

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
Факультет инфокоммуникационных технологий

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4**

по теме: Запросы на выборку данных к БД PostgreSQL.
ПРЕДСТАВЛЕНИЯ В PostgreSQL

по дисциплине: Проектирование и реализация баз данных
Специальность: 09.03.03 Мобильные и сетевые технологии

Проверил:
Говорова М.М. _____
Дата: «9» 06 2021г.
Оценка _____
Выполнил:
студент группы К3241
Кумпан В.В.

Санкт-Петербург
2021

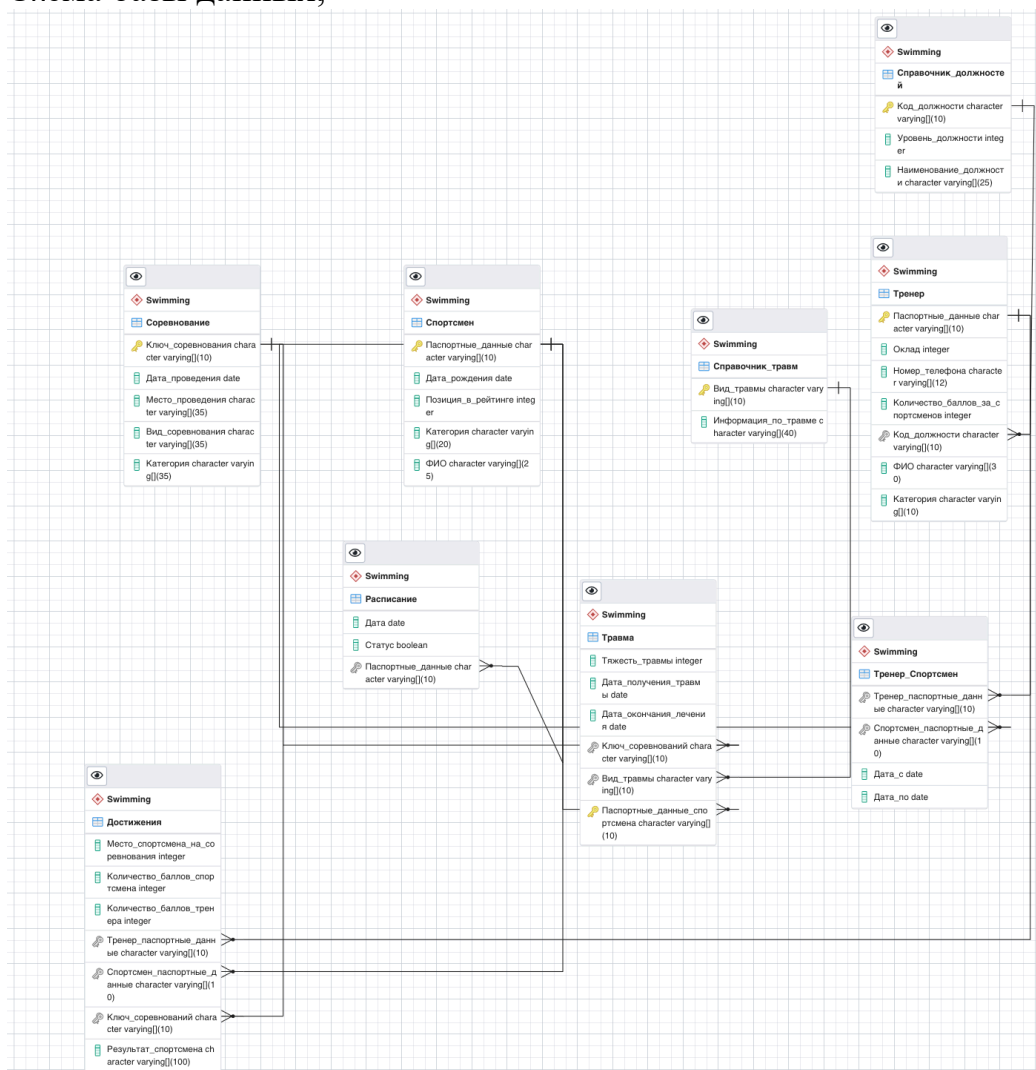
Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL и использования подзапросов при модификации данных.

Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов.
4. Просмотреть историю запросов.

Выполнение

1. Наименование БД “Swimming”
2. Схема базы данных;



3. Запросы к базе данных.

- С каким количеством спортсменов работает каждый тренер.

```
1 SELECT "Тренер_паспортные_данные", COUNT("Тренер_паспортные_данные")
2 FROM "Swimming"."Тренер_Спортсмен"
3 group by "Swimming"."Тренер_Спортсмен"."Тренер_паспортные_данные"
4
```

	Тренер_паспортные_данные... character varying[] (10)	count bigint
1	{0000000013}	1
2	{0000000011}	2
3	{0000000012}	2
4	{0000000010}	4

- Найти тренеров, чьи спортсмены не имеют травм.

```
4 SELECT "Тренер_паспортные_данные" FROM "Swimming"."Тренер_Спортсмен"
5 WHERE "Спортсмен_паспортные_данные" NOT IN (SELECT DISTINCT "Паспортные_данные_спортсмена"
6 FROM "Swimming"."Травма");
```

	Тренер_паспортные_данные... character varying[] (10)
1	{0000000010}
2	{0000000010}
3	{0000000011}
4	{0000000011}
5	{0000000012}
6	{0000000013}

- Найти тренера, получающего минимальную зарплату.

```
5 SELECT "ФИО", "Оклад" FROM "Swimming"."Тренер"
6 WHERE "Оклад" = (SELECT MIN("Оклад") from "Swimming"."Тренер")
```

	ФИО character varying[] (30)	Оклад integer
1	{"Генадий Федорович"}	40000

- Определить количество соревнований каждой категории.

```
6 SELECT "Категория", COUNT("Категория")
7 FROM "Swimming"."Соревнование"
8 group by "Категория"
```

	Категория character varying[] (35)	count bigint
1	{Городской}	1
2	{Региональный}	1
3	{Всероссийский}	1

- Найти тренера, работающего с самыми молодыми спортсменами (средний возраст спортсменов минимален).

```

6 SELECT "Тренер_паспортные_данные", "Спортсмен_паспортные_данные"
7 FROM "Swimming"."Тренер_Спортсмен"
8 WHERE "Спортсмен_паспортные_данные" = (
9     SELECT "Паспортные_данные" FROM "Swimming"."Спортсмен"
10    WHERE "Дата_рождения" = (SELECT MIN("Дата_рождения") from "Swimming"."Спортсмен")
11 )

```

	Data Output	Explain	Query History	Messages
	Тренер_паспортные_данные... character varying[] (10)		Спортсмен_паспортные_данные character varying[] (10)	
1	{0000000010}		{0000000000}	

- Сколько спортсменов участвует в соревнованиях каждой категории в заданный период.

	Data Output	Explain	Query History	Messages
	count bigint		Вид_соревнования character varying[] (35)	
1	19		{Смешанный}	
2	19		{Финал}	

```

30 SELECT COUNT("Swimming"."Соревнование"."Вид_соревнования"),
31 "Swimming"."Соревнование"."Вид_соревнования"
32 FROM "Swimming"."Достижения"
33     INNER JOIN "Swimming"."Соревнование"
34         ON "Ключ_соревнования" = "Ключ_соревнования"
35     WHERE "Дата_проведения" < '2020-06-01'
36 GROUP by ("Swimming"."Соревнование"."Вид_соревнования")

```

- Для всех спортсменов определить количество соревнований, в которых они участвовали.

	Спортсмен_паспортные_данные character varying[] (10)	count bigint
1	{0000000001}	6
2	{0000000002}	7
3	{0000000000}	6

```

29 SELECT "Спортсмен_паспортные_данные", COUNT("Спортсмен_паспортные_данные")
30 FROM "Swimming"."Достижения"
31 GROUP by ("Спортсмен_паспортные_данные")

```

4. Создание представления. Содержащее сведения обо всех тренерах, соревнованиях, в которых участвовали их спортсмены и местах, которые они заняли.

```
2 CREATE VIEW swimming_view AS
3 SELECT "Дата_проведения", "Место_проведения","Вид_соревнования", "Категория", "Место_спортсмена_на_соревнования",
4 "Количество_баллов_спортсмена", "Количество_баллов_тренера", "Тренер_паспортные_данные",
5 "Результат_спортсмена", "Степень"
6 FROM "Swimming". "Соревнование"
7 INNER JOIN "Swimming". "Достижения"
8 ON "Ключ_соревнования" = "Ключ_соревнования"
9 INNER JOIN "Swimming". "Тренер"
10 ON "Тренер_паспортные_данные" = "Тренер_паспортные_данные"
```

Data Output Explain Messages Query History

CREATE VIEW

Query returned successfully in 87 msec.

```
12 SELECT * FROM swimming_view
```

Data Output Explain Messages Query History

	Дата_проведения date	Место_проведения character varying[] (35)	Вид_соревнования character varying[] (35)	Категория character varying[] (35)
1	2020-04-01	{Бассейн атлантика}	{Смешанный}	{Региональный}
2	2020-02-01	{Бассейн водных видов сп...	{Финал}	{Всероссийский}
3	2020-06-01	{Бассейн университетский}	{Общий}	{Городской}
4	2020-04-01	{Бассейн атлантика}	{Смешанный}	{Региональный}
5	2020-02-01	{Бассейн водных видов сп...	{Финал}	{Всероссийский}

Data Output Explain Messages Query History

	Дата_проведения date	Место_проведения character varying[] (35)	Вид_соревнования character varying[] (35)	Категория character varying[] (35)	Место_спортсмена_на_соревнования integer
1	2020-04-01	{Бассейн атлантика}	{Смешанный}	{Региональный}	1
2	2020-02-01	{Бассейн водных видов сп...	{Финал}	{Всероссийский}	1
3	2020-06-01	{Бассейн университетский}	{Общий}	{Городской}	1
4	2020-04-01	{Бассейн атлантика}	{Смешанный}	{Региональный}	1
5	2020-02-01	{Бассейн водных видов сп...	{Финал}	{Всероссийский}	1

Количество_баллов_спортсмена integer	Количество_баллов_тренера integer	Тренер_паспортные_данные character varying[] (10)	Результат_спортсмена character varying[] (100)	Степень character varying[] (10)
100	10	{0000000010}	{null}	{Высшая}
100	10	{0000000010}	{null}	{Высшая}
100	10	{0000000010}	{null}	{Высшая}
100	10	{0000000010}	{null}	{Высшая}
100	10	{0000000010}	{null}	{Высшая}

5. Найти самую распространенную травму.

```
11 CREATE VIEW swimming_injury AS
12 SELECT "Swimming"."Справочник_травм"."Информация_по_травме", COUNT("Swimming"."Травма"."Вид_травмы")
13 FROM "Swimming"."Травма"
14 INNER JOIN "Swimming"."Справочник_травм"
15 ON "Swimming"."Травма"."Вид_травмы" = "Swimming"."Справочник_травм"."Вид_травмы"
16 GROUP BY "Swimming"."Справочник_травм"."Информация_по_травме"
17 ORDER BY "Swimming"."Справочник_травм"."Информация_по_травме" LIMIT 1
```

Data Output Explain Messages Query History

CREATE VIEW

Query returned successfully in 75 msec.

```
17 SELECT * FROM swimming_injury
```

Data Output Explain Messages Query History

	Информация_по_травме character varying[] (40)	count bigint
1	{Растяжение связок}	2

6. Запросы на модификацию данных UPDATE

Изменение количества баллов за спортсмена для каждого тренера по итогам всех соревнований в которых выступали их воспитанники

```
14 UPDATE "Swimming"."Тренер"
15 SET "Количество_баллов_за_спортсменов" = Balls."sum"
16 FROM (SELECT SUM("Количество_баллов_тренера")
17        FROM "Swimming"."Достижения"
18        INNER JOIN "Swimming"."Тренер"
19        ON "Тренер_паспортные_данные" = "Swimming"."Тренер"."Паспортные_данные"
20        GROUP BY "Тренер_паспортные_данные") as Balls
21 WHERE "Паспортные_данные" IN (SELECT DISTINCT "Тренер_паспортные_данные"
22        FROM "Swimming"."Тренер_Спортсмен"
23        WHERE "Спортсмен_паспортные_данные" NOT IN
24              (SELECT DISTINCT Паспортные_данные_спортсмена FROM "Swimming"."Травма"))
```

UPDATE 2

Query returned successfully in 53 msec.

До

	Паспортные_данные [PK] character varying[] (10)	Оклад integer	Номер_телефона character varying[] (12)	Количество_баллов_за_спортсменов integer	Код_должности character varying[] (10)	ФИО character varying[] (:
1	{0000000012}	44000	{+790000000002}		0 {0000000000}	{Генадий Генадьев
2	{0000000010}	50000	{+790000000000}		0 {0000000000}	{Генадий Алексеев
3	{0000000011}	45000	{+790000000001}		0 {0000000000}	{Генадий Федоров
4	{0000000013}	40000	{+790000000003}		0 {0000000003}	{Генадий Федоров

После

	Паспортные_данные [PK] character varying[] (10)	Оклад integer	Номер_телефона character varying[] (12)	Количество_баллов_за_спортсменов integer	Код_должности character varying[] (10)	ФИО character varying[] (:
1	{0000000012}	44000	{+790000000002}	0	{0000000000}	{Генадий Генадьев
2	{0000000010}	50000	{+790000000000}	0	{0000000000}	{Генадий Алексеев
3	{0000000011}	45000	{+790000000001}	31	{0000000000}	{Генадий Федоров
4	{0000000013}	40000	{+790000000003}	31	{0000000003}	{Генадий Федоров

7. Добавить спортсмена в травму получившего меньше всего очков за соревнования (id соревнований) из -за растяжения связок

```
26 INSERT INTO "Swimming"."Травма"
27 ("Тяжесть_травмы", "Дата_получения_травмы", "Дата_окончания_лечения", "Ключ_соревнований", "Вид_травмы",
28 "Паспортные_данные_спортсмена")
29 VALUES(3, ( SELECT "Дата_проведения" FROM "Swimming"."Соревнование"
30 WHERE "Ключ_соревнования" = (SELECT "Ключ_соревнований" FROM "Swimming"."Достижения"
31 WHERE "Ключ_соревнований" = '{999999999}' and "Спортсмен_паспортные_данные" = (
32 SELECT "Спортсмен_паспортные_данные" FROM "Swimming"."Достижения"
33 WHERE "Ключ_соревнований" = '{999999999}'
34 and "Количество_баллов_спортсмена" = (
35 SELECT MIN("Количество_баллов_спортсмена") FROM "Swimming"."Достижения")
36 LIMIT 1) LIMIT 1))
37 , '2020-07-10', '{999999999}', '{0505050505}', (SELECT "Спортсмен_паспортные_данные" FROM "Swimming"."Достижения"
38 WHERE "Ключ_соревнований" = '{999999999}'
39 and "Количество_баллов_спортсмена" = (
40 SELECT MIN("Количество_баллов_спортсмена") FROM "Swimming"."Достижения") LIMIT 1))
```

До





Data Output		Explain	Messages	Query History							
	Тяжесть_травмы integer		Дата_получения_травмы date		Дата_окончания_лечения date		Ключ_соревнований character varying[] (10)		Вид_травмы character varying[] (10)		Паспортные_данные [PK] character varying[]
1		5	2020-06-01		2020-07-10		{999999997}		{0101010101}		{0000000009}
2		5	2020-02-01		2020-03-10		{999999998}		{0101010101}		{0000000001}

После

Data Output		Explain	Messages	Query History		
<div>Тяжесть_травмы</div> <div>integer</div>	<div>Дата_получения_травмы</div> <div>date</div>	<div>Дата_окончания_лечения</div> <div>date</div>	<div>Ключ_соревнований</div> <div>character varying[] (10)</div>	<div>Вид_травмы</div> <div>character varying[] (10)</div>	<div>Паспортные_данные</div> <div>[PK] character varying[]</div>	
1	5 2020-06-01	2020-07-10	{999999997}	{0101010101}	{0000000009}	
2	5 2020-02-01	2020-03-10	{999999998}	{0101010101}	{0000000001}	
3	3 2020-04-01	2020-07-10	{999999999}	{0505050505}	{0000000000}	

8. Удалим из расписания тех кто находится в травме

```
44 DELETE FROM "Swimming"."Расписание"
45 WHERE "Паспортные_данные" IN (SELECT "Паспортные_данные_спортсмена" FROM "Swimming"."Травма" )
46 and "Дата" > CURRENT_DATE
```

Data Output		Explain	Messages	Query History
 Дата date		Статус boolean	 Паспортные_данные character varying[] (10)	
1	2021-11...	false	{0000000000}	
2	2021-11...	false	{0000000001}	

До

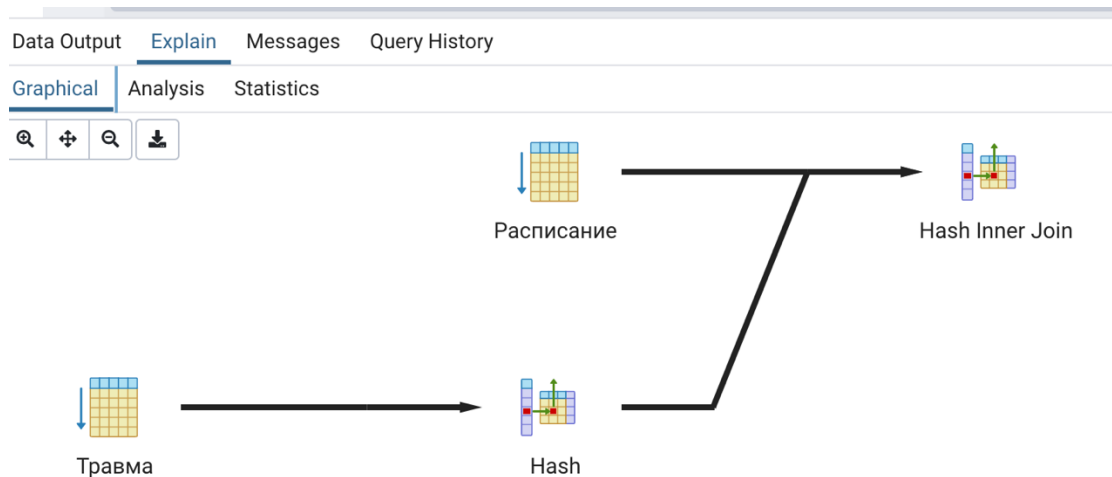
31	2021-11-06	false	{0000000000}
32	2021-11-06	false	{0000000001}

После

29	2020-01-14	false	{0000000002}
30	2020-01-14	false	{0000000002}

Data Output	Explain	Messages	Query History
DELETE 2			
Query returned successfully in 49 msec.			

Графическое представление запроса



Выводы

В процессе выполнения лабораторной работы были получены практические навыки создания представлений и запросов на выборку и модификацию данных к базе данных PostgreSQL с помощью инструмента запросов Query Tool в pgAdmin 4.