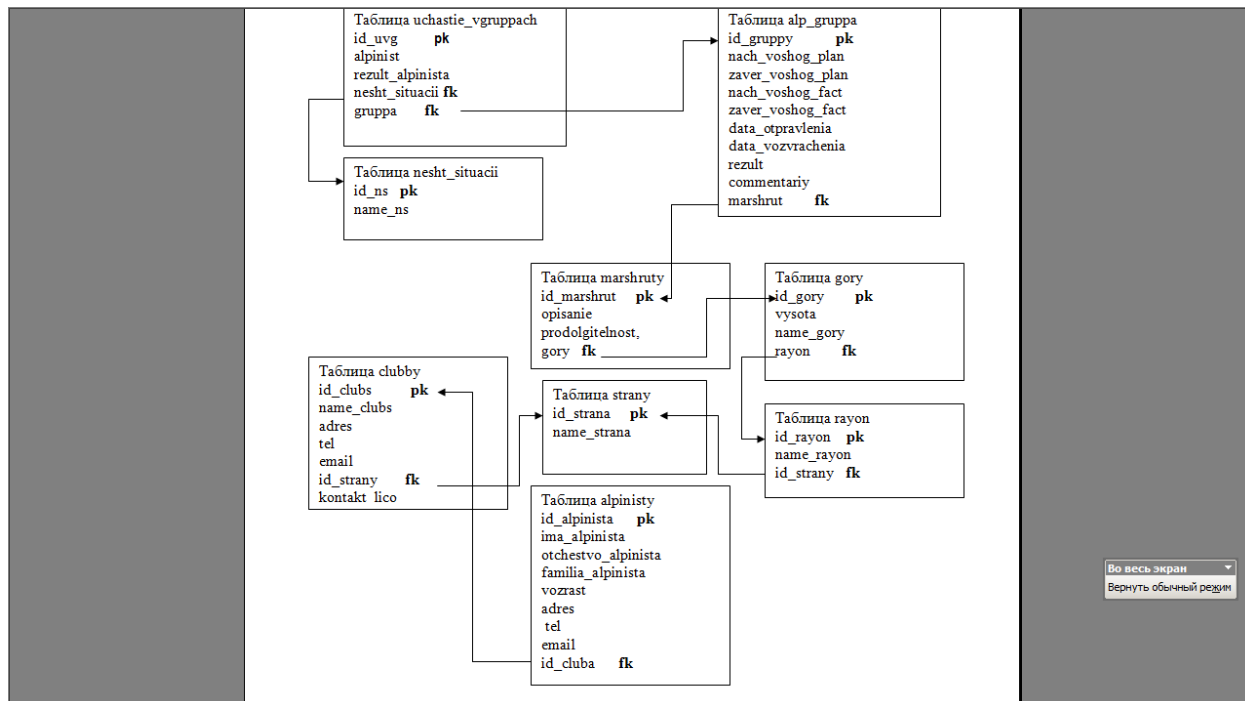


## Скрин модели базы данных альпийский клуб

Pk- primary key

Fk- foreign key

Связь один ко многим



## Задание 15

Создать программную систему, предназначенную для администратора альпинистского клуба. Альпинистский клуб организует восхождения в разных точках мира. Система должна обеспечить сохранение информации о хронике восхождений.

Для каждого восхождения формируется группа. В состав группы могут входить альпинисты из других клубов. Поэтому нужно иметь информацию о каждом клубе (название, страна, город, контактное лицо, e-mail, телефон). Необходимо иметь описание маршрута и продолжительность восхождения. Необходимо обеспечить сохранение даты/времени начала и завершения каждого восхождения (планируемого и фактического), имен и адресов участвовавших в нем альпинистов, названия и высоты горы, страны и района, где эта гора расположена. После завершения восхождения фиксируется информация об успешности восхождения для каждого участника и группы в целом. При возникновении нештатных ситуаций необходимо указать для каждого участника, что случилось (травма, пропал без вести, летальный исход и т.д.) и в пояснении о группе дать подробности.

Администратор должен иметь возможность:

- добавления сведений о новом альпинисте, новой вершине;
- изменении сведений об альпинистах и вершинах;
- формирования новых групп и внесения всей информации после завершения восхождения группой.

Перечень возможных запросов:

1. Показать список альпинистов, осуществлявших восхождение в указанный интервал дат.
2. Показать список восхождений (групп), которые осуществлялись в указанный пользователем период времени.
3. Предоставить информацию о том, сколько альпинистов побывали на каждой горе.
4. Предоставить данные о вершинах, если на них не было восхождений.
5. Показать информацию о количестве восхождений каждого альпиниста на каждую гору.

Необходимо предусмотреть возможность получения отчета, в котором для каждой горы отражается список групп, осуществлявших восхождение, в хронологическом порядке в заданный период времени. Для каждого восхождения выводится информация о количестве членов в группе и итоговое значение по участникам по всему отчету.

### Запрос №1

Выбираем маршруты

Данные из трех таблиц с подзапросом и сортировкой по убыванию высоты

Текст запроса:

```
SELECT opisanie, prodolgitelnost,goryy.vysota, goryy.name_gory, goryy.nrayon
FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory,
r.name_rayon as nrayon
FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon )as goryy
on gory=id_gory order by goryy.vysota DESC
```

Результат:

The screenshot shows the pgAdmin 4 web interface. On the left is the 'Browser' pane with a tree view of the database structure. The 'Query Editor' pane in the center contains a SQL query. The 'Data Output' pane at the bottom displays the results of the query in a table format.

**SQL Query:**

```

1 SELECT opisanie, prodolgitel'nost,goryy.vysota, goryy.name_gory, goryy.nrayon
2 FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory, r.name_rayon
3 FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon )as goryy
4 on gory=id_gory order by goryy.vysota DESC
5
6
7

```

**Data Output Table:**

	opisanie text	prodolgitel'nost numeric	vysota integer	name_gory character (100)	nrayon character (100)
1	Сказки Эльбруса		46	5642 Эльбрус	... Большой Кавказ
2	Восхождение на Джангу		49	5058 Джанга	... Большой Кавказ
3	Восхождение на Белуху		45	4509 Белуха	... Алтай
4	Алтайская сказка		46	4173 Маашей-Баш	... Алтай
5	Облегченный маршрут на Алтае		46	3738 Беркутаул	... Алтай

## Запрос №2

Выбираем маршруты , тот же запрос что и в первом варианте добавляем условия в подзапрос только район Алтайские горы и высота горы больше 4000 км

Данные из трех таблиц с подзапросом и сортировкой по убыванию высоты

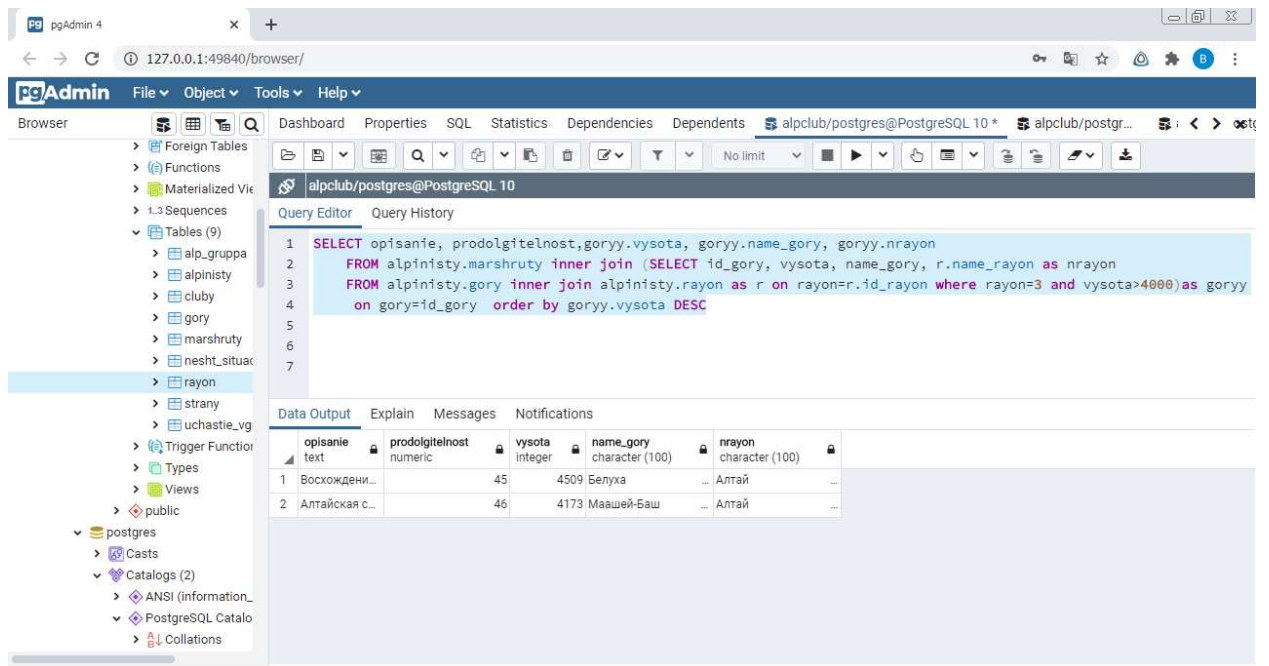
Текст запроса:

```

SELECT opisanie, prodolgitel'nost,goryy.vysota, goryy.name_gory, goryy.nrayon
FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory,
r.name_rayon as nrayon
FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon where rayon=3
and vysota>4000)as goryy
on gory=id_gory order by goryy.vysota DESC

```

Результат:



### Запрос №3

Выбираем маршруты , тот же запрос что и во втором варианте добавляем условия в подзапрос только район Алтайские горы и высота горы больше 4000 км

Добавляем конкатенацию строк (Выводим дополнительное поле с описанием Маршрута вместе с названием района)

Текст запроса:

```

SELECT opisanie, prodolgitelnost,goryy.vysota, goryy.name_gory,
goryy.nrayon,goryy.name_gory||' ||goryy.nrayon as Конкатенация
FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory,
r.name_rayon as nrayon
FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon where rayon=3
and vysota>4000)as goryy
on gory=id_gory order by goryy.vysota DESC

```

Результат:

The screenshot shows the pgAdmin 4 interface. On the left is the 'Browser' pane with a tree view of the database structure. The main area is divided into 'Query Editor' and 'Data Output'.

**Query Editor:**

```

1 SELECT opisanie, prodolgitel'nost,goryy.vysota, goryy.name_gory, goryy.nrayon,goryy.name_gory||' '||goryy.nrayon as
2 FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory, r.name_rayon as nrayon
3 FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon where rayon=3 and vysota>4000)as goryy
4 on gory=id_gory order by goryy.vysota DESC
5
6
7

```

**Data Output:**

opisanie	prodolgitel'nost	vysota	name_gory	nrayon	Конкатенация
Восхождени...		45	4509 Белуха	Алтай	Белуха Алтай
Алтайская с...		46	4173 Маашей-Баш	Алтай	Маашей-Баш Алтай

## Запрос №4

Выбираем маршруты

Извлекаем подстроку (первую букву в названии горного района) в новое поле

Текст запроса:

```

SELECT opisanie, prodolgitel'nost,goryy.vysota, goryy.name_gory,
goryy.nrayon,substring(goryy.nrayon from 1 for 1 ) as Подстрока
FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory,
r.name_rayon as nrayon
FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon )as goryy
on gory=id_gory order by goryy.vysota DESC

```

Результат:

The screenshot shows the pgAdmin 4 interface with a different query and results.

**Query Editor:**

```

1 SELECT opisanie, prodolgitel'nost,goryy.vysota, goryy.name_gory, goryy.nrayon,substring(goryy.nrayon from 1 for 1 )
2 FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory, r.name_rayon as nrayon
3 FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon )as goryy
4 on gory=id_gory order by goryy.vysota DESC
5
6
7

```

**Data Output:**

opisanie	prodolgitel'nost	vysota	name_gory	nrayon	Подстрока
Сказки Эльб...		46	5642 Эльбрус	Большой Кавказ	Б
Восхождени...		49	5058 Джанга	Большой Кавказ	Б
Восхождени...		45	4509 Белуха	Алтай	А
Алтайская с...		46	4173 Маашей-Баш	Алтай	А
Облегченны...		46	3738 Беркутаул	Алтай	А

## Запрос №5

Выбираем маршрут с самой высокой вершиной использовано 2 подзапроса и агрегатная функция max

Текст запроса:

```
SELECT opisanie, prodolgitelnost,goryy.vysota, goryy.name_gory, goryy.nrayon
FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory,
r.name_rayon as nrayon
FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon inner join
(SELECT Max(vysota)as v
FROM alpinisty.gory)as maxvysota on vysota=maxvysota.v)as goryy
on gory=id_gory
```

Результат:

The screenshot shows the pgAdmin 4 interface. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SELECT opisanie, prodolgitelnost,goryy.vysota, goryy.name_gory, goryy.nrayon
2 FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory, r.name_rayon as nrayon
3 FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon inner join (SELECT Max(vysota)as v
4 FROM alpinisty.gory)as maxvysota on vysota=maxvysota.v)as goryy
5 on gory=id_gory
6
7
8
```

Below the query editor, the 'Data Output' tab shows the results of the query. The table has 5 columns: opisanie, prodolgitelnost, vysota, name\_gory, and nrayon. The first row of data is:

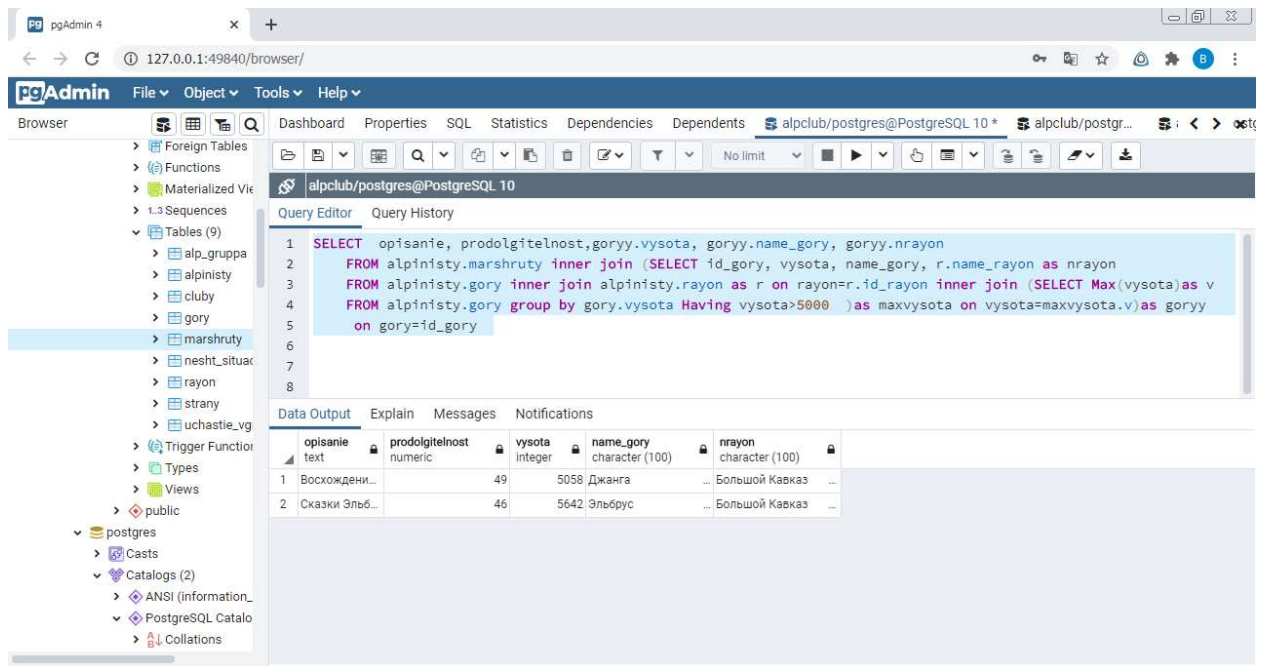
opisanie	prodolgitelnost	vysota	name_gory	nrayon
Сказки Эльб...	46	5642	Эльбрус	Большой Кавказ

## Запрос №6

Выбираем маршрут с самой высокой вершиной использовано 2 подзапроса и агрегатная функция max и предложение having (условие высота больше 5000 м)

```
SELECT opisanie, prodolgitelnost,goryy.vysota, goryy.name_gory, goryy.nrayon
FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory,
r.name_rayon as nrayon
FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon inner join
(SELECT Max(vysota)as v
FROM alpinisty.gory group by gory.vysota Having vysota>5000 )as maxvysota on
vysota=maxvysota.v)as goryy
on gory=id_gory
```

Результат:



## Запрос №7

Выбираем маршрут и горы на которые не было еще восхождений

Текст запроса:

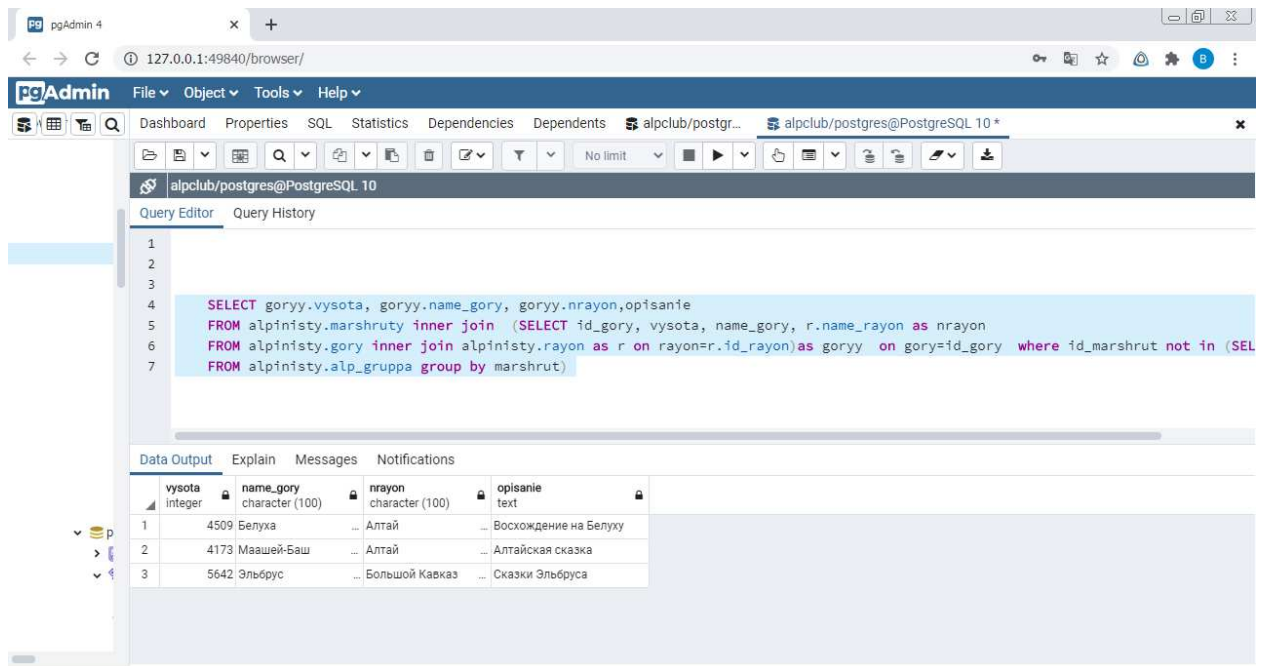
```

SELECT goryy.vysota, goryy.name_gory, goryy.nrayon, opisanie
FROM alpinisty.marshruty inner join (SELECT id_gory, vysota, name_gory,
r.name_rayon as nrayon
FROM alpinisty.gory inner join alpinisty.rayon as r on rayon=r.id_rayon) as goryy on
gory=id_gory where id_marshrut not in (SELECT marshrut
FROM alpinisty.alp_gruppa group by marshrut)

```

Результат:





## Запрос №8

Количество восхождений каждого альпиниста на каждую гору

Текст запроса:

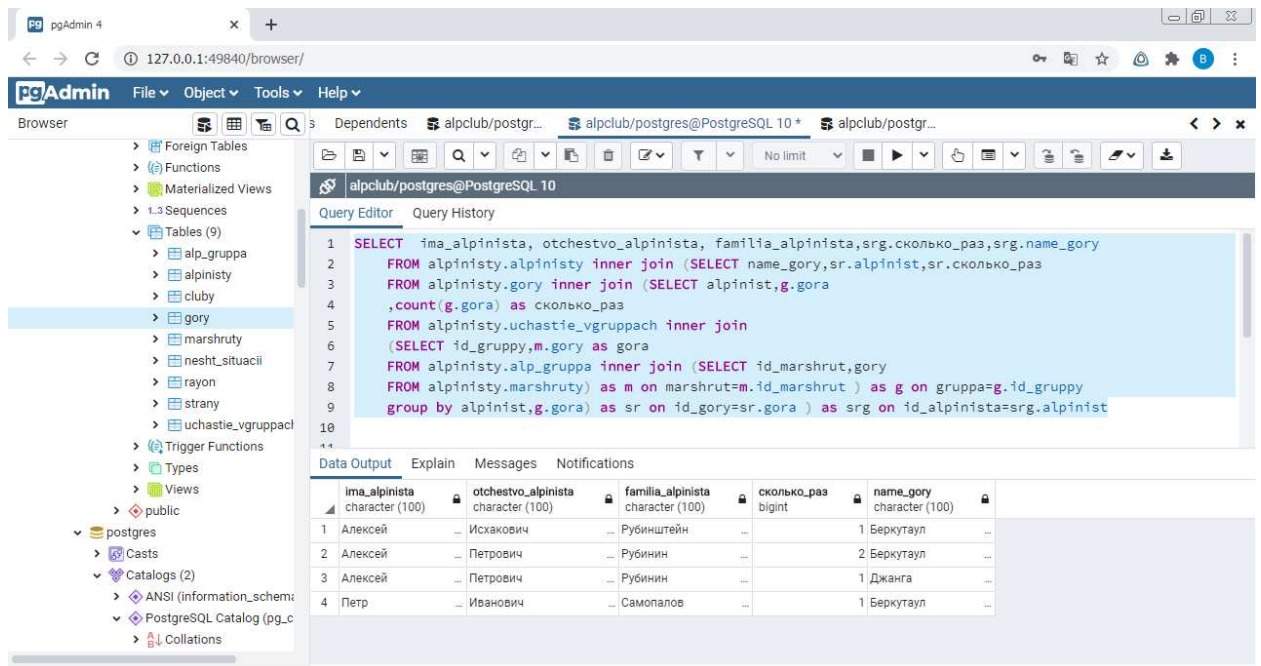
```

SELECT ima_alpinista, otchestvo_alpinista,
familia_alpinista, srg.сколько_раз, srg.name_gory
FROM alpinisty.alpinisty inner join (SELECT name_gory, sr.alpinist, sr.сколько_раз
FROM alpinisty.gory inner join (SELECT alpinist, g.gora
, count(g.gora) as сколько_раз
FROM alpinisty.uchastie_vgruppach inner join
(SELECT id_gruppy, m.gory as gora
FROM alpinisty.alp_gruppa inner join (SELECT id_marshrut, gory
FROM alpinisty.marshruty) as m on marshrut=m.id_marshrut ) as g on
gruppa=g.id_gruppy
group by alpinist, g.gora) as sr on id_gory=sr.gora ) as srg on id_alpinista=srg.alpinist

```

Результат





## Запрос №9

Сколько альпинистов побывали на каждой горе

Текст запроса:

SELECT name\_gory, sr.колич\_альпинистов

FROM alpinisty.gory inner join (select count(tt.id\_gora) as

колич\_альпинистов, tt.id\_gora as id\_goryy from (SELECT alpinist, g.gora as id\_gora

FROM alpinisty.uchastie\_vgruppach inner join

(SELECT id\_gruppy, m.gory as gora

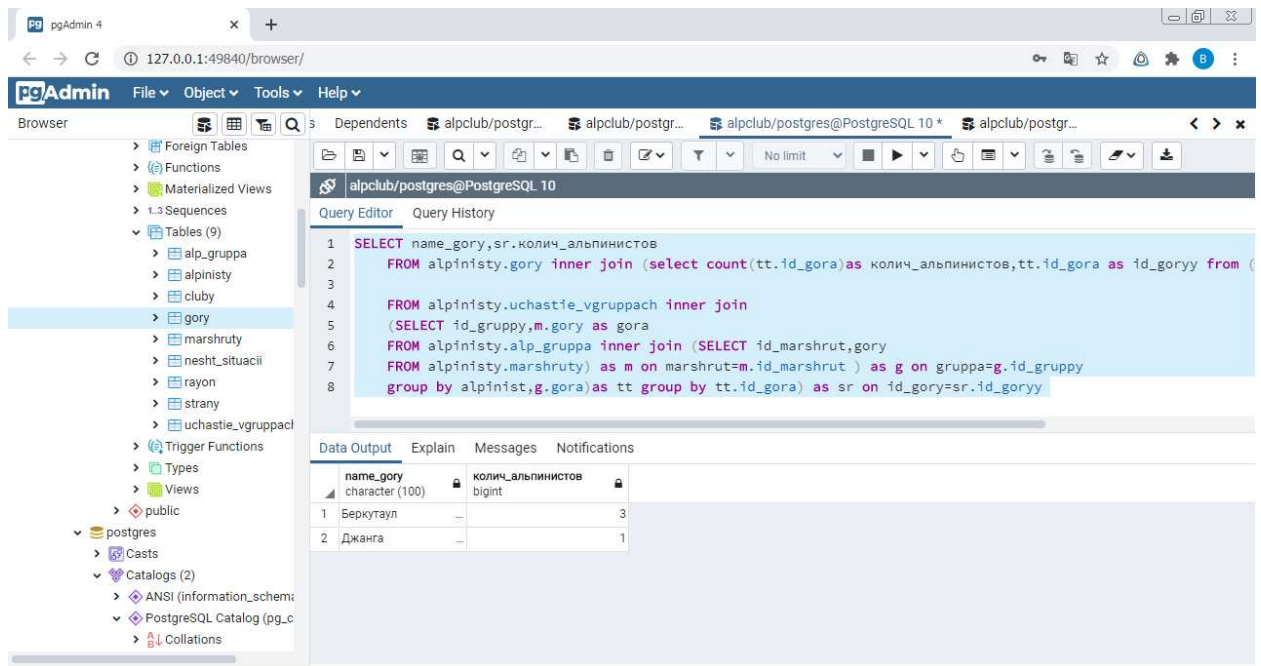
FROM alpinisty.alp\_gruppa inner join (SELECT id\_marshrut, gory

FROM alpinisty.marshruty) as m on marshrut=m.id\_marshrut ) as g on

gruppa=g.id\_gruppy

group by alpinist, g.gora) as tt group by tt.id\_gora) as sr on id\_gory=sr.id\_goryy

Результат



## Запрос №10

Список групп (восхождений) за определенное время

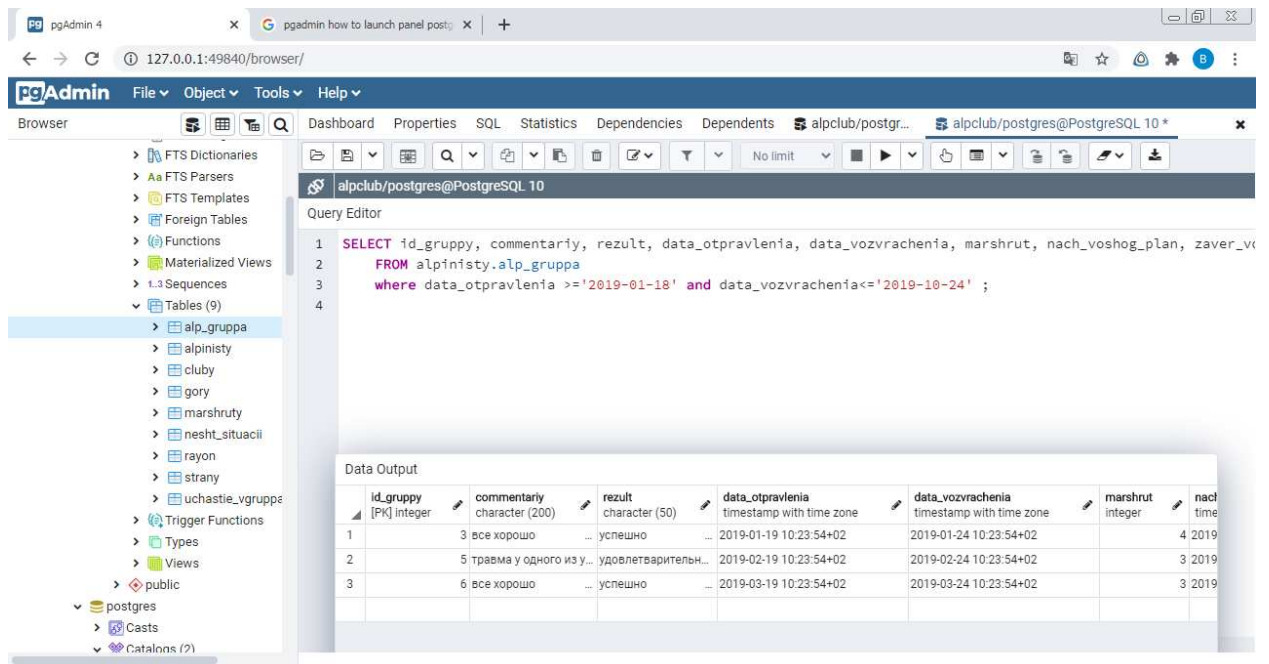
Текст запроса:

```
SELECT id_gruppy, commentariy, rezult, data_otpravlenia, data_vozvrachenia, marshrut,
nach_voshog_plan, zaver_voshog_plan, nach_voshog_fact, zaver_voshog_fact
```

```
FROM alpinisty.alp_gruppa
```

```
where data_otpravlenia >='2019-01-18' and data_vozvrachenia<='2019-10-24' ;
```

Результат



## Запрос №11

Список альпинистов осуществлявших восхождение в определенный интервал времени

Текст запроса:

```

SELECT id_gruppy, commentariy, rezult, data_otpravlenia, data_vozvrachenia, marsh.opisanie,
ima_alpinista, otchestvo_alpinista, familia_alpinista, rezult_alpinista, name_ns
FROM alpinisty.alp_gruppa
inner join (SELECT ima_alpinista, otchestvo_alpinista,
familia_alpinista, rezult_alpinista, gruppa, ns.name_ns
FROM alpinisty.uchastie_vgruppach inner join (SELECT id_ns, name_ns
FROM alpinisty.nesht_situacii) as ns on nesht_situacii=ns.id_ns inner join
(SELECT id_alpinista, ima_alpinista, otchestvo_alpinista, familia_alpinista
FROM alpinisty.alpinisty) as alp on alp.id_alpinista= alpinist)
as alp_t on id_gruppy=alp_t.gruppa inner join (SELECT id_marshrut, opisanie
FROM alpinisty.marshruty) as marsh on marshrut=marsh.id_marshrut
where data_otpravlenia >='2019-01-18' and data_vozvrachenia <='2019-12-24' order by
data_otpravlenia;

```

## Результат

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays the database structure, including tables like 'alp\_gruppa', 'alpinisty', 'cluby', 'gory', 'marshruty', 'nesht\_situacii', 'rayon', 'strany', 'uchastie\_vgruppach', 'Types', and 'Views'. The main area shows the 'Query Editor' with the following SQL query:

```

1 SELECT id_gruppy, commentariy, rezult, data_otpravlenia, data_vozvrachenia, marsh.opisanie,
2 ima_alpinista, otchestvo_alpinista, familia_alpinista, rezult_alpinista, name_ns
3 FROM alpinisty.alp_gruppa
4 inner join (SELECT ima_alpinista, otchestvo_alpinista, familia_alpinista, rezult_alpinista, gruppa, ns.name_ns
5 FROM alpinisty.uchastie_vgruppach inner join (SELECT id_ns, name_ns
6 FROM alpinisty.nesht_situacii) as ns on nesht_situacii=ns.id_ns inner join
7 (SELECT id_alpinista, ima_alpinista, otchestvo_alpinista, familia_alpinista
8 FROM alpinisty.alpinisty) as alp on alp.id_alpinista= alpinist)
9 as alp_t on id_gruppy=alp_t.gruppa inner join (SELECT id_marshrut, opisanie
10 FROM alpinisty.marshruty) as marsh on marshrut=marsh.id_marshrut
11 where data_otpravlenia >='2019-01-18' and data_vozvrachenia <='2019-12-24' order by data_otpravlenia;
12

```

Below the query editor, the 'Data Output' table is displayed, showing the results of the query. The table has 8 columns: id\_gruppy, commentariy, rezult, data\_otpravlenia, data\_vozvrachenia, opisanie, and ima\_alpinista. The results are as follows:

id_gruppy	commentariy	rezult	data_otpravlenia	data_vozvrachenia	opisanie	ima_alpinista
1	3 все хорошо	... успешно	2019-01-19 10:23:54+02	2019-01-24 10:23:54+02	Восхождение...	Алексей
2	2 все хорошо	... успешно	2019-10-19 10:23:54+03	2019-10-24 10:23:54+03	Облегченны...	Алексей
3	2 все хорошо	... успешно	2019-10-19 10:23:54+03	2019-10-24 10:23:54+03	Облегченны...	Петр
4	2 все хорошо	... успешно	2019-10-19 10:23:54+03	2019-10-24 10:23:54+03	Облегченны...	Алексей
5	4 потеряли палатку	... успешно	2019-11-19 12:23:54+02	2019-11-24 12:23:54+02	Облегченны...	Алексей