МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

**Базы данных**

Пользовательские функции. Процедурный язык PL/pgSQL

Выполнил студент группы ИВТ-31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Клюкин В.Л./

Киров 2022

1. Задание

При выполнении работы нужно использовать БД, созданную в лабораторных работах №1 и №2. Нужно выполнить следующие шаги.

1) Для любой таблицы создать функцию save\_имя\_таблицы, которая принимает на вход параметры, соответствующие её столбцам, и, если переданное значение первичного ключа равно null, выполняет запрос insert, иначе – запрос update для соответствующей строки. Функция должна вернуть значение первичного ключа вставленной или изменённой строки.

2) Для любой таблицы, на которую имеются внешние ключи, создать функцию delete\_имя\_таблицы, принимающую на вход значение первичного ключа строки и ничего не возвращающую.

3) Для таблицы, содержащей столбец с числовыми значениям, создать функцию, которая принимает на вход число – минимальное значение – и возвращает setof имя\_таблицы – множество строк, в которых значение числа больше или равно переданному аргументу.

4) Создать составной тип, содержащий не менее 2-3 полей, по крайней мере одно из которых должно быть числовым. Создать функцию, которая принимает массив объектов этого типа и минимальное значение для указанного поля. Функция должна возвращать массив отфильтрованных по переданному значению объектов.

5) Для любой таблицы создать таблицу log\_имя\_таблицы, которая будет содержать лог изменений по любому выбранному столбцу этой таблицы. Для этого нужны столбцы:

* первичный ключ;
* внешний ключ на выбранную таблицу;
* дата изменения строки;
* старое значение столбца;
* новое значение столбца.

Реализовать заполнение таблицы с логом с помощью триггеров на вставку/изменение строк.

6) Реализовать любую функцию на свой выбор, использующую для получения результата динамически формируемый запрос.

Все функции должны быть реализованы на PL/pgSQL. Отчет по лабораторной работе должен содержать код создания перечисленных функций, составного типа из шага 4 и таблицы из шага 5, а также демонстрацию работы созданных функций.

1. Ход работы

В качестве предметной области для создания БД был выбран баскетбол, а именно – НБА. Структура содержит 6 таблиц: игроки, тренеры, команды, статистика, тренерский стиль, показатели игроков. 5 таблиц имеют свой уникальный идентификатор, который будет необходим в дальнейшем.

Сначала было принято решение реализовать функцию save\_statistics, которая принимает на входе название статистики и средний показатель статистики по лиге. В случае, если данный показатель, уже есть в таблице, то его значение меняется на новое. Если же его не было, то добавляется новое значение. В случае NULL, ничего не происходит, таблица просто обновляет старое значение на себя же.

На следующем шаге реализовалась функция delete\_statistics, которая удаляет строку по заданному id. Если введенного id нет в таблице, то ничего не произойдет, будет выведено сообщение о невозможности удаления данной строки из таблицы. В случае же, если введенный идентификатор присутствует в таблице, то произойдет соответствующее удаление.

Следующая функция – фильтрация таблицы по заданному ключу. Реализация функции будет производиться для таблицы тренеров, coaches. Фильтрация будет производиться по полю age.

Функция фильтрации массива объектов будет принимать на вход массив данных и числовое значение, по которому будет фильтроваться массив данных и возвращаться из функции. Было принято решение реализовать данную функцию для таблицы coaches.

Функция логирования действий для таблицы coaches будет не простой, а триггерной. Триггерная функция будет определять какой вид запроса был произведён и будет в соответствии с ним корректировать запрос. Триггер будет срабатывать после внесения изменений в таблицу coaches.

Функция для создания динамически формируемых запросов. Созданная функция принимает на вход название таблицы, название столбца и id поля, которое будет выведено. На выходе будет возвращаться текстовая строка с содержимым поля.

Все результаты и реализация приведены в приложениях А и Б.

1. Листинг программы

Листинг программы приведен в приложении А.

1. Экранные формы

Экранные формы приведены в приложении Б.

1. Вывод

В ходе выполнения лабораторной работы были изучены основы написания пользовательских функций на процедурном языке PL/pgSQL. Освоена работа с составными типами данных и массивами. Познакомились с созданием триггеров.

Приложение А

(обязательное)

Листинг программы

* Обновление данных в строке и добавление новых

DECLARE

used\_id BIGINT;

BEGIN

IF \_id IS NULL THEN

INSERT INTO statistics (stats\_name, average\_stats)

VALUES (\_stats\_name, \_average\_stats)

RETURNING id /\* Конструкция позволяет вернуть id нового элемента \*/

INTO used\_id; /\* id нового элемента записывается в переменную used\_id \*/

ELSE

UPDATE statistics SET

stats\_name = \_stats\_name,

average\_stats = \_average\_stats

WHERE id = \_id;

used\_id := \_id; /\* Нам уже известен id, поэтому просто присвоим его \*/

END IF;

RETURN used\_id;

END;

* Удаление строки по id

BEGIN

DELETE FROM statistics WHERE id = \_id; /\* Просто пытаемся удалить строку \*/

EXCEPTION

WHEN foreign\_key\_violation THEN /\* Перехват нужного нам исключения \*/

RAISE EXCEPTION 'Невозможно выполнить удаление, так как есть внешние ссылки.';

END;

* Функция фильтрации по числовому значению

CREATE OR REPLACE FUNCTION filter\_subject\_by\_hours (

min\_val BIGINT

)

RETURNS SETOF players

AS $$

BEGIN

RETURN QUERY (SELECT \* FROM players WHERE age >= min\_val);

END;

$$ LANGUAGE plpgsql;

* Функция фильтрации массива объектов

CREATE TYPE t\_coaches AS (

id BIGINT,

first\_name VARCHAR(50),

second\_name VARCHAR(50),

experience INTEGER

);

CREATE OR REPLACE FUNCTION filter\_array\_of\_coaches (

arr t\_coaches[],

filter\_var INTEGER

)

RETURNS t\_coaches[]

AS $$

BEGIN

RETURN ARRAY( /\* Преобразуем выборку в массив \*/

SELECT (id, first\_name, second\_name, experience)::t\_coaches /\* Создаем таблицу из элементов массива \*/

FROM unnest(arr)

WHERE experience >= filter\_var

);

END;

$$ LANGUAGE plpgsql;

* Функция для создания таблицы с логом изменений

CREATE TABLE log\_coaches (

id BIGSERIAL PRIMARY KEY,

coaches\_id BIGINT REFERENCES coaches(id),

change\_datetime TIMESTAMP DEFAULT NOW(),

old\_value INT DEFAULT NULL,

new\_value INT DEFAULT NULL

);

CREATE OR REPLACE FUNCTION trigger\_func()

RETURNS TRIGGER

AS $$

DECLARE

old\_val INT;

BEGIN

/\* Определяем "старое" значение \*/

IF (TG\_OP = 'UPDATE') THEN

old\_val := OLD.experience;

ELSIF (TG\_OP = 'INSERT') THEN

old\_val := NULL;

end if;

/\* Производим запрос \*/

INSERT INTO log\_coaches

(coaches\_id, old\_value, new\_value)

VALUES

(NEW.id, old\_val, NEW.experience);

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER commit\_coaches\_change

AFTER UPDATE OR INSERT

ON coaches

FOR EACH ROW

EXECUTE PROCEDURE trigger\_func();

* Функция для создания динамически формируемого запроса

CREATE OR REPLACE FUNCTION get\_value\_by\_id (

tableName VARCHAR,

columnName VARCHAR,

id BIGINT

)

RETURNS TEXT

AS $$

DECLARE

result TEXT;

BEGIN

EXECUTE 'SELECT '|| columnName ||' FROM '|| tableName ||' WHERE id = $1' USING id INTO result;

RETURN result;

END;

$$ LANGUAGE plpgsql;

Приложение Б

(обязательное)

Экранные формы

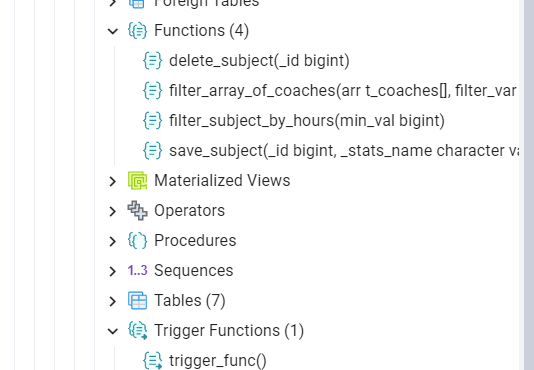


Рисунок 1 – Отображение функция в среде разработки

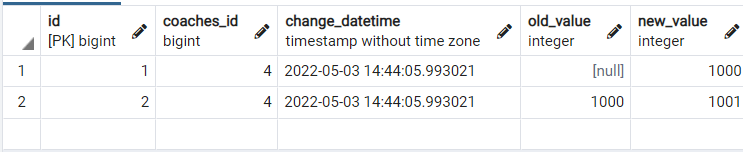


Рисунок 2- Таблица с логом изменений в таблице coaches

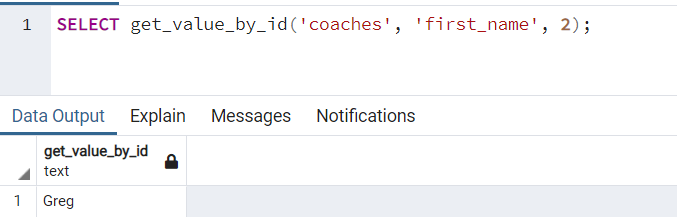


Рисунок 3 – Таблица динамически сформированного запроса