

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

Отчет

по лабораторной работе № 3
«ПРОЦЕДУРЫ, ФУНКЦИИ, ТРИГГЕРЫ В POSTGRESQL, ВАРИАНТ 1»
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 и использования процедур, функций и триггеров в базе данных PostgreSQL.

Практическое задание:

1. Создать процедуры/функции согласно индивидуальному заданию.
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.

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

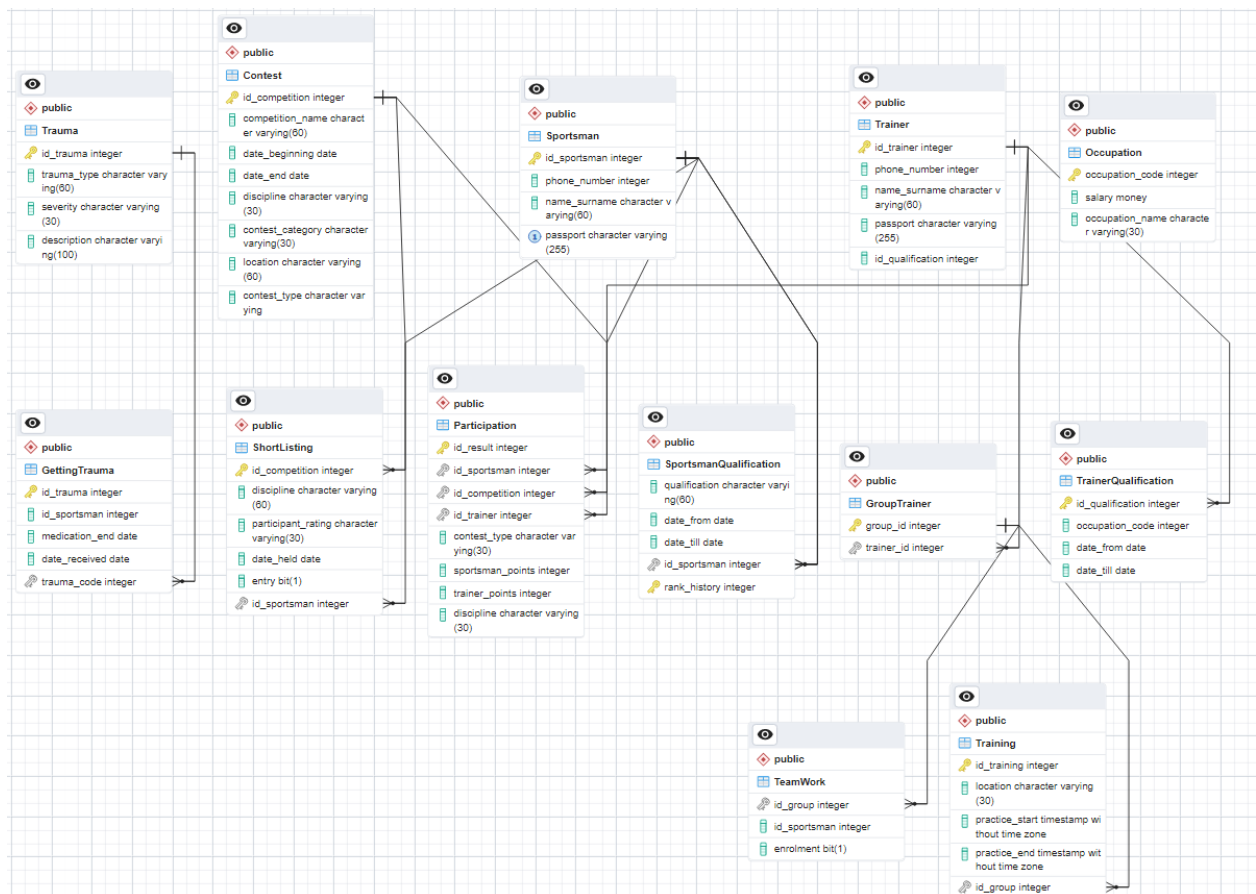


Рисунок 1 - ERD схема базы данных

Задание 1

Создать хранимые процедуры:

- Для вывода данных о результатах заданного спортсмена за прошедший год.
- Для ввода данных о соревнованиях, проводимых в первом квартале текущего года.
- Для повышения рейтинга и оклада тренера после участия в соревновании.

Вывод данных спортсмена за прошедший год

```

SportClub=# CREATE OR REPLACE FUNCTION annual_results(sportsman_id integer)
SportClub=# RETURNS TABLE (score integer)
SportClub=# AS $$
SportClub$$$ BEGIN
SportClub$$$
SportClub$$$ if sportsman_id IN (SELECT id_sportsman FROM "Sportsman") then
SportClub$$$ RETURN QUERY
SportClub$$$
SportClub$$$ SELECT "Participation".sportsman_points FROM "Participation"
SportClub$$$ LEFT JOIN "Sportsman" ON "Participation".id_sportsman = "Sportsman".id_sportsman
SportClub$$$ INNER JOIN "Contest" ON "Participation".id_competition = "Contest".id_competition
SportClub$$$ WHERE ("Sportsman".id_sportsman = sportsman_id)
SportClub$$$ AND (date_part('year' ,"Contest".date_beginning) > date_part('year', current_date) - 5);
SportClub$$$
SportClub$$$ else
SportClub$$$ raise exception 'No id matches this sportsman';
SportClub$$$ end if;
SportClub$$$ END;
SportClub$$$ $$ LANGUAGE plpgsql;
CREATE FUNCTION
  
```

Рисунок 2 - создание функции

```
SportClub=# SELECT annual_results(1) as answer;
answer
-----
      83
      16
      11
```

Рисунок 3 - результат выполнения функции

```
SportClub=#  
SportClub=# SELECT annual_results(1234) as answer;  
ОШИБКА: No sportsman
```

Рисунок 4 – ограничения

Ввод данных соревнований, проводимых в первом квартале текущего года

```

SportClub=# CREATE OR REPLACE FUNCTION add_contest(
SportClub(# id_comp integer,
SportClub(# name_comp varchar(60),
SportClub(# date_b date,
SportClub(# date_e date,
SportClub(# discipline varchar(30),
SportClub(# category varchar(30),
SportClub(# location varchar(60),
SportClub(# type varchar(30)) RETURNS VOID AS
SportClub-# $$
SportClub$$ BEGIN
SportClub$$ IF (3 >= date_part('month', date_b)) AND (date_part('month', date_b) >= 1) AND
SportClub$$ (date_part('year', date_b) = date_part('year', current_date)) THEN
SportClub$$ INSERT INTO "Contest" VALUES
SportClub$$ (id_comp, name_comp, date_b, date_e, discipline, category, location, type);
SportClub$$ ELSE
SportClub$$ RAISE EXCEPTION 'Is not quartal';
SportClub$$ end if;
SportClub$$ END;
SportClub$$ $$
SportClub-# LANGUAGE 'plpgsql';
CREATE FUNCTION

```

Рисунок 5 - создание функции

19	Ғылым мен Өнер	Ғылым мен Өнер	Ғылым мен Өнер	Ғылым мен Өнер	Ғылым мен Өнер	Ғылым мен Өнер	Ғылым мен Өнер
1488	Спортивное жекотание паука	2019-11-17	2020-12-12	Ғылым мен Өнер	Ғылым мен Өнер	Ғылым мен Өнер	Ғылым мен Өнер
(20 Ғылым)		2023-02-18	2023-02-23	Жекотание паука	Мировой	Нью-Йорк	Турнир

Рисунок 6 – результат выполнения функции

```
SportClub=#
SportClub=#
SportClub=# SELECT add_contest(102, 'Спортивное щекотание паука', '18.05.2023', '23.06.2023', 'Щекотание паука', 'Мировой', 'Нью-Йорк', 'Турнир') as answer;
ОШИБКА: Is not quartal
```

Рисунок 7 – ограничения

Изменение оклада тренера в зависимости от результатов


```

SportClub=# CREATE OR REPLACE FUNCTION update_rating_bonus() RETURNS TRIGGER AS $$
SportClub$#
SportClub$# BEGIN
SportClub$# IF NEW.id_trainer IN (SELECT "Trainer".id_trainer FROM "Trainer") THEN
SportClub$#
SportClub$# IF NEW.id_trainer IN (SELECT "Participation".id_trainer FROM "Participation") THEN
SportClub$#
SportClub$# UPDATE "Trainer" SET (rating, bonus) = (
SportClub$# (SELECT AVG(trainer_points) FROM "Participation" WHERE "Participation".id_trainer = NEW.id_trainer),
SportClub$# (
SportClub$# SELECT DISTINCT salary FROM "Occupation"
SportClub$# INNER JOIN "TrainerQualification" ON "TrainerQualification".occupation_code = "Occupation".occupation_code
SportClub$# INNER JOIN "Trainer" ON ("Trainer".id_qualification = "TrainerQualification".id_qualification) AND ("Trainer".id_trainer = NEW.id_trainer)) *
SportClub$# (SELECT AVG(trainer_points) FROM "Participation" WHERE "Participation".id_trainer = NEW.id_trainer) / 400
SportClub$# )
SportClub$# WHERE "Trainer".id_trainer = NEW.id_trainer;
SportClub$# ELSE
SportClub$# RAISE EXCEPTION 'Trainer did not participate in any competition';
SportClub$# END IF;
SportClub$# ELSE
SportClub$# RAISE EXCEPTION 'Trainer does not exist';
SportClub$# END IF;
SportClub$# RETURN NEW;
SportClub$# END;
SportClub$# $$
SportClub=# LANGUAGE 'plpgsql';
CREATE FUNCTION

```

Рисунок 12 - измененная "под триггер" функция

```

CREATE TABLE event_log (id SERIAL PRIMARY KEY,
type VARCHAR(15) NOT NULL,
tablename VARCHAR(30) NOT NULL,
record_id INTEGER NOT NULL,
record_time TIMESTAMP NOT NULL,
new_data JSON,
old_data JSON
);

```

Рисунок 13 - создадим таблицу логов

```

CREATE OR REPLACE FUNCTION log_changes() RETURNS TRIGGER AS $$
DECLARE
    action TEXT;
BEGIN
    IF TG_OP = 'INSERT' THEN
        action := 'INSERT';
    ELSEIF TG_OP = 'UPDATE' THEN
        action := 'UPDATE';
    ELSEIF TG_OP = 'DELETE' THEN
        action := 'DELETE';
    END IF;
    INSERT INTO event_log (tablename, type, new_data, old_data, record_time) VALUES
    (TG_TABLE_NAME, action, to_json(NEW), to_json(OLD), now());
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

Рисунок 14 - создадим функцию для лога

```
CREATE TRIGGER log_changes_trigger AFTER
INSERT OR UPDATE OR DELETE ON "Trainer"
FOR EACH ROW EXECUTE PROCEDURE log_changes();
```

Рисунок 15 - создадим триггер

```
SportClub=# UPDATE "Trainer" SET (id_trainer, name_surname) = (99, 'Boryc Акимов') WHERE "Trainer".id_trainer = 30;
UPDATE 1
SportClub=# SELECT * FROM event_log
SportClub=#
id | type | tablename | record_time | old_data | new_data |
-----+-----+-----+-----+-----+-----+
1 | UPDATE | Trainer | 2023-05-18 14:47:56.931226 | {"id_trainer":99,"phone_number":1364108113,"name_surname":"Boryc Акимов","passport":"79392964944","id_qualification":11,"rating":null,"bonus":null} | {"id_
trainer":30,"phone_number":1364108113,"name_surname":"Евгений Кривош", "passport":"79392964944","id_qualification":11,"rating":null,"bonus":null}
(1 #Empty)
```

Рисунок 16 - добавление записи в лог при UPDATE в таблице Trainer

Выводы:

В процессе работы ознакомился с процессом создания триггеров, функций и процедур в PSQL.