



### 1.1 Соединение двух произвольных таблиц при помощи UNION

```
SELECT airport_name as one FROM airports WHERE city = 'Москва'  
UNION  
SELECT departure_airport as one FROM flights WHERE departure_airport = 'SVO'
```

The screenshot shows a SQL query editor with the following query:

```
SELECT airport_name as one FROM airports WHERE city = 'Москва'  
UNION  
SELECT departure_airport as one FROM flights WHERE departure_airport = 'SVO'
```

Below the editor, the results are displayed in a table titled "Результат 1". The table has one column, "one", and four rows of data.

one
Домодедово
Внуково
SVO
Шереметьево

### 1.2 Соединение двух произвольных таблиц при помощи JOIN

```
SELECT DISTINCT airports.airport_name, flights.departure_airport  
FROM airports INNER JOIN flights  
ON airports.airport_code = flights.departure_airport  
WHERE airports.city = 'Москва'
```

The screenshot shows a SQL query editor with the following query:

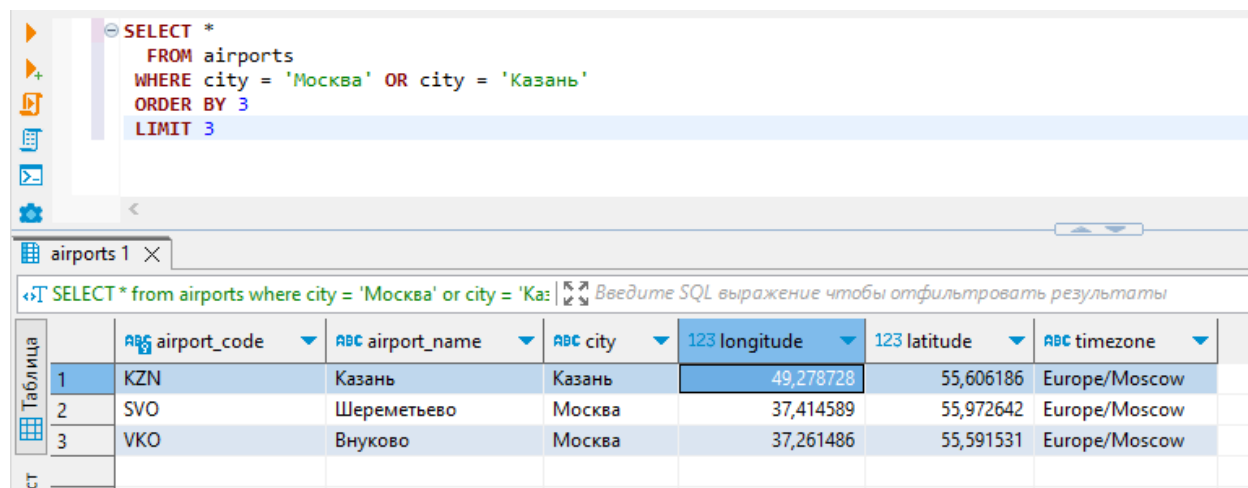
```
SELECT DISTINCT airports.airport_name, flights.departure_airport  
FROM airports INNER JOIN flights  
ON airports.airport_code = flights.departure_airport  
WHERE airports.city = 'Москва'
```

Below the editor, the results are displayed in a table titled "airports(+) 1". The table has two columns: "airport\_name" and "departure\_airport".

airport_name	departure_airport
Домодедово	DME
Внуково	VKO
Шереметьево	SVO

2. Запрос с фильтром **WHERE**, сортировкой **ORDER BY** и ограничением по количеству строк **LIMIT**

```
SELECT *  
  FROM airports  
 WHERE city = 'Москва' OR city = 'Казань'  
 ORDER BY 3  
 LIMIT 3
```

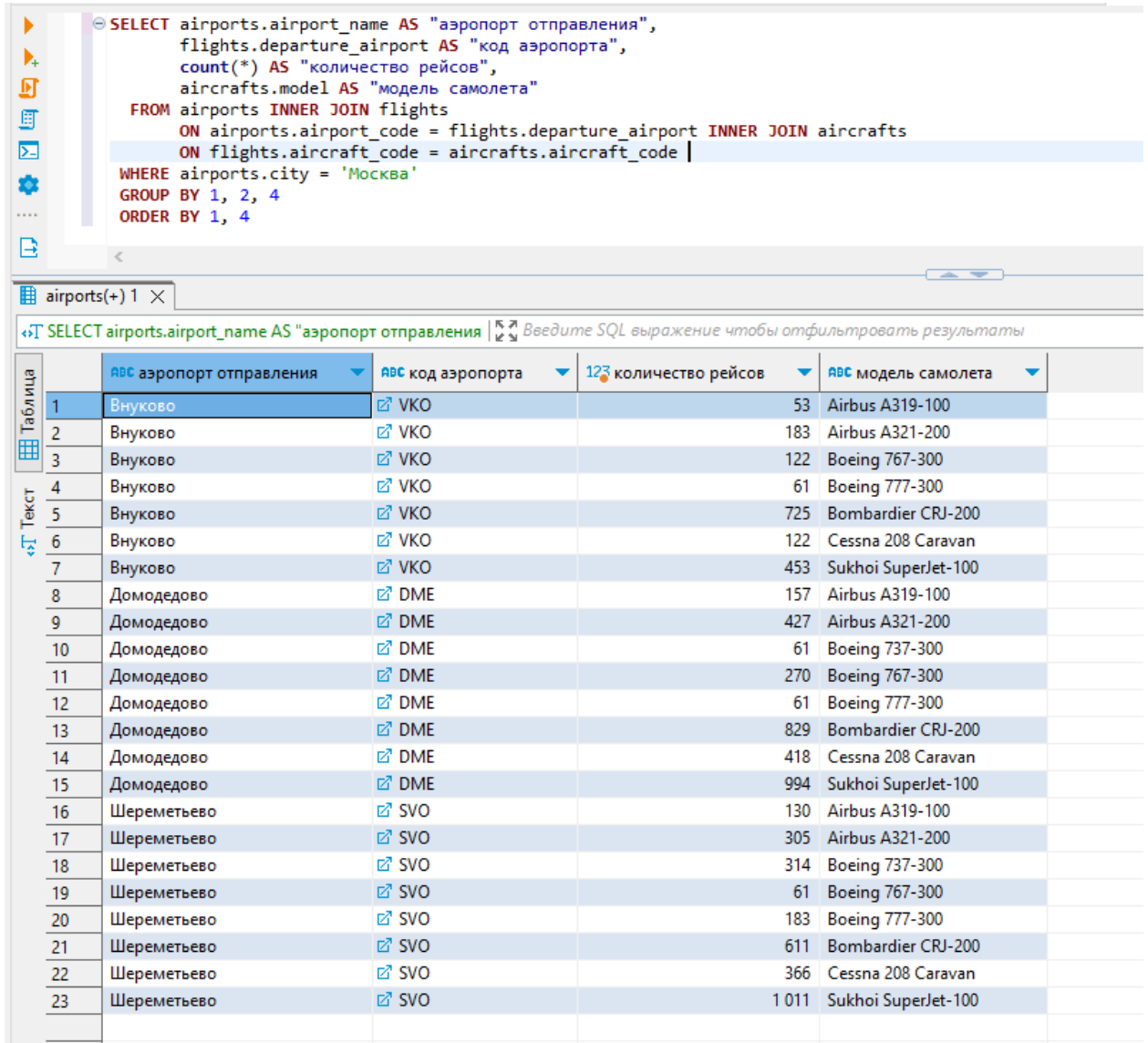


The screenshot shows a database client window with a SQL editor at the top containing the query: `SELECT * FROM airports WHERE city = 'Москва' OR city = 'Казань' ORDER BY 3 LIMIT 3`. Below the editor, a tab labeled 'airports 1' is active, displaying a table of results. The table has 7 columns: 'airport\_code', 'airport\_name', 'city', 'longitude', 'latitude', and 'timezone'. The results are sorted by the 'city' column (index 3) and limited to 3 rows. The first row is for KZN (Казань), and the next two are for SVO and VKO (both in Москва).

	airport_code	airport_name	city	longitude	latitude	timezone
1	KZN	Казань	Казань	49,278728	55,606186	Europe/Moscow
2	SVO	Шереметьево	Москва	37,414589	55,972642	Europe/Moscow
3	VKO	Внуково	Москва	37,261486	55,591531	Europe/Moscow

3. Выполнение SQL запроса в рамках JOIN оператора с использованием GROUP BY и любой из агрегатных функций (в данном случае – count(\*))

```
SELECT airports.airport_name AS "аэропорт отправления",
       flights.departure_airport AS "код аэропорта",
       count(*) AS "количество рейсов",
       aircrafts.model AS "модель самолета"
FROM airports INNER JOIN flights
      ON airports.airport_code = flights.departure_airport INNER JOIN aircrafts
      ON flights.aircraft_code = aircrafts.aircraft_code
WHERE airports.city = 'Москва'
GROUP BY 1, 2, 4
ORDER BY 1, 4
```

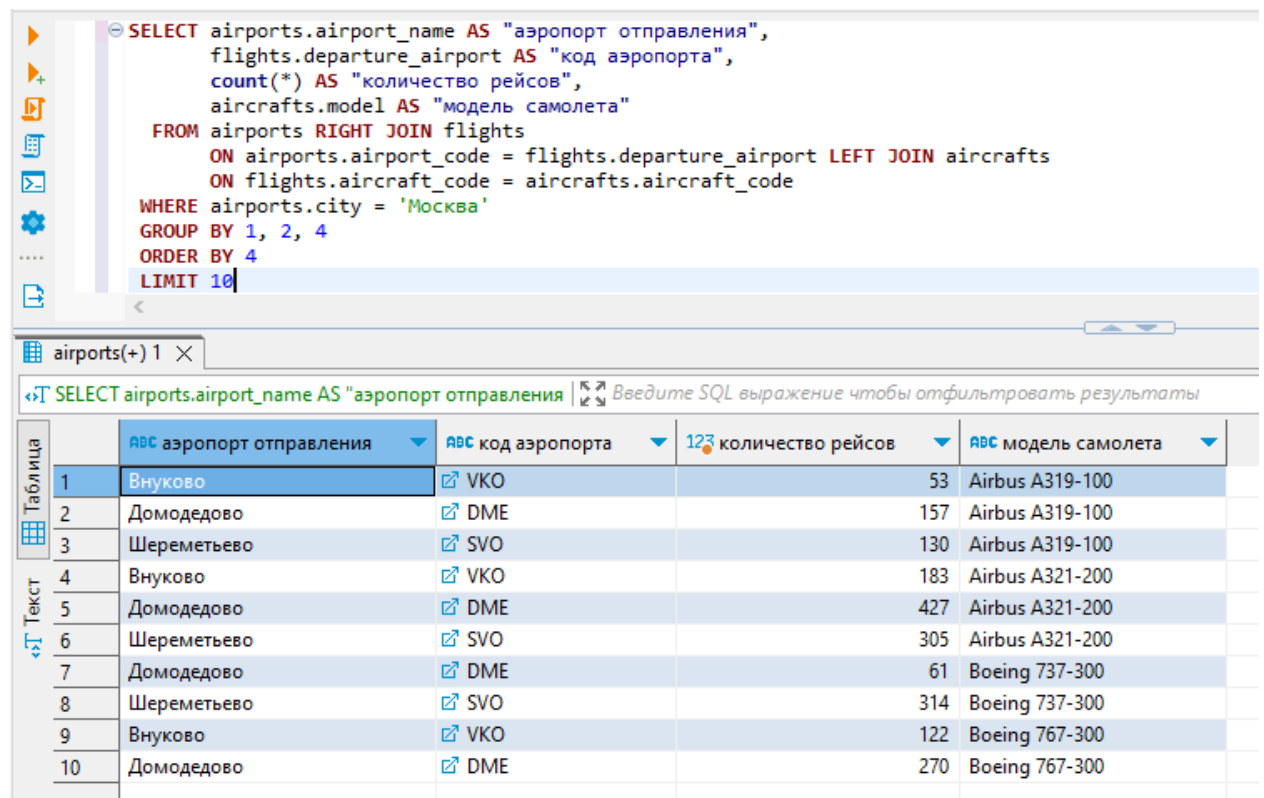


The screenshot shows a database management tool interface. At the top, a SQL query is entered in a text editor. Below the editor, the results of the query are displayed in a table view. The table has four columns: 'аэропорт отправления' (departure airport), 'код аэропорта' (airport code), 'количество рейсов' (number of flights), and 'модель самолета' (aircraft model). The results are grouped by the departure airport, with rows for Vnukovo, Domodedovo, and Sheremetyevo. The 'количество рейсов' column shows the count of flights for each group.

аэропорт отправления	код аэропорта	количество рейсов	модель самолета
Внуково	VKO	53	Airbus A319-100
Внуково	VKO	183	Airbus A321-200
Внуково	VKO	122	Boeing 767-300
Внуково	VKO	61	Boeing 777-300
Внуково	VKO	725	Bombardier CRJ-200
Внуково	VKO	122	Cessna 208 Caravan
Внуково	VKO	453	Sukhoi SuperJet-100
Домодедово	DME	157	Airbus A319-100
Домодедово	DME	427	Airbus A321-200
Домодедово	DME	61	Boeing 737-300
Домодедово	DME	270	Boeing 767-300
Домодедово	DME	61	Boeing 777-300
Домодедово	DME	829	Bombardier CRJ-200
Домодедово	DME	418	Cessna 208 Caravan
Домодедово	DME	994	Sukhoi SuperJet-100
Шереметьево	SVO	130	Airbus A319-100
Шереметьево	SVO	305	Airbus A321-200
Шереметьево	SVO	314	Boeing 737-300
Шереметьево	SVO	61	Boeing 767-300
Шереметьево	SVO	183	Boeing 777-300
Шереметьево	SVO	611	Bombardier CRJ-200
Шереметьево	SVO	366	Cessna 208 Caravan
Шереметьево	SVO	1 011	Sukhoi SuperJet-100

4. Выполнение SQL запроса с применением **LEFT JOIN** и **RIGHT JOIN** (применение **INNER JOIN** было показано в предыдущем запросе) для более чем двух (в данном случае трех) таблиц.

```
SELECT airports.airport_name AS "аэропорт отправления",
       flights.departure_airport AS "код аэропорта",
       count(*) AS "количество рейсов",
       aircrafts.model AS "модель самолета"
FROM airports RIGHT JOIN flights
      ON airports.airport_code = flights.departure_airport LEFT JOIN aircrafts
      ON flights.aircraft_code = aircrafts.aircraft_code
WHERE airports.city = 'Москва'
GROUP BY 1, 2, 4
ORDER BY 4
LIMIT 10
```



The screenshot shows a database client interface. The top pane displays the SQL query used to generate the results. The bottom pane shows the results in a table view with 10 rows and 5 columns. The columns are: 'аэропорт отправления' (airport of departure), 'код аэропорта' (airport code), 'количество рейсов' (number of flights), and 'модель самолета' (aircraft model). The results are sorted by the number of flights in descending order.

	аэропорт отправления	код аэропорта	количество рейсов	модель самолета
1	Внуково	VKO	53	Airbus A319-100
2	Домодедово	DME	157	Airbus A319-100
3	Шереметьево	SVO	130	Airbus A319-100
4	Внуково	VKO	183	Airbus A321-200
5	Домодедово	DME	427	Airbus A321-200
6	Шереметьево	SVO	305	Airbus A321-200
7	Домодедово	DME	61	Boeing 737-300
8	Шереметьево	SVO	314	Boeing 737-300
9	Внуково	VKO	122	Boeing 767-300
10	Домодедово	DME	270	Boeing 767-300

5. Создание виртуального представления **VIEW** с произвольным именем для SQL запроса из задания 2

```
CREATE VIEW second_task AS
SELECT *
FROM airports
WHERE city = 'Москва' OR city = 'Казань'
ORDER BY 3
LIMIT 3
```

CREATE VIEW second\_task AS

SELECT \*

FROM airports

WHERE city = 'Москва' OR city = 'Казань'

ORDER BY 3

LIMIT 3

Статистика 1

Name	Value
Updated Rows	0
Query	CREATE VIEW second_task AS
	SELECT *
	FROM airports
	WHERE city = 'Москва' OR city = 'Казань'
	ORDER BY 3
	LIMIT 3
Start time	Thu Aug 31 20:45:55 MSK 2023
Finish time	Thu Aug 31 20:45:55 MSK 2023

6. Подключение к БД при помощи psycopg2 с выполнением запроса из PyCharm.  
Вызвать виртуальное представление, созданное в задании 5.

```
import psycopg2
from psycopg2 import Error

try:
    # подключение к существующей базе данных
    connection = psycopg2.connect(user="postgres",
                                   password="0000",
                                   host="localhost",
                                   port="5432",
                                   database="demo")

    # курсор для выполнения операций с базой данных
    cursor = connection.cursor()
    # выполнить запрос
    cursor.execute("SELECT * FROM bookings.second_task")
    # получить результат
    record = cursor.fetchall()
    print(record)
except (Exception, Error) as error:
    print("Ошибка при работе с PostgreSQL", error)
finally:
    if connection:
        cursor.close()
        connection.close()
    print("Соединение с PostgreSQL закрыто")
```

