

SQL запросы

A

```
Select airport_code, coordinates
from airports_data
where city ->> 'ru' = 'Казань'
      or city ->> 'ru' = 'Москва'
ORDER BY airport_code DESC;
```

	airport_code	coordinates
1	VKO	(37.2615013123, 55.5914993286)
2	SVO	(37.4146, 55.972599)
3	KZN	(49.278701782227, 55.606201171875)
4	DME	(37.90629959106445, 55.40879821777344)





B

```
Select airports_data.airport_code || airports_data.city ||
airports_data.coordinates || airports_data.timezone ||
      airports_data.airport_code as "полная информация"
from airports_data
ORDER BY "полная информация";
```

	"полная информация"
1	AAQ{"en": "Anapa", "ru": "Анапа"}(37.347301483154, 45.002101898193)Europe/MoscowAAQ
2	ABA{"en": "Abakan", "ru": "Абакан"}(91.38500213623047, 53.7400016784668)Asia/KrasnoyarskABA
3	AER{"en": "Sochi", "ru": "Сочи"}(39.956600189209, 43.449901580811)Europe/MoscowAER
4	ARH{"en": "Arkhangelsk", "ru": "Архангельск"}(40.71670150756836, 64.60030364990234)Europe/MoscowARH
5	ASF{"en": "Astrakhan", "ru": "Астрахань"}(48.0063018799, 46.2832984924)Europe/SamaraASF
6	BAX{"en": "Barnaul", "ru": "Барнаул"}(83.53849792480469, 53.363800048828125)Asia/KrasnoyarskBAX
7	BQS{"en": "Blagoveschensk", "ru": "Благовещенск"}(127.41200256347656, 50.42539978027344)Asia/YakutskBQS
8	BTk{"en": "Bratsk", "ru": "Братск"}(101.697998046875, 56.370601654052734)Asia/IrkutskBTk
9	BZK{"en": "Bryansk", "ru": "Брянск"}(34.176399231, 53.214199066199996)Europe/MoscowBZK
10	CEE{"en": "Cherepovets", "ru": "Череповец"}(38.015800476100004, 59.273601532)Europe/MoscowCEE
11	CEK{"en": "Chelyabinsk", "ru": "Челябинск"}(61.5033, 55.305801)Asia/YekaterinburgCEK
12	CNN{"en": "Neryungri", "ru": "Нерюнгри"}(124.91400146484, 56.913898468018)Asia/YakutskCNN
13	CSY{"en": "Cheboksary", "ru": "Чебоксары"}(47.3473014831543, 56.090301513671875)Europe/MoscowCSY
14	DME{"en": "Moscow", "ru": "Москва"}(37.90629959106445, 55.40879821777344)Europe/MoscowDME
15	DYR{"en": "Anadyr", "ru": "Анадырь"}(177.74099731445312, 64.73490142822266)Asia/AnadyrDYR
16	EG0{"en": "Belgorod", "ru": "Белгород"}(36.5900993347168, 50.643798828125)Europe/MoscowEG0
17	ESL{"en": "Elista", "ru": "Элиста"}(44.33089828491211, 46.3739013671875)Europe/MoscowESL
18	EYK{"en": "Beloyarsky", "ru": "Белоярский"}(66.698600769, 63.686901092499994)Asia/YekaterinburgEYK
19	GDX{"en": "Magadan", "ru": "Магадан"}(150.72000122070312, 59.9109992980957)Asia/MagadanGDX
20	GDZ{"en": "Gelendzhik", "ru": "Геленджик"}(38.0124807358, 44.5820926295)Europe/MoscowGDZ
21	GOJ{"en": "Nizhniy Novgorod", "ru": "Нижегород"}(43.784000396729, 56.230098724365)Europe/MoscowGOJ
22	GRV{"en": "Grozny", "ru": "Грозный"}(45.78409957885742, 43.298099517822266)Europe/MoscowGRV
23	HMA{"en": "Khanty-Mansiysk", "ru": "Ханты-Мансийск"}(69.08609771728516, 61.028499603271484)Asia/YekaterinburgHMA

C

```
select airports_data.airport_name --> 'ru', count(*)
from flights
      join airports_data on airport_code = departure_airport
where flights.departure_airport in ('KZN', 'DME', 'OVB', 'IKT', 'LED', 'SVO')
group by airports_data.airport_name
ORDER BY count(*) Desc;
```

	 ?column? 	 count 
1	Домодедово	3217
2	Шереметьево	2981
3	Пулково	1900
4	Толмачёво	1055
5	Казань	471
6	Иркутск	366

D

```
select airports_data.airport_name ->> 'ru', count(*)
from flights
      join airports_data on airport_code = departure_airport
where flights.departure_airport not in ('KZN', 'DME', 'OVB', 'IKT', 'LED',
'SVO')
group by airports_data.airport_name
ORDER BY count(*);
```

	?column?	count
1	Хурба	18
2	Усинск	18
3	Елизово	26
4	Полярный	27
5	Нягань	27
6	Иваново-Южный	34
7	Анадырь	36
8	Магадан	36
9	Липецк	43
10	Кызыл	43
11	Нефтеюганск	44
12	Белоярский	53
13	Стрежевой	61
14	Усть-Кут	61
15	Игнатьево	61
16	Братск	61
17	Ижевск	61
18	Грозный	70
19	Орск	71
20	Байкал	71
21	Туношна	79
22	Кемерово	87
23	Горно-Алтайск	88
24	Чита	96
25	Псков	96
26	Геленджик	113

E

```
select flights.flight_no, flights.scheduled_departure, count(ticket_flights)
from flights
      JOIN ticket_flights on flights.flight_id = ticket_flights.flight_id
group by flights.flight_no, scheduled_departure
HAVING count(ticket_flights) between 27 and 90
ORDER BY flight_no DESC, scheduled_departure DESC, count(ticket_flights)
DESC;
```

	flight_no	scheduled_departure	count
1	P60710	2017-09-12 01:25:00.000000 +00:00	38
2	P60710	2017-09-05 01:25:00.000000 +00:00	57
3	P60710	2017-08-29 01:25:00.000000 +00:00	78
4	P60710	2017-08-22 01:25:00.000000 +00:00	82
5	P60710	2017-08-15 01:25:00.000000 +00:00	89
6	P60710	2017-08-08 01:25:00.000000 +00:00	86
7	P60710	2017-08-01 01:25:00.000000 +00:00	51
8	P60710	2017-07-25 01:25:00.000000 +00:00	52
9	P60709	2017-09-04 17:20:00.000000 +00:00	44
10	P60709	2017-08-28 17:20:00.000000 +00:00	63
11	P60709	2017-08-21 17:20:00.000000 +00:00	61
12	P60709	2017-08-14 17:20:00.000000 +00:00	58
13	P60709	2017-08-07 17:20:00.000000 +00:00	65
14	P60709	2017-07-31 17:20:00.000000 +00:00	41
15	P60709	2017-07-24 17:20:00.000000 +00:00	33
16	P60708	2017-09-09 14:15:00.000000 +00:00	31
17	P60708	2017-09-08 14:15:00.000000 +00:00	35
18	P60708	2017-09-07 14:15:00.000000 +00:00	31
19	P60708	2017-09-06 14:15:00.000000 +00:00	38
20	P60708	2017-09-02 14:15:00.000000 +00:00	29
21	P60708	2017-09-01 14:15:00.000000 +00:00	30
22	P60708	2017-08-31 14:15:00.000000 +00:00	36
23	P60708	2017-08-29 14:15:00.000000 +00:00	30

F

```
select tickets.passenger_name as element
from tickets
UNION ALL
select airports_data.airport_name ->> 'ru'
from airports_data
ORDER BY element DESC;
```

366 837 rows	
	element
1	Якутск
2	Элиста
3	Шереметьево
4	Чульман
5	Чита
6	Череповец
7	Челябинск
8	Чебоксары
9	Хурба
10	Храброво
11	Хомутово
12	Ханты-Мансийск
13	Хабаровск-Новый
14	Ухта
15	Уфа
16	Усть-Кут
17	Усть-Илимск
18	Усинск
19	Ульяновск-Восточный
20	Уйташ
21	Туношна
22	Толмачёво
23	Талаги

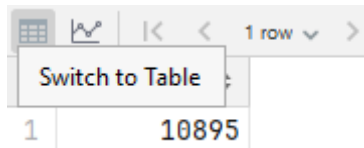
G

```
select tickets.passenger_name as element, 'Пассажир' type
from tickets
UNION ALL
select airports_data.airport_name ->> 'ru', 'Аэропорт'
from airports_data
ORDER BY type DESC, element DESC;
```

	element	type
1	ZULFIYA ZOTOVA	Пассажир
2	ZULFIYA ZOTOVA	Пассажир
3	ZULFIYA ZHURAVLEVA	Пассажир
4	ZULFIYA ZAYCEVA	Пассажир
5	ZULFIYA ZAYCEVA	Пассажир
6	ZULFIYA ZAKHAROVA	Пассажир
7	ZULFIYA ZAKHAROVA	Пассажир
8	ZULFIYA VOROBEOVA	Пассажир
9	ZULFIYA VOLKOVA	Пассажир
10	ZULFIYA VLASOVA	Пассажир
11	ZULFIYA VASILEVA	Пассажир
12	ZULFIYA VASILEVA	Пассажир
13	ZULFIYA VASILEVA	Пассажир
14	ZULFIYA TIKHONOVA	Пассажир
15	ZULFIYA TARASOVA	Пассажир
16	ZULFIYA SOROKINA	Пассажир
17	ZULFIYA SOROKINA	Пассажир
18	ZULFIYA SOKOLOVA	Пассажир
19	ZULFIYA SOKOLOVA	Пассажир
20	ZULFIYA SOKOLOVA	Пассажир
21	ZULFIYA SMIRNOVA	Пассажир
22	ZULFIYA SIDOROVA	Пассажир
23	ZULFIYA SERGEEVA	Пассажир

H

```
select count(*)
from flights
      LEFT JOIN ticket_flights on flights.flight_id =
ticket_flights.flight_id
where ticket_flights.flight_id is null;
```



A screenshot of a database query result interface. At the top, there are navigation icons and a dropdown menu showing '1 row'. Below this, a button labeled 'Switch to Table' is visible. The main area displays a single row with the index '1' and the value '10895'.

1	10895
---	-------

I

```
select airport_name -->> 'ru' as airport_name,
      (
        Select AVG(param)
        FROM (
          select count(seats) as param from seats
          join aircrafts_data on seats.aircraft_code =
aircrafts_data.aircraft_code
          join flights on aircrafts_data.aircraft_code =
flights.aircraft_code
          join airports_data as air on flights.departure_airport =
air.airport_code
          where air.airport_code = airports_data.airport_code and
(select extract(MONTH FROM scheduled_departure)) = 9
          group by flight_id
        ) as alias
      ) as average_seats,
      (
        Select AVG(param)
        FROM (
          select count(ticket_flights) as param from ticket_flights
          join flights on ticket_flights.flight_id =
flights.flight_id
          join airports_data as air on flights.departure_airport =
air.airport_code
          where air.airport_code = airports_data.airport_code and
(select extract(MONTH FROM scheduled_departure)) = 9
          group by flights.flight_id
        ) as alias
      ) as average_tickets
from airports_data
-- where (select extract(MONTH FROM scheduled_departure)) = 9
group by airport_name -->> 'ru', airport_code
order by average_seats desc, average_tickets desc;
```

	airport_name	average_seats	average_tickets
1	Елизово	222	39
2	Хурба	222	9.5
3	Магадан	159.5	26.5
4	Краснодар	135.2592592592593	70.375
5	Владивосток	123.6341463414634146	18.4324324324324324
6	Курумоч	121.3333333333333333	60.2142857142857143
7	Сочи	116.5671641791044776	64.3623188405797101
8	Анадырь	116	41.5
9	Братск	116	29
10	Витязево	114.6	48.6666666666666667
11	Баратаевка	110.7777777777777778	40.1290322580645161
12	Шереметьево	110.0058479532163743	24.1182795698924731
13	Минеральные Воды	107.9166666666666667	33.5853658536585366
14	Кольцово	107.9113924050632911	45.3854166666666667
15	Пермь	107.8169014084507042	41.6956521739130435
16	Пулково	107.1192660550458716	26.5486111111111111
17	Хабаровск-Новый	106.5217391304347826	39.3066666666666667
18	Домодедово	101.3766937669376694	24.4856046065259117
19	Храброво	99.3888888888888889	35.2121212121212121
20	Внуково	99.2385786802030457	17.7931034482758621
21	Сыктывкар	98.4137931034482759	47.2105263157894737
22	Петрозаводск	97	37.1153846153846154
23	Брянск	97	35.2622950819672131
24	Игнатьево	97	22.7857142857142857
25	Уйташ	88.6521739130434783	30.25
26	Толмачёво	87.7024793388429752	43.1526717557251908
27	Иркутск	87.0476190476190476	44.9545454545454545

J

```
-- J
Select flight_no, min(amount) as min_price, max(amount) as max_price from
ticket_flights
join flights on ticket_flights.flight_id = flights.flight_id
group by flight_no
order by flight_no;
```

	flight_no	min_price	max_price
1	P60012	12300	13500
2	P60013	14000	42100
3	P60014	3300	9800
4	P60015	18700	20600
5	P60016	18700	20600
6	P60019	9500	10500
7	P60020	9500	10500
8	P60029	5300	5300
9	P60030	5300	5300
10	P60032	5300	5300
11	P60035	8700	8700
12	P60038	8700	8700
13	P60039	3200	9700
14	P60040	3200	9700
15	P60041	20900	23000
16	P60042	23600	26000
17	P60043	9900	10900
18	P60044	9900	10900
19	P60045	14600	16100
20	P60046	14600	16100
21	P60049	12300	13500
22	P60050	7200	7900
23	P60051	7200	7900