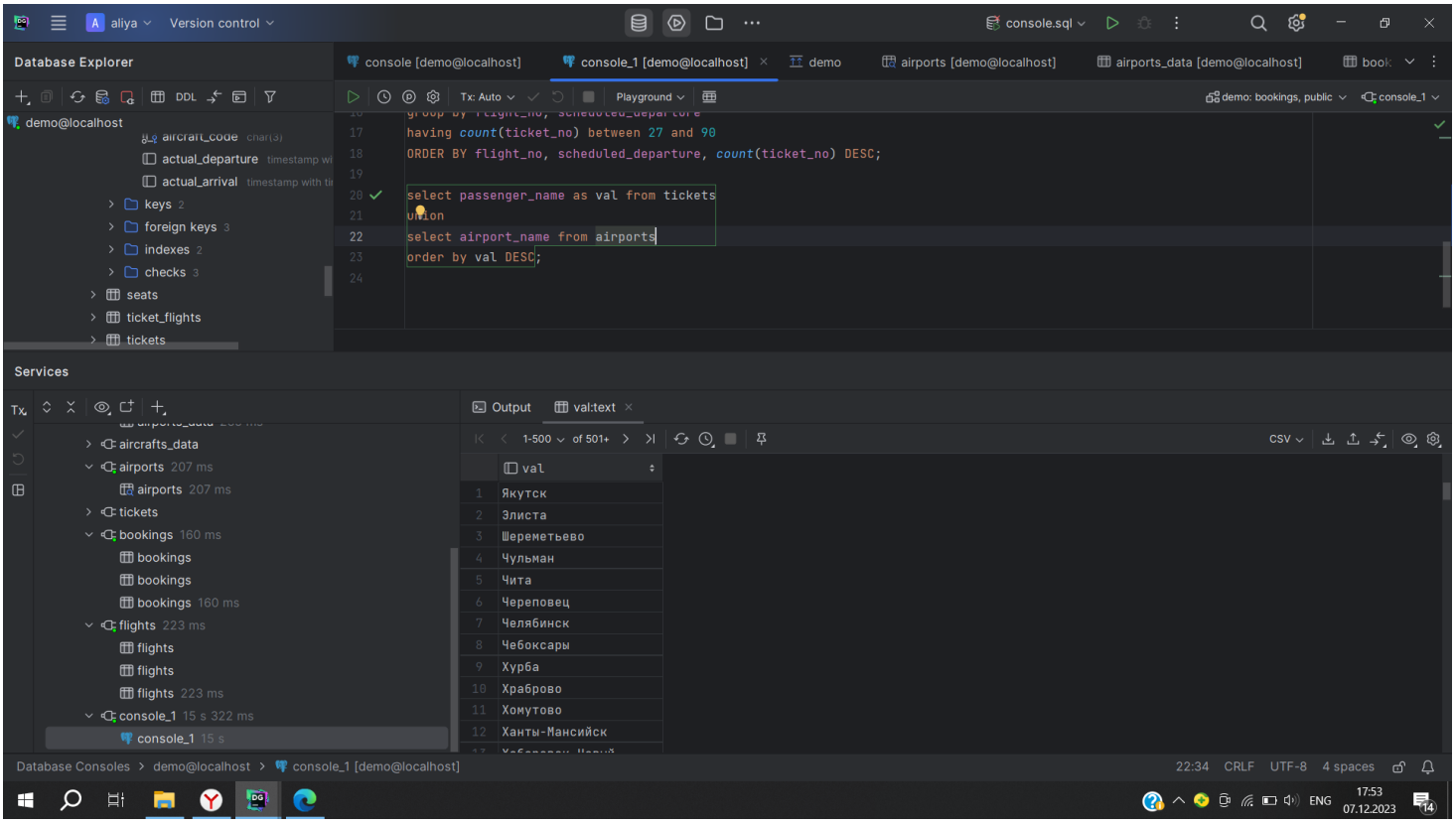
D)


SELECT (airport_code || airport_name || city || coordinates || timezone) AS full_information
FROM airports_data
ORDER BY full_information;
SELECT airport_name, count(departure_airport) from airports_data a, flights f
where f.departure_airport = a.airport_code and a.airport_code not in ('KZN', 'DME', 'OVB', 'IKT', 'LED', 'SVO')
group by airport_name
order by count(departure_airport);
airport_name
count
{"en": "Usinsk Airport", "ru": "Усинск"} 34
{"en": "Komsomolsk-on-Amur Airport", "ru": "Хурба"} 35
{"en": "Nyagan Airport", "ru": "Нягань"} 51
{"en": "Polyarny Airport", "ru": "Полярный"} 51
{"en": "Yelizovo Airport", "ru": "Елизово"} 52
{"en": "Ivanovo South Airport", "ru": "Иваново-Южный"} 68
{"en": "Sokol Airport", "ru": "Магадан"} 70
{"en": "Ugolny Airport", "ru": "Анадырь"} 70
{"en": "Kyzyl Airport", "ru": "Кызыл"} 86
{"en": "Lipetsk Airport", "ru": "Липецк"} 86
{"en": "Nefteyugansk Airport", "ru": "Нефтеюганск"} 87
{"en": "Beloyarskiy Airport", "ru": "Белоярский"} 104

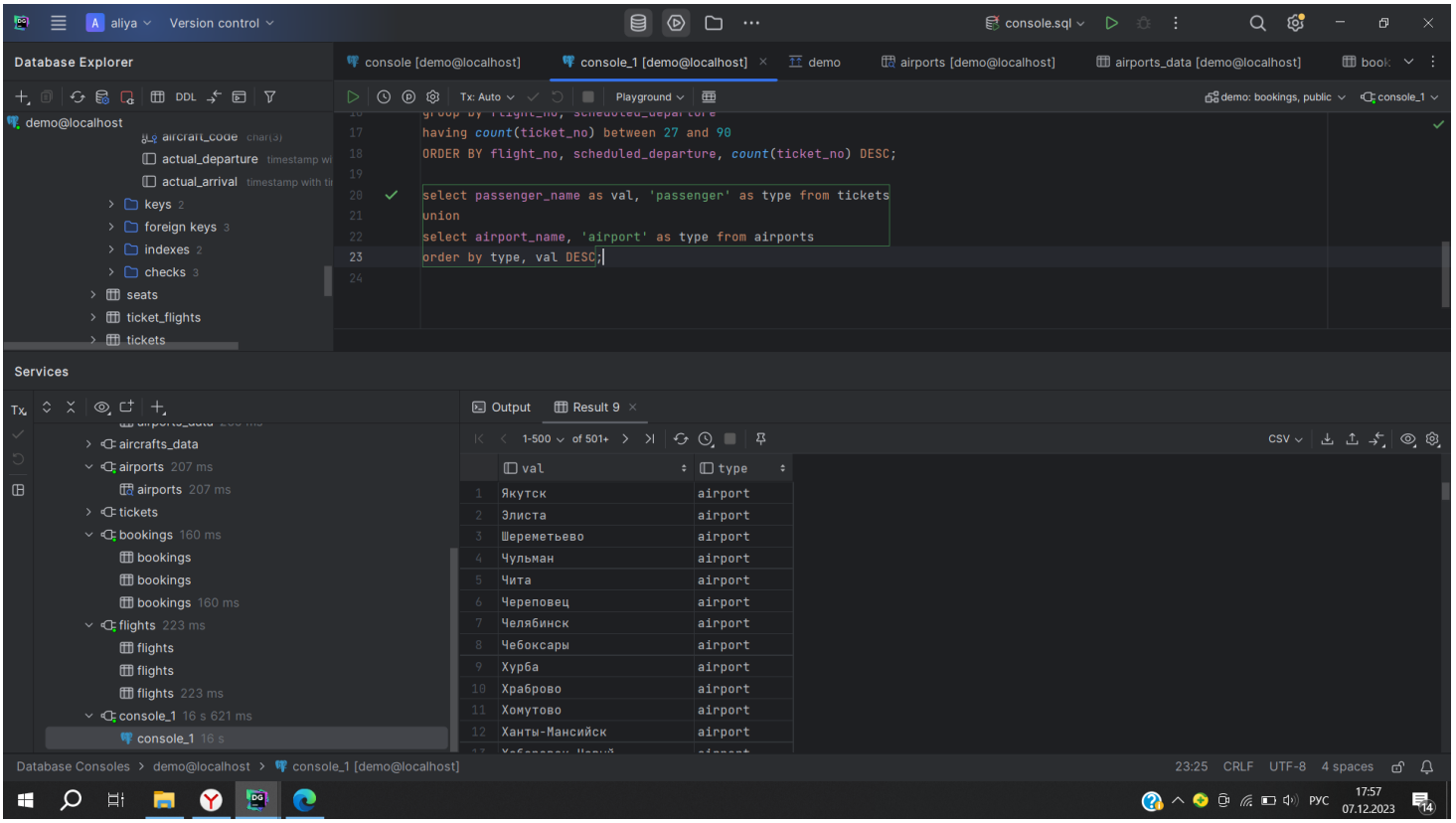
E)


Database Explorer
console [demo@localhost]
console_1 [demo@localhost]
demo
airports [demo@localhost]
airports_data [demo@localhost]
group by airport_name
order by count(departure_airport);
select flight_no, scheduled_departure, count(ticket_no) from flights f, ticket_flights t
where t.flight_id = f.flight_id
group by flight_no, scheduled_departure
having count(ticket_no) between 27 and 90
ORDER BY flight_no, scheduled_departure, count(ticket_no) DESC;
Services
Output
Result 7
flight_no
scheduled_departure
count
PG0012 2017-07-11 07:55:00.000000 +00:00 31
PG0013 2017-09-10 15:15:00.000000 +00:00 68
PG0013 2017-09-11 15:15:00.000000 +00:00 50
PG0013 2017-09-12 15:15:00.000000 +00:00 53
PG0013 2017-09-13 15:15:00.000000 +00:00 43
PG0014 2017-05-17 04:30:00.000000 +00:00 36
PG0014 2017-05-18 04:30:00.000000 +00:00 40
PG0014 2017-05-19 04:30:00.000000 +00:00 37
PG0014 2017-05-20 04:30:00.000000 +00:00 39
PG0014 2017-05-21 04:30:00.000000 +00:00 38
PG0014 2017-05-22 04:30:00.000000 +00:00 38
PG0014 2017-05-23 04:30:00.000000 +00:00 39

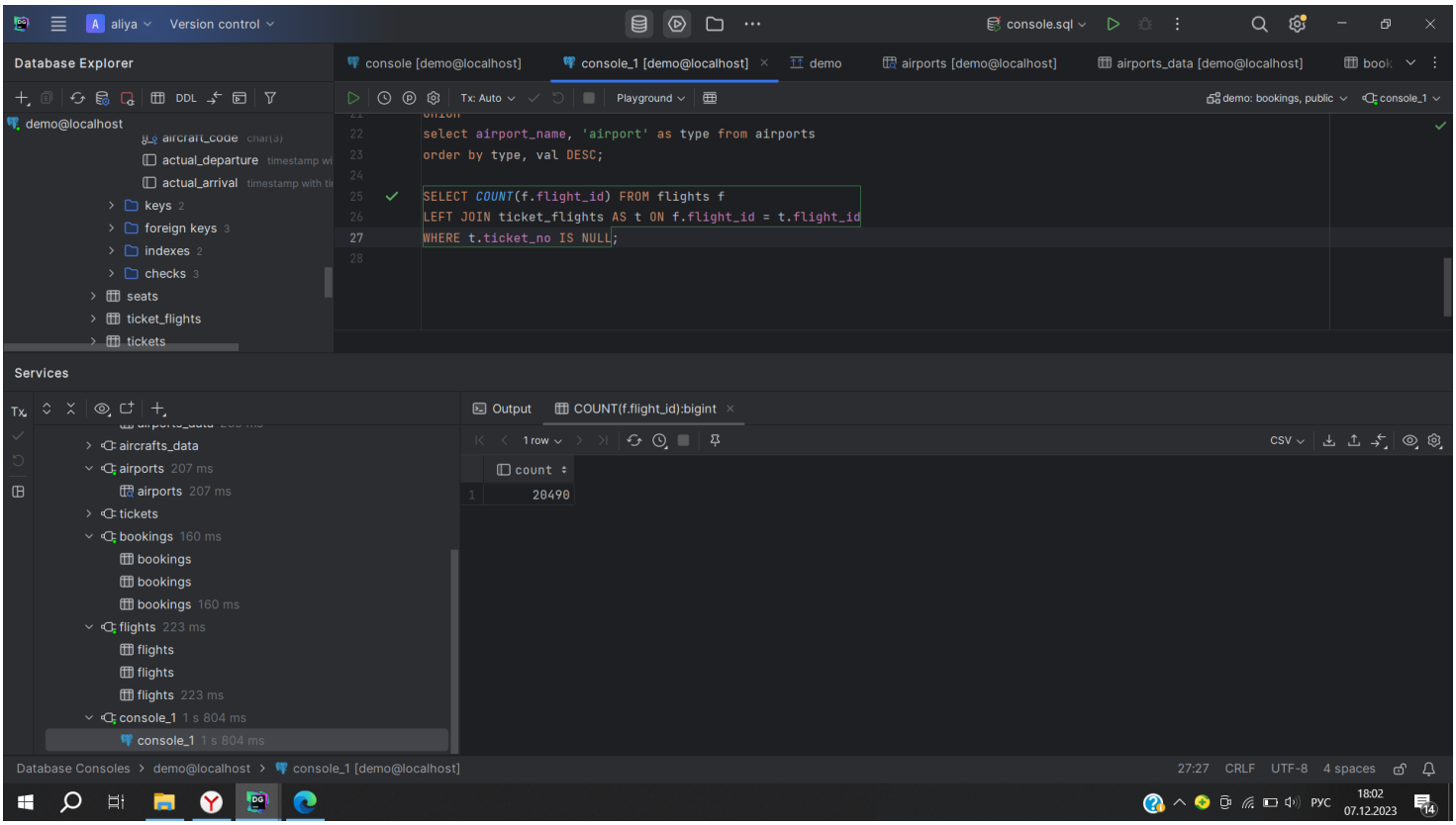
F)


Database Explorer
console [demo@localhost]
console_1 [demo@localhost]
demo
airports [demo@localhost]
airports_data [demo@localhost]
having count(ticket_no) between 27 and 90
ORDER BY flight_no, scheduled_departure, count(ticket_no) DESC;
select passenger_name as val from tickets
union
select airport_name from airports
order by val DESC;
Services
Output
val:text
val
Якутск
Элиста
Шереметьево
Чульман
Чита
Череповец
Челябинск
Чебоксары
Хурба
Храброво
Хомутово
Ханты-Мансийск

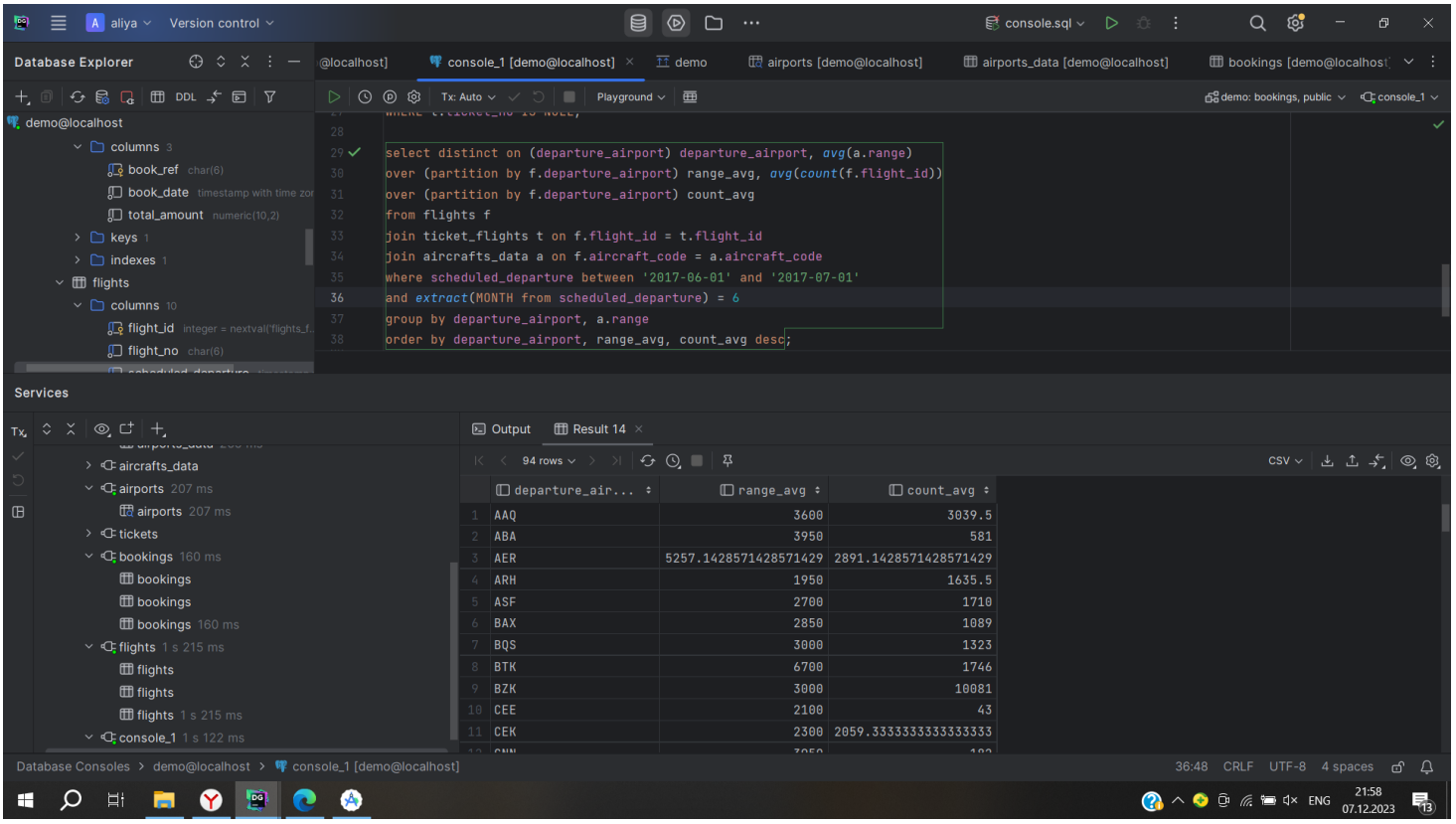
G)



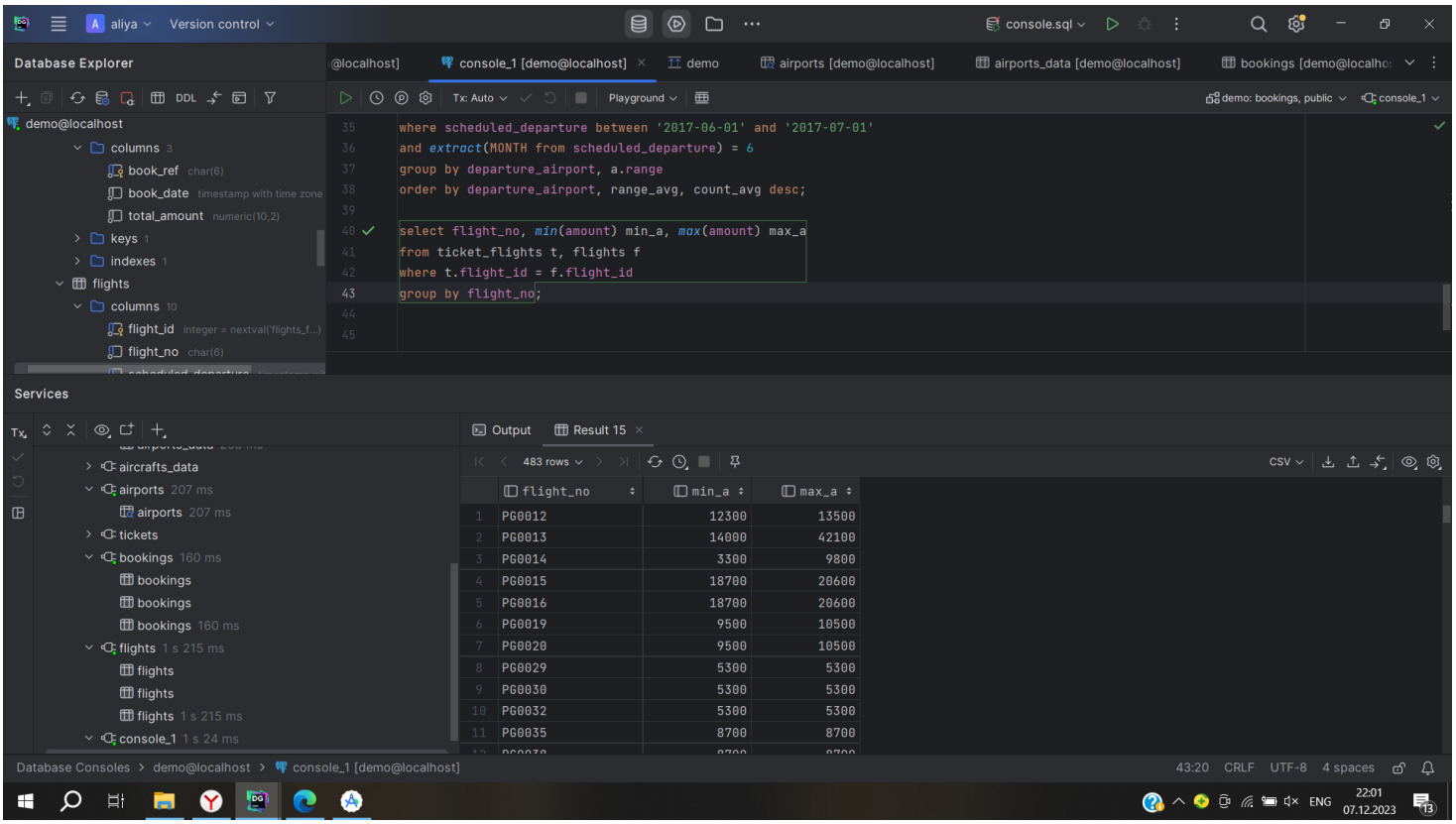
H) *у меня medium версия



I)


Database Explorer
@localhost
console_1 [demo@localhost]
demo
airports [demo@localhost]
airports_data [demo@localhost]
bookings [demo@localhost]
Tx: Auto
Playground
demo: bookings, public
console_1
demo@localhost
columns 3
book_ref char(6)
book_date timestamp with time zor
total_amount numeric(10,2)
keys 1
indexes 1
flights
columns 10
flight_id integer = nextval('flights_f...
flight_no char(6)
28
29 select distinct on (departure_airport) departure_airport, avg(a.range)
30 over (partition by f.departure_airport) range_avg, avg(count(f.flight_id))
31 over (partition by f.departure_airport) count_avg
32 from flights f
33 join ticket_flights t on f.flight_id = t.flight_id
34 join aircrafts_data a on f.aircraft_code = a.aircraft_code
35 where scheduled_departure between '2017-06-01' and '2017-07-01'
36 and extract(MONTH from scheduled_departure) = 6
37 group by departure_airport, a.range
38 order by departure_airport, range_avg, count_avg desc;
Services
aircrafts_data
airports 207 ms
airports 207 ms
tickets
bookings 160 ms
bookings
bookings
bookings 160 ms
flights 1 s 215 ms
flights
flights
flights 1 s 215 ms
console_1 1 s 122 ms
Output
Result 14
94 rows
CSV
departure_air...
range_avg
count_avg
1 AAQ 3600 3039.5
2 ABA 3950 581
3 AER 5257.1428571428571429 2891.1428571428571429
4 ARH 1950 1635.5
5 ASF 2700 1710
6 BAX 2850 1089
7 BQS 3000 1323
8 BTK 6700 1746
9 BZK 3000 10081
10 CEE 2100 43
11 CEK 2300 2059.3333333333333333
Database Consoles > demo@localhost > console_1 [demo@localhost]
36:48 CRLF UTF-8 4 spaces
ENG
21:58
07.12.2023

J)


Database Explorer
@localhost
console_1 [demo@localhost]
demo
airports [demo@localhost]
airports_data [demo@localhost]
bookings [demo@localhost]
Tx: Auto
Playground
demo: bookings, public
console_1
demo@localhost
columns 3
book_ref char(6)
book_date timestamp with time zone
total_amount numeric(10,2)
keys 1
indexes 1
flights
columns 10
flight_id integer = nextval('flights_f...)
flight_no char(6)
35 where scheduled_departure between '2017-06-01' and '2017-07-01'
36 and extract(MONTH from scheduled_departure) = 6
37 group by departure_airport, a.range
38 order by departure_airport, range_avg, count_avg desc;
39
40 select flight_no, min(amount) min_a, max(amount) max_a
41 from ticket_flights t, flights f
42 where t.flight_id = f.flight_id
43 group by flight_no;
44
45
Services
aircrafts_data
airports 207 ms
airports 207 ms
tickets
bookings 160 ms
bookings
bookings
bookings 160 ms
flights 1 s 215 ms
flights
flights
flights 1 s 215 ms
console_1 1 s 24 ms
Output
Result 15
483 rows
CSV
flight_no
min_a
max_a
1 PG0012 12300 13500
2 PG0013 14000 42100
3 PG0014 3300 9800
4 PG0015 18700 20600
5 PG0016 18700 20600
6 PG0019 9500 10500
7 PG0020 9500 10500
8 PG0029 5300 5300
9 PG0030 5300 5300
10 PG0032 5300 5300
11 PG0035 8700 8700
Database Consoles > demo@localhost > console_1 [demo@localhost]
43:20 CRLF UTF-8 4 spaces
ENG
22:01
07.12.2023