**Задача 1: Расчет стипендии для студентов**

Т.к. в задаче не было явно указано куда сохраняется результат расчетов то была добавленна колонка scholarship в таблицу students

create table public.students (

id serial not null,

"name" text null,

total\_score integer null,

scholarship integer null,

constraint students\_pk primary key (id)

);

create table public.activity\_scores (

student\_id integer not null,

activity\_type text null,

score integer null

);

**внесение первоначальных данных**

insert into students (name,total\_score,scholarship) values('Иван Иванов',0,0);

insert into students (name,total\_score,scholarship) values('Екатерина Андреева',0,0);

insert into students (name,total\_score,scholarship) values('Анна Потапова',0,0);

insert into students (name,total\_score,scholarship) values('Ильяс Мухаметшин',0,0);

insert into students (name,total\_score,scholarship) values('Сергей Петров',0,0);

insert into activity\_scores (student\_id,activity\_type,score) values(1,'Home work',5);

insert into activity\_scores (student\_id,activity\_type,score) values(2,'Home work',15);

insert into activity\_scores (student\_id,activity\_type,score) values(3,'Home work',10);

insert into activity\_scores (student\_id,activity\_type,score) values(4,'Home work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(5,'Home work',25);

insert into activity\_scores (student\_id,activity\_type,score) values(1,'Class work',5);

insert into activity\_scores (student\_id,activity\_type,score) values(2,'Class work',15);

insert into activity\_scores (student\_id,activity\_type,score) values(3,'Class work',10);

insert into activity\_scores (student\_id,activity\_type,score) values(4,'Class work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(5,'Class work',25);

insert into activity\_scores (student\_id,activity\_type,score) values(1,'Exam',5);

insert into activity\_scores (student\_id,activity\_type,score) values(2,'Exam',15);

insert into activity\_scores (student\_id,activity\_type,score) values(3,'Exam',10);

insert into activity\_scores (student\_id,activity\_type,score) values(4,'Exam',20);

insert into activity\_scores (student\_id,activity\_type,score) values(5,'Exam',25);

insert into activity\_scores (student\_id,activity\_type,score) values(1,'Ext work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(2,'Ext work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(3,'Ext work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(4,'Ext work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(5,'Ext work',20);

**Создание функции calculate\_scholarship**

create or replace procedure calculate\_scholarship()

language plpgsql

as $$

begin

-- stored procedure body

with sub as (

select student\_id, sum(score) as total\_score from activity\_scores group by student\_id

)

update students set total\_score = sub.total\_score,

scholarship = (case when sub.total\_score >=90 then 1000

when sub.total\_score >=80 and sub.total\_score <90 then 500 else 0 end)

from sub

where students.id = sub.student\_id;

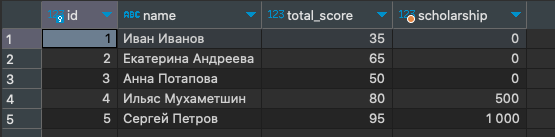
end; $$

**Вызов функции calculate\_scholarship**

call calculate\_scholarship()

**Проверка результата**

select \* from students order by id



**создание тригера**

create or replace function update\_scholarship\_trigger()

returns trigger

language plpgsql

as $$

begin

-- trigger logic

call public.calculate\_scholarship();

return new;

end;

$$

create trigger update\_scholarship\_trigger

after insert or update

on activity\_scores

for each row

execute procedure update\_scholarship\_trigger();

**добавляение данных для проверки работы тригера**

insert into activity\_scores (student\_id,activity\_type,score) values(1,'Exam 2',10);

insert into activity\_scores (student\_id,activity\_type,score) values(2,'Exam 2',20);

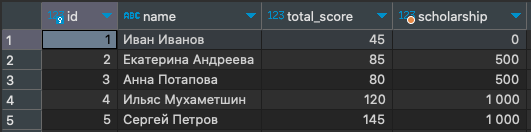
insert into activity\_scores (student\_id,activity\_type,score) values(3,'Exam 2',30);

insert into activity\_scores (student\_id,activity\_type,score) values(4,'Exam 2',40);

insert into activity\_scores (student\_id,activity\_type,score) values(5,'Exam 2',50);

**проверка результата**

select \* from students order by id



**Задача 2: Учет баллов студентов**

Создание таблиц

create table public.students (

id serial not null,

"name" text null,

total\_score integer null,

constraint students\_pk primary key (id)

);

create table public.activity\_scores (

student\_id integer not null,

activity\_type text null,

score integer null

);

alter table public.activity\_scores add constraint activity\_scores\_fk foreign key (student\_id) references public.students(id);

**внесение первоначальных данных**

insert into students (name,total\_score) values('Иван Иванов',0);

insert into students (name,total\_score) values('Екатерина Андреева',0);

insert into students (name,total\_score) values('Анна Потапова',0);

insert into students (name,total\_score) values('Ильяс Мухаметшин',0);

insert into students (name,total\_score) values('Сергей Петров',0);

insert into activity\_scores (student\_id,activity\_type,score) values(1,'Home work',5);

insert into activity\_scores (student\_id,activity\_type,score) values(2,'Home work',15);

insert into activity\_scores (student\_id,activity\_type,score) values(3,'Home work',10);

insert into activity\_scores (student\_id,activity\_type,score) values(4,'Home work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(5,'Home work',25);

insert into activity\_scores (student\_id,activity\_type,score) values(1,'Class work',5);

insert into activity\_scores (student\_id,activity\_type,score) values(2,'Class work',15);

insert into activity\_scores (student\_id,activity\_type,score) values(3,'Class work',10);

insert into activity\_scores (student\_id,activity\_type,score) values(4,'Class work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(5,'Class work',25);

insert into activity\_scores (student\_id,activity\_type,score) values(1,'Exam',5);

insert into activity\_scores (student\_id,activity\_type,score) values(2,'Exam',15);

insert into activity\_scores (student\_id,activity\_type,score) values(3,'Exam',10);

insert into activity\_scores (student\_id,activity\_type,score) values(4,'Exam',20);

insert into activity\_scores (student\_id,activity\_type,score) values(5,'Exam',25);

insert into activity\_scores (student\_id,activity\_type,score) values(1,'Ext work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(2,'Ext work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(3,'Ext work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(4,'Ext work',20);

insert into activity\_scores (student\_id,activity\_type,score) values(5,'Ext work',20);

**Создание функции update\_total\_score**

create or replace procedure update\_total\_score(v\_student\_id integer)

language plpgsql

as $$

declare

-- variable declaration

score\_rec record;

total integer;

begin

-- stored procedure body

total = 0;

for score\_rec in (select score from activity\_scores where student\_id=v\_student\_id)

loop

-- raise notice '%', score\_rec.score;

total = total + score\_rec.score;

end loop;

update students set total\_score=total where id=v\_student\_id;

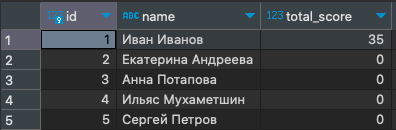
end; $$

**Вызов функции update\_total\_score**

call update\_total\_score(1)

**Проверка результата**

select \* from students order by id



**создание тригера**

create or replace function trigger\_update\_total\_score()

returns trigger

language plpgsql

as $$

begin

-- trigger logic

call public.update\_total\_score(new.student\_id);

return new;

end;

$$

create trigger update\_total\_score

after insert or update

on activity\_scores

for each row

execute procedure trigger\_update\_total\_score();

**добавляение данных для проверки работы тригера**

insert into activity\_scores (student\_id,activity\_type,score) values(1,'Exam 2',10);

insert into activity\_scores (student\_id,activity\_type,score) values(2,'Exam 2',20);

insert into activity\_scores (student\_id,activity\_type,score) values(3,'Exam 2',30);

insert into activity\_scores (student\_id,activity\_type,score) values(4,'Exam 2',40);

insert into activity\_scores (student\_id,activity\_type,score) values(5,'Exam 2',50);

**проверка результата**

select \* from students order by id

