**ФПИиКТ**

**Отчет по дисциплине ИСиБД**

**Лабораторной работе №1**

**Вариант №179**

Ибадуллаев Алибаба Эльбрус оглы

Группа: P33131

Преподаватель: Байрамова Хумай Бахруз

Санкт-Петербург

2022г.

# Задание

Для выполнения лабораторной работы №1 необходимо:

1. На основе предложенной предметной области (текста) составить ее описание. Из полученного описания выделить сущности, их атрибуты и связи.
2. Составить инфологическую модель.
3. Составить даталогическую модель. При описании типов данных для атрибутов должны использоваться типы из СУБД PostgreSQL.
4. Реализовать даталогическую модель в PostgreSQL. При описании и реализации даталогической модели должны учитываться ограничения целостности, которые характерны для полученной предметной области.
5. Заполнить созданные таблицы тестовыми данными.

# Описание предметной области

## Текст

Проведя несколько месяцев в изолированной группе, человек становится необычайно чувствителен к настроениям остальных. Что-то в отношении окружающих к Флойду изменилось; вновь всплыло даже обращение "доктор Флойд", которого он не слышал так давно, что отвык на него откликаться.

## Мое понимание

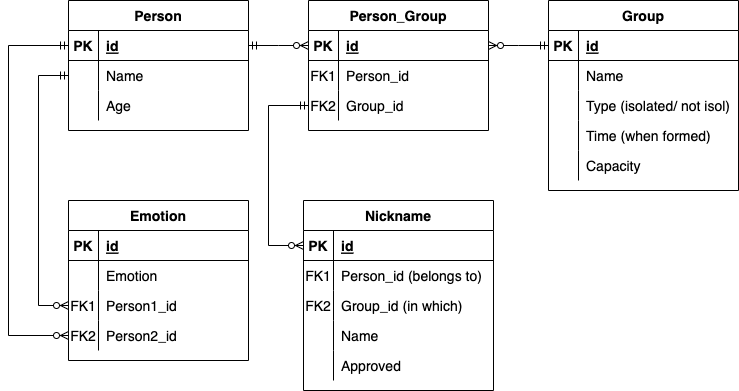
**Человек** может состоять в **группах**, которые могут быть изолированнными или не изолированными. У каждого же человека есть **эмоции** к какому либо человеку. В каждой же группе могут быть **клички** которые присваиваются людям внутри нее.

Человек, группа – стержневые таблицы

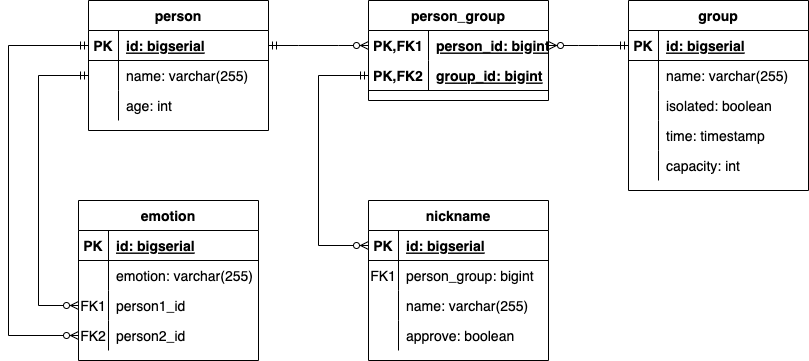
клички – характеристики

Человек\_Группа, Эмоции – ассоциация

# Инфологическая модель



# Даталогическая модель



# Реализация

### Создание таблиц

create table persons(  
 id bigserial primary key,  
 name varchar(255) not null,  
 surname varchar(255) not null,  
 age int not null  
);  
  
create table groups(  
 id bigserial primary key,  
 name varchar(255) not null,  
 isolated boolean not null,  
 created timestamp not null,  
 capacity int not null  
);  
  
create table persons\_groups(  
 person\_id bigint not null references persons(id) on delete cascade,  
 group\_id bigint not null references groups(id) on delete cascade,  
 primary key (person\_id, group\_id),

joined timestamp

);  
  
create table emotions(  
 id bigserial primary key,  
 person1\_id bigint not null references persons(id) on delete cascade,  
 person2\_id bigint not null references persons(id) on delete cascade,  
 emotion varchar(255) not null  
);  
  
create table nickname(  
 id bigserial primary key,  
 person\_id bigint,  
 group\_id bigint,  
 FOREIGN KEY (person\_id, group\_id) references persons\_groups(person\_id, group\_id) on delete cascade,  
 nickname varchar(255) not null,  
 approve boolean not null  
);

### Вставка данных

INSERT INTO s312200.persons (name, surname, age) VALUES ( 'Kick', 'Butovskiy', 45);  
INSERT INTO s312200.persons (name, surname, age) VALUES ('Floyd', 'Vahovskiy', 33);  
INSERT INTO s312200.persons (name, surname, age) VALUES ('Pet', 'Losenko', 55);  
INSERT INTO s312200.persons (name, surname, age) VALUES ('John', 'Sport', 42);  
INSERT INTO s312200.persons (name, surname, age) VALUES ('Michel', 'Maywather', 38);  
  
INSERT INTO s312200.groups (name, isolated, created, capacity) VALUES ('Professors', true, '2021-11-07 06:02:45.772000', 5);  
INSERT INTO s312200.groups (name, isolated, created, capacity) VALUES ('Sportsmans', false, '2021-11-01 06:02:45.772000', 10);  
  
INSERT INTO s312200.persons\_groups (person\_id, group\_id, joined) VALUES (2, 1, '2022-11-05 06:02:45.962181');  
INSERT INTO s312200.persons\_groups (person\_id, group\_id, joined) VALUES (3, 1, '2022-10-24 06:02:45.962181');  
INSERT INTO s312200.persons\_groups (person\_id, group\_id, joined) VALUES (1, 1, '2022-11-04 06:02:45.962181');  
INSERT INTO s312200.persons\_groups (person\_id, group\_id, joined) VALUES (1, 2, '2022-11-05 06:02:45.962181');  
INSERT INTO s312200.persons\_groups (person\_id, group\_id, joined) VALUES (4, 2, '2022-11-07 06:02:45.962000');  
INSERT INTO s312200.persons\_groups (person\_id, group\_id, joined) VALUES (4, 1, '2022-11-02 12:40:42.000000');  
  
INSERT INTO s312200.nickname (person\_id, group\_id, nickname, approve) VALUES (2, 1, 'Doctor', false);  
INSERT INTO s312200.nickname (person\_id, group\_id, nickname, approve) VALUES (4, 2, 'Kachok', true);  
INSERT INTO s312200.nickname (person\_id, group\_id, nickname, approve) VALUES (4, 1, 'NULLER', true);  
  
INSERT INTO s312200.emotions (person1\_id, person2\_id, emotion) VALUES (2, 3, 'joyful');  
INSERT INTO s312200.emotions (person1\_id, person2\_id, emotion) VALUES (2, 4, 'amused');  
INSERT INTO s312200.emotions (person1\_id, person2\_id, emotion) VALUES (2, 1, 'unsure');  
INSERT INTO s312200.emotions (person1\_id, person2\_id, emotion) VALUES (2, 5, 'secure');

## Запрос

Вывести самого младшего человека у которого больше всех никнеймов, а в последнюю группу он вступил не раньше недели назад. При этом, чтобы эмоций на других людей у него было 4 с названием эмоции из 6 символов

DO  
$$  
 DECLARE  
 count\_persons numeric:= (select count(\*) from persons);  
 min numeric := 0;  
 curr\_id bigint := 0;  
 curr\_emotion text := '';  
 flag\_emotions bool := true;  
 BEGIN  
 FOR i IN 1..count\_persons LOOP  
 curr\_id = (select id from persons order by age desc limit 1 offset count\_persons - i);  
 raise NOTICE 'number of nicknames for id - %: %', curr\_id, (select count(\*) from nickname where person\_id = curr\_id);  
**-- here find person with largest number of nicknames**  
 if ((select count(\*) from nickname where person\_id = curr\_id) > (select count(\*) from nickname where nickname.person\_id = min)) then  
**-- here check that the time of last joined group that lower than one week**  
 if (  
 select ((current\_timestamp -  
 (select MAX(joined) from persons\_groups where person\_id = 2 )  
 ) < interval '7 days') as joined\_not\_before\_one\_week  
 ) is true then  
**-- here check that person has 4 emotions the length of which is 6 char**  
 if (  
 (select count(\*) from emotions where person1\_id = curr\_id) = 4  
 ) then  
 for i in 0..3 loop  
 curr\_emotion =(select emotion from emotions where person1\_id = curr\_id limit 1 offset i);  
 raise notice 'emotion: %, %', curr\_emotion, length(curr\_emotion);  
 if ( length(curr\_emotion) != 6) then  
 flag\_emotions = false ;  
 continue;  
 end if;  
 end loop;  
 if (flag\_emotions) is true then  
 min = curr\_id;  
 end if;  
 flag\_emotions = true;  
 end if;  
 end if;  
 end if;  
 END LOOP;  
 raise warning  
 '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Youngest person  
 with largest number of nicknames  
 with time of last joined group that lower than one week.  
 which has 4 emotions the length of which is 6 char  
 Is --> %', (select (name) from persons where id = min);  
 END  
$$;

## Ответ на запрос

[2022-11-08 13:20:39] [00000] number of nicknames for id - 2: 1

[2022-11-08 13:20:39] [00000] emotion: joyful, 6

[2022-11-08 13:20:39] [00000] emotion: amused, 6

[2022-11-08 13:20:39] [00000] emotion: secure, 6

[2022-11-08 13:20:39] [00000] emotion: unsure, 6

[2022-11-08 13:20:39] [00000] number of nicknames for id - 5: 0

[2022-11-08 13:20:39] [00000] number of nicknames for id - 4: 2

[2022-11-08 13:20:39] [00000] number of nicknames for id - 1: 0

[2022-11-08 13:20:39] [00000] number of nicknames for id - 3: 0

[2022-11-08 13:20:39] [01000] \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[2022-11-08 13:20:39] Youngest person

[2022-11-08 13:20:39] with largest number of nicknames

[2022-11-08 13:20:39] with time of last joined group that lower than one week.

[2022-11-08 13:20:39] which has 4 emotions the length of which is 6 char

[2022-11-08 13:20:39] Is --> **Floyd**