

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

Отчет

по Лабораторной
Работе № 5 по
дисциплине «**Базы
Данных**» Вариант
16, Спортивный клуб

Автор: Корчагин Вадим Сергеевич

Факультет:

ФИКТ

Группа:

К3241

Преподаватель: Говорова Марина Михайловна



Введение

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:

- Создать процедуры/функции согласно индивидуальному заданию (часть 4).
- 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.

Процедуры/функции

1. Для вывода данных о результатах заданного спортсмена за предыдущий год.

```
CREATE OR REPLACE FUNCTION get_sportsman_results(sportsman_id integer)
RETURNS TABLE (
    competition_name character varying(100),
    sportsman_name character varying(20),
    sportsman_surname character varying(20),
    prize_place numeric,
    sportsmans_points integer,
    competition_date timestamp without time zone
) AS $$
BEGIN
    RETURN QUERY
    SELECT
        c.name_competition,
        s.name,
        s.surname,
        cc.prize_place,
        cc.sportsmans_points,
        cc.date_time
    FROM
        competition.competition_conducting cc
    JOIN
        competition.competition c ON cc.competition_code = c.competition_code
    JOIN
        sportsman.sportsman s ON cc.sportsman_code = s.sportsman_code
    WHERE
        s.sportsman_code = sportsman_id
        AND cc.status = 'passed'
        AND EXTRACT(YEAR FROM cc.date_time) = EXTRACT(YEAR FROM
CURRENT_DATE) - 1;
END;
$$ LANGUAGE plpgsql;
```

```
new=# SELECT * FROM get_sportsman_results(75890);
competition_name | sportsman_name | sportsman_surname | prize_place | sportsmans_points | competition_date
-----+-----+-----+-----+-----+-----
Tracy            | Gloria         | Renee              | 1           | 160               | 2022-01-12 00:00:01
David            | Gloria         | Renee              | 1           | 1                 | 2022-01-12 00:00:01
(2 ёсЁюьш)
```

2. Для вывода данных о соревнованиях, проводимых в первом квартале текущего года.

```
CREATE OR REPLACE FUNCTION get_competitions_in_first_quarter()
RETURNS TABLE (
    competition_code integer,
    name_competition character varying(100),
    date_time timestamp without time zone,
    venue character varying(60),
    category character varying(20),
    type_competition character varying(20),
    status character varying(10)
)
AS $$
BEGIN
    RETURN QUERY
    SELECT
        c.competition_code,
        c.name_competition,
        c.date_time,
        c.venue,
        c.category,
        c.type_competition,
        c.status
    FROM
        competition.competition c
    WHERE
        EXTRACT(QUARTER FROM c.date_time) = 1
        AND EXTRACT(YEAR FROM c.date_time) = EXTRACT(YEAR FROM
CURRENT_DATE);
END;
$$ LANGUAGE plpgsql;
```

```
new=# SELECT * FROM get_competitions_in_first_quarter();
 competition_code | name_competition |      date_time      | venue  | category  | type_competition | status
-----
11656 | Julia           | 2023-01-21 00:00:01 | Denise | AprilJoke | Brian           | passed
15398 | Leslie          | 2023-01-22 00:00:01 | Adrienne | EXE       | Matthew         | passed
2843 | Jacob           | 2023-01-23 00:00:01 | Lonnie  | EXE       | Lance           | passed
```

3. Для повышения рейтинга и оклада тренера после участия в соревновании.

```
CREATE OR REPLACE FUNCTION public.update_coach_performance(  
    p_competition_code integer)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
    COST 100  
    VOLATILE PARALLEL UNSAFE  
AS $BODY$  
DECLARE  
    coach_code_val INTEGER;  
    prize_place_val INTEGER;  
    coach_rating_val INTEGER;  
    coach_salary_val NUMERIC;  
    multiplier NUMERIC;  
BEGIN  
    SELECT cc.coach_code, cc.prize_place  
    INTO coach_code_val, prize_place_val  
    FROM competition.competition_conducting cc  
    WHERE cc.competition_conducting_code = p_competition_code;  
  
    SELECT cr.rating_number  
    INTO coach_rating_val  
    FROM coach.coach_rating cr  
    WHERE cr.coach_code = coach_code_val;  
  
    SELECT c.salary  
    INTO coach_salary_val  
    FROM coach.coach c  
    WHERE c.coach_code = coach_code_val;  
  
    UPDATE coach.coach_rating  
    SET rating_number = CASE prize_place_val  
        WHEN 1 THEN coach_rating_val + 10  
        WHEN 2 THEN coach_rating_val + 7  
        WHEN 3 THEN coach_rating_val + 5  
        ELSE coach_rating_val  
    END  
    WHERE coach_code = coach_code_val;  
  
    IF coach_salary_val >= 30000 THEN  
        multiplier := 1;  
    ELSE  
        CASE prize_place_val  
            WHEN 1 THEN multiplier := 1.1;  
            WHEN 2 THEN multiplier := 1.07;  
            WHEN 3 THEN multiplier := 1.05;  
            ELSE multiplier := 1;  
        END CASE;  
    END IF;  
END $BODY$;
```

END IF;

```
UPDATE coach.coach
SET salary = coach_salary_val * multiplier
WHERE coach_code = coach_code_val;
END;
$BODY$;
```

```
new=# SELECT * FROM coach.coach WHERE coach_code = 86978;
 salary | coach_code | passport_number | name | surname | patronymic | phone_number
-----+-----+-----+-----+-----+-----+-----
 1963.52734996 |      86978 | 2590531 | Christina | Gavin | Katelyn | 34483
(1 ÷)
(1 ÷)
```

```
new=# SELECT * FROM coach.coach_rating WHERE coach_code = 86978;
 rating_code | from_date | to_date | coach_code | rating_number
-----+-----+-----+-----+-----
      4396 | 2029-04-30 | 2029-05-01 |      86978 | 23195
(1 ÷)
(1 ÷)
```

```
new=# SELECT update_coach_performance(98023)
new=# ;
 update_coach_performance
-----
(1 ÷)
(1 ÷)
```

```
new=# SELECT * FROM coach.coach WHERE coach_code = 86978;
 salary | coach_code | passport_number | name | surname | patronymic | phone_number
-----+-----+-----+-----+-----+-----+-----
 2159.880084956 |      86978 | 2590531 | Christina | Gavin | Katelyn | 34483
(1 ÷)
(1 ÷)
```

```
new=# SELECT * FROM coach.coach_rating WHERE coach_code = 86978;
 rating_code | from_date | to_date | coach_code | rating_number
-----+-----+-----+-----+-----
      4396 | 2029-04-30 | 2029-05-01 |      86978 | 23205
(1 ÷)
(1 ÷)
```

Актив
Чтобы а
"Парам

Триггеры

Триггер на обновление данных в таблице `competition.competition_conducting`

1. Если статус соревнования равен 'passed', то нельзя обновить данные.
2. Если статус соревнования равен 'in_progress', то можно обновить только поля `coachs_points` и `sportsmans_points`.
3. Если статус соревнования равен 'not_passed', то можно обновить все поля, кроме `prize_place`.

```
CREATE OR REPLACE FUNCTION
public.before_update_competition_conducting()
  RETURNS trigger
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
  IF OLD.status = 'passed' THEN
    RAISE EXCEPTION 'Нельзя обновить данные прошедших соревнований.';
  ELSIF OLD.status = 'in_progress' THEN
    NEW.coachs_points = COALESCE(NEW.coachs_points, OLD.coachs_points);
    NEW.sportsmans_points = COALESCE(NEW.sportsmans_points,
OLD.sportsmans_points);
  ELSIF OLD.status = 'not_passed' THEN
    NEW.prize_place = COALESCE(NEW.prize_place, OLD.prize_place);
  END IF;

  RETURN NEW;
END;
$BODY$;

ALTER FUNCTION public.before_update_competition_conducting()
  OWNER TO postgres;
```

```
new=# UPDATE competition.competition_conducting SET coachs_points = 60, sportsmans_points = sportsmans_points + 10 WHERE competition_conducting_code
= 1;
ОШИБКА: Нельзя обновить данные прошедших соревнований.
КОНТЕКСТ: функция PL/pgSQL before_update_competition_conducting(), строка 4, оператор RAISE
```

Активация Windows

Чтобы активировать Windows, перейдите в раздел
"Параметры"

Триггер на обновление данных в таблице `competition.competition_conducting`

1. Если соревнование имеет статус пройденного, то у спортсмена обязательно должно стоять место.
2. При вставке спортсмена, спортсмен должен иметь запись в таблице квалификаций.
3. Дата проведения соревнования не может быть меньше даты соревнований.

```
-- FUNCTION: public.check_competition_conducting()
```

```
-- DROP FUNCTION IF EXISTS public.check_competition_conducting();
```

```
CREATE OR REPLACE FUNCTION public.check_competition_conducting()
```

```
    RETURNS trigger
```

```
    LANGUAGE 'plpgsql'
```

```
    COST 100
```

```
    VOLATILE NOT LEAKPROOF
```

```
AS $BODY$
```

```
BEGIN
```

```
    IF NEW.status = 'passed' AND NEW.prize_place IS NULL THEN
```

```
        RAISE EXCEPTION 'Призовое место должно быть указано для  
завершенных соревнований.';
```

```
    END IF;
```

```
    IF NOT EXISTS (
```

```
        SELECT 1
```

```
        FROM competition.qualifying
```

```
        WHERE
```

```
            sportsman_code = NEW.sportsman_code
```

```
            AND competition_code = NEW.competition_code
```

```
    ) THEN
```

```
        RAISE EXCEPTION 'Спортсмен не имеет права для участия в  
соревнованиях без квалификации.';
```

```
    END IF;
```

```
        IF NEW.date_time > (SELECT date_time FROM competition.competition  
WHERE competition_code = NEW.competition_code) THEN
```

```
            RAISE EXCEPTION 'Дата соревнования не может быть больше даты  
соревнования проведения.';
```

```
        END IF;
```

```
    RETURN NEW;
```

```
END;
```

```
$BODY$;
```

```
ALTER FUNCTION public.check_competition_conducting()
```

```
    OWNER TO postgres;
```



```
new=# INSERT INTO competition.competition_conducting (
new(#      competition_code, sportsman_code, coach_code, sport_code, status, coachs_points, sportsmans_points, date_time, prize_place
new(# ) VALUES ( 273, 79500, 70473, 96165, 'passed', 41568, 97423, '2024-01-01 00:00:01', NULL);
ОШИБКА: Призовое место должно быть указано для завершенных соревнований.
КОНТЕКСТ: функция PL/pgSQL check_competition_conducting(), строка 4, оператор RAISE
new=#
```

Активация Windows
Чтобы активировать Windows, перейдите в раздел
"Параметры".

```
new=# INSERT INTO competition.competition_conducting (
new(#      competition_code, sportsman_code, coach_code, sport_code, status, coachs_points, sportsmans_points, date_time, prize_place
new(# ) VALUES ( 273, 79500, 70473, 96165, 'passed', 41568, 97423, '2024-01-01 00:00:01', 1);
ОШИБКА: Спортсмен не имеет права для участия в соревнованиях без квалификации.
КОНТЕКСТ: функция PL/pgSQL check_competition_conducting(), строка 14, оператор RAISE
new=#
```

Активация Windows
Чтобы активировать Windows, перейдите в раздел
"Параметры".

Логирование insert, update, delete

```
-- FUNCTION: public.log_trigger()
```

```
-- DROP FUNCTION IF EXISTS public.log_trigger();
```

```
CREATE OR REPLACE FUNCTION public.log_trigger()
```

```
RETURNS trigger
```

```
LANGUAGE 'plpgsql'
```

```
COST 100
```

```
VOLATILE NOT LEAKPROOF
```

```
AS $BODY$
```

```
DECLARE
```

```
old_data JSONB;
```

```
new_data JSONB;
```

```
BEGIN
```

```
IF TG_OP = 'INSERT' THEN
```

```
old_data = NULL;
```

```
new_data = to_jsonb(NEW);
```

```
ELSIF TG_OP = 'UPDATE' THEN
```

```
old_data = to_jsonb(OLD);
```

```
new_data = to_jsonb(NEW);
```

```
ELSIF TG_OP = 'DELETE' THEN
```

```
old_data = to_jsonb(OLD);
```

```
new_data = NULL;
```

```
END IF;
```

```
INSERT INTO log_table (table_name, operation_type, old_data, new_data)
```

```
VALUES (TG_TABLE_NAME, TG_OP, old_data, new_data);
```

```
RETURN NULL;
```

```
END;
```

```
$BODY$;
```

```
ALTER FUNCTION public.log_trigger()
```

```
OWNER TO postgres;
```

```
new=# SELECT * FROM log_table;
```

log_id	table_name	operation_type	operation_time	old_data	new_data
1	coach	DELETE	2023-12-09 15:30:13.676046	{ "name": "Лтрэ", "salary": 50000, "surname": "Лтрэут", "coach_code": 87218, "patronymic": "Лтрэуты", "phone_number": 1234567890, "passport_number": "AB123456" }	
2	coach	DELETE	2023-12-09 15:31:18.783664	{ "name": "John", "salary": 5000, "surname": "Doe", "coach_code": 1, "patronymic": "Smith", "phone_number": 123456789, "passport_number": "ABC123" }	
3	coach	INSERT	2023-12-09 15:31:23.887268		{ "name": "John", "salary": 5000, "surname": "Doe", "coach_code": 1, "patronymic": "Smith", "phone_number": 123456789, "passport_number": "ABC123" }

(3 стр.)

Активация Windows

Чтобы активировать Windows, перейдите в раздел

Вывод

В процессе выполнения лабораторной работы были успешно разработаны и протестированы процедуры и триггеры в базе данных PostgreSQL. Эта работа позволила приобрести практические навыки программирования на SQL и глубже понять принципы функционирования триггеров.