**Лабораторная работа по теме “ЯзыкPL/pgSQL”**

1.

**Создаем таблицы:**

create table students(

id SERIAL primary key,

name text, total\_score integer default 0

);

create table activity\_scores (

student\_id integer,

activity\_type text,

score integer,

CONSTRAINT activity\_scores\_student\_id FOREIGN KEY(student\_id) REFERENCES students(id)

);

**Создаем функция обновления:**

CREATE OR REPLACE FUNCTION update\_total\_score(student\_id integer)

returns void AS

$$

DECLARE

activity\_scores\_row activity\_scores%ROWTYPE;

cursor CURSOR (id integer) FOR SELECT \*

FROM activity\_scores

where activity\_scores.student\_id = id;

score integer;

BEGIN

score := 0;

OPEN cursor(student\_id);

LOOP

FETCH cursor INTO activity\_scores\_row;

EXIT WHEN NOT FOUND;

score := score + activity\_scores\_row.score;

END LOOP;

CLOSE cursor;

UPDATE students set total\_score = score where id = student\_id;

END;

$$ LANGUAGE plpgsql;

**Создаем триггерную функцию:**

CREATE OR REPLACE FUNCTION update\_total\_score\_trigger\_function()

RETURNS TRIGGER AS $$

DECLARE student\_id students.id%TYPE;

BEGIN

IF (TG\_OP = 'DELETE') THEN

student\_id = OLD.student\_id;

ELSE

student\_id = NEW.student\_id;

END IF;

PERFORM update\_total\_score(student\_id);

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

**Создаем триггер:**

CREATE OR REPLACE TRIGGER update\_total\_score\_trigger

AFTER INSERT OR UPDATE OR DELETE ON activity\_scores

FOR EACH ROW

EXECUTE FUNCTION update\_total\_score\_trigger\_function()

**Заполняем данные:**

insert into students(name)

values ('Сергей'),

('Иван'),

('Дарья'),

('Антон'),

('Юлия');

insert into activity\_scores values

(1, 'футбол', 1),

(1, 'Прогулка', 1),

(1, 'Бег', 1),

(1, 'Кино', 2),

(1, 'Театр', 1),

(2, 'футбол', 3),

(2, 'Прогулка', 1),

(2, 'Бег', 4),

(2, 'Кино', 1),

(2, 'Театр', 6),

(3, 'футбол', 3),

(3, 'Прогулка', 2),

(3, 'Бег', 6),

(3, 'Кино', 8),

(3, 'Театр', 3),

(4, 'футбол', 3),

(4, 'Прогулка', 2),

(4, 'Бег', 6),

(4, 'Кино', 8),

(4, 'Театр', 3),

(5, 'футбол', 3),

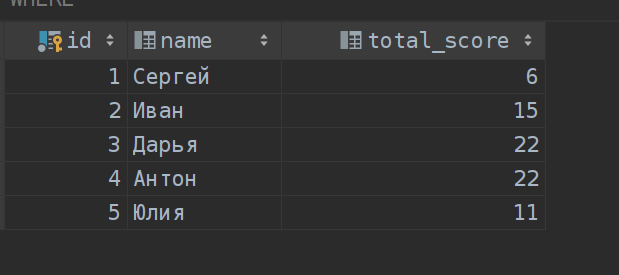
(5, 'Прогулка', 2),

(5, 'Бег', 1),

(5, 'Кино', 2),

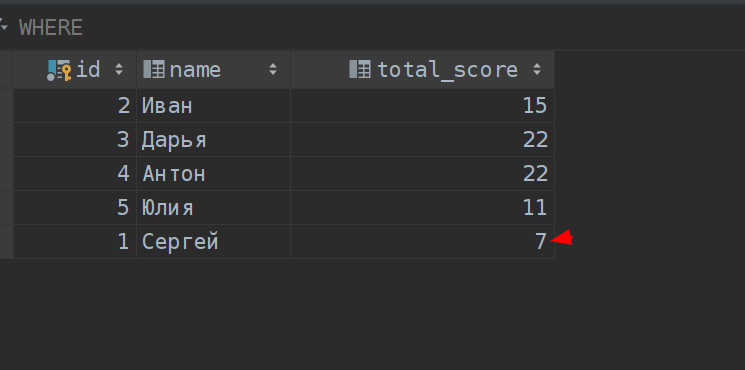
(5, 'Театр', 3);

**Проверяем результат после вставки**

****

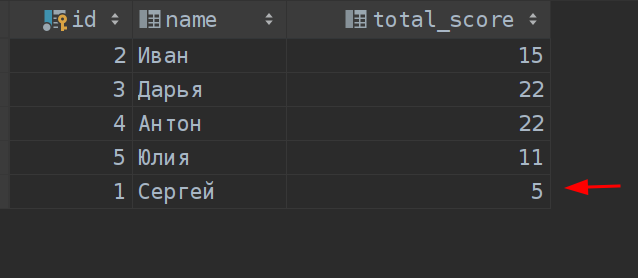
**Проверяем результат после обновления:**

update activity\_scores set score = score + 1 where student\_id=1 and activity\_type = 'футбол';



**Проверяем результат после удаления:**

delete from activity\_scores where student\_id=1 and activity\_type = 'футбол';



2.

**Добавляем колонку:**

ALTER TABLE students ADD COLUMN scholarship integer default 0;

Добавим функцию расчета стипендии:

CREATE OR REPLACE FUNCTION calculate\_scholarship(score integer)

RETURNS integer AS

$$

DECLARE

scholarship integer;

BEGIN

IF score >= 90 THEN

scholarship = 1000;

ELSEIF score >= 80 AND score < 90 THEN

scholarship = 500;

ELSE

scholarship = 0;

END IF;

RETURN scholarship;

END;

$$ LANGUAGE plpgsql;

**Учитывая что у нас уже обновляется total\_score в students - повесим триггер на students. А чтобы избежать двойного update - повесим триггер как BEFORE**

CREATE OR REPLACE FUNCTION calculate\_scholarship\_trigger\_function()

RETURNS TRIGGER AS $$

BEGIN

IF (TG\_OP = 'INSERT') THEN

NEW.scholarship = calculate\_scholarship(NEW.total\_score);

ELSEIF (NEW.total\_score != OLD.total\_score) THEN

NEW.scholarship = calculate\_scholarship(NEW.total\_score);

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER calculate\_scholarship\_trigger

BEFORE INSERT OR UPDATE ON students

FOR EACH ROW

EXECUTE FUNCTION calculate\_scholarship\_trigger\_function()

**Проверим результат после обновления/добавления записей:**

