

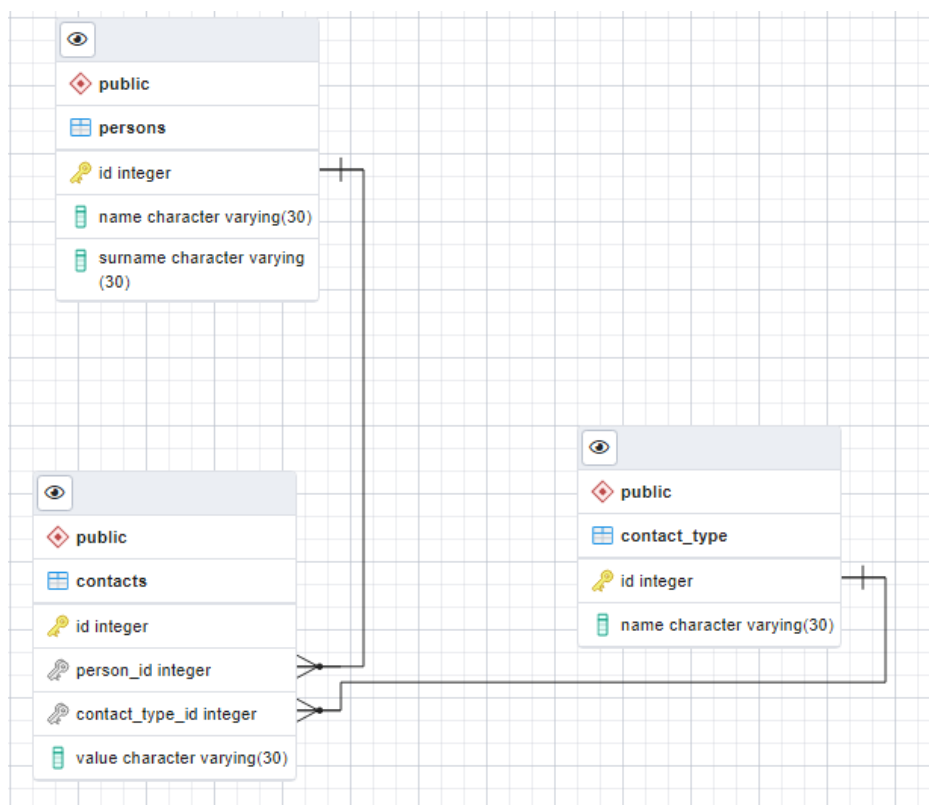
## Задание:

Создать и наполнить базу данных, содержащую связь типа М:М. Связь, помимо внешних ключей, должна иметь дополнительный атрибут. Создать представление, объединяющее таблицы, соединённых связью М:М, но не включающее идентификаторы. Написать триггеры для представления, которые

- при вставке нового отношения вставляют новые записи в исходные таблицы, если они там ранее отсутствовали;
- при обновлении обновляют соответствующие записи в таблице-связке;
- при удалении удаляют соответствующие записи в таблице-связке.

## Решение:

Для реализации лр была создана база данных записной книжки



БД включает в себя три таблицы:

- persons (\*id, name, surname)
- contact\_type (\*id, name)
- Таблица-связка contacts (\*id, person\_id, contact\_type\_id, value)

Столбцы person\_id и contact\_type\_id – внешние ключи для создания связи с таблицами persons и contact\_type.

Столбцы id каждой таблицы – первичные ключи (+ автоинкремент)

### Создание представления

```
CREATE OR REPLACE VIEW my_view AS
```

```
SELECT p.name, p.surname, ct.name as contact_type, c.value
```

```
FROM contacts AS c
```

```
JOIN contact_type AS ct ON c.contact_type_id = ct.id
```

```
JOIN persons AS p ON c.person_id = p.id
```

Т.к. изменять данные в представлении самостоятельно не получится, были написаны триггеры для операций вставки, изменения и удаления. При попытке совершить операцию с данными в представлении, триггеры будут вызывать соответствующие функции для совершения соответствующих операций с соответствующими данными в соответствующих таблицах БД. На основании изменений, внесенных в таблицы, будет изменяться и само представление. В этом и заключается суть лабораторной работы.

Для реализации были созданы 3 триггера и по 1 триггерной функции для каждого триггера (можно было сделать одну общую функцию, мне показалось удобнее сделать по отдельности)

Триггеры буду срабатывать для каждой новой строки, чтобы в случае операции сразу с несколькими строками все они обрабатывались корректно. Для этого воспользуемся ключевым словом **FOR EACH ROW**. Так же, в отличие от лр по триггерам тут мы будем использовать ключевое слово **INSTEAD OF {операция} ON** для того, чтобы вместо запрашиваемой операции у нас вызывалась соответствующая триггерная функция

Для вставки были написаны проверки на наличие записей в соответствующих таблицах. Для удаления и изменения проверки оказались необязательными.

Т.к. все записи в наших таблицах уникальны, то для определения person\_id в таблице contacts используется следующий запрос:

Вставка и обновление:

```
SELECT p.id FROM persons AS p WHERE p.name = new.name AND p.surname = new.surname
```

Удаление:

```
SELECT p.id FROM persons AS p WHERE p.name = OLD.name AND p.surname = OLD.surname
```

Для определения contact\_type\_id в таблице contacts используется следующий запрос:

Вставка и обновление:

```
SELECT ct.id FROM contact_type AS ct WHERE ct.name = new.contact_type
```

Удаление:

```
SELECT ct.id FROM contact_type AS ct WHERE ct.name = OLD.contact_type
```

## Тестирование:

Общий вид таблиц:

persons

	id [PK] integer	name character varying (30)	surname character varying (30)
1	1	Alexandr	Smirnov
2	2	Maksim	Petrov
3	6	testname	testsurname
4	9	Andrey	Semenov

contact\_type

	id [PK] integer	name character varying (30)
1	1	discord
2	2	phone_number
3	3	email
4	4	icq
5	6	index

## Contacts

	id [PK] integer	person_id integer	contact_type_id integer	value character varying (30)
1	1	1	1	alexsmir
2	2	1	2	89824884765
3	3	1	3	smirnov@yandex.ru
4	4	2	2	898434563421
5	5	2	3	maxpetr@mail.ru
6	13	9	6	614066

## View

	name character varying (30)	surname character varying (30)	contact_type character varying (30)	value character varying (30)
1	Alexandr	Smirnov	phone_number	89824884765
2	Alexandr	Smirnov	email	smirnov@yandex.ru
3	Maksim	Petrov	phone_number	898434563421
4	Maksim	Petrov	email	maxpetr@mail.ru
5	Alexandr	Smirnov	discord	alexsmir
6	Andrey	Semenov	index	614066

Попробуем добавить новую строчку в представление:

insert into my\_view values('testname', 'testsurname', 'icq', '64323');

Таблица persons не изменилась т.к. такие ФИ уже есть:

	id [PK] integer	name character varying (30)	surname character varying (30)
1	1	Alexandr	Smirnov
2	2	Maksim	Petrov
3	6	testname	testsurname
4	9	Andrey	Semenov

Таблица contact\_type не изменилась т.к. тип контакта icq уже есть:

	id [PK] integer	name character varying (30)
1	1	discord
2	2	phone_number
3	3	email
4	4	icq
5	6	index

В contacts появилась соответствующая запись с правильными айдишниками

	id [PK] integer	person_id integer	contact_type_id integer	value character varying (30)
1	1	1	1	alexsmir
2	2	1	2	89824884765
3	3	1	3	smirnov@yandex.ru
4	4	2	2	898434563421
5	5	2	3	maxpetr@mail.ru
6	13	9	6	614066
7	14	6	4	64323

В представлении так же появилась соответствующая запись

	name character varying (30)	surname character varying (30)	contact_type character varying (30)	value character varying (30)
1	Alexandr	Smirnov	phone_number	89824884765
2	Alexandr	Smirnov	email	smirnov@yandex.ru
3	Maksim	Petrov	phone_number	898434563421
4	Maksim	Petrov	email	maxpetr@mail.ru
5	Alexandr	Smirnov	discord	alexsmir
6	Andrey	Semenov	index	614066
7	testname	testsurname	icq	64323

Попробуем изменить тип контакста discord на skype:

`update my_view set contact_type = 'skype' where contact_type = 'discord'`

person и contacts не изменились

	id [PK] integer	name character varying (30)	surname character varying (30)
1	1	Alexandr	Smirnov
2	2	Maksim	Petrov
3	6	testname	testsurname
4	9	Andrey	Semenov

	id [PK] integer	person_id integer	contact_type_id integer	value character varying (30)
1	1	1	1	alexsmir
2	2	1	2	89824884765
3	3	1	3	smirnov@yandex.ru
4	4	2	2	898434563421
5	5	2	3	maxpetr@mail.ru
6	13	9	6	614066
7	14	6	4	64323

## contact\_type

	id [PK] integer	name character varying (30)
1	1	skype
2	2	phone_number
3	3	email
4	4	icq
5	6	index

Представление:

	name character varying (30)	surname character varying (30)	contact_type character varying (30)	value character varying (30)
1	Alexandr	Smirnov	phone_number	89824884765
2	Alexandr	Smirnov	email	smirnov@yandex.ru
3	Maksim	Petrov	phone_number	898434563421
4	Maksim	Petrov	email	maxpetr@mail.ru
5	Andrey	Semenov	index	614066
6	testname	testsurname	icq	64323
7	Alexandr	Smirnov	skype	alexsmir

Результат соответствует ожидаемому.

Попробуем удалить запись

```
delete from my_view where name = 'testname' and surname = 'testsurname'
```

Операции удаления затрагивают только таблицу-связку(contacts)

Поэтому persons и contact\_type не меняются

	id [PK] integer	name character varying (30)	surname character varying (30)
1	1	Alexandr	Smirnov
2	2	Maksim	Petrov
3	6	testname	testsurname
4	9	Andrey	Semenov

	id [PK] integer	name character varying (30)
1	1	skype
2	2	phone_number
3	3	email
4	4	icq
5	6	index

## contacts

	id [PK] integer	person_id integer	contact_type_id integer	value character varying (30)
1	1	1	1	alexsmir
2	2	1	2	89824884765
3	3	1	3	smirnov@yandex.ru
4	4	2	2	898434563421
5	5	2	3	maxpetr@mail.ru
6	13	9	6	614066

## Представление

	name character varying (30)	surname character varying (30)	contact_type character varying (30)	value character varying (30)
1	Alexandr	Smirnov	phone_number	89824884765
2	Alexandr	Smirnov	email	smirnov@yandex.ru
3	Maksim	Petrov	phone_number	898434563421
4	Maksim	Petrov	email	maxpetr@mail.ru
5	Andrey	Semenov	index	614066
6	Alexandr	Smirnov	skype	alexsmir

Соответствующие записи удалились из представления и таблицы-связки как и ожидалось.

Вставим несколько записей сразу так, чтобы в каждой таблице появились новые записи:

```
insert into my_view values('test2', 'test2', 'discord', 'test2discord'),
```

```
('test3', 'test3', 'phone_number', '89996665423'),
```

```
('test3', 'test3', 'address', 'Lenina-70');
```

## persons

	id [PK] integer	name character varying (30)	surname character varying (30)
1	1	Alexandr	Smirnov
2	2	Maksim	Petrov
3	6	testname	testsurname
4	9	Andrey	Semenov
5	10	test2	test2
6	11	test3	test3

## contact\_type

	id [PK] integer	name character varying (30)
1	1	skype
2	2	phone_number
3	3	email
4	4	icq
5	6	index
6	7	discord
7	8	address

## contacts

	id [PK] integer	person_id integer	contact_type_id integer	value character varying (30)
1	1	1	1	alexsmir
2	2	1	2	89824884765
3	3	1	3	smirnov@yandex.ru
4	4	2	2	898434563421
5	5	2	3	maxpetr@mail.ru
6	13	9	6	614066
7	15	10	7	test2discord
8	16	11	2	89996665423
9	17	11	8	Lenina-70

## Представление

	name character varying (30)	surname character varying (30)	contact_type character varying (30)	value character varying (30)
1	Alexandr	Smirnov	phone_number	89824884765
2	Alexandr	Smirnov	email	smirnov@yandex.ru
3	Maksim	Petrov	phone_number	898434563421
4	Maksim	Petrov	email	maxpetr@mail.ru
5	Andrey	Semenov	index	614066
6	Alexandr	Smirnov	skype	alexsmir
7	test2	test2	discord	test2discord
8	test3	test3	phone_number	89996665423
9	test3	test3	address	Lenina-70

Все тесты отработывают корректно



## Скрипт:

--создание view

CREATE OR REPLACE VIEW my\_view AS

SELECT p.name, p.surname, ct.name as contact\_type, c.value

FROM contacts AS c

JOIN contact\_type AS ct ON c.contact\_type\_id = ct.id

JOIN persons AS p ON c.person\_id = p.id

--триггерная функция для вставки

CREATE OR REPLACE FUNCTION trigger\_function\_insert()

RETURNS trigger AS

\$\$

BEGIN

--new: name, surname, contact\_type, value

--если нет человека с ФИ - вставляем добавляем запись в persons

IF NOT EXISTS (SELECT p.name, p.surname FROM persons AS p WHERE p.name = new.name AND p.surname = new.surname)

THEN

INSERT INTO persons VALUES(DEFAULT, new.name, new.surname);

END IF;

--если нет такого типа контакта, то добавляем запись в contact\_type

IF NOT EXISTS (SELECT ct.name FROM contact\_type AS ct WHERE ct.name = new.contact\_type)

THEN

INSERT INTO contact\_type VALUES(DEFAULT, new.contact\_type);

END IF;

--данные о типе контакта и человеке уже гарантированно будут. Добавляем запись в contacts, с совпадающими айдишниками в соотв. таблицах

IF NOT EXISTS (SELECT c.value FROM contacts AS c WHERE c.value = new.value)

THEN

INSERT INTO contacts VALUES(

```

        DEFAULT,
        (SELECT p.id FROM persons AS p WHERE p.name = new.name AND
p.surname = new.surname),
        (SELECT ct.id FROM contact_type AS ct WHERE ct.name =
new.contact_type),
        new.value
    );
END IF;
RETURN NULL;
END;
$$ LANGUAGE plpgsql;

```

--триггерная функция для обновления

```

CREATE OR REPLACE FUNCTION trigger_function_update()
RETURNS trigger AS
$$
BEGIN
    update persons
        set name = new.name, surname = new.surname
        where name = old.name and name = old.surname;

    UPDATE contact_type
        SET name = new.contact_type
        WHERE name = old.contact_type;

    UPDATE contacts
        SET value = new.value
        WHERE value = old.value
        AND (SELECT p.id FROM persons AS p WHERE p.name = new.name AND p.surname =
new.surname) = person_id
        AND (SELECT ct.id FROM contact_type AS ct WHERE ct.name = new.contact_type) =
contact_type_id;

```

```

RETURN NULL;
END;
$$ LANGUAGE plpgsql;
```

--триггерная функция для удаления

```
CREATE OR REPLACE FUNCTION trigger_function_delete()
```

```
RETURNS trigger AS
```

```
$$
```

```
DECLARE
```

```
BEGIN
```

```
DELETE FROM contacts
```

```
where person_id = (SELECT p.id FROM persons AS p WHERE p.name = OLD.name AND
p.surname = OLD.surname)
```

```
AND contact_type_id = (SELECT ct.id FROM contact_type AS ct WHERE ct.name =
OLD.contact_type)
```

```
AND value = OLD.value;
```

```
RETURN NULL;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

--триггер вставки

```
CREATE TRIGGER view_insert
```

```
INSTEAD OF INSERT ON my_view
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION trigger_function_insert();
```

--триггер обновления

```
CREATE TRIGGER view_update
```

INSTEAD OF UPDATE ON my\_view

FOR EACH ROW

EXECUTE FUNCTION trigger\_function\_update();

--триггер удаления

CREATE TRIGGER view\_delete

INSTEAD OF DELETE ON my\_view

FOR EACH ROW

EXECUTE FUNCTION trigger\_function\_delete();

insert into my\_view values('testname', 'testsurname', 'icq', '64323');

update my\_view set contact\_type = 'skype' where contact\_type = 'discord'

delete from my\_view where name = 'testname' and surname = 'testsurname'

insert into my\_view values('test2', 'test2', 'discord', 'test2discord'),

('test3', 'test3', 'phone\_number', '89996665423'),

('test3', 'test3', 'address', 'Lenina-70');

delete from my\_view where surname = 'test2' or surname = 'test3'

update my\_view set contact\_type = 'gggg' where contact\_type = 'hhhh'

update my\_view

set name = format('%s\_changed', name)

where surname like 'test%'

insert into my\_view values('Andrey', 'Semenov', 'index', '614066');