**Федеральное государственное автономное образовательное учреждение высшего образования Университет ИТМО**

**Факультет программной инженерии и компьютерных технологий**

# «Информационные системы и базы данных»

### Отчёт по Курсовой работе

**Этап 2**

**Управление умным домом**

**Студенты:**

Ляо Ихун

Скакун Артем Андреевич

**Гр. P33131**

**Преподаватель:**

Байрамова Хумай Бахруз Кызы

**Этап 2**

# Текст задачи:

Нарисовать ER-диаграмму предметной области. ER-модель должна соответствовать описанию, представленному в рамках первого этапа курсовой работы.

На основе ER-модели построить даталогическую модель.

# Описание предметной области

На текущий момент, на рынке достаточно много приложений для управления умным домом. У всех компанией, которые создают умные дома, есть свои приложения для своих продуктов. Их процессы работы по сути похожи. Они позволяют пользователям самостоятельно модернизировать и создавать новые сценарии, которые будут удовлетворять потребности клиента. Например, увлажнитель начинает работать, если влажность воздуха ниже нормы или включается обогрев, когда температура опускается ниже заданной.

Но тут есть следующие проблемы. Во-первых, написание сценария может быть достаточно сложным, и пользователям трудно его создать, особенно, когда в доме достаточно много разнообразной умной мебели, и ее взаимодействие между собой сложное. Во-вторых, в существующих приложениях нет возможности делиться сценариями. Поэтому мы хотим создать такое приложение, которое поддерживает общие сценарии и поставляется с готовыми, созданными программистами, чтобы помочь пользователям комфортно и эффективно управлять своим умным домом.

# Бизнес-Процессы

1. Регистрация пользователя.
2. Авторизация пользователя.
3. Пользователь добавляет себя в дом в приложение. Пользователь добавляет дом по его адресу или просто присоединяется к другому дому.
4. Добавление комнат в дом. Пользователь добавляет комнаты в квартиру, и данные о них
5. Добавление умных вещей в помещение. Пользователь добавляет умные вещи в комнату.
6. Создание сценария по условию - умная вещь начинает работать, если выполняется какое-то условие (например, увлажнитель начинает работать, если влажность воздуха ниже нормы). Сценарий создан пользователем или программистом.
7. Создание сценария по расписанию - работа умных вещей по заданному расписанию. Сценарий создан пользователем или программистом.
8. Пользователь может поделиться сценариями с другими пользователями.
9. Пользователь управляет состоянием вещей при помощи сценариев.
10. Пользователь управляет состоянием вещей вручную.
11. Пользователь сообщают о проблемах, служба поддержки их рашают.

# Описание предментной области:

|  |  |
| --- | --- |
| **Сушность** | **Классификация** |
| Script | Стержневая |
| House | Стержневая |
| Room | Стержневая |
| Furniture | Стержневая |
| Problem | Стержневая |
| Support\_man | Стержневая |
| Action | Стержневая |
| Contact | Характеристика |
| Address | Характеристика |
| User | Стержневая |
| Condition\_script | Характеристика |
| Schedule\_script | Характеристика |
| List\_Script\_User | Ассоциативная |
| List\_Action\_script | Ассоциативная |
| List\_User\_House | Ассоциативная |

## Стрежневая:

1. House
   1. id serial
   2. address text
   3. type\_house enum
2. Room
   1. id serial
   2. house\_id int
   3. area float
   4. height int
   5. Is\_filled boolean
   6. Type enum
3. Furniture
   1. id serial
   2. room\_id int
   3. type enum
   4. manufacture varchar(258)
   5. Available boolean
4. Script
   1. id serial
   2. Creator\_name varchar(64)
   3. Type enum
5. Action
   1. id serial
   2. Type\_furniture enum
   3. type enum
   4. description text
6. Support\_man
   1. Id serial
   2. name varchar(64)
   3. password varchar(256)
   4. is\_free boolean
7. Problem
   1. id serial
   2. description text
   3. user\_id int
   4. type enum
   5. support\_id int
   6. is\_finished boolean
8. User
   1. id int
   2. password varchar(256)
   3. Name varchar(64)
   4. Age int
   5. Address\_id int

## Характеристика:

1.Schedule\_script

* 1. script\_id int
  2. start\_time time
  3. end\_time time

1. Condition\_Script
   1. script\_id int
   2. condition text
2. Contact
   1. User\_id int
   2. Email varchar(128)
   3. Phone varchar(64)
3. Address
   1. id serial
   2. Contry enum
   3. City enum
   4. Street varchar(64)

## Ассоциативная:

1. List\_Action\_script
   1. id serial
   2. script\_id int
   3. action\_id int
2. List\_Script\_User
   1. id serial
   2. user\_id int
   3. script\_id int
3. List\_User\_house
   1. id serial
   2. user\_id int
   3. house\_id int

# Функциональные зависимости

## 1. User

id -> password

id -> name

id -> gender

id -> age

## 2.Contact

User\_id -> phone\_number

User\_id -> email

phone\_number -> user\_id

Email -> user\_id

## 3. Problem

Id->description

Id-> user\_id

Id-> type

Id-> is\_finished

Id-> support\_id

1. Before Insert:

Если у support\_man 4 проблем, состояние is\_finished которых true, то состояние is\_free support\_man становится false.Создая новую проблему, если у support\_man состояние is\_finished является false, то дальше создание не будет успешно.

1. After Update:

состояние is\_finish проблемы на true. Одновременно, свое состояние is\_free становится true, если его состояние было false.

Если мы хотим перевести проблему к дргому support\_man, то тоже нужно проверять состояния его и первого человека и делать соотвественное измнение и на превого человека, и на второго.

1. Before Update:

Если хотим перевести проблему к занятом. То от такого обновления отказывается.

## 4. House

Id->address\_id

Id->type\_house

Address\_id -> id

## 5.Room

Id->area

Id->house\_id

Id->height

Id->is\_filled

Id -> type

## 6. Furniture

Id->room\_id

Id->type

Id->manufacture

Id->available

1. Before Insert:

Количество мебелей в комнате не больше 20. Также если комната уже заполнена мебели. То новая мебель не добавляется.

## 7.Action

Id->furniture\_id

Id->type

Id->Description

-

## 8. Support\_man

Id -> name

Id->password

Id->is\_free

## 9. Script

Id->creator\_name

Id->type

-

## 10. Schedule\_script

Script\_id -> start\_time

Script\_id -> end\_time

-

## 11. Condition\_Script

Script\_id -> condition

-

## 12. List\_Script\_User

Id -> user\_id

Id -> script\_id

-

## 13. List\_User\_House

Id -> user\_id

Id -> house\_id

-

## 14. List\_Action\_script

Id -> script\_id

Id -> action\_id

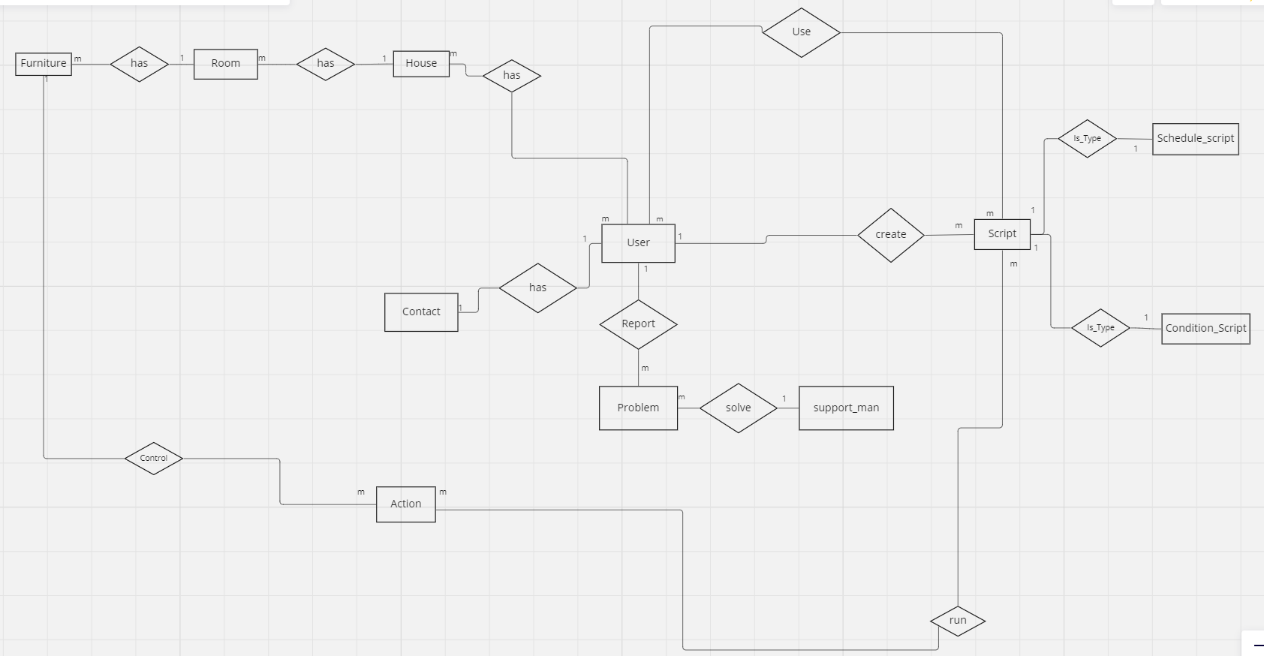
## Address

Id->contry

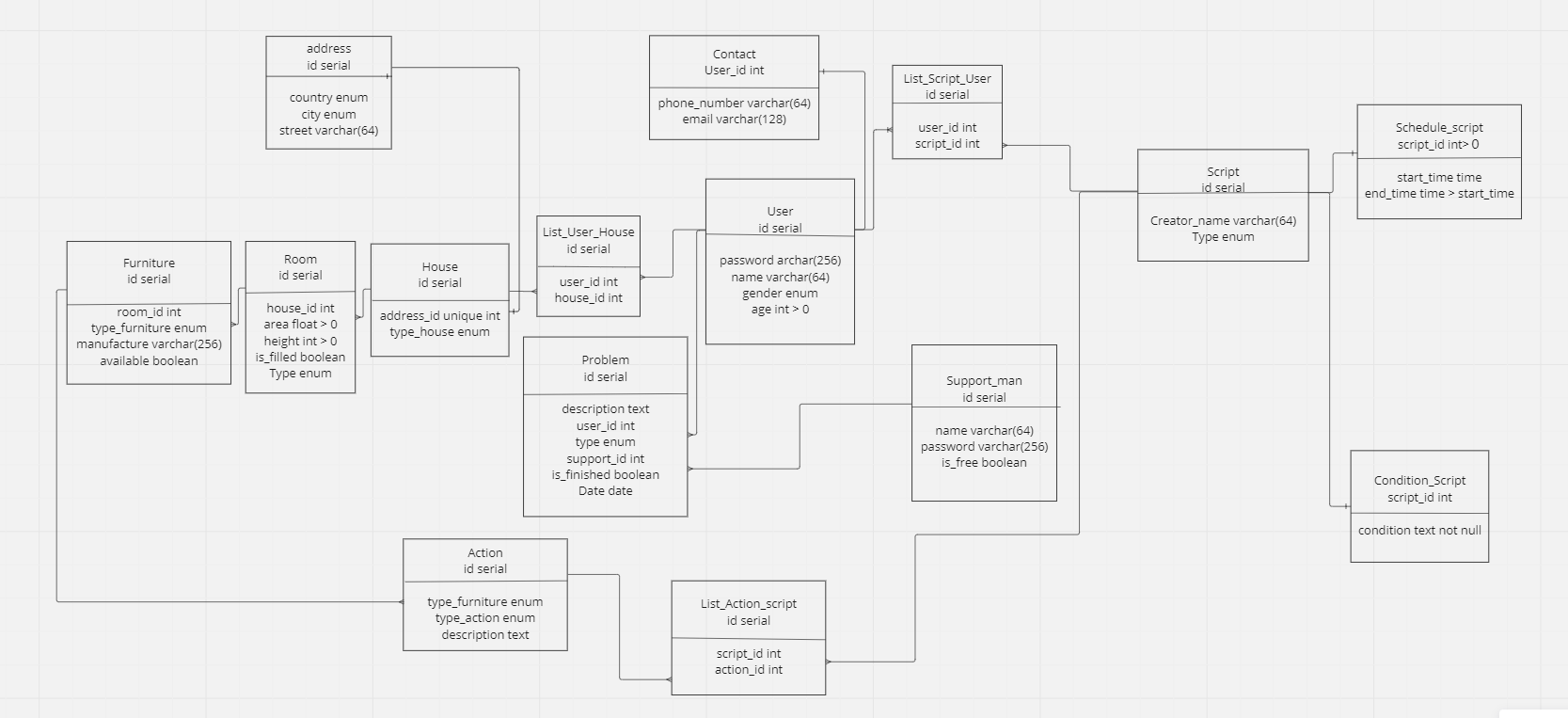
Id->city

Id->street

# Инфологическая модель



# Даталогическую модель



**Вывод:**

Все таблицы нормальзованы до 3 НФ кроме Address. Это было сделано умышлено, чтобы повысить производительность. У остальных таблиц все атомарные атрибуты, нет частичных зависимостей или транзитивных зависимостей от первичного ключа.