

Скачать и применить dump (резервную копию) на базу данных, установленную локально.

```
Командная строка - psql -U × + ▾
Microsoft Windows [Version 10.0.22621.2428]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\nikol>psql -U postgres -d HHW5 -f "C:\Users\nikol\OneDrive\Рабочий стол\demo-small-20170815.sql"
Пароль пользователя postgres:
SET
SET
SET
SET
SET
SET
SET
SET
SET
DROP DATABASE
CREATE DATABASE
Вы подключены к базе данных "demo" как пользователь "postgres".
SET
SET
SET
SET
SET
SET
SET
SET
CREATE SCHEMA
COMMENT
CREATE EXTENSION
COMMENT
SET
CREATE FUNCTION
CREATE FUNCTION
COMMENT
SET
SET
CREATE TABLE
COMMENT
COMMENT
COMMENT
COMMENT
CREATE VIEW
```

Объединить данные из двух произвольных таблиц используя UNION [ALL], отсортировать результат в порядке убывания

The screenshot shows a database IDE interface with the following components:

- Database Explorer:** Shows the database structure for HHW5@localhost, including a demo database with bookings and tables.
- SQL Editor:** Contains the following SQL query:

```
1 SELECT ticket_no FROM tickets
2 UNION ALL
3 SELECT ticket_no FROM ticket_flights
4 ORDER BY ticket_no DESC;
```
- Services:** Shows the execution timeline for the query, including the time taken for each table scan and the overall query execution time.
- Output:** Displays the results of the query, showing a list of ticket numbers sorted in descending order.

ticket_no
0005435999873
0005435999873
0005435999873
0005435999872
0005435999872
0005435999872
0005435999871
0005435999871
0005435999871
0005435999871
0005435999870
0005435999870
0005435999870
0005435999870
0005435999869
0005435999869
0005435999869

Запрос с любым фильтром WHERE к произвольной таблице, отсортировать результат с ограничением вывода по количеству строк (LIMIT)

The screenshot shows a database client interface with a dark theme. The top bar includes a menu, a version control dropdown, and icons for database, playground, and file operations. The left sidebar, titled 'Database Explorer', shows a tree view of the database structure: 'HHW5@localhost' (2 of 17) contains 'demo' (1 of 4), which contains 'bookings' (8 tables), including 'aircrafts\_data' and 'airports\_data'. The main editor displays a SQL query in a file named 'SQL\_1.sql' for the 'ticket\_flights' table. The query is: 

```
1 SELECT * FROM ticket_flights
2 WHERE fare_conditions='Comfort'
3 ORDER BY amount
4 LIMIT 20;
```

 Below the editor, the 'Services' panel shows a tree view of the database structure, including 'HHW5@localhost' and 'postgres@localhost'. The 'Output' panel shows the results of the query, displaying 20 rows of data. The results are sorted by 'amount' in descending order. The columns are 'ticket\_no', 'flight\_id', 'fare\_conditions', and 'amount'.

	ticket_no	flight_id	fare_conditions	amount
1	0005434876636	3490	Comfort	19900.00
2	0005434876580	3490	Comfort	19900.00
3	0005434876583	3490	Comfort	19900.00
4	0005434876621	3490	Comfort	19900.00
5	0005434876558	3490	Comfort	19900.00
6	0005434876650	3490	Comfort	19900.00
7	0005434876565	3490	Comfort	19900.00
8	0005432798969	3490	Comfort	19900.00
9	0005432798961	3490	Comfort	19900.00
10	0005435625877	3490	Comfort	19900.00
11	0005434937796	3490	Comfort	19900.00
12	0005435625873	3490	Comfort	19900.00
13	0005435125665	3490	Comfort	19900.00
14	0005435125651	3490	Comfort	19900.00
15	0005434876614	3490	Comfort	19900.00
16	0005435625858	3490	Comfort	19900.00

Напишите аналитический запрос к связке из двух произвольных таблиц, получите какой-либо вывод при помощи агрегатных функций с использованием GROUP BY. Связка таблиц выполнять при помощи IN/EXISTS

The screenshot shows a database IDE interface. The top bar includes a menu, a version control dropdown, and icons for database, playground, and file operations. The left sidebar, titled 'Database Explorer', shows a tree view of the database structure: HHW5@localhost (2 of 17) > demo (1 of 4) > bookings > tables (8). The tables listed are aircrafts\_data, airports\_data, boarding\_passes, bookings, flights, and seats. The main editor area shows a SQL query in a file named 'SQL\_1.sql'. The query is: 

```
1 select book_ref, count(book_ref) from tickets
2 where book_ref in
3     (select bookings.book_ref from bookings where total_amount = '200000')
4 group by book_ref
5
```

 The bottom section, titled 'Services', shows a list of database connections and their execution times. The 'Output' pane displays the results of the query, showing 8 rows with columns 'book\_ref' and 'count'.

	book_ref	count
1	010153	2
2	115B21	2
3	401304	2
4	4913F8	2
5	80D5F9	2
6	A11BC8	2
7	CD755F	2
8	DCF4D2	2

Найдите все уникальные неупорядоченные пары аэропортов связанные рейсами в таблице flights. В этом пункте нельзя использовать DISTINCT. Неупорядоченная пара, значит что DME–KZN = KZN–DME. Все уникальные пары, значит что вне зависимости от количества рейсов по маршруту KZN-DME, в выводе запроса будет фигурировать только 1 экземпляр данной пары.

The screenshot shows a database IDE interface. On the left, the 'Database Explorer' pane shows a tree structure with 'HHW5@localhost' as the root, containing a 'demo' database with a 'bookings' table. The 'bookings' table is expanded, showing its schema with columns: aircrafts\_data, airports\_data, boarding\_passes, bookings, and flights. The 'SQL\_1.sql' file is open in the editor, containing the following SQL query:

```
1 select
2     least(f1.departure_airport, f1.arrival_airport) as airport1,
3     greatest(f1.departure_airport, f1.arrival_airport) as airport2
4 from flights f1
5 group by airport1, airport2;
```

The 'Services' pane at the bottom shows the 'Output' tab with 'Result 2' containing 309 rows. The results are displayed in a table with two columns: 'airport1' and 'airport2'. The first 13 rows are shown below:

	airport1	airport2
1	DME	GDZ
2	AAQ	EG0
3	GDZ	ROV
4	LPK	PES
5	GOJ	NAL
6	AER	PEZ
7	KJA	NUX
8	OSW	VK0
9	IKT	VV0
10	NOJ	NUX
11	CSY	SVO
12	GDZ	VK0
13	DME	ULV