

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

Отчет

по Лабораторной Работе №4

по дисциплине «**БазыДанных**»

Вариант 16, Спортивный клуб

Автор: Корчагин Вадим Сергеевич

Факультет: ФИКТ

Группа: K3241

Преподаватель: Говорова Марина Михайловна

ИТМО

Санкт-Петербург

2023

1. Цель работы:

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

2. Практическое задание:

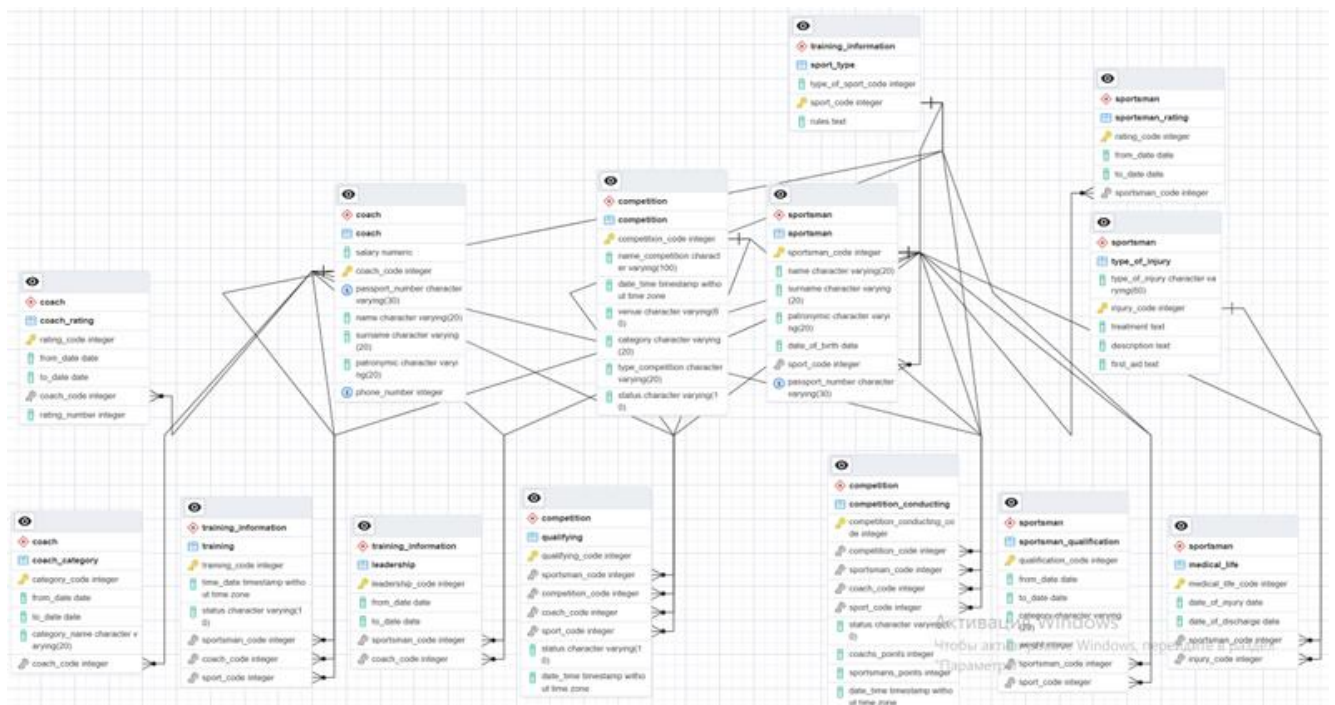
1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).

2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.

3. Изучить графическое представление запросов и просмотреть 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.

3. Схема базы данных:



4. Выполнение:

Создать запросы:

- 1) С каким количеством спортсменов работает каждый тренер:

Query

Query History

```
1 SELECT c.coach_code, c.name, c.surname, COUNT(*) AS total_sportsmen
2 FROM coach.coach c
3 JOIN training_information.training t ON c.coach_code = t.coach_code
4 JOIN sportsman.sportsman s ON t.sportsman_code = s.sportsman_code
5 GROUP BY c.coach_code, c.name, c.surname;
```

Data Output

Messages

Notifications

	coach_code [PK] integer	name character varying (20)	surname character varying (20)	total_sportsmen bigint
1	86978	Christina	Gavin	4
2	70473	Stephen	Raymond	15
3	76196	Paul	Melissa	3
4	87217	Gregory	Andrew	1

- 2) Найти тренеров, чьи спортсмены не имеют травм.

Sport_clubs/postgres@PostgreSQL 16

Query Query History

```

1 SELECT c.coach_code, c.name, c.surname
2 FROM coach.coach c
3 WHERE NOT EXISTS (
4     SELECT 1
5     FROM training_information.training tt
6     LEFT JOIN sportsman.medical_life m ON tt.sportsman_code = m.sportsman_code
7     WHERE tt.coach_code = c.coach_code AND m.sportsman_code IS NOT NULL
8 );
9

```

Data Output Messages Notifications

	coach_code [PK] integer	name character varying (20)	surname character varying (20)
1	87217	Gregory	Andrew

3) Найти тренера, получающего минимальную зарплату.

Query Query History

```

1 SELECT c.salary, c.coach_code, c.name, c.surname
2 FROM coach.coach c
3 ORDER BY salary
4 LIMIT 1;
5

```

Data Output Messages Notifications

	salary numeric	coach_code [PK] integer	name character varying (20)	surname character varying (20)
1	916	87217	Gregory	Andrew

4) Определить количество соревнований каждой категории.

Query Query History

```

1 SELECT category, COUNT(*) AS competition_count
2 FROM competition.competition
3 GROUP BY category;

```

Data Output Messages Notifications

	category character varying (20)	competition_count bigint
1	SCVM	5
2	AprilJoke	7
3	EXE	2
4	Yeah	9

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

Query

Query History

1

2

3

4

5

6

7

8

```
SELECT c.coach_code, c.name, c.surname, AVG(EXTRACT(YEAR FROM AGE(CURRENT_DATE, s.date_of_birth))) AS avg_age
FROM coach.coach c
JOIN training_information.training t ON c.coach_code = t.coach_code
JOIN sportsman.sportsman s ON t.sportsman_code = s.sportsman_code
GROUP BY c.coach_code, c.name, c.surname
ORDER BY avg_age ASC
LIMIT 1;
```

Data Output

Messages

Explain ×

Notifications

	coach_code [PK] integer	name character varying (20)	surname character varying (20)	avg_age numeric
1	86978	Christina	Gavin	24.2500000000000000

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

Query Query History

```
1 SELECT
2     c.competition_code,
3     c.category,
4     COUNT(cc.sportsman_code) AS num_participants
5 FROM
6     competition.competition_conducting cc
7 JOIN
8     competition.competition c ON cc.competition_code = c.competition_code
9 WHERE
10    c.date_time BETWEEN '2024-11-01 00:00:01' AND '2024-11-9 00:00:01'
11 GROUP BY
12     c.competition_code,
13     c.category;
14
```

Data Output Messages Notifications

	competition_code [PK] integer	category character varying (20)	num_participants bigint
1	273	Yeah	3
2	12273	EXE	1
3	19818	AprilJoke	2

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

Query Query History

```
1 SELECT
2     s.sportsman_code,
3     s.name,
4     s.surname,
5     COUNT(cc.competition_code) AS competitions_participated
6 FROM
7     sportsman.sportsman s
8 JOIN
9     competition.competition_conducting cc ON s.sportsman_code = cc.sportsman_code
10 GROUP BY
11     s.sportsman_code, s.name, s.surname
12 ORDER BY
13     s.sportsman_code;
14
```

Data Output Messages Notifications

	sportsman_code [PK] integer	name character varying (20)	surname character varying (20)	competitions_participated bigint
1	4286	Steve	James	1
2	5411	Renee	Amy	1
3	21045	Kendra	Cassandra	1
4	28770	Kyle	Mark	1
5	29165	Sarah	Robert	1
6	39395	Dakota	Angel	1
7	66895	David	Kimberly	1
8	75890	Gloria	Renee	1
9	79500	Denise	Michele	1
10	80315	James	Emma	1

Создать представление:

- 1) Содержащее сведения обо всех тренерах, соревнованиях, в которых участвовали их спортсмены и местах, которые они заняли.

Query

Query History

```
1 CREATE VIEW coach_competition_places AS
2 SELECT c.coach_code,
3        co.competition_code,
4        sp.sportsman_code,
5        cc.prize_place
6 FROM coach.coach c
7 JOIN competition.competition_conducting cc ON c.coach_code = cc.coach_code
8 JOIN sportsman.sportsman sp ON sp.sportsman_code = cc.sportsman_code
9 JOIN competition.competition co ON co.competition_code = cc.competition_code;
```

Data Output

Messages

Notifications

CREATE VIEW

Query returned successfully in 87 msec.

Query

Query History

```
1 SELECT * FROM public.coach_competition_places
```

Data Output

Messages

Notifications

	coach_code integer	competition_code integer	sportsman_code integer	prize_place numeric
1	70473	12273	4286	1
2	70473	42472	5411	2
3	86978	42472	21045	2
4	86978	273	28770	3
5	70473	19818	29165	2
6	70473	273	39395	3
7	76196	19818	66895	3
8	86978	42472	75890	1
9	70473	273	79500	1
10	76196	42472	80315	3

2) Найти самую распространенную травму.

Query

Query History

1

```
CREATE VIEW most_common_injury AS
SELECT
  type_of_injury,
  COUNT(*) AS injury_count
FROM sportsman.medical_life
JOIN sportsman.type_of_injury ON sportsman.medical_life.injury_code = sportsman.type_of_injury.injury_code
GROUP BY type_of_injury
ORDER BY injury_count DESC
LIMIT 1;
```

Data Output

Messages

Notifications

CREATE VIEW

Query returned successfully in 35 msec.

1

```
SELECT * FROM public.most_common_injury
```

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	type_of_injury character varying (60) 🔒	injury_count bigint 🔒
1	Bruise	7

Запросы на модификацию данных:

1) Insert (Добавим нового тренера в таблицу):

```
Query Query History
1 INSERT INTO coach.coach (salary, coach_code, passport_number, name, surname, patronymic, phone_number)
2 VALUES (50000, (SELECT COALESCE(MAX(coach_code), 0) + 1 FROM coach.coach), 'AB123456', 'Иван', 'Иванов', 'Иванович', 1234567890);
3
```

- До запроса:

	salary numeric	coach_code [PK] integer	passport_number character varying (30)	name character varying (20)	surname character varying (20)	patronymic character varying (20)	phone_number integer
1	33292	70473	9614643	Stephen	Raymond	Ashley	43919
2	89399	76196	1885811	Paul	Melissa	Tracey	85935
3	27212	86978	2590531	Christina	Gavin	Katelyn	34483
4	916	87217	5384211	Gregory	Andrew	Katherine	87122

- После запроса:

	salary numeric	coach_code [PK] integer	passport_number character varying (30)	name character varying (20)	surname character varying (20)	patronymic character varying (20)	phone_number integer
1	33292	70473	9614643	Stephen	Raymond	Ashley	43919
2	89399	76196	1885811	Paul	Melissa	Tracey	85935
3	27212	86978	2590531	Christina	Gavin	Katelyn	34483
4	916	87217	5384211	Gregory	Andrew	Katherine	87122
5	50000	87218	AB123456	Иван	Иванов	Иванович	1234567890

2) Update (Изменим зарплату тренера, который обладает самым высоким рейтингом)

```
Query Query History
1 UPDATE coach.coach
2 SET salary = salary * 1.1
3 WHERE coach_code = (SELECT coach_code FROM coach.coach_rating ORDER BY rating_number DESC LIMIT 1);
4
```

- До запроса:

Data Output Messages Notifications							
	salary numeric	coach_code [PK] integer	passport_number character varying (30)	name character varying (20)	surname character varying (20)	patronymic character varying (20)	phone_number integer
1	33292	70473	9614643	Stephen	Raymond	Ashley	43919
2	89399	76196	1885811	Paul	Melissa	Tracey	85935
3	27212	86978	2590531	Christina	Gavin	Katelyn	34483
4	916	87217	5384211	Gregory	Andrew	Katherine	87122

Data Output Messages Notifications					
	rating_code [PK] integer	from_date date	to_date date	coach_code integer	rating_number integer
1	4396	2029-04-30	2029-05-01	86978	23115
2	32694	2029-04-29	2029-05-01	76196	18009
3	70738	2029-02-06	2029-02-25	87217	63453
4	84980	2029-04-30	2029-05-03	70473	76558

- После запроса:

Data Output Messages Notifications							
	salary numeric	coach_code [PK] integer	passport_number character varying (30)	name character varying (20)	surname character varying (20)	patronymic character varying (20)	phone_number integer
1	36621.2	70473	9614643	Stephen	Raymond	Ashley	43919
2	89399	76196	1885811	Paul	Melissa	Tracey	85935
3	27212	86978	2590531	Christina	Gavin	Katelyn	34483
4	916	87217	5384211	Gregory	Andrew	Katherine	87122
5	50000	87218	AB123456	Иван	Иванов	Иванович	1234567890

3) Delete (Удалим из medical_life записи спортсменов, у которых определенный вид травмы):

Query Query History	
1	DELETE FROM sportsman.medical_life
2	WHERE injury_code IN (SELECT injury_code FROM sportsman.type_of_injury WHERE type_of_injury = 'Fracture');
3	

• До запроса:

Data Output Messages Notifications						
	medical_life_code [PK] integer	date_of_injury date	date_of_discharge date	sportsman_code integer	injury_code integer	
1	54617	2029-01-10	2029-02-24	79500	2927	
2	61904	2029-02-02	2029-04-29	63004	2927	
3	70538	2029-04-18	2029-05-03	68731	2927	
4	72134	2029-04-18	2029-05-03	5411	2927	
5	74350	2029-03-11	2029-03-23	75890	2927	
6	85574	2029-02-03	2029-03-28	80315	2927	
7	89362	2029-03-29	2029-04-28	94897	2927	
8	92366	2029-02-05	2029-03-28	21045	5412	
9	92479	2029-04-14	2029-05-02	6609	5412	
10	96428	2029-04-19	2029-04-27	29165	6261	
11	99925	2029-03-02	2029-03-22	91008	6261	

Data Output Messages Notifications			
	type_of_injury character varying (60)	injury_code [PK] integer	treatment text
1	Bruise	2927	Second stock career their. Skin interview almost recognize back drop.
2	Dislocation	5412	Probably way PM. Stay hair center yes determine.
3	Fracture	6261	Tax long research manager health visit available. See former them indeed. Everyone continue line a

• После запроса:

Query

Query History

1

SELECT * FROM sportsman.medical_life

2

ORDER BY medical_life_code ASC

Data Output

Messages

Notifications

	medical_life_code [PK] integer	date_of_injury date	date_of_discharge date	sportsman_code integer	injury_code integer
1	54617	2029-01-10	2029-02-24	79500	2927
2	61904	2029-02-02	2029-04-29	63004	2927
3	70538	2029-04-18	2029-05-03	68731	2927
4	72134	2029-04-18	2029-05-03	5411	2927
5	74350	2029-03-11	2029-03-23	75890	2927
6	85574	2029-02-03	2029-03-28	80315	2927
7	89362	2029-03-29	2029-04-28	94897	2927
8	92366	2029-02-05	2029-03-28	21045	5412
9	92479	2029-04-14	2029-05-02	6609	5412

Создание индексов:

1. Первый индекс

Query	Query History
1 2	<pre>CREATE INDEX idx_salary ON coach.coach (salary);</pre>
Data Output	Messages
CREATE INDEX	Query returned successfully in 50 msec.


Без индекса:

Query Query History

```
1 EXPLAIN ANALYZE SELECT * FROM coach.coach WHERE salary > 50000;  
2
```

Data Output Messages Notifications



	QUERY PLAN text 
1	Seq Scan on coach (cost=0.00..1.29 rows=8 width=292) (actual time=0.010..0.011 rows=1 loops...
2	Filter: (salary > '50000'::numeric)
3	Rows Removed by Filter: 4
4	Planning Time: 1.534 ms
5	Execution Time: 0.020 ms

С индексом:

Query Query History

1

EXPLAIN ANALYZE SELECT * FROM coach.coach WHERE salary > 50000;

2

Data Output Messages Notifications

+

📄

▼

📋

▼

🗑️

📦

⬇️

📈

QUERY PLAN

text

🔒

1	Seq Scan on coach (cost=0.00..1.06 rows=2 width=292) (actual time=0.009..0.010 rows=1 loops...
2	Filter: (salary > '50000'::numeric)
3	Rows Removed by Filter: 4
4	Planning Time: 0.957 ms
5	Execution Time: 0.018 ms

2. Второй индекс:

Query Query History

1

CREATE INDEX idx_competition_venue_datetime ON competition.competition(venue, date_time);

2

Data Output Messages Notifications

CREATE INDEX

Query returned successfully in 34 msec.

Без индекса:

Query Query History

1

EXPLAIN ANALYZE

2

SELECT * FROM competition.competition WHERE venue = 'Cool_stadium';

Data Output Messages Notifications

+

📄

▼

📋

▼

🗑️

📦

⬇️

📈

QUERY PLAN

text

🔒

1	Seq Scan on competition (cost=0.00..1.29 rows=1 width=522) (actual time=0.019..0.019 rows=3 loops...
2	Filter: ((venue)::text = 'Cool_stadium'::text)
3	Rows Removed by Filter: 20
4	Planning Time: 0.839 ms
5	Execution Time: 0.033 ms

С индексом:

Query Query History

```
1 EXPLAIN ANALYZE
2 SELECT * FROM competition.competition WHERE venue = 'Cool_stadium';
```

Data Output Messages Notifications



QUERY PLAN		text	
1	Seq Scan on competition	(cost=0.00..1.29 rows=1 width=522) (actual time=0.013..0.013 rows=3 loops=...	
2	Filter:	((venue)::text = 'Cool_stadium'::text)	
3	Rows Removed by Filter:	20	
4	Planning Time:	0.057 ms	
5	Execution Time:	0.022 ms	

Вывод:

В ходе данной лабораторной работы были разработаны разнообразные вариации запросов к базе данных. Эти запросы включали в себя операции группировки, фильтрации, агрегации, сортировки и присоединения. В работе с подзапросами удалось осуществить их применение как в запросах на чтение, так и в запросах на запись, удаление и обновление данных. Кроме того, было проведено сравнение времени выполнения запросов до и после создания индексов. Полученные результаты демонстрировали ускорение операций чтения при использовании индексов.