МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет *компьютерных наук*

Кафедра *программирования и информационных технологий*

*Лабораторная работа 2 “ERWIN”*

Руководитель *Д.Н. Борисов*

Обучающийся *В.М. Беспалов, группа 3.1*

Воронеж 2022

Содержание

[Содержание 2](#_Toc99021129)

[Описание 3](#_Toc99021130)

[1. Диаграмма «Сущность-Связь» (Entity Relationship Diagram) 5](#_Toc99021131)

[2. Диаграмма модели данных, основанная на данных (KB) 6](#_Toc99021132)

[3. Полная атрибутивная модель (FA) 7](#_Toc99021133)

[4. Трансформационная модель (TM) 8](#_Toc99021134)

[5. Модель СУБД (DBMS Model) 9](#_Toc99021135)

[6. Документирование модели 11](#_Toc99021136)

Описание

Требуется разработать средствами Rational Rose модель программного обеспечения встроенного микропроцессора учрежденческой мини-АТС (автоматической телефонной станции).

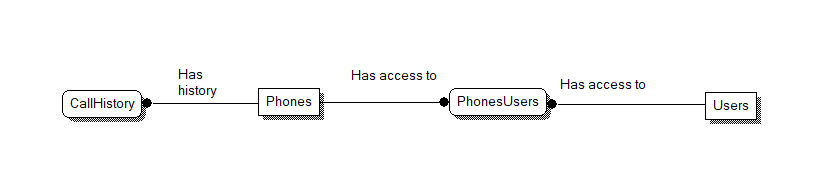
Мини-АТС осуществляет связь между служащими учреждения. Каждый абонент подключен к ней линией связи. Мини-АТС соединяет линии абонентов (осуществляет коммутацию линий). Абоненты имеют номера, состоящие из трех цифр. Специальный номер 9 зарезервирован для внешней связи. Телефонное соединение абонентов производится следующим образом. Абонент поднимает трубку телефона, и мини-АТС получает сигнал «Трубка». В ответ мини-АТС посылает сигнал «Тон». Приняв этот сигнал, абонент набирает телефонный номер (посылает три сигнала «Цифра»). Мини-АТС проверяет готовность вызываемого абонента. Если абонент не готов (его линия занята), мини-АТС посылает вызывающему абоненту сигнал «Занято». Если абонент готов, мини-АТС посылает обоим абонентам сигнал «Вызов». При этом телефон вызываемого абонента начинает звонить, а вызывающий абонент слышит в трубке длинные гудки. Вызываемый абонент снимает трубку, и мини-АТС получает от него сигнал «Трубка», после чего осуществляет коммутацию линии. Абоненты обмениваются сигналами «Данные», которые мини-АТС должна передавать от одного абонента к другому. Когда один из абонентов опускает трубку, мини-АТС получает сигнал «Конец» и посылает другому абоненту сигнал «Тон».

В любой момент абонент может положить трубку, при этом мини-АТС получает сигнал «Конец». После получения этого сигнала сеанс обслуживания абонента завершается.

Если абонент желает соединиться с абонентом за пределами учреждения, то он набирает номер «9». Мини-АТС посылает по линии, соединяющей с внешней (городской) АТС, сигнал «Трубка» и в дальнейшем служит посредником между телефоном абонента и внешней АТС. Она принимает и передает сигналы и данные между ними, не внося никаких изменений. Единственное исключение касается завершения сеанса. Получив от городской АТС сигнал «Конец», мини-АТС посылает абоненту сигнал «Тон», и ждет сигнала «Конец» для завершения обслуживания абонента. Если же вызывавший абонент первым вешает трубку, то мини-АТС получает сигнал «Конец», передает его городской АТС и завершает сеанс.

Мини-АТС может получить сигнал «Вызов» от городской АТС. Это происходит, когда нет соединений с внешними абонентами. Сигнал «Вызов» от городской АТС передается абоненту с кодом «000». Только этот абонент может отвечать на внешние звонки.

1. Диаграмма «Сущность-Связь» (Entity Relationship Diagram)



1. Диаграмма «Сущность-Связь»

На данном этапе проектирования выявлены следующие сущности:

* Call History – история звонков
* Phones – список всех телефонов
* Users – список всех пользователей
* PhonesUsers – промежуточная сущность для реализации связи многие-ко-многим

Диаграмма имеет следующие отношения между сущностями:

* Users-PhonesUsers  
  Название: имеет доступ к

Описание: определяет, какие телефоны могут быть использованы пользователем.

Тип связи: один-ко-многим (один пользователь может использовать несколько телефонов)

* Phones-PhonesUsers

Название: имеет доступ к

Описание: определяет, какими пользователями может использоваться телефон.

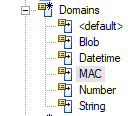
Тип связи: один-ко-многим (Один телефон может состоять в этой таблице в множественном экземпляре)

* CallHistory-Phones  
  Название: Содержится в истории

Описание: отображает вхождение телефона в список истории звонков

Тип связь: один-ко-многим (Один телефон содержится в многих записях, при этом каждая запись должна содержать только один телефон)

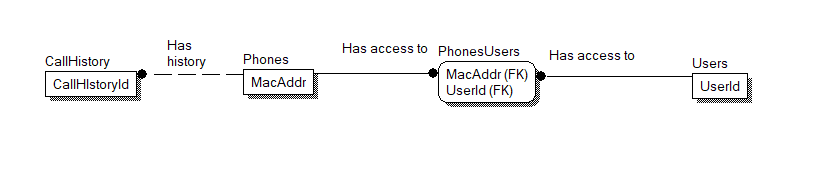
При проектировании сущностей были обозначены следующие домены:



1. Домены атрибутов

* Datetime – дата (число, месяц, день) и время (часы, минуты, секунды)
* Number – целое число
* MAC – целое число 48 бит
* String – строка текста

1. Диаграмма модели данных, основанная на данных (KB)

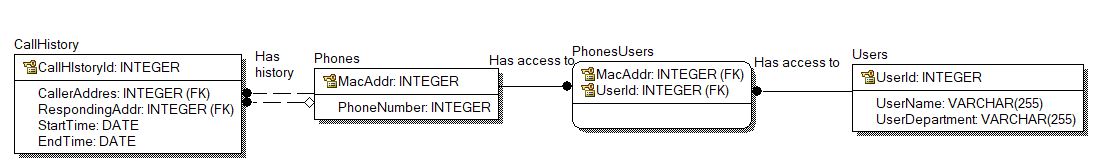


1. Диаграмма модели данных, основанная на данных (KB)

В процессе разработки модели данных, основанной на ключах, сущности, выделенные ранее, получили следующий набор ключей:

* CallHisory  
  CallHisoryId – уникальный идентификатор записи звонка.
* Phones  
  MacAddr – заводской уникальный идентификатор телефона.
* Users  
  UserId – уникальный идентификатор пользователя. (может быть номером трудового договора)
* PhonesUsers  
  <MacAddr, UserId> - составной ключ, определяющий факт того, что пользователь может использовать данный телефон и телефон может быть использован пользоватем.

1. Полная атрибутивная модель (FA)



1. Диаграмма полной атрибутивной модели (FA)

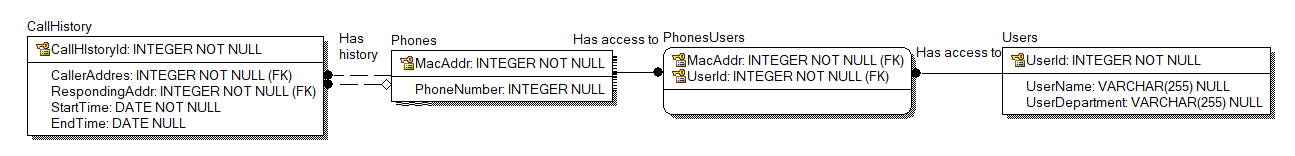
На данном этапе моделирования, к спроектированной раннее модели, были добавлены следующие атрибуты:

* Phones  
  PhoneNumber – номер телефона (от 0 до 300), который АТС присваивает телефон.
* Users  
  UserName – имя пользователя.

UserDepartment – отдел, в котором работает пользователь.

* CallHsitory  
  StartTime – время начала разговора (для экономии места в базе данных, этот атрибут, также, как и вся запись, производится только в том случае, если абонент ответит на звонок).  
  EndTime – время завершения разговора (время, когда один из пользователей повесил трубку, после начавшегося разговора).  
  CallerAddress – MAC адрес звонящего телефона (это сделано для того, чтобы однозначно определить телефон звонящего, в т.ч. идентифицировать замену телефона. В случае замены, у телефона будет выставлен короткий адрес как null).  
  RespondingAddr – MAC адрес отвечающего телефона (Вообще введение MAC адресов обещает проблемы с последующим администрированием. Нужно будет постоянно регулировать списки доступа. Такой подход мне кажется более безопасным, так как никто извне не должен получать доступ).

1. Трансформационная модель (TM)



1. Диаграмма трансформационной модели

В качестве СУБД была выбрана MySQL ввиду высокой популярности среди разработчиков и открытого исходного кода. Трансформационная модель для данного типа СУБД представляет информацию о колонках таблиц (их названиях, типах, ограничениях на значения), ключах, связях между таблицами.

Набор атрибутов и ключей сущностей при разработке трансформационной модели не изменился.

1. Модель СУБД (DBMS Model)

С помощью инструментов прямой разработки ERWIN Data Modeler 7.3 была сгенерирована база данных в виде набора последовательно исполняемых SQL-запросов к предполагаемой среде развертывания, создающих структуру таблиц и отношений между ними.

**CREATE** **TABLE** CallHistory

**(**

CallHIstoryId INTEGER **NOT** **NULL,**

StartTime DATE **NOT** **NULL,**

EndTime DATE **NULL,**

CallerAddres INTEGER **NOT** **NULL,**

RespondingAddr INTEGER **NOT** **NULL**

**);**

**ALTER** **TABLE** CallHistory

**ADD** **PRIMARY** **KEY** **(**CallHIstoryId**);**

**CREATE** **TABLE** Phones

**(**

MacAddr INTEGER **NOT** **NULL,**

PhoneNumber INTEGER **NULL**

**);**

**ALTER** **TABLE** Phones

**ADD** **PRIMARY** **KEY** **(**MacAddr**);**

**CREATE** **TABLE** PhonesUsers

**(**

MacAddr INTEGER **NOT** **NULL,**

UserId INTEGER **NOT** **NULL**

**);**

**ALTER** **TABLE** PhonesUsers

**ADD** **PRIMARY** **KEY** **(**MacAddr**,**UserId**);**

**CREATE** **TABLE** Users

**(**

UserId INTEGER **NOT** **NULL,**

UserName VARCHAR**(**255**)** **NULL,**

UserDepartment VARCHAR**(**255**)** **NULL**

**);**

**ALTER** **TABLE** Users

**ADD** **PRIMARY** **KEY** **(**UserId**);**

**ALTER** **TABLE** CallHistory

**ADD** **FOREIGN** **KEY** R\_3 **(**CallerAddres**)** **REFERENCES** Phones **(**MacAddr**);**

**ALTER** **TABLE** CallHistory

**ADD** **FOREIGN** **KEY** R\_10 **(**RespondingAddr**)** **REFERENCES** Phones **(**MacAddr**);**

**ALTER** **TABLE** PhonesUsers

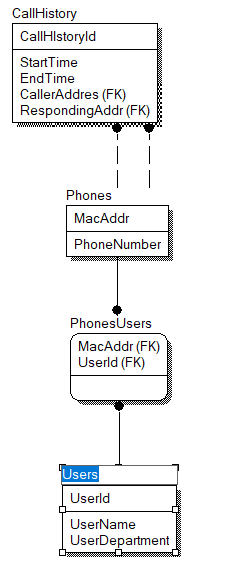
**ADD** **FOREIGN** **KEY** R\_5 **(**MacAddr**)** **REFERENCES** Phones **(**MacAddr**);**

**ALTER** **TABLE** PhonesUsers

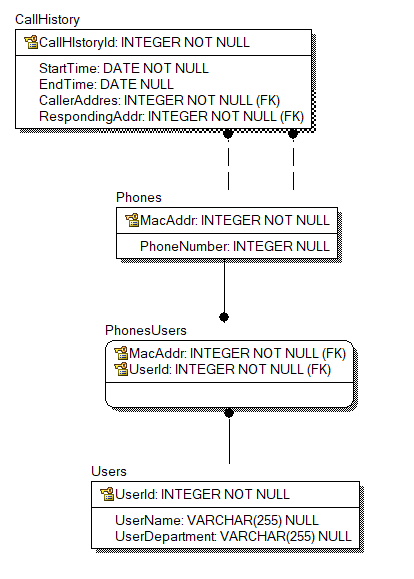
**ADD** **FOREIGN** **KEY** R\_6 **(**UserId**)** **REFERENCES** Users **(**UserId**);**

С помощью инструментов обратной разработки была получена

следующая модель базы данных:



1. Полная атрибутивная модель (FA)

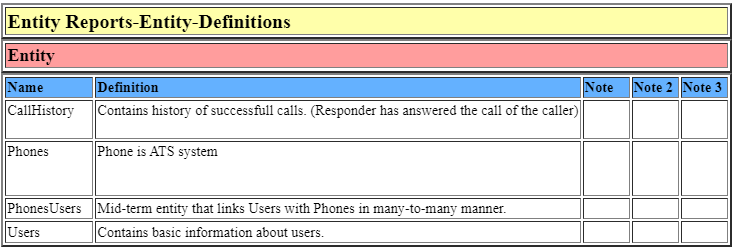


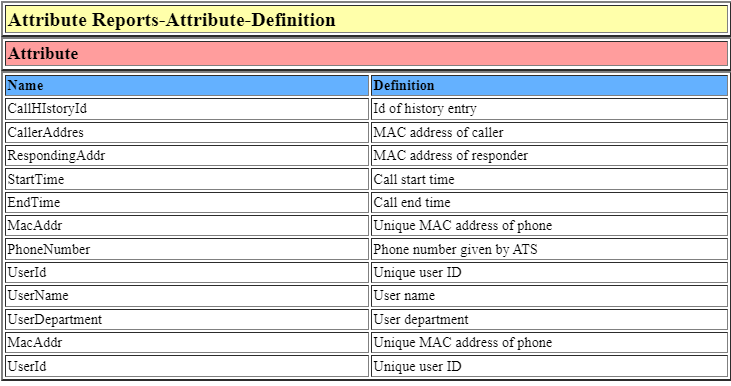
1. Трансформационная модель (TA)

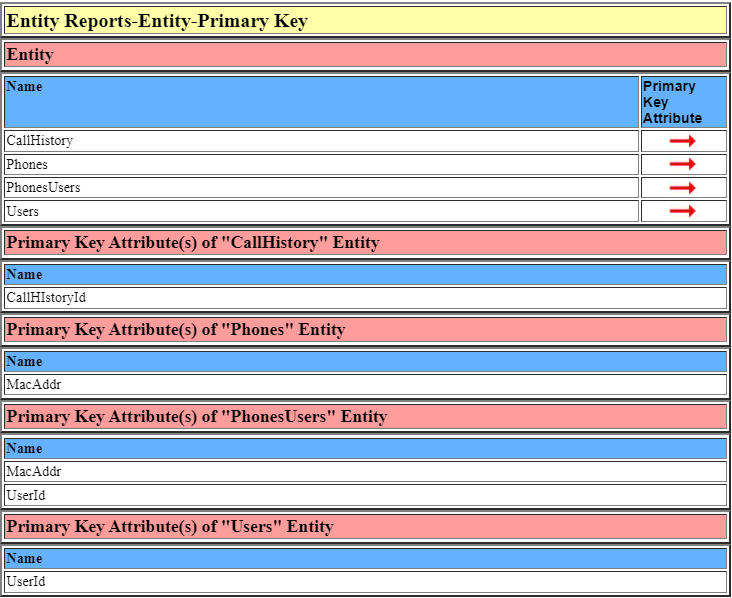
Данная модель соответствует рассмотренной ранее трансформационной схеме базы данных.

1. Документирование модели

* CallHistory – история успешных звонков, включающая в себя MAC телефона звонящего, звонившего, а также время начала и окончания звонка.
* Phones – данные о заводском номере телефона и о номере, выданном АТС
* Users – данные о работниках, которые допущены для использования каких-либо телефонов.
* PhonesUsers – данные о том, каким телефоном может пользоваться какой-либо пользователь и наоборот, может ли данный телефон быть использован этим пользователем.







1. Сгенерированная документация.