CREATE TABLE "student"

(

id int GENERATED BY DEFAULT AS IDENTITY not null PRIMARY KEY,

fio varchar not null UNIQUE

);

CREATE TABLE "groups"

(

id int GENERATED BY DEFAULT AS IDENTITY not null PRIMARY KEY,

title varchar not null UNIQUE

);

CREATE TABLE "institut"

(

student\_id integer NOT NULL REFERENCES "student" (id),

groups\_id integer NOT NULL REFERENCES "groups" (id),

vuz varchar not null,

PRIMARY KEY (student\_id, groups\_id)

);

--Создание представления "UNIVERS" , которое включает в себя поля fio, title, "institut".vuz

create or replace view "UNIVERS" as

select fio, title, "institut".vuz from "student"

join "institut" on student.id = "institut".student\_id

join "groups" on "groups".id ="institut".groups\_id

order by fio;

--Создаём триггерную функцию

CREATE OR REPLACE FUNCTION "ListOfClassesViewр"() RETURNS trigger

AS

$$

declare

id1 int; id2 int;

BEGIN

--Действия функции при использовании операции insert (Вставляем значения во все таблицы, при заполнении "UNIVERS" )

IF ( tg\_op = 'INSERT') THEN

INSERT INTO "student"

VALUES (DEFAULT, new.fio)

ON CONFLICT (fio) DO NOTHING;

INSERT INTO "groups"

VALUES (DEFAULT, new.title)

ON CONFLICT (title) DO NOTHING;

INSERT INTO "institut" (student\_id, groups\_id, vuz )

SELECT (SELECT id

FROM "student"

WHERE fio = new.fio),

(SELECT id

FROM "groups"

WHERE title = new.title),

new.vuz

ON CONFLICT (student\_id, groups\_id)

DO UPDATE SET vuz = new.vuz;

--Действия функции при использовании операции update (Обновляются значения только в представлении )

elsif (tg\_op = 'UPDATE') then

update "student" set fio = new.fio where fio = old.fio;

update "groups" set title = new.title where title = old.title;

select id from "student" where fio=new.fio into id1;

select id from "groups" where title=new.title into id2;

update "institut" set vuz=new.vuz where student\_id =id1 and groups\_id =id2;

--Действия функции при использовании операции delete (Удаляются значения только в представлении )

ELSIF (tg\_op = 'DELETE') THEN

DELETE

FROM "institut"

WHERE student\_id IN (SELECT id FROM "student" WHERE fio = old.fio)

AND groups\_id IN (SELECT id FROM "groups" WHERE title = old.title);

END IF;

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

--Триггер вызываемый при всех операциях

CREATE TRIGGER "ListOfClassesView\_upsert"

INSTEAD OF INSERT OR UPDATE OR DELETE

ON "UNIVERS"

FOR EACH ROW

EXECUTE FUNCTION "ListOfClassesViewр"();

--Тестирование

select \* from "UNIVERS";

--Вставка

insert into "UNIVERS" values ('Юрий','ПМИ', 'ПГУ'), ('Иван','МОВС', 'ПГУ'),('Владимир','ПМИ', 'ПГУ'), ('Семён','КМБ', 'ПНИПУ');

insert into "UNIVERS" values ('Станислав','ЕНФ','ПГГПУ');

--Изменение

update "UNIVERS" set vuz='МГУ' where title='ПМИ';

update "UNIVERS" set fio='Михаил' where fio='Станислав';

--Удаление

delete from "UNIVERS" where fio='Владимир';

delete from "UNIVERS" where title='ПМИ';

drop view "UNIVERS";

drop table "institut" cascade;

drop table "student" cascade;

drop table "groups" cascade;