

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего  
образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**ОТЧЕТ**

по лабораторной работе №3

«Работа с триггерами и функциями в POSTGRESQL»

по дисциплине «Проектирование и реализация баз данных»

Автор:

Петросян Анна Мнацакановна

Факультет:

Инфокоммуникационных технологий

Группа:

К3242

Преподаватель: Говорова Марина Михайловна



**УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург

2022

**Цель работы:** овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL, SQL Shell (psql).

**Практическое задание:**

### Вариант 1

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

[Функция - результаты заданного спортсмена](#)

[Функция - соревнования](#)

[Функция - категория](#)

[Функция - зарплата](#)

[Триггер](#)

Выполнение:

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

- Для вывода данных о результатах заданного спортсмена за прошедший год (поменяла на 4, потому что пустой запрос).

### Функция - результаты заданного спортсмена

Создание типа данных:

```
CREATE TYPE t_resul AS (fullname VARCHAR(100), place INT, res INT, status VARCHAR(50));
```

```
postgres=# \c sport
Вы подключены к базе данных "sport" как пользователь "postgres".
sport=# CREATE TYPE t_resul AS (fullname VARCHAR(100), place INT, res INT, status VARCHAR(50));
CREATE TYPE
```

Создание функции:

```
CREATE OR REPLACE FUNCTION resul(ida integer) RETURNS SETOF t_resul AS
```

\$\$

BEGIN

RETURN QUERY

SELECT sportclub.athlete.fullname, sportclub.part.place, sportclub.part.res,  
sportclub.part.status

FROM sportclub.athlete, sportclub.part, sportclub.competition

WHERE sportclub.athlete.idath=ida AND sportclub.athlete.idath=sportclub.part.idath

AND sportclub.competition.name=sportclub.part.namecompetition

AND sportclub.competition.date BETWEEN (NOW()-INTERVAL '4 YEAR') AND  
(SELECT CURRENT\_DATE);

END;

\$\$ LANGUAGE plpgsql;

```
sport=# CREATE OR REPLACE FUNCTION resul(ida integer) RETURNS SETOF t_resul AS
sport=# $$
sport=# BEGIN
sport=# RETURN QUERY
sport=# SELECT sportclub.athlete.fullname, sportclub.part.place, sportclub.part.res, sportclub.part.status
sport=# FROM sportclub.athlete, sportclub.part, sportclub.competition
sport=# WHERE sportclub.athlete.idath=ida AND sportclub.athlete.idath=sportclub.part.idath
sport=# AND sportclub.competition.name=sportclub.part.namecompetition
sport=# AND sportclub.competition.date BETWEEN (NOW()-INTERVAL '4 YEAR') AND (SELECT CURRENT_DATE);
sport=# END;
sport=# $$ LANGUAGE plpgsql;
CREATE FUNCTION
```

Вызов:

SELECT \* FROM resul(347445);

```
sport=# SELECT * FROM resul(347445);
      fullname      | place | res | status
-----+-----+-----+-----
 Клубничкина Хюррем Ивановна |    4 |   5 | Участник
 Клубничкина Хюррем Ивановна |    7 |  12 | Участник
(2 строки)
```

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

## Функция - соревнования

Создание типа данных:

CREATE TYPE **t\_competition** AS (competition\_name VARCHAR(100),  
competition\_date date, place VARCHAR(20),  
cat VARCHAR(20), sort VARCHAR(20));

```
sport=# CREATE TYPE t_competition AS (competition_name VARCHAR(100),
sport=#   competition_date date, place VARCHAR(20),
sport=#   cat VARCHAR(20), sort VARCHAR(20));
CREATE TYPE
```

#### Создание функции:

```
CREATE OR REPLACE FUNCTION competition() RETURNS SETOF t_competition
AS
$$
BEGIN
    RETURN QUERY
    SELECT * FROM sportclub.competition WHERE
    (EXTRACT(month FROM sportclub.competition.date) BETWEEN 1 AND 3)
AND
    (EXTRACT(year FROM sportclub.competition.date))= EXTRACT(YEAR
FROM current_date);
END;
$$ LANGUAGE plpgsql;
```

```
sport=# CREATE OR REPLACE FUNCTION competition() RETURNS SETOF t_competition AS
sport=# $$
sport=# BEGIN
sport=# RETURN QUERY
sport=# SELECT * FROM sportclub.competition WHERE
sport=# (EXTRACT(month FROM sportclub.competition.date) BETWEEN 1 AND 3) AND
sport=# (EXTRACT(year FROM sportclub.competition.date))= EXTRACT(YEAR FROM current_date);
sport=# END;
sport=# $$ LANGUAGE plpgsql;
CREATE FUNCTION
```

#### Вызов:

```
SELECT * FROM competition();
```

```
sport=# SELECT * FROM competition();
  competition_name | competition_date | place | cat | sort
-----+-----+-----+-----+-----
Взбитосливский заплыв | 2022-02-13 | Лос-Анджелес, США | Взрослые | Городское
Апельсинкино сопротивление | 2022-03-15 | Мурманск | Юниоры | Региональное
(2 строки)
```

- Для повышения рейтинга (*категория*) и оклада тренера после участия в соревновании

## Функция - категория

#### Категория:

#### Создание типа:

```
CREATE TYPE t_cat AS (idcoach int, fullname VARCHAR(100), level VARCHAR(100),
passport VARCHAR(100), phone VARCHAR(100));
```

```
sport=# CREATE TYPE t_cat AS (idcoach int, fullname VARCHAR(100), level VARCHAR(100), passport VARCHAR(100), phone VARCH
AR(100));
CREATE TYPE
```

#### Создание функции:

```

CREATE OR REPLACE FUNCTION cat() RETURNS SETOF t_cat AS
$$
BEGIN
    RETURN QUERY
    UPDATE sportclub.coach
    SET level=CASE
    WHEN sportclub.coach.idcoach IN
        (SELECT sportclub.coach.idcoach
         FROM sportclub.coach, sportclub.part, sportclub.salary_coach
         WHERE sportclub.coach.idcoach=sportclub.part.idcoach
         AND sportclub.coach.level='Средняя')
    THEN 'Высшая'
    WHEN sportclub.coach.idcoach IN
        (SELECT sportclub.coach.idcoach
         FROM sportclub.coach, sportclub.part, sportclub.salary_coach
         WHERE sportclub.coach.idcoach=sportclub.part.idcoach
         AND sportclub.coach.level='Низшая')
    THEN 'Средняя'
    ELSE sportclub.coach.level
    END;
END;
$$ LANGUAGE plpgsql;

```

```

sport=# CREATE TYPE t_cat AS (idcoach int, fullname VARCHAR(100), level VARCHAR(100), passport VARCHAR(100), phone VARCHAR(100));
CREATE TYPE
sport=# CREATE OR REPLACE FUNCTION cat() RETURNS SETOF t_cat AS
sport=# $$
sport=# BEGIN
sport=# UPDATE sportclub.coach
sport=# SET level=CASE
sport=# WHEN sportclub.coach.idcoach IN
sport=# (SELECT sportclub.coach.idcoach
sport=# FROM sportclub.coach, sportclub.part, sportclub.salary_coach
sport=# WHERE sportclub.coach.idcoach=sportclub.part.idcoach
sport=# AND sportclub.coach.level='Средняя')
sport=# THEN 'Высшая'
sport=# WHEN sportclub.coach.idcoach IN
sport=# (SELECT sportclub.coach.idcoach
sport=# FROM sportclub.coach, sportclub.part, sportclub.salary_coach
sport=# WHERE sportclub.coach.idcoach=sportclub.part.idcoach
sport=# AND sportclub.coach.level='Низшая')
sport=# THEN 'Средняя'
sport=# ELSE sportclub.coach.level
sport=# END;
sport=# END;
sport=# $$ LANGUAGE plpgsql;
CREATE FUNCTION

```

ДО:

```

sport=# SELECT * FROM sportclub.coach ORDER BY fullname ASC;

```

idcoach	fullname	level	passport	phone
197295	Вафелька Эол Николаевич	Средняя	28CV04832	89678896115
137829	Весёлая Миузки Андреевна	Высшая	4010897523	89215674030
100100	Вишня Оливия	Высшая	4015676767	89435678979
890890	Добрый Алексей Викторович	Низшая	403456789	89765672345
567567	Круассанов Олег	Средняя	4028765430	89546785213
924378	Родриго Пересвет Александрович	Высшая	4012389222	89112378654

(6 строк)

после: (Вафелька и Добрый)

idcoach	fullname	level	passport	phone
197295	Вафелька Эол Николаевич	Высшая	28CV04832	89678896115
137829	Весёлая Миузки Андреевна	Высшая	4010897523	89215674030
100100	Вишня Оливия	Высшая	4015676767	89435678979
890890	Добрый Алексей Викторович	Средняя	403456789	89765672345
567567	Круассанов Олег	Средняя	4028765430	89546785213
924378	Родриго Пересвет Александрович	Высшая	4012389222	89112378654

(6 строк)

## Функция - зарплата

### Создание типа данных:

```
CREATE TYPE t_slry AS (idcoach int, slr bigint);
```

```
sport=# CREATE TYPE t_slry AS (idcoach int, slr bigint);
CREATE TYPE
```

### Создание функции:

```
CREATE OR REPLACE FUNCTION sal() RETURNS SETOF t_slry AS
```

```
$$
```

```
BEGIN
```

```
    UPDATE sportclub.salary_coach
```

```
        SET slr=CASE
```

```
        WHEN sportclub.salary_coach.idcoach IN
```

```
            (SELECT sportclub.salary_coach.idcoach
```

```
            FROM sportclub.coach, sportclub.part, sportclub.salary_coach
```

```
            WHERE sportclub.salary_coach.idcoach=sportclub.part.idcoach
```

```
AND
```

```
            sportclub.part.idcoach=sportclub.coach.idcoach
```

```
AND
```

```
    sportclub.salary_coach.idcoach=sportclub.coach.idcoach
```

```
        AND sportclub.coach.level='Высшая')
```

```
    THEN slr+15000
```

```
    WHEN sportclub.salary_coach.idcoach IN
```

```
        (SELECT sportclub.salary_coach.idcoach
```

```
        FROM sportclub.coach, sportclub.part, sportclub.salary_coach
```

```

WHERE sportclub.salary_coach.idcoach=sportclub.part.idcoach
AND
sportclub.part.idcoach=sportclub.coach.idcoach
AND
sportclub.salary_coach.idcoach=sportclub.coach.idcoach
AND sportclub.coach.level='Средняя')
THEN slr+10000
ELSE slr+0
END;
END;
$$ LANGUAGE plpgsql;

```

```

sport=# CREATE OR REPLACE FUNCTION sal() RETURNS SETOF t_slr AS
sport=# $$
sport=# BEGIN
sport=# UPDATE sportclub.salary_coach
sport=# SET slr=CASE
sport=# WHEN sportclub.salary_coach.idcoach IN
sport=# (SELECT sportclub.salary_coach.idcoach
sport=# FROM sportclub.coach, sportclub.part, sportclub.salary_coach
sport=# WHERE sportclub.salary_coach.idcoach=sportclub.part.idcoach AND
sport=# sportclub.part.idcoach=sportclub.coach.idcoach AND sportclub.salary_coach.idcoach=sportclub.coach.idcoach
sport=# AND sportclub.coach.level='Высшая')
sport=# THEN slr+15000
sport=# WHEN sportclub.salary_coach.idcoach IN
sport=# (SELECT sportclub.salary_coach.idcoach
sport=# FROM sportclub.coach, sportclub.part, sportclub.salary_coach
sport=# WHERE sportclub.salary_coach.idcoach=sportclub.part.idcoach AND
sport=# sportclub.part.idcoach=sportclub.coach.idcoach AND sportclub.salary_coach.idcoach=sportclub.coach.idcoach
sport=# AND sportclub.coach.level='Средняя')
sport=# THEN slr+10000
sport=# ELSE slr+0
sport=# END;
sport=# END;
sport=# $$ LANGUAGE plpgsql;
CREATE FUNCTION

```

до:

fullname	level	idcoach	slr
Вафелька Эол Николаевич	Высшая	197295	60000
Весёлая Миузки Андреевна	Высшая	137829	60000
Вишня Оливия	Высшая	100100	60000
Добрый Алексей Викторович	Средняя	890890	60000
Круассанов Олег	Средняя	567567	60000
Родриго Пересвет Александрович	Высшая	924378	60000

(6 строк)

после:

fullname	level	idcoach	slr
Вафелька Эол Николаевич	Высшая	197295	75000
Весёлая Миузки Андреевна	Высшая	137829	75000
Вишня Оливия	Высшая	100100	75000
Добрый Алексей Викторович	Средняя	890890	70000
Круассанов Олег	Средняя	567567	60000
Родриго Пересвет Александрович	Высшая	924378	75000

(6 строк)

проверим:

принимали участие:

```
-----
fullname
-----
Вафелька Эол Николаевич
Весёлая Миузки Андреевна
Вишня Оливия
Добрый Алексей Викторович
Круассанов Олег
Родриго Пересвет Александрович
(6 строк)
```

не принимали участие:

```
-----
fullname
-----
Круассанов Олег
(1 строка)
```

- **Вафельке** повысили на 15к, так как он был средней и в соревнованиях участие принимал, значит, сейчас у него высшая категория
- **Весёлой** повысили, потому что она принимала участие в соревнованиях (тк у неё высшая категория, то в предыдущем запросе категория не изменилась)
- **Вишня** - то же самое
- **Доброму** повысили на 10к, потому что была низшая, принимал участие, =>, средняя (см. пред. запрос)
- **Круассанову** не повысили (была средняя и участия в соревнованиях не принимал, что видно по предыдущей функции)
- **Родриго** - см. Весёлая

### Задание 5. Создать необходимые триггеры.

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.

## Триггер

Создана таблица logs

Триггерная функция из лекции:

```
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.id;
        mstr := 'Add new';
        retstr := mstr || astr;
        INSERT INTO
sportclub.logs(text,added,table_name) values
(retstr,NOW(), TG_TABLE_NAME);
        RETURN NEW;
    ELSIF TG_OP = 'UPDATE' THEN
        astr = NEW.id;
        mstr := 'Update';
        retstr := mstr || astr;
        INSERT INTO
sportclub.logs(text,added,table_name) values
(retstr,NOW(), TG_TABLE_NAME);
        RETURN NEW;
    ELSIF TG_OP = 'DELETE' THEN
        astr = OLD.id;
        mstr := 'Remove';
```

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



```

        retstr := mstr || astr;
        INSERT INTO
sportclub.logs(text,added,table_name) values
(retstr,NOW(), TG TABLE_NAME);
        RETURN OLD;
    END IF;
END;
$$ LANGUAGE plpgsql;

```

CREATE TRIGGER t\_ijk AFTER INSERT OR UPDATE OR DELETE ON sportclub.injbook FOR EACH ROW EXECUTE PROCEDURE add\_to\_log();

```

sport=# CREATE TRIGGER t_aiu AFTER INSERT OR UPDATE OR DELETE ON sportclub.injbook FOR EACH ROW EXECUTE PROCEDURE add_to_log();
CREATE TRIGGER

```

INSERT INTO sportclub.injbook(code, sort, descr) VALUES ('5', 'Перелом ноги', 'Внешняя деформация');

UPDATE sportclub.injbook SET descr='рентген' WHERE code=2;

DELETE FROM sportclub.injbook WHERE code=5;

```

sport=# SELECT * FROM sportclub.logs;

```

text	added	table_name
Add new(5,"Перелом ноги","Внешняя деформация")	2022-05-02 20:17:59.706719	
Add new(5,"Перелом ноги","Внешняя деформация")	2022-05-02 20:17:59.706719	
Update(2,"Перелом руки",рентген)	2022-05-02 20:20:11.907193	
Update(2,"Перелом руки",рентген)	2022-05-02 20:20:11.907193	
Remove(5,"Перелом ноги","Внешняя деформация")	2022-05-02 20:25:17.552688	injbook
Remove(5,"Перелом ноги","Внешняя деформация")	2022-05-02 20:25:17.552688	injbook
Add new(5,"Перелом ноги","Внешняя деформация")	2022-05-02 20:29:35.668115	injbook
Add new(5,"Перелом ноги","Внешняя деформация")	2022-05-02 20:29:35.668115	injbook
Update(2,"Перелом руки",рентген)	2022-05-02 20:29:50.832425	injbook
Update(2,"Перелом руки",рентген)	2022-05-02 20:29:50.832425	injbook
Remove(5,"Перелом ноги","Внешняя деформация")	2022-05-02 20:29:59.221584	injbook
Remove(5,"Перелом ноги","Внешняя деформация")	2022-05-02 20:29:59.221584	injbook

(12 строк)

Удаление триггерной функции:

```

sport=# DROP TRIGGER t_ijk ON sportclub.injbook;
DROP TRIGGER

```

## Выводы:

Созданы функции и триггеры на запрос, вставку, удаление и редактирование данных.