Державний університет «Одеська Політехніка»

Інститут комп’ютерних систем

Кафедра інформаційних систем

Лабораторна робота №6

з дисципліни «Організація баз даних та знань»

Тема: «Маніпулювання даними. Тригери.

Варіант 4

Підготувала студентка

групи АІ 212

Козуб К.О.

Перевірив:

Дрозд М.А

Одеса 2022

1. Опис завдання

Запишіть SQL-запити для маніпулювання даними з таблиць, що створені у

лабораторній роботі 1. В роботі обов’язково відобразити:

1) тригер на операцію модифікації даних INSERT;

2) тригер на операцію модифікації даних DELETE;

3) тригер на операцію модифікації даних UPDATE;

4) користувальницьку функцію (збережену процедуру), яка викликається

оператором select.

1. тригер на операцію модифікації даних INSERT;

Замінити в таблиці УЧЕНЬ ,первинний ключ УМЕНЬ на 0, якщо значення, яке вставляють в цей атрибут, менше або дорівнює 26

SQL Код (створення функції та самого тригера) :

REATE OR REPLACE FUNCTION chek\_year\_of\_birth()

RETURNS TRIGGER AS

$$

BEGIN

IF new.student\_id <= 26 THEN

Update student SET student\_id= 0 WHERE student\_id=new.student\_id;

END IF;

RETURN new;

END;

$$

LANGUAGE plpgsql ;

CREATE TRIGGER chek\_year\_of\_birth

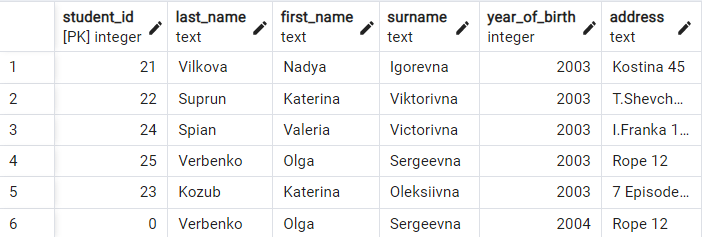
AFTER INSERT ON student

FOR EACH ROW EXECUTE PROCEDURE chek\_year\_of\_birth();

INSERT INTO student VALUES

(5,'Verbenko','Olga','Sergeevna','2004','Rope 12' )

Результат виконання команди:



1. тригер на операцію модифікації даних DELETE

В разі видалення, зменшити значення відповідної послідовності на один з таблиці Учень

SQL Код:

CREATE OR REPLACE FUNCTION student\_deleted() RETURNS trigger

AS $$

DECLARE

old\_id INTEGER;

BEGIN

SELECT student\_id INTO old\_id FROM old WHERE student\_id = old.student\_id;

ALTER SEQUENCE s\_student RESTART WITH old\_id;

RETURN old;

END;

$$ LANGUAGE 'plpgsql';

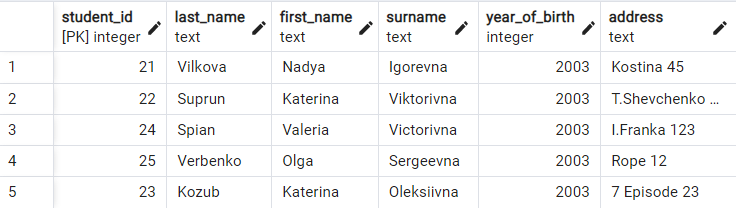
CREATE OR REPLACE TRIGGER student\_deleted

AFTER DELETE ON student

FOR EACH STATEMENT

EXECUTE PROCEDURE student\_deleted()

Результат виконання команди:



1. тригер на операцію модифікації даних UPDATE;

В цьому прикладі я намагаюсь замінити в таблиці Учень значення рік народження на 0, якщо рік народження більше або дорівнює 21.

CREATE FUNCTION checkk\_student\_id() RETURNS trigger

AS $$ BEGIN

IF new.student\_id >= 21 THEN

UPDATE student SET year\_of\_birth= '0' WHERE student\_id =new.student\_id;

END IF;

RETURN new;

END;

$$ LANGUAGE 'plpgsql';

CREATE TRIGGER checkk\_student\_id

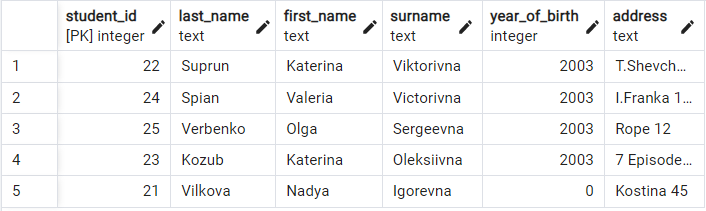
AFTER UPDATE OF student\_id ON student

FOR EACH ROW

EXECUTE PROCEDURE checkk\_student\_id();

Update student SET student\_id = 21

WHERE student\_id = 21;



1. користувальницьку функцію (збережену процедуру), яка викликається оператором select.

Ми створили функцію, яка підраховує кількість років народження, та визвали її

create FUNCTION teacher\_year () RETURNS bigint as

$$

BEGIN

return sum(year\_of\_birth)

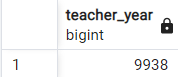
from teacher;

end;

$$

LANGUAGE 'plpgsql';

select teacher\_year();



Висновки: В ході виконання лабораторної роботи було, використовуючи мову SQL створено тригери на операції модифікації даних INSERT, DELETE, UPDATE, а також користувальницьку функцію (збережену процедуру), яка викликається оператором SELECT