

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе № 2
**«ЗАПРОСЫ ФНА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ,
ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В POSTGRESQL»**
по дисциплине **«Базы данных»**

Автор: Чаптыков Николай

Факультет: ИКТ

Группа: K32422

Преподаватель: Говорова М.М.



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2023

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

Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL
2. Составить три запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов
3. Изучить графическое представление запросов и посмотреть историю запросов
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

Описание варианта:

1) БД «Спортивный клуб»

2) Состав реквизитов сущностей

Спортсмен (ФИО, паспортные данные, телефон, ID спортсмена, квалификация спортсмена)

Тренер (ФИО, номер паспорта, телефон, ID тренера, квалификация тренера)

Тренировка (ID тренировки, время конца тренировки, время начала тренировки, дата проведения, место проведения)

Соревнование (ID соревнования, место проведения соревнования, дата начала проведения соревнования, вид соревнования, категория соревнования, вид спорта, дата конца проведения соревнования, название соревнования)

Квалификация тренера (код квалификации, С, По, код должности)

Должность (Код должности, оклад тренера, название должности)

Квалификация спортсмена (история квалификаций, рейтинг спортсмена, квалификация спортсмена)

Ход работы:

Задание 1

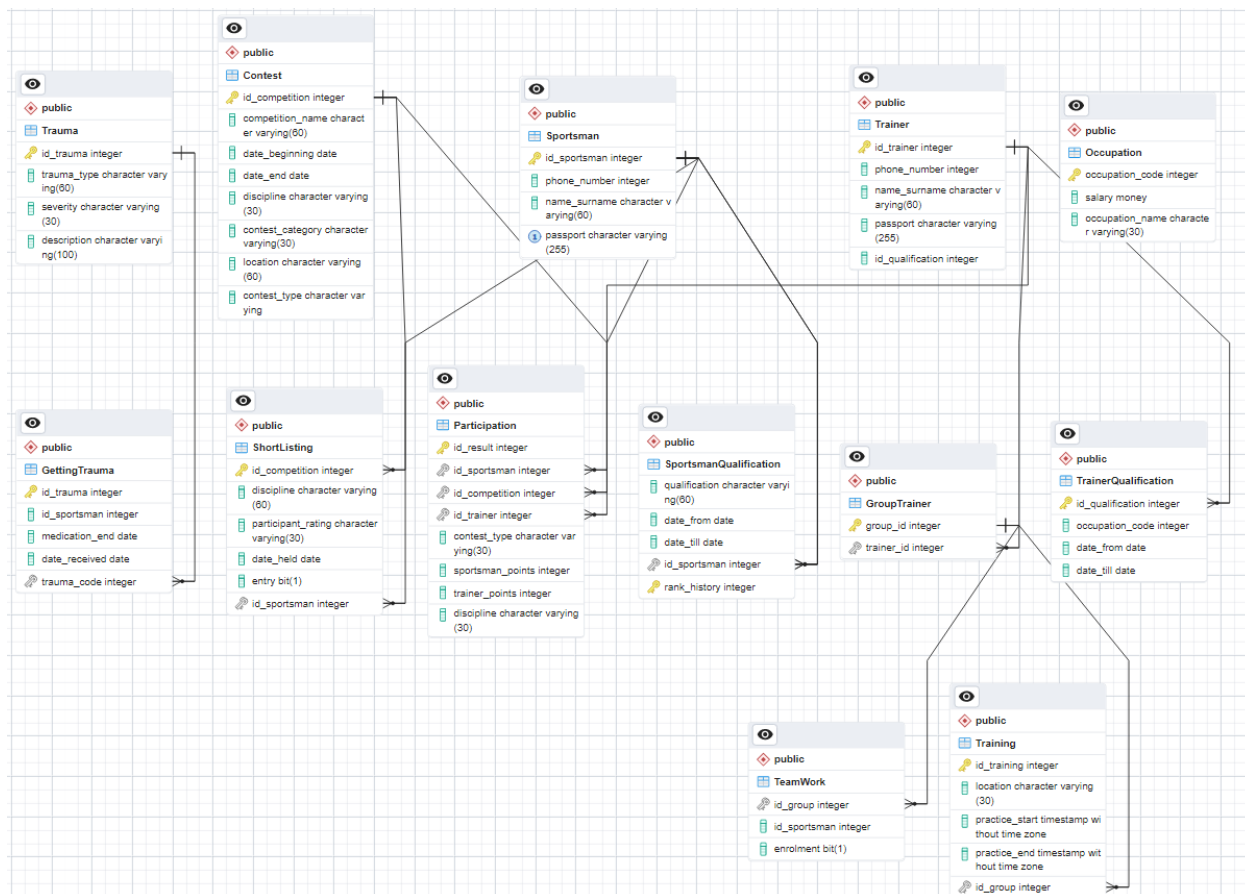


Рисунок 1 - ERD схема базы данных

Задание 2

С каким количеством спортсменов работает каждый тренер:

```

1 SELECT "Trainer".id_trainer as "ID",
2     "Trainer".name_surname as "Имечко фамилия",
3     COUNT("TeamWork".id_sportsman) as "Количество" from "Trainer"
4 LEFT JOIN "GroupTrainer" ON "Trainer".id_trainer = "GroupTrainer".trainer_id
5 LEFT JOIN "TeamWork" ON "TeamWork".id_group = "GroupTrainer".group_id
6 GROUP BY "Trainer".id_trainer
7 ORDER BY "Количество" DESC
  
```

	ID integer	Имечко фамилия character varying (60)	Количество bigint
1	17	Захар Абдурозик	10
2	5	Артем Поклонский	10
3	28	Аяз Поклонский	10
4	3	Нурлан Любимов	10
5	7	Кирилл Поклонский	9
6	41	Василий Благородный	0
7	0	Аяз Харитонович	0
8	40	Аяз Гарт	0
9	46	Кирилл Иванов	0
10	43	Артем Иванов	0
11	32	Нурлан Любимов	0
12	9	Василий Абдурозик	0
13	10	Артем Харитонович	0
14	35	Захар Иванов	0
15	45	Артем Харитонович	0

Найти тренеров, чьи спортсмены не имеют травм:

```
1 SELECT DISTINCT "GroupTrainer".trainer_id AS "ID тренера", "Trainer".name_surname AS "Имячко фамилия" FROM "GroupTrainer"
2 INNER JOIN "TeamWork" ON "TeamWork".id_group = "GroupTrainer".group_id
3 INNER JOIN "Trainer" ON "Trainer".id_trainer = "GroupTrainer".trainer_id
4 WHERE NOT "TeamWork".id_sportsman = ANY (SELECT "GettingTrauma".id_sportsman FROM "GettingTrauma")
```

	ID тренера integer	Имячко фамилия character varying (60)
1	3	Нурлан Любимов
2	5	Артём Поклонский
3	7	Кирилл Поклонский
4	17	Захар Абдурозик
5	28	Аяз Поклонский

Найти тренера, получающего минимальную зарплату:

```
1 SELECT "Trainer".id_trainer as "ID", "Trainer".name_surname as "Имячко фамилия", "Occupation".salary as "Оклад" FROM "Trainer"
2 LEFT JOIN "TrainerQualification" ON "Trainer".id_qualification = "TrainerQualification".id_qualification
3 INNER JOIN "Occupation" ON "Occupation".occupation_code = "TrainerQualification".occupation_code
4 WHERE "Occupation".salary = (SELECT MIN(salary) FROM "Occupation")
5 GROUP BY "Trainer".id_trainer, "Occupation".salary
```

	ID integer	Имячко фамилия character varying (60)	Оклад money
1	19	Василий Гарт	5 731,00 ?

Определить количество соревнований каждой категории:

```
1 SELECT DISTINCT "Contest".contest_category AS "Категория", COUNT("Contest".id_competition) AS "Количество" FROM "Contest"
2 GROUP BY "Contest".contest_category
3
```

	Категория character varying (30)	Количество bigint
1	Мировой	2
2	Региональный	5
3	Галактический	3
4	Школьный	5
5	Муниципальный	4

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

```

1 SELECT trainer_id as "ID", name_surname as "Имячко фамилия", avg_date as "Дата" FROM (
2 SELECT "GroupTrainer".trainer_id, "Trainer".name_surname,
3     to_timestamp(avg(extract(epoch from "Sportsman".birthdate))) as avg_date FROM "GroupTrainer"
4 -- avg от date почему-то не хотел работать
5 INNER JOIN "TeamWork" ON "GroupTrainer".group_id = "TeamWork".id_group
6 INNER JOIN "Sportsman" ON "TeamWork".id_sportsman = "Sportsman".id_sportsman
7 INNER JOIN "Trainer" ON "GroupTrainer".trainer_id = "Trainer".id_trainer
8 GROUP BY "GroupTrainer".trainer_id, "Trainer".name_surname) as q
9 WHERE avg_date = (SELECT MAX(avg_date) FROM (
10 SELECT "GroupTrainer".trainer_id as "ID", "Trainer".name_surname as "Имячко фамилия",
11     to_timestamp(avg(extract(epoch from "Sportsman".birthdate))) as avg_date FROM "GroupTrainer"
12 INNER JOIN "TeamWork" ON "GroupTrainer".group_id = "TeamWork".id_group
13 INNER JOIN "Sportsman" ON "TeamWork".id_sportsman = "Sportsman".id_sportsman
14 INNER JOIN "Trainer" ON "GroupTrainer".trainer_id = "Trainer".id_trainer
15 GROUP BY "GroupTrainer".trainer_id, "Trainer".name_surname) AS f)

```

	ID integer	Имячко фамилия character varying (60)	Дата timestamp with time zone
1	3	Нурлан Любимов	2002-10-27 07:48:00+03

Сколько спортсменов участвует в соревнованиях каждой категории в заданный период:

```

1 SELECT DISTINCT "Contest".contest_category AS "Категория", COUNT("Contest".id_competition) AS "Количество" FROM "Contest"
2 GROUP BY "Contest".contest_category
3

```

	Категория character varying (30)	Количество bigint
1	Мировой	2
2	Региональный	5
3	Галактический	3
4	Школьный	5
5	Муниципальный	4

Для всех спортсменов определить количество соревнований, в которых они участвовали:

```
1 SELECT DISTINCT "Sportsman".id_sportsman as "ID",
2     "Sportsman".name_surname as "Имячко фамилия",
3     COUNT("Participation".id_sportsman) as "Количество" FROM "Sportsman"
4 LEFT JOIN "Participation" ON "Sportsman".id_sportsman = "Participation".id_sportsman
5 GROUP BY "Sportsman".id_sportsman
6 ORDER BY "Количество" DESC
```

	ID integer	Имячко фамилия character varying (60)	Количество bigint
1	18	Света Вольных	6
2	28	Анастасия Хёнгю	5
3	21	Станислав Цой	4
4	12	Саян Бондаренко	3
5	15	Мария Оникиенко	3
6	16	Саян Бондаренко	3
7	1	Сергей Вольных	2
8	7	Станислав Бондаренко	2
9	9	Света Хёнгю	2
10	10	Мария Бондаренко	2
11	23	Станислав Цой	2
12	24	Николай Оникиенко	2
13	30	Мария Захарченко	2
14	0	Мария Хёнгю	1
15	2	Саян Оникиенко	1
16	5	Анастасия Оникиенко	1

Задание 3

Необходимо составить следующие представления:

1. Представление, содержащее сведения обо всех тренерах, соревнованиях, в которых участвовали их спортсмены и местах, которые они заняли:

```
1 SELECT DISTINCT "Trainer".name_surname AS "тренер",
2     "Trainer".id_trainer AS "номер тренера",
3     "Sportsman".name_surname AS "спортсмен",
4     "Sportsman".id_sportsman AS "номер спортсмена",
5     "Contest".competition_name AS "соревнование",
6     "Participation".sportsman_points AS "место"
7 FROM "Trainer"
8 LEFT JOIN "GroupTrainer" ON "GroupTrainer".trainer_id = "Trainer".id_trainer
9 LEFT JOIN "TeamWork" ON "GroupTrainer".group_id = "TeamWork".id_group
10 JOIN "Sportsman" ON "TeamWork".id_sportsman = "Sportsman".id_sportsman
11 LEFT JOIN "Participation" ON "Sportsman".id_sportsman = "Participation".id_sportsman
12 LEFT JOIN "Contest" ON "Participation".id_competition = "Contest".id_competition
13 GROUP BY "Trainer".name_surname, "Trainer".id_trainer, "Sportsman".id_sportsman, "Sportsman".name_surname,
14 |"Participation".sportsman_points, "Contest".competition_name
15 ORDER BY "Trainer".name_surname;
```


1	SELECT * FROM ContestResults					
	тренер character varying (60)	номер_тренера integer	спортсмен character varying (60)	номер_спортсмена integer	соревнование character varying (60)	место integer
1	Артём Поклонский	5	Мария Бондаренко	10	Галактический Турнир имени Пол...	12
2	Артём Поклонский	5	Мария Бондаренко	10	Школьный Командный турнир им...	1
3	Артём Поклонский	5	Мария Оникиенко	15	Галактический Турнир-Пятиборье ...	52
4	Артём Поклонский	5	Мария Оникиенко	15	[null]	12
5	Артём Поклонский	5	Мария Оникиенко	15	[null]	97
6	Артём Поклонский	5	Мария Оникиенко	19	Муниципальный Турнир имени Дз...	17
7	Артём Поклонский	5	Мария Хёнгю	17	Школьный Турнир-Семиборье име...	66
8	Артём Поклонский	5	Саян Бондаренко	12	Региональный Турнир-Семиборье ...	56
9	Артём Поклонский	5	Саян Бондаренко	12	Региональный Турнир имени Циол...	27
10	Артём Поклонский	5	Саян Бондаренко	12	Школьный Командный турнир им...	9
11	Артём Поклонский	5	Саян Бондаренко	16	Региональный Турнир-Семиборье ...	59

2. Найти самую распространенную травму:

```

1 SELECT q.trauma_type AS "Тип травмы",
2       q.severity AS "Суровость",
3       q.cnt AS "Количество"
4 FROM ( SELECT DISTINCT "Trauma".trauma_type,
5       "Trauma".severity,
6       count("GettingTrauma".trauma_code) AS cnt
7 FROM "Trauma"
8       LEFT JOIN "GettingTrauma" ON "GettingTrauma".trauma_code = "Trauma".id_trauma
9 GROUP BY "Trauma".trauma_type, "Trauma".severity
10 ORDER BY (count("GettingTrauma".trauma_code)) DESC) q
11 WHERE q.cnt = (( SELECT max(f.cnt) AS max
12 FROM ( SELECT DISTINCT "Trauma".trauma_type,
13       "Trauma".severity,
14       count("GettingTrauma".trauma_code) AS cnt
15 FROM "Trauma"
16       LEFT JOIN "GettingTrauma" ON "GettingTrauma".trauma_code = "Trauma".id_trauma
17 GROUP BY "Trauma".trauma_type, "Trauma".severity) f));

```

1	SELECT * FROM FrequentTrauma		
	Тип травмы character varying (60)	Суровость character varying (30)	Количество bigint
1	Перелом	Незначительная	7
2	Перелом	Тяжелая	7

Задание 4

Выполнение INSERT:

```

1 --Создадим новую группу для тренера--
2 INSERT INTO "GroupTrainer" (group_id, trainer_id)
3 VALUES (420, (SELECT id_trainer FROM "Trainer" WHERE id_trainer = 15))

```

	group_id [PK] integer	trainer_id integer
1	7	28
2	8	5
3	9	3
4	10	17
5	11	7
6	12	13

	group_id [PK] integer	trainer_id integer
1	7	28
2	8	5
3	9	3
4	10	17
5	11	7
6	12	13
7	420	15

Рисунок 2 – до/после INSERT

Выполнение UPDATE:

```

1  --Сменим тренера для группы Аяза Поклонского--
2  UPDATE "GroupTrainer" SET trainer_id = 15
3  WHERE "GroupTrainer".group_id = (SELECT trainer_id FROM "Trainer" WHERE name_surname = "Аяз Поклонский")

```

	group_id [PK] integer	trainer_id integer
1	7	28

	group_id [PK] integer	trainer_id integer
1	7	15

Рисунок 3 – до/после выполнения UPDATE

Выполнение DELETE:

```

1  --Удалим старую группу нового тренера--
2  DELETE FROM "GroupTrainer"
3  WHERE "trainer_id" = (SELECT id_trainer FROM "Trainer" WHERE "Trainer".name_surname = 'Василий Любимов')
4  and "group_id" = 420
5

```

	group_id [PK] integer	trainer_id integer
1	7	15
2	8	5
3	9	3
4	10	17
5	11	7
6	12	13
7	420	15

	group_id [PK] integer	trainer_id integer
1	7	15
2	8	5
3	9	3
4	10	17
5	11	7
6	12	13

Рисунок 4 – до/после выполнения DELETE

Задание 4


```
1 EXPLAIN ANALYZE SELECT id_sportsman, sportsman_points
2 FROM "Participation"
3 WHERE sportsman_points > 80
```

Data Output Messages Notifications

QUERY PLAN text

1	Seq Scan on "Participation" (cost=0.00..1.63 rows=6 width=8) (actual time=0.052..0.066 rows=7 loops...
2	Filter: (sportsman_points > 80)
3	Rows Removed by Filter: 43
4	Planning time: 0.713 ms
5	Execution time: 0.472 ms

Рисунок 5 – выполнение запроса до создания индексов

```
1 create index col_a on "Participation"(id_sportsman);
2 create index col_b on "Participation"(sportsman_points);
```

Рисунок 6 - создание индексов

```

1
2 EXPLAIN ANALYZE SELECT id_sportsman, sportsman_points
3 FROM "Participation"
4 WHERE sportsman_points > 80

```

Data Output Messages Notifications

QUERY PLAN text

1	Seq Scan on "Participation" (cost=0.00..1.63 rows=6 width=8) (actual time=0.023..0.033 rows=7 loops...
2	Filter: (sportsman_points > 80)
3	Rows Removed by Filter: 43
4	Planning time: 0.941 ms
5	Execution time: 0.063 ms

Рисунок 7 – выполнение запроса после создания индексов

```

1 CREATE INDEX bogus ON "Sportsman"(id_sportsman, birthdate);

```

Рисунок 8 - создание составного индекса

QUERY PLAN		
	text	
1	Seq Scan on "Sportsman" (cost=0.00..1.75 rows=6 width=8) (actual time=0.024..0.036 rows=22 loops...	
2	Filter: ((birthdate > '2000-05-18'::date) AND (id_sportsman > 2))	
3	Rows Removed by Filter: 28	
4	Planning time: 0.406 ms	
5	Execution time: 0.072 ms	

Рисунок 9 - до создания индексов


	QUERY PLAN	
	text	
1	Seq Scan on "Sportsman" (cost=0.00..1.75 rows=6 width=8) (actual time=0.029..0.042 rows=22 loops...	
2	Filter: ((birthdate > '2000-05-18'::date) AND (id_sportsman > 2))	
3	Rows Removed by Filter: 28	
4	Planning time: 0.190 ms	
5	Execution time: 0.070 ms	

Рисунок 10 - после создания индексов

Запросы выполняются значительно быстрее после создания индексов. Целесообразно создавать индексы при фильтрации и сортировке. С другой стороны, редактирование данных таблицы может усилить нагрузку на систему потому, что индексированные данные тоже необходимо изменить.

Выводы:

Глубже ознакомился с программой Pgadmin. Научился делать представления и запросы на выборку данных в среде PostgreSQL, использовать индексы. На практике эти навыки позволяют провести качественную оптимизацию работы БД, что прекрасно.