

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 №5

тема: Процедуры, функции и триггеры в PostgreSQL

дисциплина: «ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗ ДАННЫХ»

Выполнила:

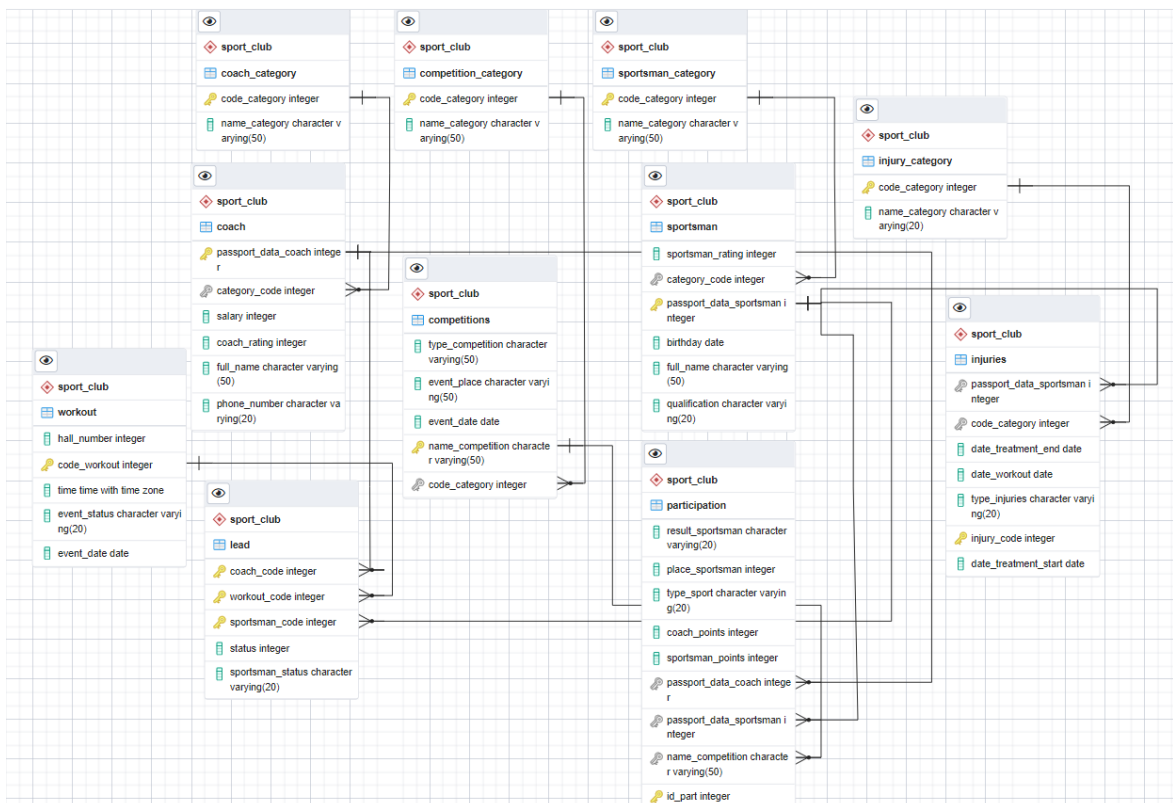
студентка II курса ИКТ группы К3241

Лорс Хава Абуевна

Проверила:

Говорова Марина Михайловна

Санкт-Петербург
2022



Выполнение:

1. Создать процедуры/функции согласно индивидуальному заданию и (согласно индивидуальному заданию, часть 4):

Для вывода данных о результатах заданного спортсмена за прошедший год:

```
CREATE FUNCTION result_s(ids int)
RETURNS table(full_name VARCHAR(50), place_sportsman INT,
result_sportsman VARCHAR(20), sportsman_points INT)
AS
$$
BEGIN
RETURN QUERY
SELECT sport_club.sportsman.full_name,
sport_club.participation.place_sportsman,
sport_club.participation.result_sportsman,
sport_club.participation.place_sportsman
FROM sport_club.sportsman, sport_club.participation,
sport_club.competitions
WHERE sport_club.sportsman.passport_data_sportsman=ids
AND
sport_club.sportsman.passport_data_sportsman=sport_club.participation.passport_data_sportsman
AND
sport_club.competitions.name_competition=sport_club.participation.name_competition
AND sport_club.competitions.event_date BETWEEN
(NOW()-INTERVAL '1YEAR')
AND (SELECT CURRENT_DATE);
END;
$$ LANGUAGE plpgsql;

select * from result_s(1285277485)
```

sport_club/postgres@PostgreSQL 12 ▾

Query Editor История запросов

```

1 CREATE FUNCTION result_s(ids int) RETURNS table(full_name VARCHAR(50), place_sportsman INT, result_sportsman VARCHAR(20), sportsman_points INT)
2 AS
3 $$
4 BEGIN
5 RETURN QUERY
6 SELECT sport_club.sportsman.full_name, sport_club.participation.place_sportsman,
7 sport_club.participation.result_sportsman,
8 sport_club.participation.place_sportsman
9 FROM sport_club.sportsman, sport_club.participation, sport_club.competitions
10 WHERE sport_club.sportsman.passport_data_sportsman=ids
11 AND sport_club.sportsman.passport_data_sportsman=sport_club.participation.passport_data_sportsman
12 AND sport_club.competitions.name_competition=sport_club.participation.name_competition
13 AND sport_club.competitions.event_date BETWEEN (NOW()-INTERVAL '1YEAR')
14 AND (SELECT CURRENT_DATE);
15 END;
16 $$ LANGUAGE plpgsql;
17
18 select * from result_s(1285277485)

```

Результат План выполнения Сообщения Notifications

	full_name character varying	place_sportsman integer	result_sportsman character varying	sportsman_points integer
1	ДОБРЮЛЮБОВА АННА	1	ВЫСОКИЙ	1

Для ввода данных о соревнованиях, проводимых в первом квартале текущего года:

```

CREATE FUNCTION data_()
RETURNS table(name_competition VARCHAR(50), event_place
VARCHAR(50), event_date DATE, type_competition VARCHAR(50),
code_category INT)
AS
$$
BEGIN
RETURN QUERY
SELECT * FROM sport_club.competitions
WHERE (EXTRACT(month FROM
sport_club.competitions.event_date) BETWEEN 1 AND 3)
AND (EXTRACT(year FROM sport_club.competitions.event_date))=
EXTRACT(year FROM current_date);
END;
$$ LANGUAGE plpgsql;

select * from data_();

```

[Query Editor](#)
[История запросов](#)

```

1 CREATE FUNCTION data_() RETURNS table(name_competition VARCHAR(50), event_place VARCHAR(50),
2                                     event_date DATE, type_competition VARCHAR(50), code_category INT)
3 AS
4 $$
5 BEGIN
6 RETURN QUERY
7 SELECT * FROM sport_club.competitions
8 WHERE (EXTRACT(month FROM sport_club.competitions.event_date) BETWEEN 1 AND 3)
9 AND (EXTRACT(year FROM sport_club.competitions.event_date))=
10 EXTRACT(year FROM current_date);
11 END;
12 $$ LANGUAGE plpgsql;
13
14 select * from data_();
15
16
  
```

[Результат](#)
[План выполнения](#)
[Сообщения](#)
[Notifications](#)

	name_competition character varying	event_place character varying	event_date date	type_competition character varying	code_category integer
1	ОДИНОЧНОЕ	СК ГОРЬКОВСКАЯ	2022-03-24	ФИГУРНОЕ КАТАНИЕ ПО СТРАНАМ	3
2	КОМАНДНОЕ	СК ВЯЗЕМСКИЙ	2022-03-25	МЕЖДУНАРОДНЫЕ СОРЕВНОВАНИЯ ПО ВОЛЕЙБОЛУ	2

Для повышения рейтинга и оклада тренера после участия в соревновании:

до:

	passport_data_coach [PK] integer	category_code integer	salary integer	coach_rating integer	full_name character varying (50)	phone_number character varying (20)
1	1101574677	3	82000	1	ЕРМОЛОВА ВАСИЛИСА	936581855
2	1510082405	1	82000	1	ПОПОВ ДМИТРИЙ	936789595
3	1610082400	4	66000	2	ИВАНОВ ИВАН	936787410
4	1610082401	5	86000	4	ИВАНОВА НАТАЛЬЯ	936787411
5	1610082402	6	86000	3	ВОЛОДИН АЛЕКСЕЙ	936787412
6	1610082404	8	90000	8	ПЕТРОВ ДМИТРИЙ	936787414
7	1610082405	2	82000	1	ДУРОВ АНТОН	936787412
8	1610082408	3	81000	1	ПЛЮЩЕНКО АЛЕКСЕЙ	936787417
9	1610082409	8	90000	2	ВАСИЛЕНКО АНТОН	936787413
10	1610082410	5	66000	2	КОЗЛОВ ИВАН	936787414

```

CREATE OR REPLACE FUNCTION prot_comp()
RETURNS table(passport_data_coach INT, full_name
VARCHAR(50), coach_rating INT, phone_number VARCHAR(20),
salary INT, code_category INT)
AS
$$
BEGIN
RETURN QUERY
UPDATE sport_club.coach
  
```

```

SET coach_rating = coach_rating - 1, salary = salary + 1000
WHERE coach.coach_rating > 1
AND (SELECT sport_club.coach.passport_data_coach
FROM sport_club.coach, sport_club.participation
WHERE sport_club.coach.passport_data_coach =
sport_club.participation.passport_data_coach);
END;

$$ LANGUAGE plpgsql;

select * from sport_club.coach;

```

ПОСЛЕ:

Query Editor

Query History

```
1 CREATE OR REPLACE FUNCTION prot_comp()
2 RETURNS table(passport_data_coach INT, full_name VARCHAR(50),
3 coach_rating INT, phone_number VARCHAR(20), salary INT, code_category INT)
4 AS
5 $$
6 BEGIN
7 RETURN QUERY
8 UPDATE sport_club.coach
9 SET coach_rating = coach_rating - 1, salary = salary + 1000
10 WHERE coach.coach_rating > 1 AND
11 (SELECT sport_club.coach.passport_data_coach FROM sport_club.coach, sport_club.participation
12 WHERE sport_club.coach.passport_data_coach=sport_club.participation.passport_data_coach);
13 END;
14
15 $$ LANGUAGE plpgsql;
16
17 select * from sport_club.coach;
```

Data Output

Explain

Messages

Notifications

	passport_data_coach [PK] integer	category_code integer	salary integer	coach_rating integer	full_name character varying (50)	phone_number character varying (20)
1	1610082400	4	66000	2	ИВАНОВ ИВАН	936787410
2	1610082401	5	86000	4	ИВАНОВА НАТАЛЬЯ	936787411
3	1610082402	6	86000	3	ВОЛОДИН АЛЕКСЕЙ	936787412
4	1610082404	8	90000	8	ПЕТРОВ ДМИТРИЙ	936787414
5	1610082408	3	81000	1	ПЛЮЩЕНКО АЛЕКСЕЙ	936787417
6	1610082409	8	90000	2	ВАСИЛЕНКО АНТОН	936787413
7	1610082410	5	66000	2	КОЗЛОВ ИВАН	936787414
8	1510082405	1	82000	1	ПОПОВ ДМИТРИЙ	936789595
9	1610082405	2	82000	1	ДУРОВ АНТОН	936787412
10	1101574677	3	82000	1	ЕРМОЛОВА ВАСИЛИСА	936581855

В базе данных не нашлось тренеров, участвовавших в соревнования с рейтингом больше единицы. Все три тренера оказались с рейтингом = 1, поэтому после вызова данной функции рейтинг и оклады не изменились.

2. Создать триггер для логирования событий вставки, удаления, редактирования данных в базе данных PostgreSQL (согласно индивидуальному заданию, часть 5).

Допустимо создать универсальный триггер или отдельные триггеры на логирование действий.

1) Создана таблица logs;

2) Триггерная функция:

```
CREATE OR REPLACE FUNCTION add_to_log()  
RETURNS TRIGGER AS $$  
DECLARE  
mstr varchar(30);  
astr varchar(100);  
retstr varchar(254);  
BEGIN  
IF TG_op = 'INSERT' THEN  
astr = NEW;  
mstr := 'Add new';  
retstr := mstr || astr;  
INSERT INTO  
sport_club.logs(text, added, table_name) values  
(retstr, NOW(), TG_TABLE_NAME);  
RETURN NEW;  
ELSEIF TG_OP = 'UPDATE' THEN  
astr = NEW;  
mstr := 'Update';  
retstr := mstr || astr;  
INSERT INTO  
sport_club.logs(text, added, table_name) values  
(retstr, NOW(), TG_TABLE_NAME);  
RETURN NEW;  
ELSEIF TG_OP = 'DELETE' THEN  
astr = OLD;  
mstr := 'Remove';  
retstr := mstr || astr;  
INSERT INTO  
sport_club.logs(text, added, table_name) values  
(retstr, NOW(), TG_TABLE_NAME);  
RETURN OLD;  
END IF;  
END;  
$$ LANGUAGE plpgsql;
```

Query Editor	Query History
<pre> 19 retstr := mstr astr; 20 INSERT INTO 21 sport_club.logs(text, added, table_name) values 22 (retstr, NOW(), TG_TABLE_NAME); 23 RETURN NEW; 24 ELSEIF TG_OP = 'DELETE' THEN 25 astr = OLD; 26 mstr := 'Remove'; 27 retstr := mstr astr; 28 INSERT INTO 29 sport_club.logs(text, added, table_name) values 30 (retstr, NOW(), TG_TABLE_NAME); 31 RETURN OLD; 32 END IF; 33 END; 34 \$\$ LANGUAGE plpgsql; 35 </pre>	
Data Output	Explain
	Messages
	Notifications
<p>CREATE FUNCTION</p> <p>Query returned successfully in 43 msec.</p>	

Рис. 1 - Создание функции логирования `add_to_log`

```

CREATE TRIGGER inj_tr AFTER INSERT OR UPDATE OR DELETE
ON sport_club.injuries
FOR EACH ROW EXECUTE PROCEDURE add_to_log();

```

sport_club/postgres@PostgreSQL 12	
Query Editor	Query History
<pre> 1 CREATE TRIGGER inj_tr AFTER INSERT OR UPDATE OR DELETE 2 ON sport_club.injuries FOR EACH ROW EXECUTE PROCEDURE add_to_log(); </pre>	
Data Output	Explain
	Messages
	Notifications
<p>CREATE TRIGGER</p> <p>Query returned successfully in 41 msec.</p>	

Рис. 2 - Создание триггера

```

INSERT INTO sport_club.injuries(passport_data_sportsman,
code_category date_treatment_end, date_workout,
type_injuries, injury_code, date_treatment_start)

VALUES (1285277485, 3, '2014-04-22', '2014-04-05', 'ЛЕГКИЙ',
105, '2014-04-05');

UPDATE sport_club.injuries

SET code_category=1

WHERE injury_code=104;

```



```
DELETE FROM sport_club.injuries WHERE code_category=3;
```

```
select * from sport_club.logs;
```

Query Editor

Query History

```
1 INSERT INTO sport_club.injuries(passport_data_sportsman, code_category, date_treatment_end,
2                                     date_workout, type_injuries, injury_code, date_treatment_start)
3 VALUES (1285277485, 3, '2014-04-22', '2014-04-05', 'ЛЕГКИЙ', 105, '2014-04-05');
4 UPDATE sport_club.injuries
5 SET code_category=1
6 WHERE injury_code=104;
7 DELETE FROM sport_club.injuries WHERE code_category=3;
8
9 select * from sport_club.logs;
```

Data Output

Explain

Messages

Notifications





	 text text	 added time with time zone	 table_name text	
1	Add new(1285277485,2,2014-04-22,2014-04-05,ЛЕГКИЙ,104,2014-04-05)	18:20:10.331643+03:00	injuries	
2	Update(1501256372,3,2019-09-09,2019-08-20,ЛЕГКИЙ,103,2019-08-20)	18:20:10.331643+03:00	injuries	
3	Add new(1285277485,3,2014-04-22,2014-04-05,ЛЕГКИЙ,105,2014-04-05)	18:28:47.499229+03:00	injuries	
4	Update(1285277485,1,2014-04-22,2014-04-05,ЛЕГКИЙ,104,2014-04-05)	18:28:47.499229+03:00	injuries	
5	Remove(1501256372,3,2019-09-09,2019-08-20,ЛЕГКИЙ,103,2019-08-20)	18:28:47.499229+03:00	injuries	
6	Remove(1285277485,3,2014-04-22,2014-04-05,ЛЕГКИЙ,105,2014-04-05)	18:28:47.499229+03:00	injuries	

Рис. 3 - Проверка

```
DROP TRIGGER inj_tr ON sport_club.injuries;
```

Query Editor	Query History
1	DROP TRIGGER inj_tr ON sport_club.injuries;

Data Output	Explain	Messages	Notifications
DROP TRIGGER			
Query returned successfully in 39 msec.			

Рис. 4 - Удаление триггера

Вывод:

В данной работе были изучены функции и процедуры, а также созданы триггеры для логирования событий вставки, удаления и редактирования данных.