

**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. Р33131

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 компаний, 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 предметной области:

Сущность	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

Стержневая:

1. House

- a. id serial
- b. address text
- c. type_house enum

2. Room

- a. id serial
- b. house_id int
- c. area float
- d. height int
- e. Is_filled boolean

- f. Type enum

3. Furniture

- a. id serial
- b. room_id int
- c. type enum
- d. manufacture varchar(258)
- e. Available boolean

4. Script

- a. id serial
- b. Creator_name varchar(64)
- c. Type enum

5. Action

- a. id serial
- b. Type_furniture enum
- c. type enum
- d. description text

6. Support_man

- a. Id serial
- b. name varchar(64)
- c. password varchar(256)
- d. is_free boolean

7. Problem

- a. id serial
- b. description text
- c. user_id int
- d. type enum
- e. support_id int
- f. is_finished boolean

8. User

- a. id int
- b. password varchar(256)
- c. Name varchar(64)
- d. Age int
- e. Address_id int

Характеристика:

1. Schedule_script
 - f. script_id int
 - g. start_time time
 - h. end_time time
2. Condition_Script
 - a. script_id int
 - b. condition text
3. Contact
 - a. User_id int
 - b. Email varchar(128)
 - c. Phone varchar(64)
4. Address
 - a. id serial
 - b. Contry enum
 - c. City enum
 - d. Street varchar(64)

Ассоциативная:

1. List_Action_script
 - a. id serial
 - b. script_id int
 - c. action_id int
2. List_Script_User
 - a. id serial
 - b. user_id int
 - c. script_id int
3. List_User_house
 - a. id serial
 - b. user_id int
 - c. 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. Создавая новую проблему, если у support_man состояние is_finished является false, то дальше создание не будет успешно.

2) 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 другому support_man, то тоже нужно проверять состояния его и первого человека и делать соответственное изменение и на первого человека, и на второго.

3) Before Update:

Если хотим перевести проблему к занятому. 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 мебели 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

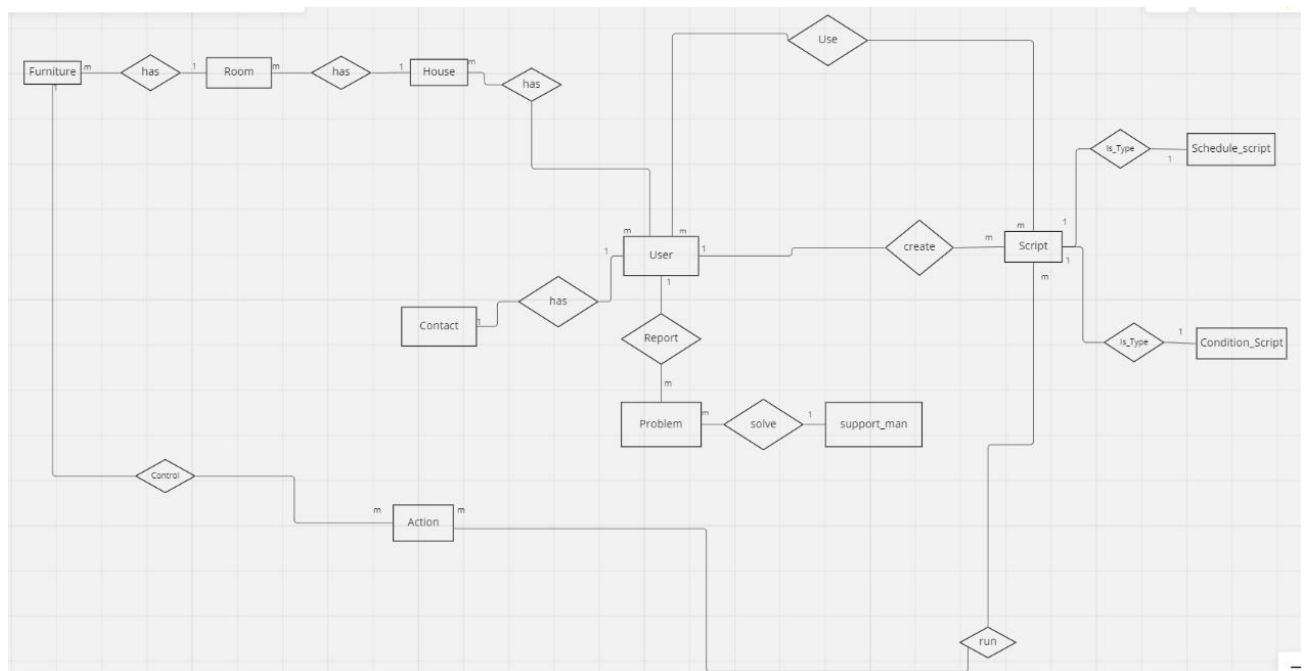
15. Address

Id->contry

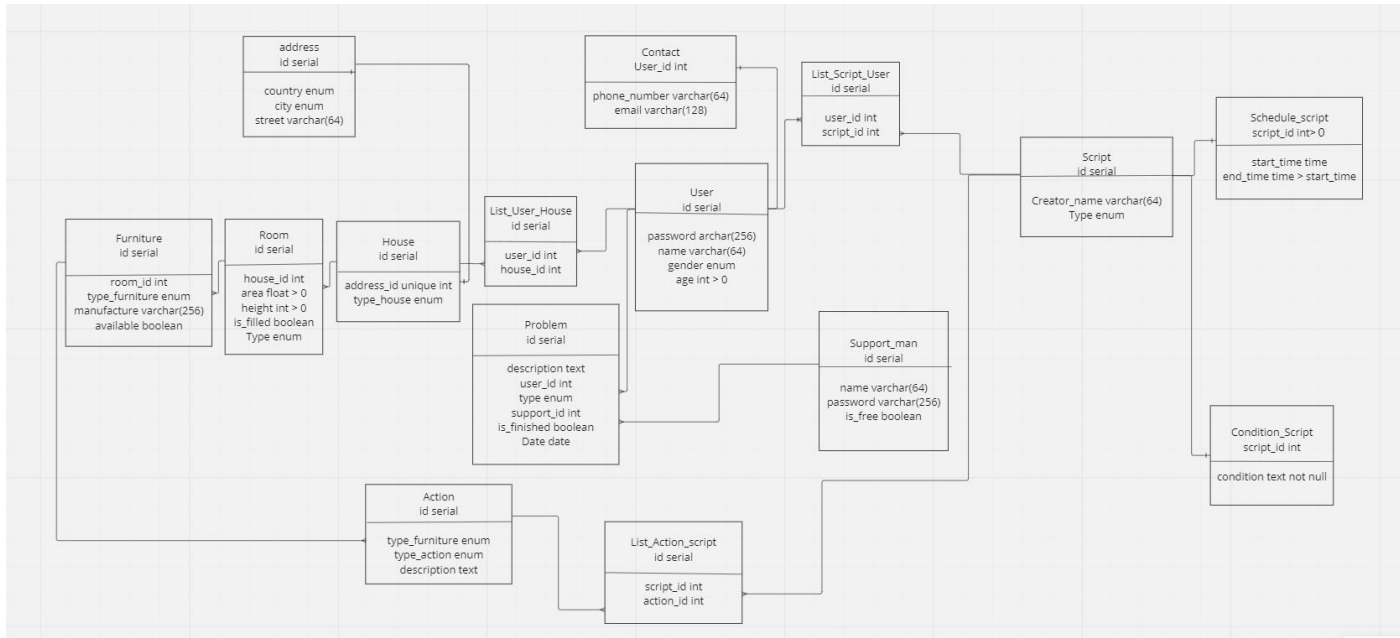
Id->city

Id->street

Инфологическая модель



Даталогическую модель



Вывод:

Все таблицы нормализованы 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.