

Министерство образования и науки Российской Федерации  
федеральное государственное автономное образовательное учреждение  
высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИТМО»  
Факультет среднего профессионального образования

ОТЧЁТ  
ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 5  
по теме: Создание запросов  
по дисциплине: Основы проектирования баз данных

Специальность 09.02.07 Информационные системы и программирование

Проверил:  
\_\_\_\_\_ Говоров А. И.  
Дата: «\_\_\_\_\_» \_\_\_\_\_ 2020г.  
Оценка \_\_\_\_\_

Выполнил:  
студент группы Y2336  
\_\_\_\_\_ Наумов М.А.

Санкт-Петербург 2020

## ЦЕЛЬ РАБОТЫ

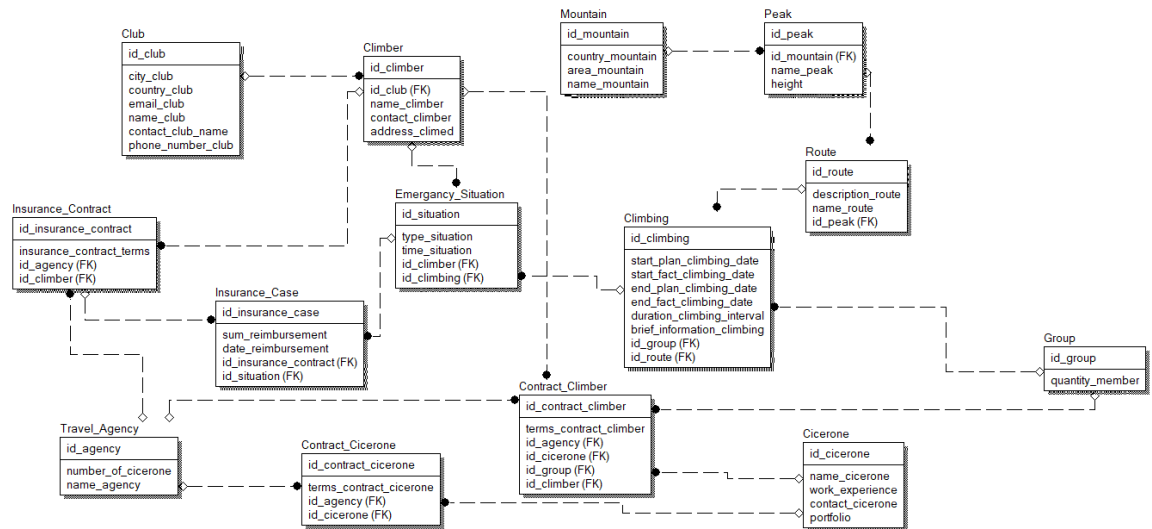
Цель практической работы №5: овладеть практическими навыками создания запросов к базе данных PostgreSQL.

## ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Необходимо реализовать некоторое количество SQL-запросов. В лабораторной работе №4 Вам требуется написать определенное количество запросов на определенное количество баллов, зависимое от оценки, на которую Вы претендуете. . Примерный набор требуемых запросов:

1. выбор значений, заданных атрибутов из более, чем двух таблиц, с сортировкой – от 1 балла;
2. использование условий WHERE, состоящих из более, чем одного условия – от 1 балла;
3. использование функций для работы с датами – от 2 баллов;
4. использование строковых функций – от 3 баллов;
5. запрос с использованием подзапросов – от 2 баллов (многострочный подзапрос - от 3 баллов);
6. вычисление групповой (агрегатной) функции – от 1 балла (с несколькими таблицами – от 3 баллов);
7. вычисление групповой (агрегатной) функции с условием HAVING – от 2 баллов;
8. использование предикатов EXISTS, ALL, SOME и ANY - от 4 баллов;
9. использование запросов с операциями реляционной алгебры (объединение, пересечение и т.д.) - от 3 баллов;
10. использование объединений запросов (inner join и т.д.) - от 3 баллов. Запросов должно быть не менее 10 (30 баллов на 5, 25 баллов на 4, 20 баллов на 3)(5, 4, 3 - это оценки, а не суммы баллов в журнале).

## 1. База данных



## 2. Запросы:

Вывести всех альпинистов:

```
select "name_climber" from "climber";
```

	name_climber "char[]"
1	{M}
2	{M}
3	{A}
4	{A}
5	{J}

Вывести все клубы из Санкт-Петербурга:

```
select "name_club" from "club" where "city_club"[1]='S';
```

	name_club "char[]"
1	{T}
2	{M}

Вывести самую высокую вершину

```
select "name_peak", "hight_peak" from "peak" where "hight_peak"=(select  
max("hight_peak") from "peak");
```

	name_peak "char[]"	hight_peak integer
1	{R}	5423

Вывести самую низкую вершину в России

```
select "name_peak", "hight_peak" from "peak" where "hight_peak"=(select min("hight_peak") from "peak" as p, "mountain" as m where p."id_mountain"=m."id_mountain" and "country_mountain"[1]='R');
```

	name_peak "char[]"	hight_peak integer
1	{R}	1234

Вывести количество несштатных ситуаций за последний год и сгруппировать по типу

```
select "type_situation", count(*) from "emergancy_situation" where "time_situation" between '00:00' and '12:00' group by "type_situation";
```

	type_situatio "char[]"	count bigint
1	{C}	1
2	{E}	1
3	{D}	1

Вывести количество контрактов по гидам(гид-агенство) и сортировать по увеличению стажа

```
select "name_cicerone", count(*) from "cicerone" as c, "contract_cicerone" as cc where c."id_cicerone"=cc."id_cicerone" group by c."id_cicerone" order by "work_experience";
```

	name_cicero "char[]"	count bigint
1	{C}	1
2	{A}	1
3	{D}	1
4	{B}	1
5	{E}	1

Вывести суммарные выплаты страховых по альпинистам

```
select "name_climber", sum("sum_reimbursement") as reimbursement from "climber" as c, "insurance_contract" as ic, "insurance_case" as i_case where c."id_climber"=ic."id_climber" and ic."id_insurance_contract"=i_case."id_insurance_contract" group by c."id_climber" order by reimbursement desc;
```

	name_climber "char[]"	reimbursement bigint
1	{A}	90000
2	{J}	70000
3	{M}	55000
4	{A}	38000
5	{M}	31500

Вывести маршрут, на котором чаще всего происходят несчастные ситуации

```
select "name_route" from "route" as r where ((select count(*) from "climbing" as c, "emergancy_situation" as es where r."id_route"=c."id_route" and c."id_climbing"=es."id_climbing"))=(select max(c_es) from ( select count(*) as c_es from "route" as rt, "climbing" as c, "emergancy_situation" as es where rt."id_route"=c."id_route" and c."id_climbing"=es."id_climbing" group by rt."id_route") foo);
```

	name_route "char[]"
1	{A}

Вывести группы, которые ходили на самую высокую вершину

```
select g."id_group" from "group" as g, "climbing" as c, "route" as r, "peak" as p where r."Id_peak"=p."id_peak" and r."id_route"=c."id_route" and g."id_group"=c."id_group" and "hight_peak" = (select max("hight_peak") from "route" as rt, "peak" as pk where rt."Id_peak"=pk."id_peak");
```

	id_group [PK] integer
1	5

Вывести среднюю высоту вершин находящихся в России

```
select avg("hight_peak") from "peak" inner join "mountain" using(id_mountain) where "country_mountain"[1]='R';
```

	avg numeric
1	3184.0000000000000000

## ВЫВОД

В практической работе №5 были получены практические навыки создания запросов к базам данных PostgreSQL.