**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Поволжский государственный университет телекоммуникаций и информатики»**

**Факультет Информационных систем и технологий**

**Направление Информатика и вычислительная техника**

**Кафедра Программной инженерии**

**Лабораторная работа №4**

**Проверил Питерсков А.П.**

**Выполнил Морзюков. М.А.**

Самара 2024

**Цель работы**:

* Знакомство с языком PL/pgSQL.
* Создание процедур, функций и триггеров,
* Использование транзакции для обеспечения целостности.

**Ход работы:**

1. Сделать таблицу с историей изменений company\_name, в которую при обновлении через триггер записываются прежние значения и дата окончания их действий (дата обновления). Сделайте запрос, который показывает значения для конкретного кортежа на заданный момент времени.

2. Создать функцию с входным параметром - имя актера. Если актер с таким именем не найден, функция должна вернуть 0. Если найден, то вычислить через person\_info birth date (info\_type\_id=21) и death date (info\_type\_id=23) возраст актера. Если актер еще не умер, то вычислить, сколько сейчас лет. Функция должна вернуть целое число прожитых лет.

3. Используя функцию, созданную ранее, создайте процедуру, которая по имени актера выводит в лог текст следующего содержания:

Name: ...

Nicknames: nickname1, nickname2, ... строка выводится только если у человека есть клички, выводятся все клички через запятую

Age: ... вычисляется из функции

First appear: ... фильм и год выхода, в котором актер впервые снялся.

Если актер не найден или не удалось определить его первую роль, то процедура должна завершиться с ошибкой и вывести в лог "Invalid data"

1. **Создаём таблицу**

CREATE TABLE company\_name\_history

(

id SERIAL PRIMARY KEY,

finish\_time TIMESTAMP,

company\_id INTEGER,

name TEXT NOT NULL,

country\_code varchar(200),

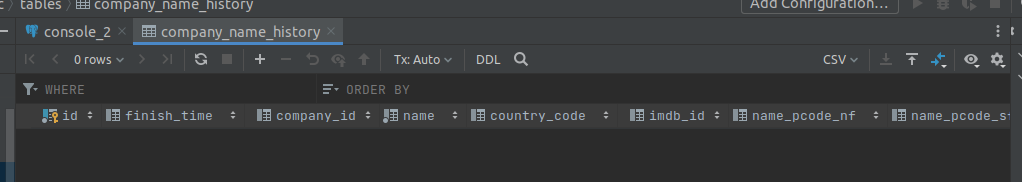
imdb\_id INTEGER,

name\_pcode\_nf varchar(10),

name\_pcode\_sf varchar(10),

md5sum varchar(50)

)



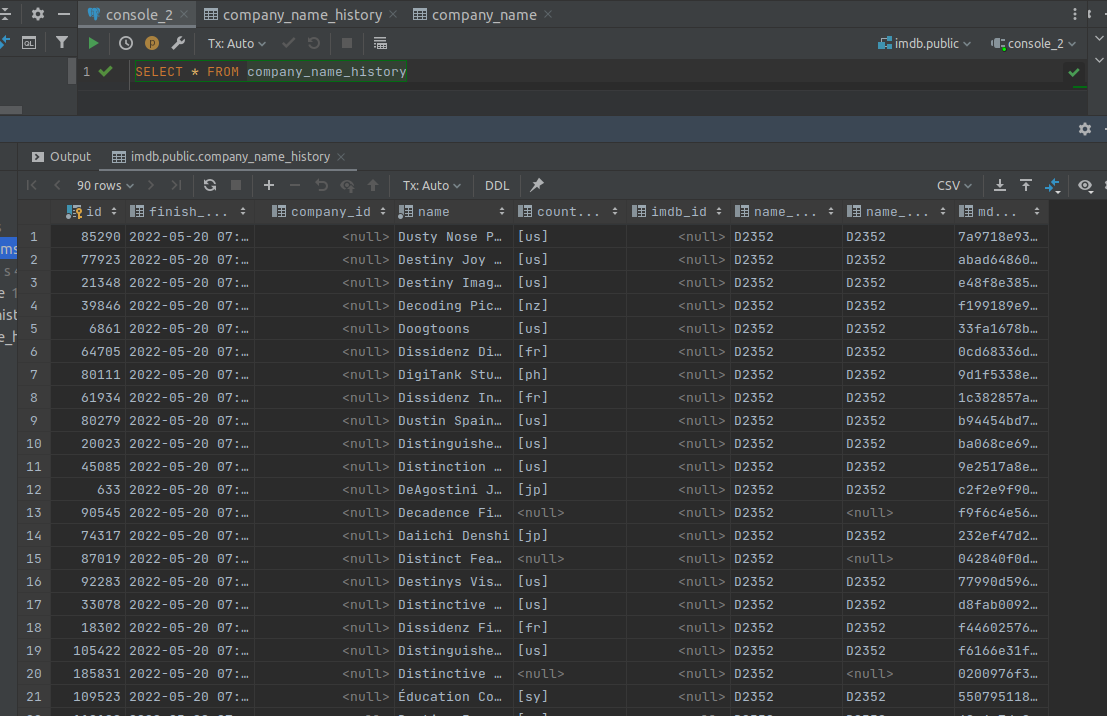
**Создаём функцию и триггер**

CREATE OR REPLACE FUNCTION company\_function() returns trigger  
AS  
$$  
BEGIN  
INSERT INTO company\_name\_history(finish\_time, company\_id,name,country\_code, imdb\_id,name\_pcode\_nf, name\_pcode\_sf, md5sum)  
VALUES (now(),[old.company](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fold.company)\_id,[old.name](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fold.name), [old.country](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fold.country)\_code,[old.imdb](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fold.imdb)\_id,[old.name](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fold.name)\_pcode\_nf, [old.name](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fold.name)\_pcode\_sf,old.md5sum);  
RETURN new;  
end;  
$$ LANGUAGE plpgsql;

———————————-

DROP TRIGGER IF EXISTS tr\_new\_company\_name on company\_name;  
CREATE TRIGGER tr\_new\_company\_name AFTER UPDATE OR DELETE ON company\_name  
FOR EACH ROW EXECUTE PROCEDURE company\_function();

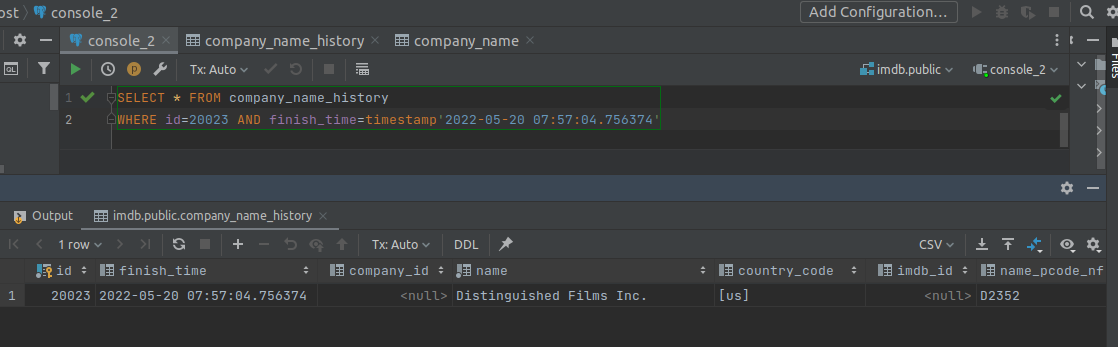
**Проверка**  
SELECT \* FROM company\_name\_history



**Выполнение запроса:**

SELECT \* FROM company\_name\_history

WHERE id=20023 AND finish\_time=timestamp'2024-03-1 07:57:04.756374'



**2) Создание функции для подсчета возраста актёра**

create or replace function sum\_of\_years(looking\_for\_name text) returns integer as

$$

declare

person\_for\_id integer := (select id

from name

where name = looking\_for\_name

limit 1);

begin

if (select id from person\_info where person\_id = person\_for\_id limit 1) is null then

return 0;

elsif (select info from person\_info where info\_type\_id = 23 and person\_id = person\_for\_id) is null then

return now() - (select to\_date(info, 'DD Month YYYY')

from person\_info

where info\_type\_id = 21

and info like '\_% % \_\_\_\_'

and person\_id = person\_for\_id);

else

return ((select to\_date(info, 'DD Month YYYY')

from person\_info

where info\_type\_id = 23

and info like '\_% % \_\_\_\_'

and person\_id = person\_for\_id) - (select to\_date(info, 'DD Month YYYY')

from person\_info

where info\_type\_id = 21

and info like '\_% % \_\_\_\_'

and person\_id = person\_for\_id));

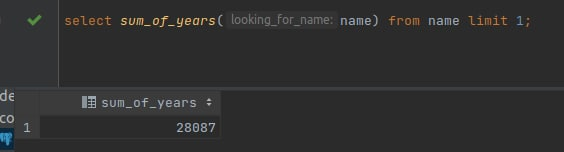
end if;

end;

$$ language plpgsql;

**Проверка:**

SELECT sum\_of\_years(name) FROM name limit 1;

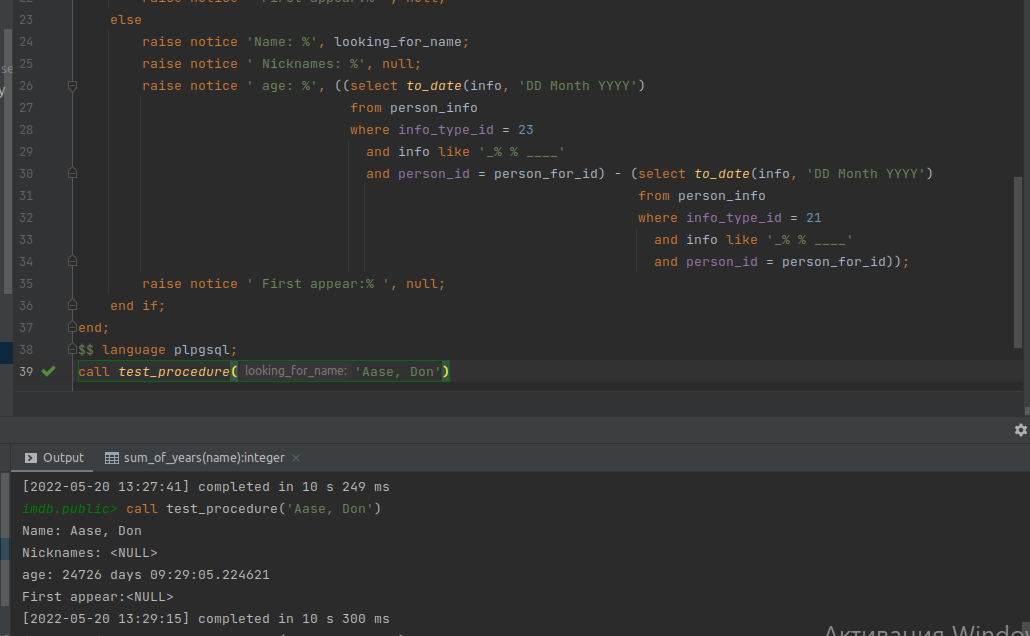


**3) Создание процедуры**

create or replace procedure test\_procedure(looking\_for\_name text) as  
$$  
declare  
    person\_for\_id integer := (select id  
                              from name  
                              where name = looking\_for\_name  
                              limit 1);  
begin  
    if (select id from person\_info where person\_id = person\_for\_id limit 1) is null then  
        raise notice 'Name: %', looking\_for\_name;  
        raise notice ' Nicknames: %', null;  
        raise notice ' age: %', 0;  
        raise notice ' First appear:% ', null;  
    elsif (select info from person\_info where info\_type\_id = 23 and person\_id = person\_for\_id) is null then  
        raise notice 'Name: %', looking\_for\_name;  
        raise notice ' Nicknames: %', null;  
        raise notice ' age: %', now() - (select to\_date(info, 'DD Month YYYY')  
                                         from person\_info  
                                         where info\_type\_id = 21  
                                           and info like '\_% % \_\_\_\_'  
                                           and person\_id = person\_for\_id);  
        raise notice ' First appear:% ', null;  
    else  
        raise notice 'Name: %', looking\_for\_name;  
        raise notice ' Nicknames: %', null;  
        raise notice ' age: %', ((select to\_date(info, 'DD Month YYYY')  
                                  from person\_info  
                                  where info\_type\_id = 23  
                                    and info like '\_% % \_\_\_\_'  
                                    and person\_id = person\_for\_id) - (select to\_date(info, 'DD Month YYYY')  
                                                                      from person\_info  
                                                                      where info\_type\_id = 21  
                                                                        and info like '\_% % \_\_\_\_'  
                                                                        and person\_id = person\_for\_id));  
        raise notice ' First appear:% ', null;  
    end if;  
end;  
$$ language plpgsql;113:19

**Проверка:**

CALL test\_procedure('Aase, Don')



**Вывод:**

Мы научились работать

* с функциями
* триггерами
* процедурами
* применили свои знания на практике, выполнив запросы с ними.