**ФПИиКТ**

**Отчет по дисциплине ИСиБД**

**Лабораторной работе №2**

**Вариант №179**

Ибадуллаев Алибаба Эльбрус оглы

Группа: P33131

Преподаватель: Байрамова Хумай Бахруз кызы

Санкт-Петербург

2022г.

# Задание

Для отношений, полученных при построении предметной области из

лабораторной работы №1, выполните следующие действия:

• опишите функциональные зависимости для отношений полученной

схемы (минимальное множество);

• приведите отношения в 3NF (как минимум). Постройте схему на основе

полученных отношений;

• опишите изменения в функциональных зависимостях, произошедшие

после преобразования в 3NF

• какие денормализации будут полезны для вашей схемы? Приведите

подробное описание;

# До нормализации

## Функц. зависимости

Person:

Id -> name

Id -> age

Nickname: (каждому человеку в группе соответствует только единый никнейм =>

Запись из человек\_группа может являтся составным суррогатным ключом)

Id -> person\_id

Id -> group\_id

Id -> name

Id -> approve

**{group\_id, person\_id}** -> id

**{group\_id, person\_id}** -> name

**{group\_id, person\_id}** -> approve

Group: (Название группы может одназначно быть определено по типу, в один и тот же момент может быть создана только одна группа. Если время создание группы совпадает с временем распостронения короновируса то группа является изолированной)

Id -> name

Id -> type

Id -> time

Id -> capacity

time -> type

{name, time } -> id

{name, time } -> type

{name, time } -> capacity

Emotion: (каждый человек может иметь неограниченное количество эмоций по отношению к другому человеку(возможны одинаковые пары))

Id -> emotion

Id -> person1

Id -> person2

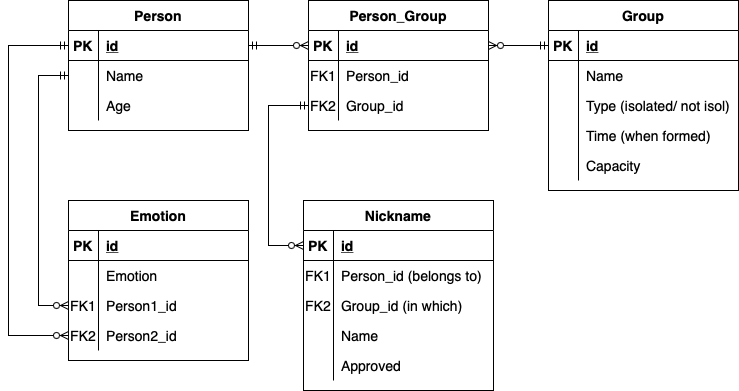
Person\_group: (каждый человек может состоять во множестве групп)

id -> person\_id

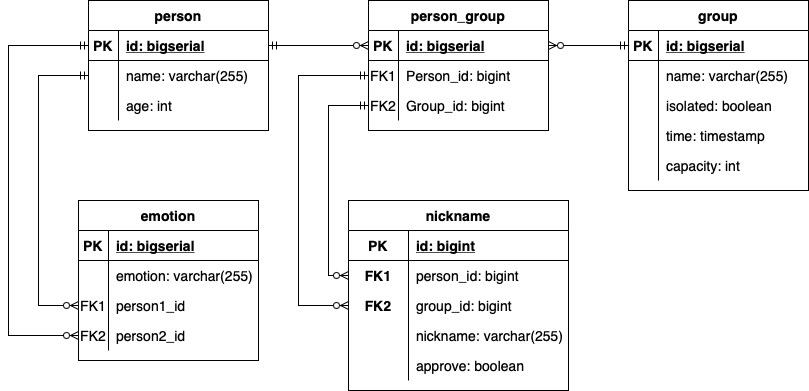
id -> group\_id

## Схема

Инфологическая



Даталогическая



# После нормализации

## Функц. зависимости

Person:

id -> name

id -> age

Nickname:

{group\_id, person\_id} -> name

{group\_id, person\_id} -> approve

Group:

Id -> name

Id -> type

Id -> time

id -> capacity

{name, time} -> id

{name, time} -> capacity

Group\_type:

time -> type

Emotion:

id -> emotion

id -> person1

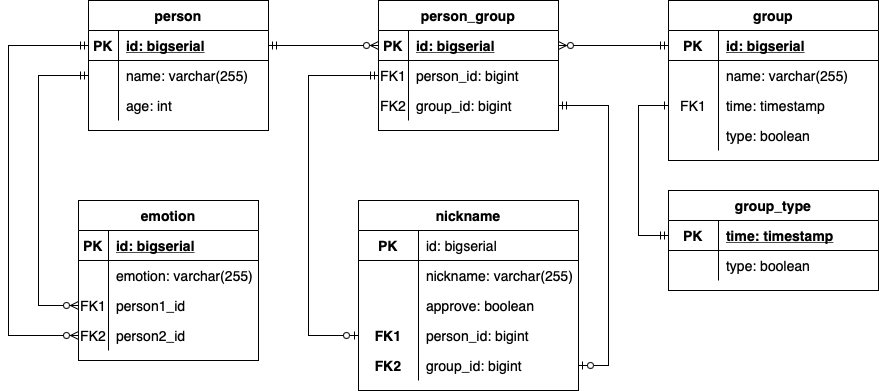
id -> person2

Person\_group

id -> person\_id

id -> group\_id

## Схема



## Изменения

1) В группе после приведения к 1 вынес атрибут type в отдельную сущность из-за частичной зависимости от составного ключа(время создание и имени)

**Отношения находятся в 3НФ.**

* Требования первой нормальной формы выполняются: нет повторяющихся значений в таблице.
* Требования второй нормальной формы выполняются, так как все неключевые атрибуты входят в какой-то из потенциальных ключей.
* Требования третьей нормальной формы выполняются: неключевые атрибуты зависят нетранзитивно от первичного ключа.

## Денормализация

Композиция group\_type и group повысила бы производительность работы с данной сущностью. Также избавились бы от избыточной связанности.

## Вывод

Я познакомился и применил на практике понятия функциональных зависимостей, а также поработал надо нормализацией таблицы и провел анализ того как можно было денормализовать наши сущности.

# explained

Person:

id -> name

id -> age

Nickname: (каждому человеку в группе соответствует только единый никнейм =>

Запись из человек\_группа может являтся составным суррогатным ключом)

~~id -> person\_id (композиция)~~

~~id -> group\_id (композиция)~~

id -> {group\_id, person\_id}

~~id -> name (транзитивная зависимость)~~

~~id -> approve (транзитивная зависимость)~~

**{group\_id, person\_id}** -> name

**{group\_id, person\_id}** -> approve

Group: (Название группы может одназначно быть определено по типу, в один и тот же момент может быть создана только одна группа. Если время создание группы совпадает с временем распостронения короновируса то группа является изолированной)

id -> name

id -> time

~~id -> type (транзитивная зависимость)~~

~~id -> capacity (транзитивная зависимость)~~

time -> type (делаем отдельной сущностью из-за не зависимости от составного первичного ключа)

{name, time} -> id

~~{name, time} -> type (из за избыточночти)~~

{name, time} -> capacity

Group\_type:

time -> type

Emotion: (каждый человек может иметь неограниченное количество эмоций по отношению к другому человеку(возможны одинаковые пары))

id -> emotion

id -> person1

id -> person2

Person\_group: (каждый человек может состоять во множестве групп)

id -> person\_id

id -> group\_id

# Было раньше (после нормализации)

Person:

Id -> name

Id -> age

Nickname:

~~Id -> group~~

~~Id -> name~~

**{group, name}** -> person

**{group, name}** -> approve

Group:

~~Id -> name~~

**name** -> isolated

**name** -> time

**name** -> capacity

Emotion:

Id -> emotion

Id -> person1

Id -> person2

**\***

**Person\_nicknames:**

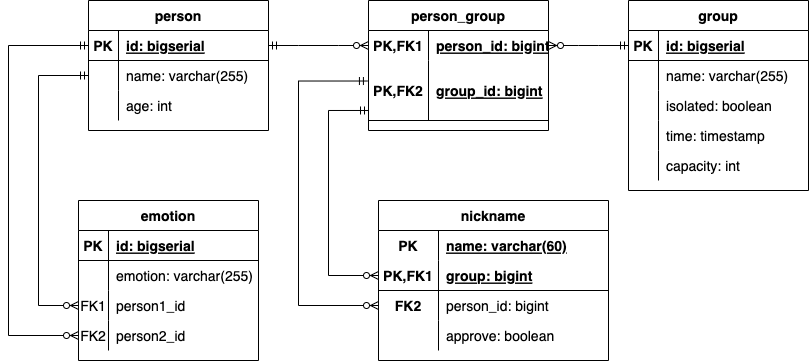
**{person\_id, group\_id} -> nickname**

**{person\_id, group\_id} -> approved**

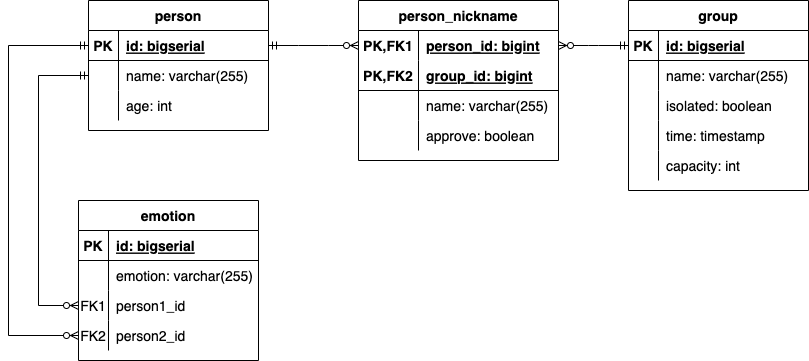
**{person\_id, nickname, group\_id} -> {person\_id, group\_id}**

**Тогда избавляемся от таблицы nickname**

## Схема



Альтернативный вариант



## Изменения

1) Потенциальный ключ **name** в таблице **group** выведен как первичный ключ (так как группы не могут иметь одинаковые названия)

Внутри таблицы **nickname** потенциальный ключ **name** и внешний ключ **group** составили составной первичный ключ (так как в группах могут быть только уникальные клички (**не могут повторяться внутри одной и той же группы!**)). Тем самым нормализовали до 3 НФ.

Избавились от лишних суррогатный первичных ключей, вместо них используются потенциальные. (Уменьшили занимаемую память в двумя таблицах (2\*8 байта\*n))

n – количество записей в таблице

\* 2) Заметил, что можно сделать композицию двух таблиц nickname и person\_group (это возможно, так как внутри одной группы не могут повторяться клички)