Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

*Факультет Программной инженерии и компьютерной техники*

Дисциплина: Базы данных

**Лабораторная работа №3**

Вариант «Табличный»

Группа: P3132

Выполнил: Волков Григорий Алексеевич

Преподаватель: Харитонова Анастасия Евгеньевна

г. Санкт-Петербург

2023 г.

Оглавление

[Задание 3](#_Toc133171810)

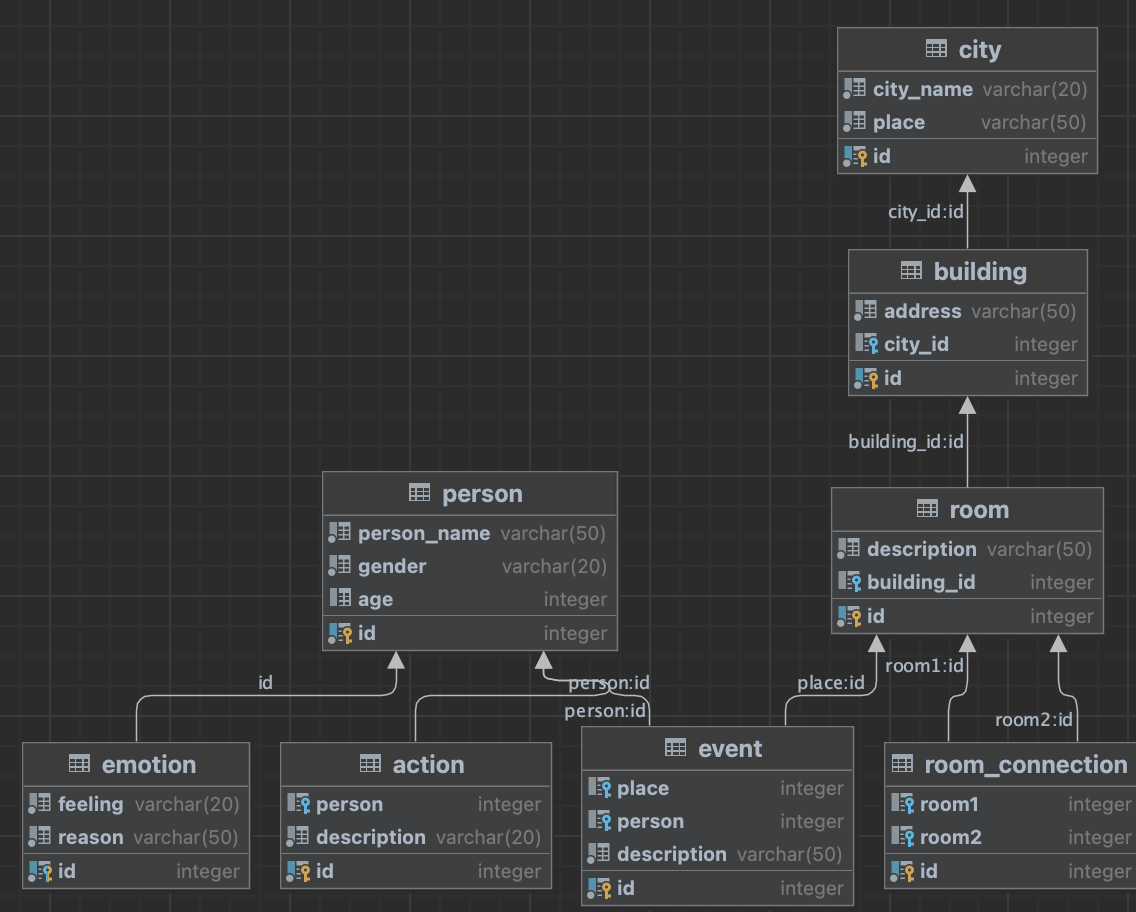
[Выполнение работы 5](#_Toc133171811)

# **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 NF(какминимум).Постройте схему на основе полученных отношений;
* опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе NF;
* преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF;
* какие денормализации будут полезны для вашей схемы? Приведите подробное описание;

Написать для данной базы данных логирующие триггеры для любой таблицы на выбор. Логи должны храниться в отдельной таблице. Тип триггера – табличный. В лог-таблице должно храниться время, тип crud-операции и первичный ключ строки, где произошло какое-то событие



# **Выполнение работы**

1. **Функциональные зависимости**

person: id (person\_name, gender, age)

emotion: id (feeling, reason)

action: id (person, description)

event: id (place, person, description)

city: id (city\_name, place)

building: id (address, city\_id)

room: id (description, building\_id)

room\_connection: id (room1, room2)

1. **Привести отношение в 3NF**

**1NF**: Отношение находится в 1NF, если в таблице нет дублирующийся строк, в каждой ячейки находится атомарное значение, данные в столбце одного типа, отсутствуют массивы и списки. Моя модель находится в отношении 1NF, так как нет дубликатов строк, массивов, а данные в столбцах одного типа и в ячейках находятся только атомарные значения

**2NF**: Отношение находится в 2NF, если оно уже находится в 1NF, присутствуют ключи и всего не ключевые атрибуты полностью функционально зависят от первичного ключа. Моя база находится в отношении 2NF, так как она находится уже в 1NF, присутствуют ключи, по которым можно идентифицировать строку, а составных ключей в базе данных нет

**3NF**: Отношение находится в 3NF, если оно уже находится в 2NF и отсутствуют транзитивные зависимости. Моя модель удовлетворяет данным условиям, так как все не ключевые атрибуты зависят только от первичных ключей и не содержит транзитивных взаимосвязей

Следовательно моя модель изначально удовлетворяла условиям отношения 3NF, вносить изменения не потребовалось, схема остается та же, как и функциональные зависимости

1. **Преобразуем отношения в BCNF**

Так как в моей модели отсутствуют составные первичные ключи, то моя база автоматически приведена в нормальную форму Бойса-Кодда.

1. **Какие денормализации могут быть полезны для таблицы**

Объединение таблиц: иногда объединение таблиц может сильно повысить эффективность базы данных, а именно уменьшить количество довольно ресурсоемких операций в виде JOIN. Так, например можно объединить таблицы action и event, так как таблицы содержат информацию, по смыслу тесно связанную между собой.

1. **Написать логирующие триггеры для любой таблицы на выбор**

Я выбрал таблицу person и написал один триггер, который вызывается при внесении данных в таблицу или их изменении

**Код**

create table if not exists person\_log\_table(  
 id serial primary key,  
 date varchar(20),  
 type text  
);  
create or replace function *person\_trigger\_function*()  
returns trigger as $$  
 begin  
 if (tg\_op = 'INSERT') then  
 insert into person\_log\_table(type, date) values (tg\_op, *current\_date*);  
 return new;  
 elsif (tg\_op = 'UPDATE') then  
 insert into person\_log\_table(type, date) values (tg\_op, *current\_date*);  
 return new;  
 end if;  
 end  
 $$  
 language plpgsql;  
  
create or replace trigger person\_log\_table  
 after insert or update on person execute procedure *person\_trigger\_function*();