

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра комплексной информационной безопасности электронно-  
вычислительных систем (КИБЭВС)

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА С УЧЕБНО-  
ИССЛЕДОВАТЕЛЬСКОЙ БАЗОЙ ДАННЫХ ДЛЯ ОПЕРАТОРА СОТОВОЙ  
СВЯЗИ

Курсовая работа  
по дисциплине «Безопасность систем баз данных»

Пояснительная записка

Студент гр. 720-1

\_\_\_\_\_ А.А. Шарапов  
(подпись)

\_\_\_\_\_  
(дата)  
Руководитель

Старший преподаватель кафедры  
КИБЭВС

\_\_\_\_\_ Н.А. Новгородова  
(оценка) (подпись)  
\_\_\_\_\_  
(дата)

Томск 2023

## Реферат

Курсовая работа содержит 52 страницы, 33 рисунка, 11 таблиц, 2 источника, 2 приложения.

Ключевые слова: БАЗА ДАННЫХ, СУБД, C#, SQL, MICROSOFT SQL, IDEF0, IDEF1X, .NET FRAMEWORK.

Объектом разработки является автоматизированная система для мобильного оператора.

Цель работы – разработать автоматизированную систему с базой данных для мобильного оператора.

Результаты работы:

- определены необходимые технологии и проработана архитектура БД;
- разработано приложение для работы с БД.

В качестве инструментария для выполнения данной работы были использованы: язык программирования C#, СУБД Microsoft SQL Server, IDE Visual Studio, IDE Rider, фреймворк .NET Framework.

Ссылка на репозиторий: <https://github.com/Alexman356/MobileOperator>.

Курсовая работа оформлена в текстовом редакторе Microsoft Word 2016 согласно ОС ТУСУР 01-2021. [1]

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ  
И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности электронно-  
вычислительных систем (КИБЭВС)

УТВЕРЖДАЮ

Заведующий кафедрой КИБЭВС,  
д-р техн. наук, профессор

\_\_\_\_\_ А.А. Шелупанов

“ \_\_\_\_ ” \_\_\_\_\_ 2022г.

### ЗАДАНИЕ

на курсовую работу по дисциплине «Безопасность систем баз данных» студенту  
Шарапову Александру Анатольевичу группы 720-1 факультета безопасности.

1 Тема работы: Автоматизированная информационная система с  
учебно-исследовательской базой данных: Оператор сотовой связи;

2 Исходные данные к работе:

2.1 Реляционная СУБД Microsoft SQL Server;

2.2 Данные по предметной области

Сотрудники, абоненты, тарифы, контракты, номера

3 Срок сдачи студентом законченной работы: \_\_\_\_\_

4 Содержание курсовой работы:

4.1 Проектирование инфологической модели данных:

– описание и структуризация предметной области (описание бизнес-процессов, диаграммы IDEF0);

- представление модели «Сущность-связь» (ER-модель);
- сценарий пользовательского интерфейса.

#### 4.2 Проектирование даталогической (логической) модели данных:

- проектирование реляционной базы данных на основе принципов нормализации;
- проектирование концептуальной модели данных (использование методологии IDEF1X);
- составление глоссария модели.

#### 4.3 Физическое проектирование БД:

- создание базы данных и ее необходимых элементов;
- описание ограничений на базу данных;
- сопоставление логических и физических имен.

#### 4.4 Написание программы обработки и работы с данными:

- генерация программы меню, реализующей пользовательский интерфейс;
- режим просмотра данных с использованием экранных форм;
- использование режимов редактирования данных;
- процедуры поиска и манипулирования данными (сортировки, фильтры и пр.);
- использование SQL операторов (SQL запросы, операторы определения данных, операторы манипулирования данными);
- обеспечение безопасности данных.

#### 5 Содержание пояснительной записки:

- титульный лист;
- реферат на русском языке;
- задание;
- содержание;
- введение;
- вопросы проектирования БД;

- обоснование выбора программных средств;
- руководство пользователя;
- описание прикладной программы;
- заключение;
- список использованных источников;
- приложения (экранные формы, листинг программы и др.).

Пояснительная записка должна быть оформлена в соответствии со стандартом ТУСУР.

В конверте на обложке приложить диск с БД, исходными текстами программы с соответствующими файлами, исполнительными файлами, пояснительной запиской, презентацией.

6 Дата выдачи задания:

«14» сентября 2022 г.

Задание согласовано:

Руководитель работы

Новгородова Н.А., старший преподаватель кафедры КИБЭВС

“14” сентября 2022г. \_\_\_\_\_

Задание принято к исполнению

“14” сентября 2022г. \_\_\_\_\_

## Оглавление

Введение.....	8
1 КОНЦЕПТУАЛЬНОЕ (ИНФОЛОГИЧЕСКОЕ) ПРОЕКТИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ (ПО).....	9
1.1 Неформальное описание ПО с использованием естественного языка ....	9
1.2 Описание бизнес-процессов ПО «на текущий момент» в методологии функционального моделирования IDEF0.....	9
1.3 Цель и актуальность внедрения автоматизации ПО.....	11
1.3.1 Выделение бизнес-процессов, подлежащих автоматизации.....	11
1.3.2 Описание бизнес-процессов ПО в IDEF0 после внедрения автоматизированной информационной системы.....	12
1.4 Концептуальная информационная модель данных для ПО.....	14
1.4.1 Основные объекты ПО, информация о которых будет накапливаться в БД. Их характеристики и свойства (атрибуты).....	14
1.4.2 Определение связей между объектами ПО.....	15
1.4.3 Графическое представление концептуальной информационной модели данных .....	15
1.5 Политика безопасности по работе с данными.....	16
1.5.1 Типы пользователей.....	16
1.5.2 Ограничения пользователей по работе с объектами ПО.....	17
2 ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ БАЗЫ ДАННЫХ.....	18
2.1 Логическое (дatalogическое) проектирование модели данных .....	18
2.1.1 Определение отношений и связей между отношениями на основе концептуальной информационной модели. Первичные и внешние ключи.....	18
2.1.2 Нормализация логической модели данных .....	18
2.1.3 Графическое представление логической модели данных в методологии IDEF1x .....	22
2.2 Физическое проектирование с учетом выбранной СУБД .....	23

2.2.1	Определение типов данных атрибутов отношений.....	23
2.2.2	Графическое представление физической модели данных в методологии IDEF1x .....	29
2.3	Представление физической модели данных на языке SQL .....	30
3	Программный комплекс для работы с БД .....	31
3.1	Описание прикладной программы .....	31
3.1.1	Окно авторизации и главное окно приложения .....	31
3.1.2	Окно «Abonents» .....	32
3.1.3	Окно «Rates» .....	34
3.1.4	Окно «Employees» .....	35
3.1.5	Окно «Contracts» .....	37
3.1.6	Окно «Users» .....	39
3.1.7	Окно «Profile» .....	41
3.2	Руководства администратора .....	42
3.3	Руководство программиста.....	43
	Заключение.....	44
	Список использованных источников.....	45
	Приложение А (обязательное) Физическая модель данных на языке SQL .....	46
	Приложение Б (обязательное) Листинг кода окна авторизации.....	50

## **Введение**

Цель работы – автоматизация процесса работы с абонентами оператора сотовой связи путём проектирования и разработки приложения и базы данных, а также углубление знаний в области проектирования баз данных и получение практических навыков их разработки.

Объектом автоматизации является работа с абонентами оператора сотовой связи, а также обработка, хранение, накопление данных о них.

Автоматизация осуществляется благодаря взаимодействию пользователей с графическим интерфейсом и работой с БД.



# **1 КОНЦЕПТУАЛЬНОЕ (ИНФОЛОГИЧЕСКОЕ) ПРОЕКТИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ (ПО)**

## **1.1 Неформальное описание ПО с использованием естественного языка**

В данной курсовой работе предметной областью является разработка базы данных для регистрирования и обслуживания абонентов оператора сотовой связи.

Оператор сотовой связи — компания, предоставляющая услуги сотовой связи для сотовых телефонов своих абонентов. При подключении клиента к сотовой связи следует зарегистрировать информацию о клиенте: фамилию и адрес клиента, паспортные данные и указать тариф подключения. Данные вносятся в договор. В договорах имеется информация об абоненте, сотруднике, который оформлял договор, номер телефона, статус и дата подключения. Если абонент уже существует в базе данных, то сотруднику необходимо идентифицировать его, выдать ему номер и, при желании абонента, подключить тариф.

## **1.2 Описание бизнес-процессов ПО «на текущий момент» в методологии функционального моделирования IDEF0**

Графически представленная функциональная модель ПО «Оператор сотовой связи» в методологии IDEF0 до внедрения автоматизированной системы описана на рисунках 1.1 – 1.3.

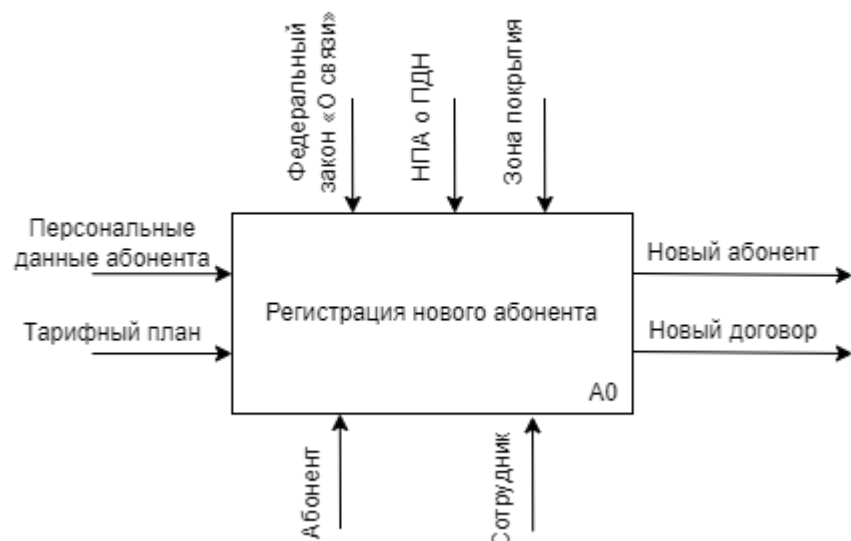


Рисунок 1.1 – Функциональная модель ПО «Оператор сотовой связи»

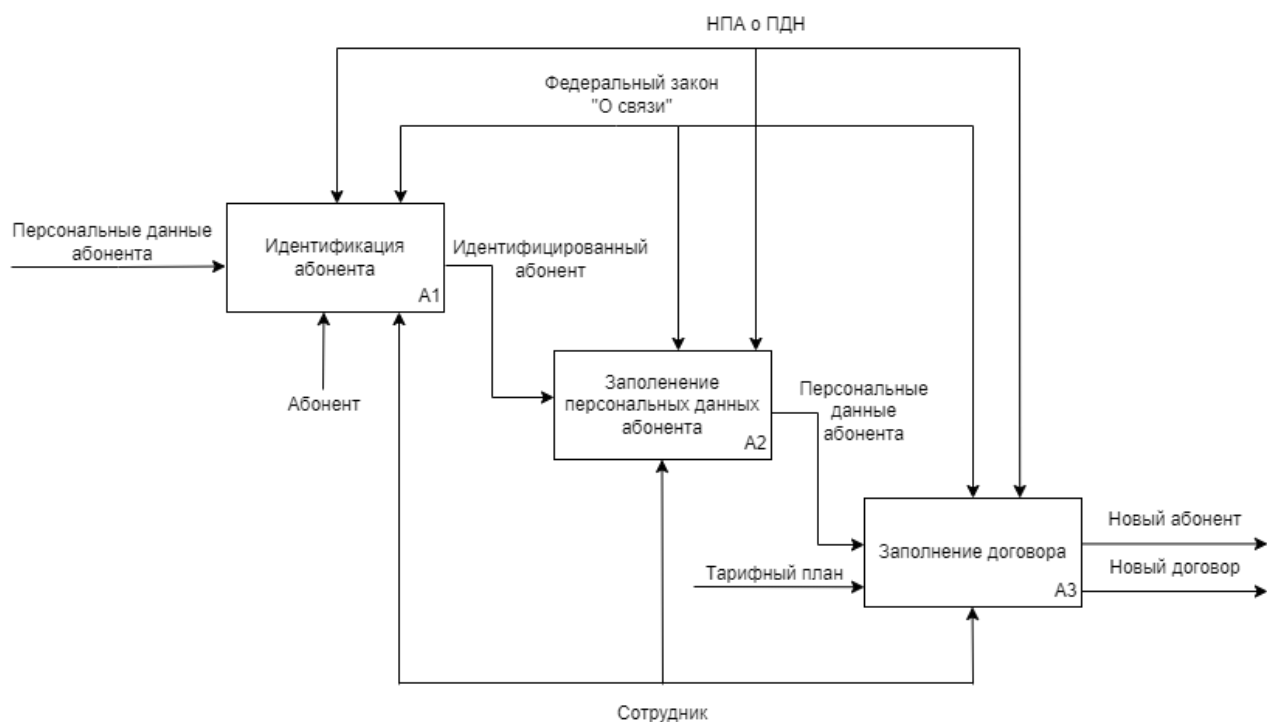


Рисунок 1.2 – Декомпозированная функциональная модель ПО «Оператор сотовой связи» (при регистрации нового абонента)

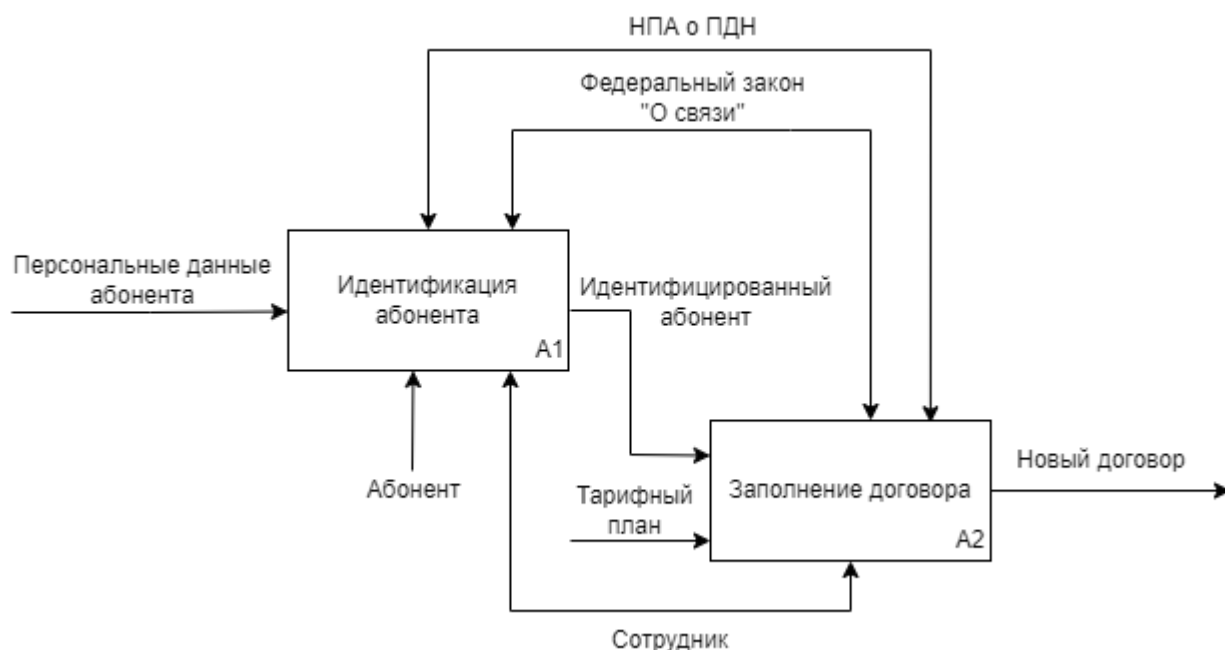


Рисунок 1.3 – Декомпозированная функциональная модель ПО «Оператор сотовой связи» (при добавлении нового номера абоненту)

### 1.3 Цель и актуальность внедрения автоматизации ПО

#### 1.3.1 Выделение бизнес-процессов, подлежащих автоматизации

В последние годы количество абонентов сотовой связи увеличилось в несколько раз. И как следствие в разы увеличилась нагрузка на операторов сотовой связи. В частности, это касается регистрации новых абонентов. Автоматизация этого процесса значительно уменьшит нагрузку на операторов, а также ускорит обработку данных.

В соответствии с предметной областью система строится с учётом следующих особенностей:

- у каждого абонента может быть несколько номеров;
- оформление номера происходит через договор на одного абонента;
- у каждого номера может быть подключен только один тариф или не подключен вообще;
- абонент может заблокировать свой номер;

- каждый оператор может составлять несколько договоров;
- каждый договор не может быть составлен несколькими операторами;
- сотрудник уже существует в БД.

Данная система использует модель, содержащую информацию об абонентах данного оператора и ее сотрудниках, существующих тарифов, а также все договора, составленными сотрудниками.

Информация об абонентах и сотрудниках содержит все необходимые персональные данные.

Необходимая информация о договорах содержит: номер заявления, номер телефона, код сотрудника, код абонента, дата обращения, статус договора (он может быть в следующих состояниях: «Активен» или «Не активен»).

Информация о тарифах содержит такие данные как: название тарифа, размер ежемесячной платы, размер пакета интернет трафика, количество минут и SMS.

### **1.3.2 Описание бизнес-процессов ПО в IDEF0 после внедрения автоматизированной информационной системы**

Графически представленная функциональная модель ПО «Оператор сотовой связи» в методологии IDEF0 после внедрения автоматизированной системы описана на рисунках 1.4 – 1.6.



Рисунок 1.4 – Функциональная модель ПО «Оператор сотовой связи» с АИС

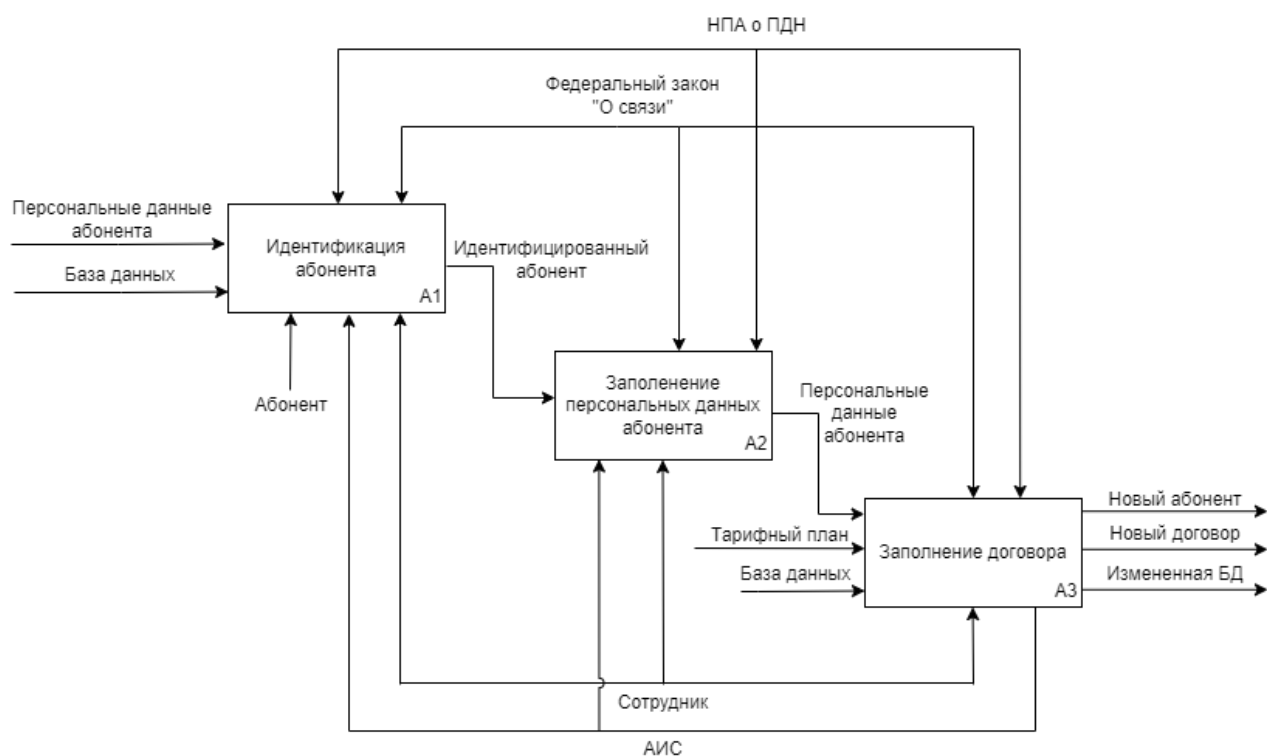


Рисунок 1.5 – Декомпозированная функциональная модель ПО «Оператор сотовой связи» (при регистрации нового абонента) с АИС

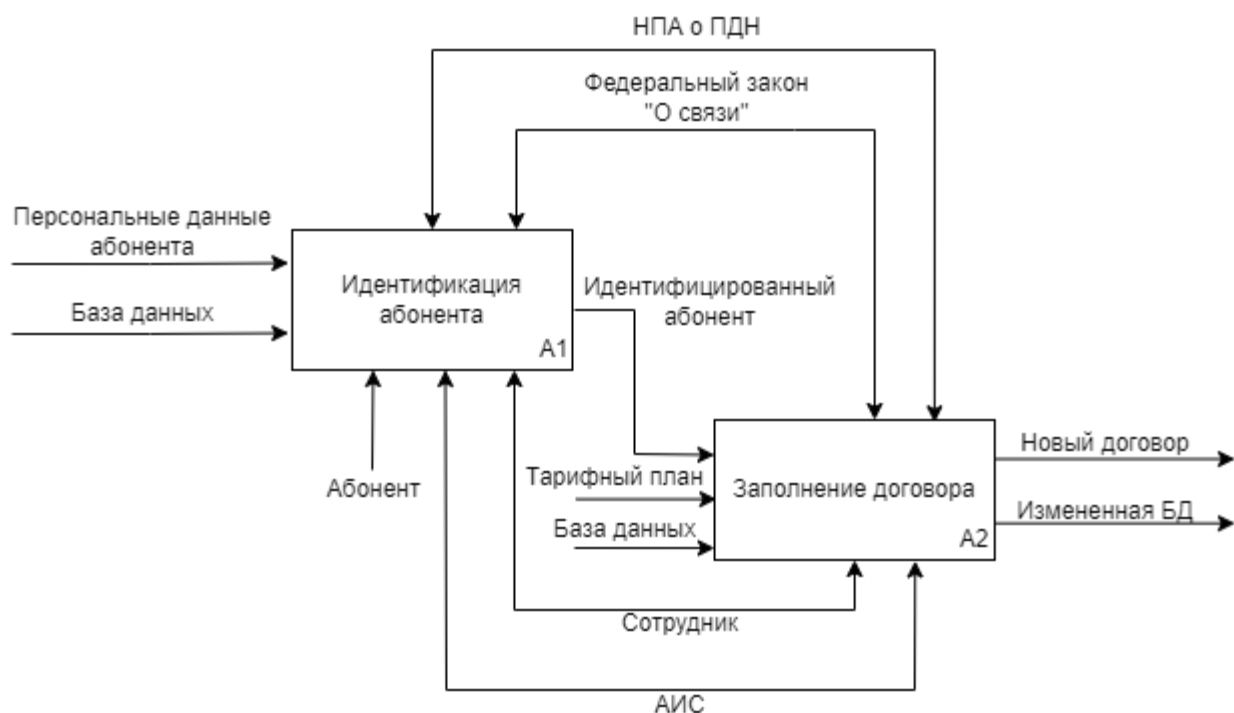


Рисунок 1.6 – Декомпозированная функциональная модель ПО «Оператор сотовой связи» (при добавлении нового номера абоненту) с АИС

## 1.4 Концептуальная информационная модель данных для ПО

### 1.4.1 Основные объекты ПО, информация о которых будет накапливаться в БД. Их характеристики и свойства (атрибуты)

В создаваемой базе данных основными объектами, информация о которых будет накапливаться в ней, будут являться сущности: «Сотрудник», «Абонент», «Тариф», «Договор».

Необходимо рассмотреть каждую сущность отдельно и выделить атрибуты каждой сущности.

Сущность «Сотрудник». Её атрибутами будут являться: логин, пароль, электронная почта, дата рождения, пол, зарплата, фамилия, имя, отчество, должность, адрес, паспортные данные.

Сущность «Абонент». Её атрибутами будут являться: логин, пароль, электронная почта, дата рождения, пол, фамилия, имя, отчество, должность, паспортные данные, адрес.

Сущность «Тариф». Её атрибутами будут являться: SMS, голосовая связь, пакет интернет-трафика, ежемесячная оплата, название.

Сущность «Договор». Её атрибутами будут являться: код сотрудника, код абонента, номер телефона, дата обращения, статус заявки.

#### **1.4.2 Определение связей между объектами ПО**

Связи между сущностями:

Можно ли в договоре использовать несколько тарифов? – Нет. Может ли тариф использоваться в нескольких договорах? – Да.

Связь между сущностями «Тариф» и «Договор»: 1:М.

Может ли сотрудник обрабатывать несколько договоров? – Да. Может ли договор обрабатываться несколькими сотрудниками? – Нет.

Связь между сущностями «Сотрудник» и «Договор»: 1:М.

Может ли абонент заполнять несколько договоров? – Да. Может ли договор заполняться несколькими абонентами? – Нет.

Связь между сущностями «Абонент» и «Договор»: 1:М.

#### **1.4.3 Графическое представление концептуальной информационной модели данных**

Графически представленная концептуальная информационная модель данных изображена на рисунке 1.7.

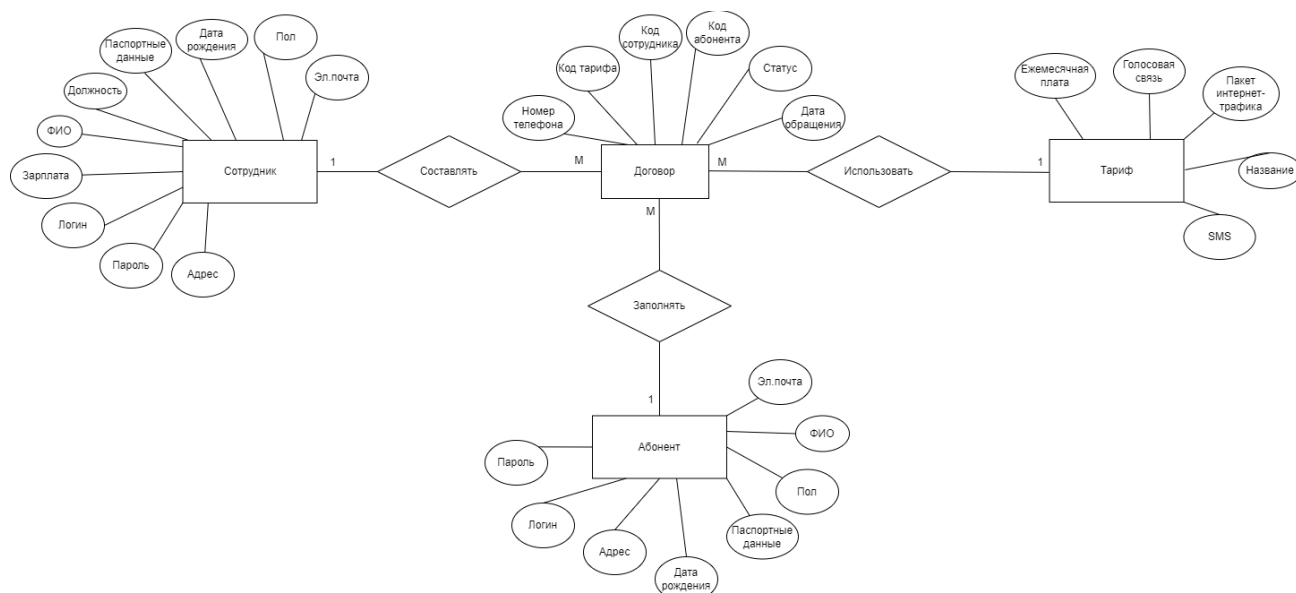


Рисунок 1.7 – Концептуальная информационная модель данных для ПО

## 1.5 Политика безопасности по работе с данными

### 1.5.1 Типы пользователей

Определены следующие типы пользователей, их основные задачи и запросы к БД:

Администратор:

- приём/увольнение сотрудников;
- внесение изменений в данные о сотрудниках;
- получение списка всех абонентов;
- добавление, редактирование и обновление тарифов.

Оператор:

- составление, обновление или удаление договоров;
- получение списка всех абонентов и тарифов и изменение или удаление данных об абонентах.

Абонент:

- обновление договоров;
- получение списка всех тарифов и изменение.



### 1.5.2 Ограничения пользователей по работе с объектами ПО

Пользователи разных ролей ограничены в возможностях работы с таблицами из базы данных в целях сохранения безопасности информации. Далее были приведены точные ограничения пользователей по работе с базой данных. Для удобства восприятия данные представлены в виде таблиц 1.1 – 1.3.

Таблица 1.1 – Разделение прав доступа для роли «Абонент»

Права	«Договоры»	«Тарифы»	«Сотрудники»	«Абоненты»
Просмотр	–	+	–	О себе
Обновление	+	–	–	–
Добавление	–	–	–	–
Удаление	–	–	–	–

Таблица 1.2 – Разделение прав доступа для роли «Администратор»

Права	«Договоры»	«Тарифы»	«Сотрудники»	«Абоненты»
Просмотр	+	+	+	+
Обновление	+	+	+	+
Добавление	+	+	+	+
Удаление	+	+	+	+

Таблица 1.3 – Разделение прав доступа для роли «Оператор»

Права	«Договоры»	«Тарифы»	«Сотрудники»	«Абоненты»
Просмотр	+	+	О себе	+
Обновление	+	–	–	+
Добавление	+	–	–	+
Удаление	+	–	–	+

## **2 ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ БАЗЫ ДАННЫХ**

### **2.1 Логическое (дatalogическое) проектирование модели данных**

#### **2.1.1 Определение отношений и связей между отношениями на основе концептуальной информационной модели. Первичные и внешние ключи**

Для каждой таблицы нужно выделить первичный ключ. Для таблицы «Договоры» первичным ключом будет являться поле «contract\_id». Для таблицы «Тарифы» первичным ключом будет являться поле «rate\_id». Для таблицы «Абоненты» первичным ключом будет являться поле «abonent\_id». Для таблицы «Сотрудники» первичным ключом будет являться поле «employee\_id».

Внешний ключ связывает таблицы, поэтому нужно выделить все внешние ключи. В таблице «Договор» будет три внешних ключа «rate\_id» (связь с таблицей «Тарифы»), «abonent\_id» (связь с таблицей «Абоненты»), «employee\_id» (связь с таблицей «Сотрудники»).

#### **2.1.2 Нормализация логической модели данных**

Нормальная форма – свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Для схемы БД используются обозначения, представленные на рисунке 2.1.

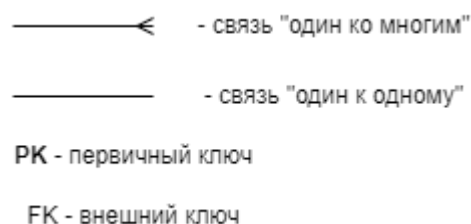


Рисунок 2.1 – Обозначения, используемые на схеме базы данных

Были определены отношения между сущностями, а также атрибуты, первичные и внешние ключи в отношениях. Полученные таблицы представлены на рисунке 2.2.

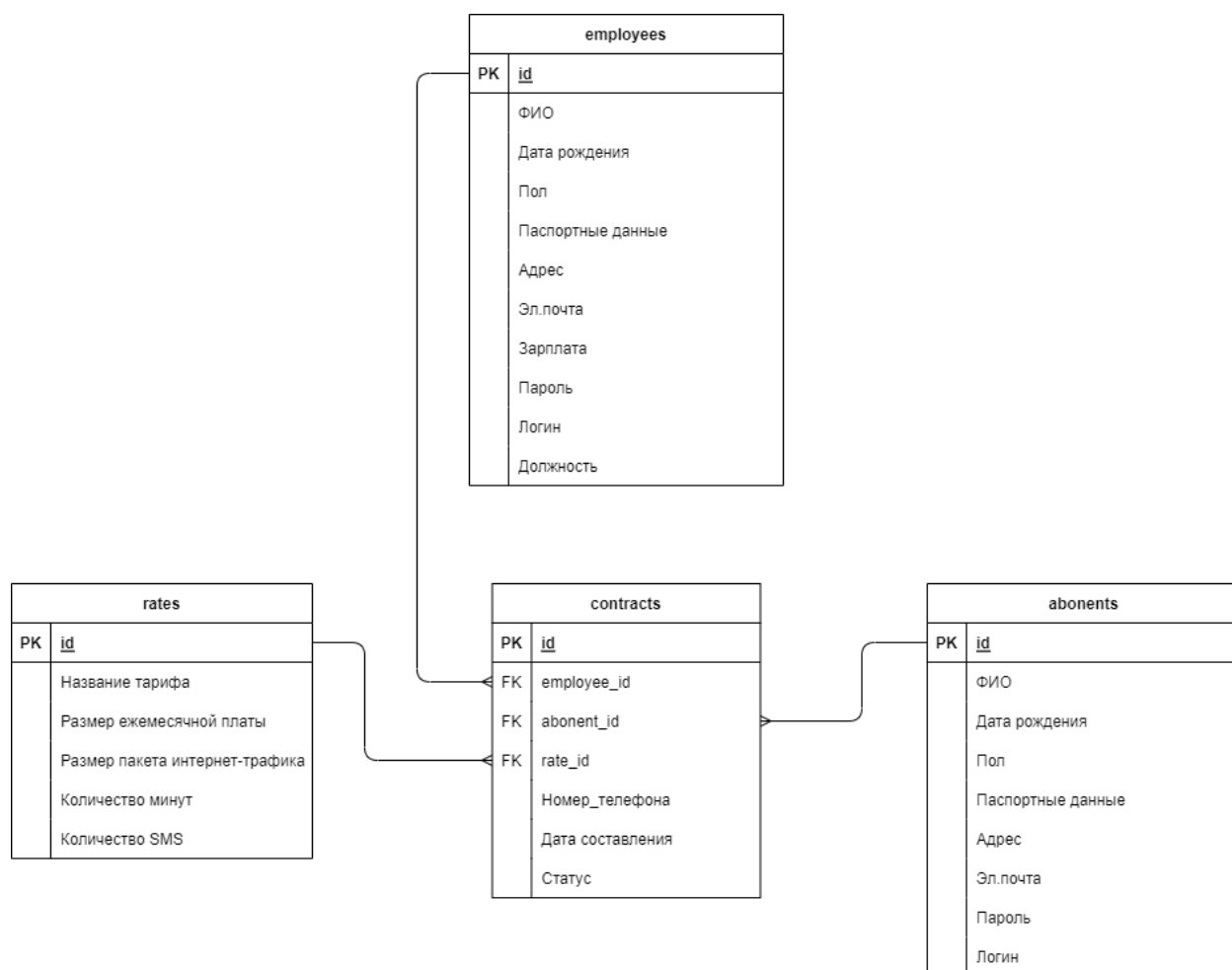


Рисунок 2.2 – Таблицы сущностей

Отношение находится в первой нормальной форме (сокращённо 1НФ), если все его атрибуты атомарны, то есть если ни один из его атрибутов нельзя разделить на более простые атрибуты, которые соответствуют каким-то другим свойствам описываемой сущности. Первая нормальная форма представлена на рисунке 2.3.

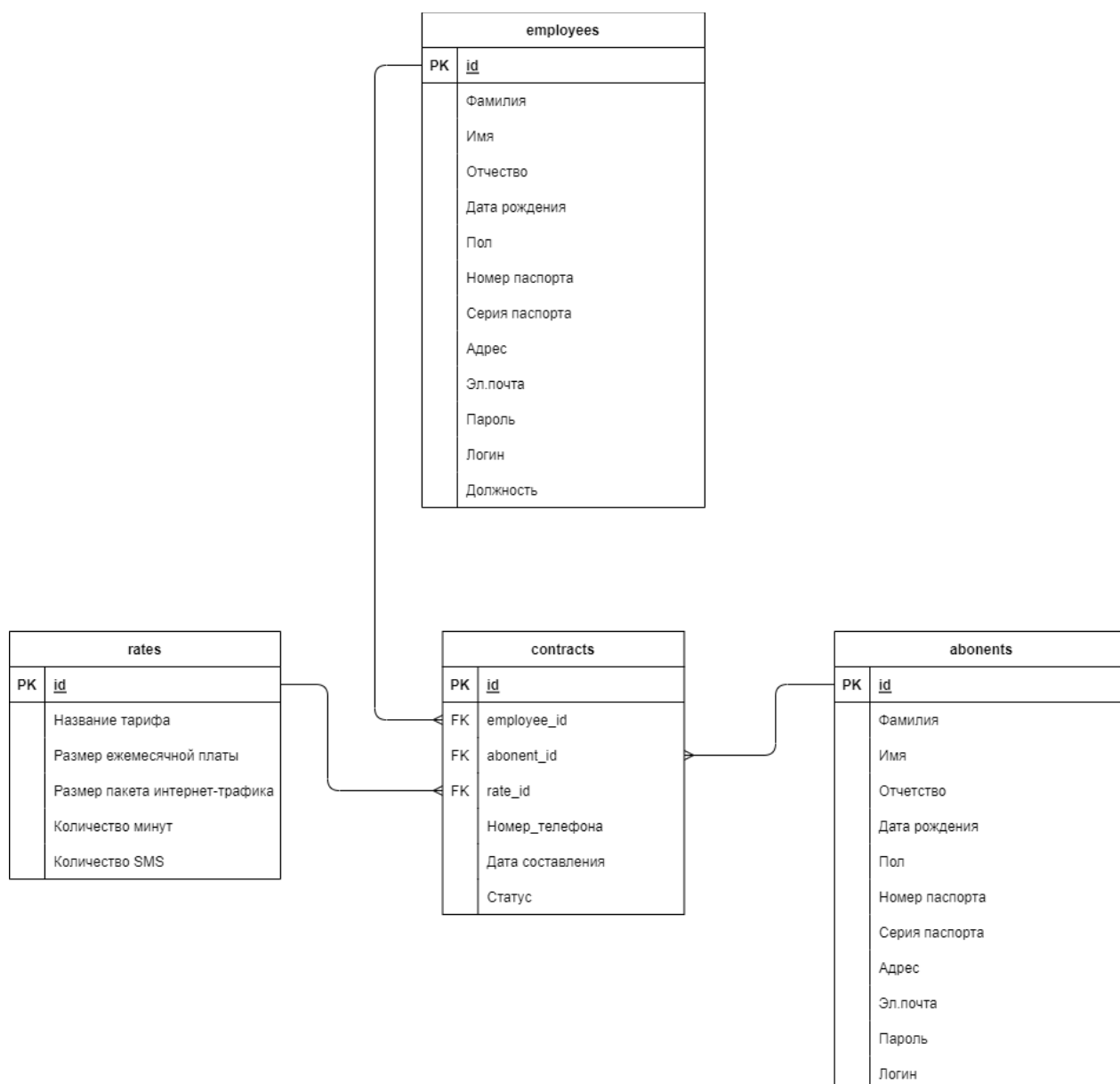


Рисунок 2.3 – Первая нормальная форма

Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый неключевой атрибут неприводимо зависит от (каждого) её потенциального ключа. Для того чтобы привести таблицу к 2НФ найдены атрибуты, частично зависящие от ключа и вынесены в новую сущность. Вторая нормальная форма представлена на рисунке 2.4.

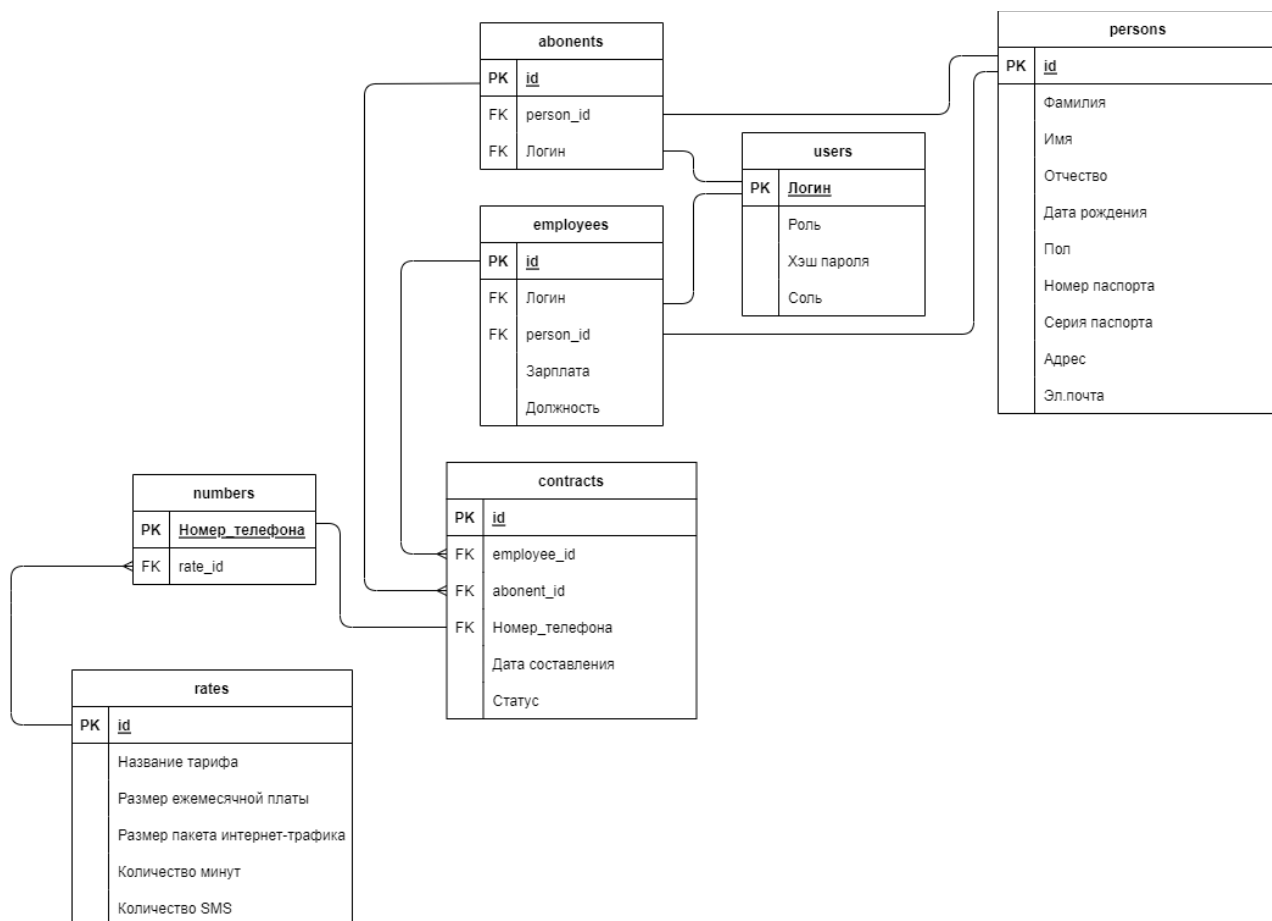


Рисунок 2.4 – Вторая нормальная форма

Отношение находится в третьей нормальной форме, когда находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа. Если в отношении существует транзитивная зависимость между атрибутами, то транзитивно зависимые атрибуты удаляются из него и помещаются в новое отношение вместе с копией их детерминанта. Третья нормальная форма представлена на рисунке 2.5.

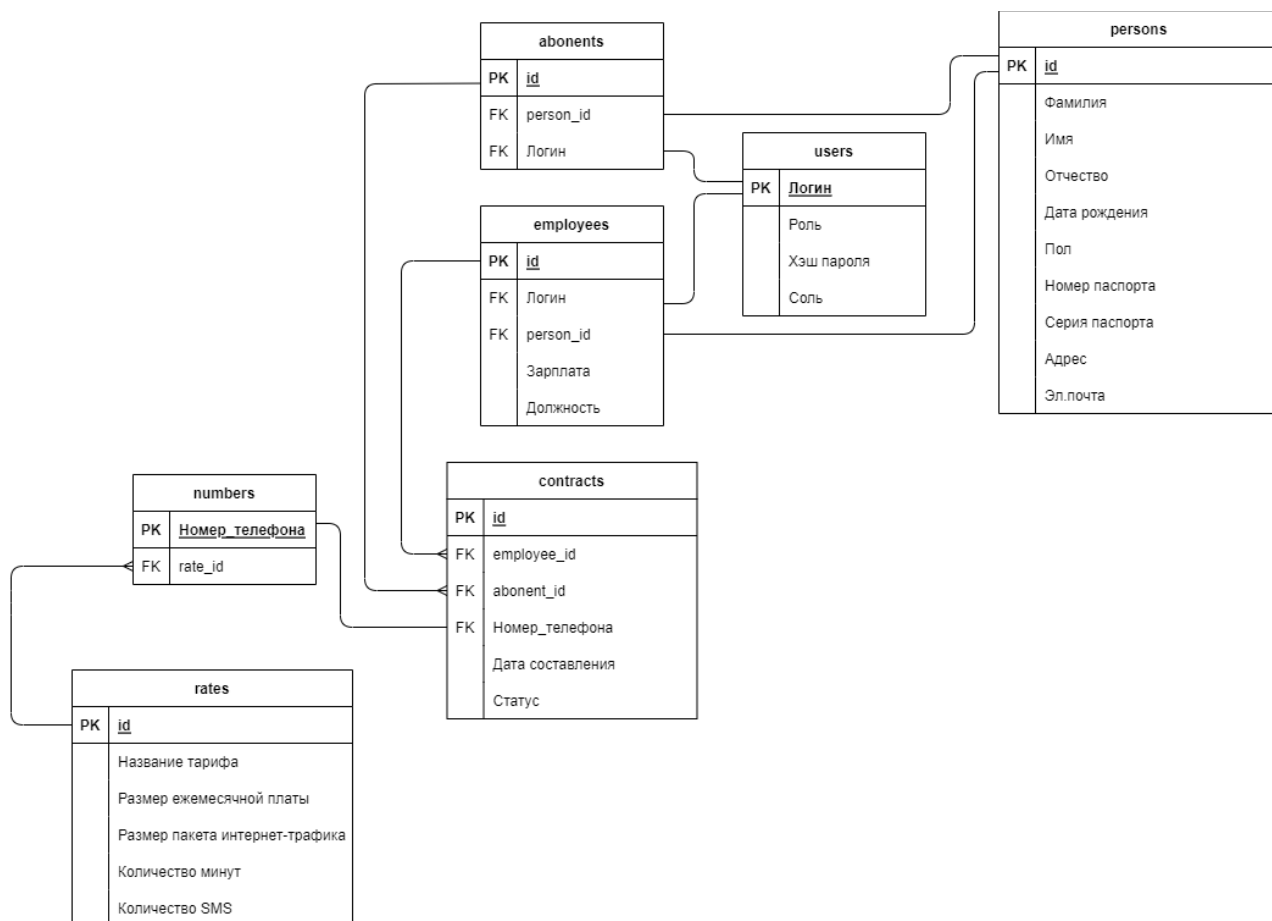


Рисунок 2.5 – Третья нормальная форма

### 2.1.3 Графическое представление логической модели данных в методологии IDEF1x

Реляционная модель данных в методологии IDEF1x представлена на рисунке 2.6.

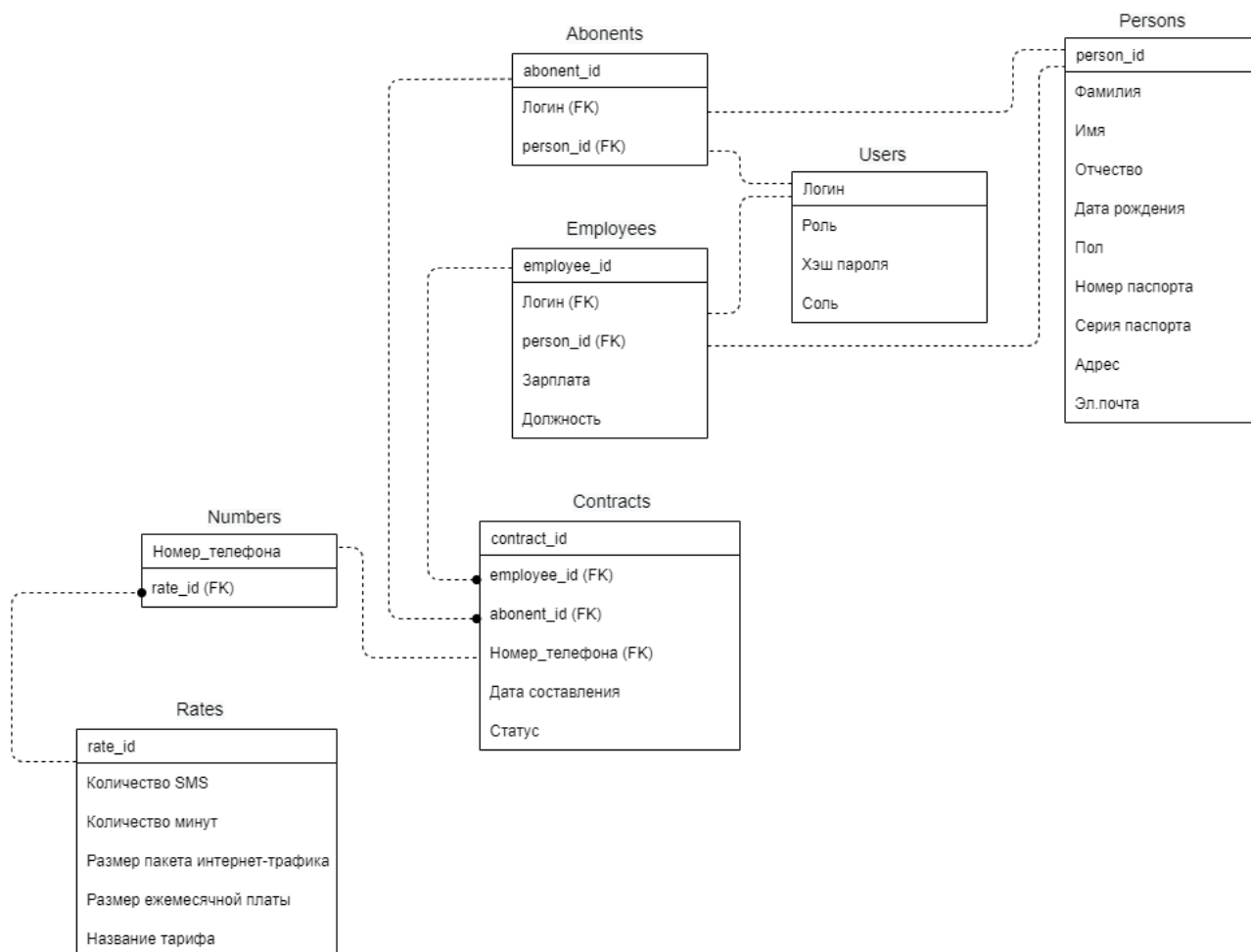


Рисунок 2.6 – Графическое представление логической модели данных в методологии IDEF1x

## 2.2 Физическое проектирование с учетом выбранной СУБД

Для работы была выбрана СУБД Microsoft SQL Server. Microsoft SQL Server – система управления реляционными базами данных, разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов с расширениями.

### 2.2.1 Определение типов данных атрибутов отношений

Каждое реляционное отношение соответствует одной сущности и в него вносятся все атрибуты этой сущности. Для каждого отношения определяются

первичный ключ и внешние ключи (в соответствии со схемой БД). Отношения приведены в таблицах 2.1 – 2.7. Для каждого отношения указаны атрибуты с их внутренним названием, типом и условиями на значение.

Таблица 2.1 – Описание таблицы «Rates»

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
rate_id	Счетчик	Уникальный идентификатор тарифа	Первичный ключ. Условие на значение: $\geq 0$	rate_ID
Название тарифа	Текстовый	Название тарифа для абонентов	Обязательное поле. Размер поля 100.	Name_rate
Размер ежемесячной платы	Целочисленный	Размер ежемесячной платы за тариф	Обязательное поле. Значение по умолчанию 0. Условие на значение: $\geq 0$	Cost
Размер пакета интернет-трафика	Целочисленный	Размер доступного пакета интернет-трафика в рамках тарифа	Значение по умолчанию 0. Условие на значение: $\geq 0$	Internet
Количество минут	Целочисленный	Количество минут на разговоры в рамках тарифа	Значение по умолчанию 0. Условие на значение: $\geq 0$	Minutes
Количество SMS	Целочисленный	Количество сообщений в рамках тарифа	Значение по умолчанию 0. Условие на значение: $\geq 0$	SMS



Таблица 2.2 – Описание таблицы «Numbers»

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
Номер телефона	Текстовый	Номер телефона абонента	Первичный ключ. Условие на значение: длина номера телефона = 12	Number_telephone
rate_id	Счетчик	Код тарифа, подключенного к номеру телефона абонента	Внешний ключ. Условие на значение: >=0	rate_ID

Таблица 2.3 – Описание таблицы «Abonents»

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
abonent_id	Счетчик	Уникальный идентификатор абонента	Первичный ключ. Условие на значение: >=0	abonent_ID
person_id	Целочисленный	Код человека в БД	Внешний ключ. Обязательное поле. Условие на значение: >=0	person_ID
Логин	Текстовый	Логин пользователя БД	Внешний ключ. Размер поля 50	user_login

Таблица 2.4 – Описание таблицы «Users»

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
Логин	Текстовый	Логин пользователя для работы с БД	Первичный ключ. Обязательное поле. Размер поля 50	Login

Окончание таблицы 2.4

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
Хэш пароля	Текстовый	Пароль пользователя в виде хэша	Обязательное поле. Размер поля 44	PasswordHash
Соль	Текстовый	Соль для пароля пользователя	Обязательное поле. Размер поля 10	PasswordSalt
Роль	Текстовый	Пароль пользователя для работы с БД	Обязательное поле. Размер поля 50	Role

Таблица 2.5 – Описание таблицы «Persons»

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
person_id	Счетчик	Уникальный идентификатор человека	Первичный ключ. Условие на значение: $\geq 0$	person_ID
Фамилия	Текстовый	Фамилия человека	Обязательное поле. Размер поля 50	Last_name
Имя	Текстовый	Имя человека	Обязательное поле. Размер поля 50	First_name
Отчество	Текстовый	Отчество человека	Размер поля 50	Middle_name
Дата рождения	Дата	Дата рождения человека	Обязательное поле	Birthdate
Пол	Текстовый	Пол человека	Обязательное поле. Условие на значение: Пол = Мужской или Пол = Женский	Gender

Окончание таблицы 2.5

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
Серия паспорта	Текстовый	Серия паспорта человека	Обязательное поле. Условие на значение: Длина = 4	Series_passport
Номер паспорта	Текстовый	Номер паспорта человека	Обязательное поле. Условие на значение: Длина = 6	Number_passport
Адрес	Текстовый	Место жительства человека	Обязательное поле. Размер поля 300	Address
Электронная почта	Текстовый	Электронная почта человека	Размер поля 100	Email

Таблица 2.6 – Описание таблицы «Employees»

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
employee_id	Счетчик	Уникальный идентификатор сотрудника	Первичный ключ. Условие на значение: $\geq 0$	abonent_ID
person_id	Целочисленный	Код человека в БД	Внешний ключ. Обязательное поле. Условие на значение: $\geq 0$	person_ID
Логин	Текстовый	Код пользователя БД	Внешний ключ. Обязательное поле. Размер поля 50	user_login
Зарплата	Целочисленный	Зарплата сотрудника	Обязательное поле.	Salary

Окончание таблицы 2.6

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
Должность	Текстовый	Должность, занимаемая сотрудником	Обязательное поле. Условие на значение: Должность = Стажер ИЛИ Администратор ИЛИ Оператор	Post

Таблица 2.7 – Описание таблицы «Contracts»

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
contract_id	Счетчик	Уникальный идентификатор договора	Первичный ключ. Условие на значение: $\geq 0$	contract_ID
abonent_id	Целочисленный	Код абонента	Внешний ключ. Обязательное поле. Условие на значение: $\geq 0$	abonent_ID
employee_id	Целочисленный	Код сотрудника	Внешний ключ. Обязательное поле. Условие на значение: $\geq 0$	person_ID
number_Номер_телефона	Целочисленный	Номер телефона для абонента	Внешний ключ. Обязательное поле. Условие на значение: длина номера телефона = 12	Number_telephone

Окончание таблицы 2.7

Имя столбца	Тип данных	Описание	Ограничения	Внутреннее название столбца
Дата составления	Дата	Дата составления договора	Обязательное поле	Date
Статус	Целочисленный	Статус договора	Обязательное поле. Условие на значение: Статус = 1 ИЛИ Статус = 0	Status

### 2.2.2 Графическое представление физической модели данных в методологии IDEF1x

В СУБД Microsoft SQL Server была построена диаграмму базы данных. Результат построения представлен на рисунке 2.7.

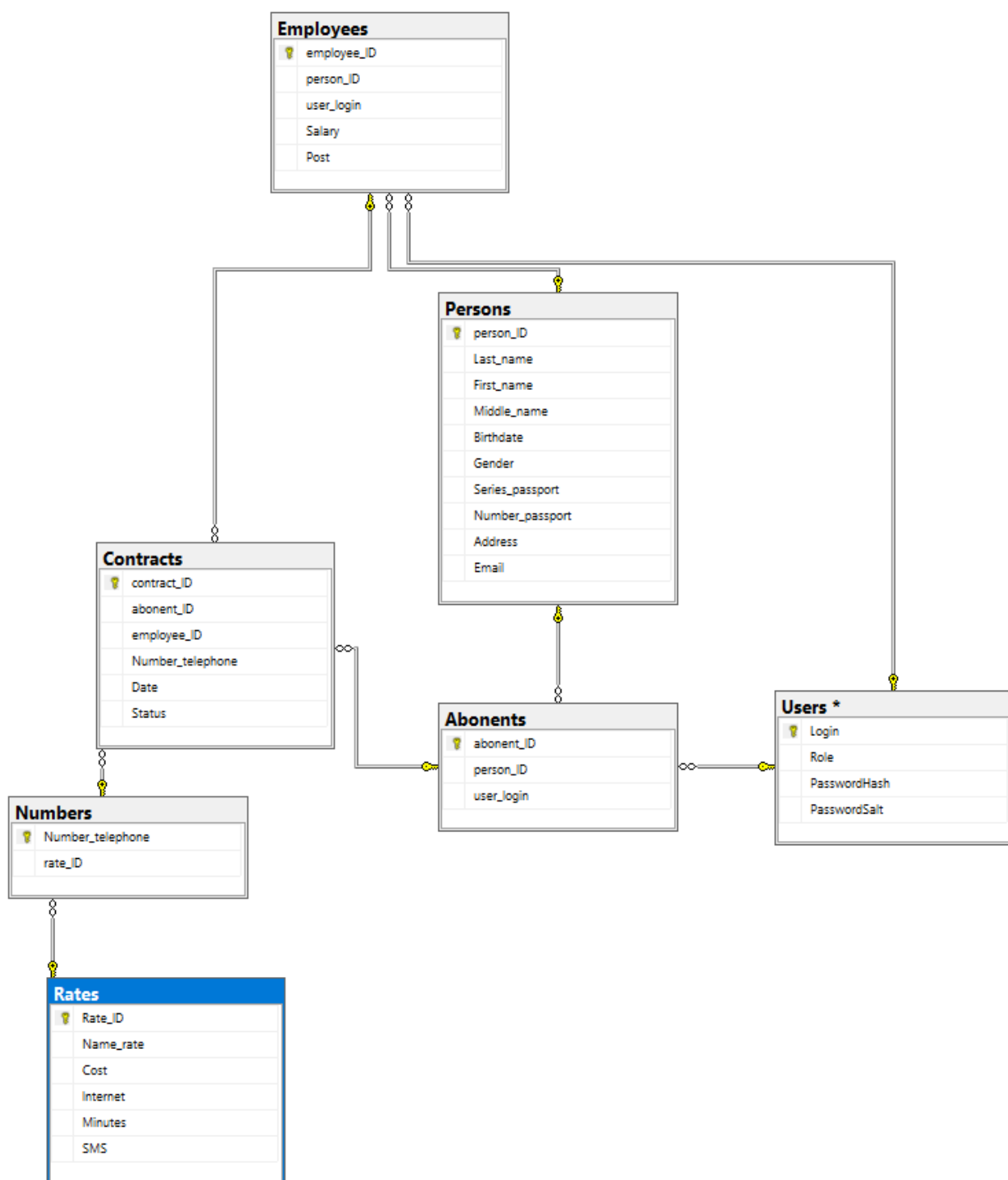


Рисунок 2.7 – Диаграмма базы данных

## 2.3 Представление физической модели данных на языке SQL

Физическая модель данных на языке SQL представлена в приложении А.

## 3 Программный комплекс для работы с БД

### 3.1 Описание прикладной программы

#### 3.1.1 Окно авторизации и главное окно приложения

Перед началом использования приложения, необходимо авторизоваться. Окно авторизации представлено на рисунке 3.1. Оно содержит два поля – для ввода логина и для ввода пароля. После ввода данных необходимо нажать на кнопку «Log in». Если данные введены корректно и пользователь существует, откроется окно авторизации, в противном случае появится сообщение о неверно введенных данных.

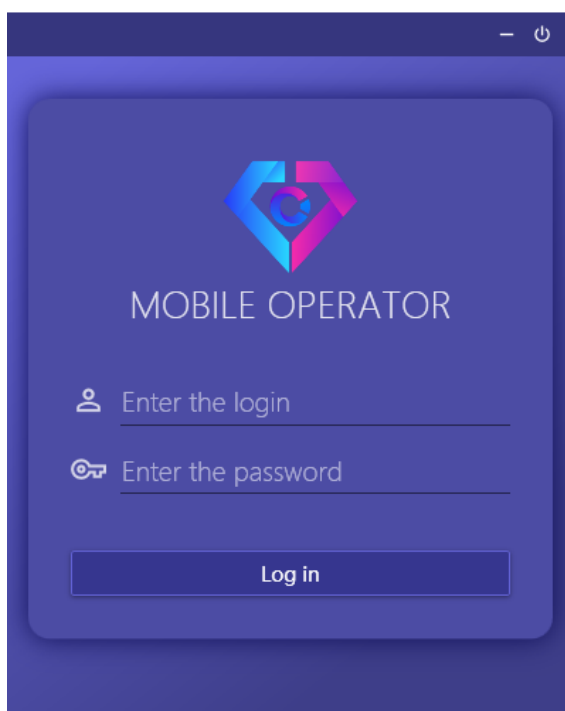


Рисунок 3.1 – Окно авторизации

В открывшемся главном окне приложения слева на панели расположено 7 кнопок: «Abonents», «Rates», «Employees», «Contracts», «Users», «Profile», «Log out» (Рисунок 3.2). Каждая кнопка отвечает за определенные действия. После нажатия на кнопку «Log out» происходит выход из учетной записи и открытие окна с авторизацией.

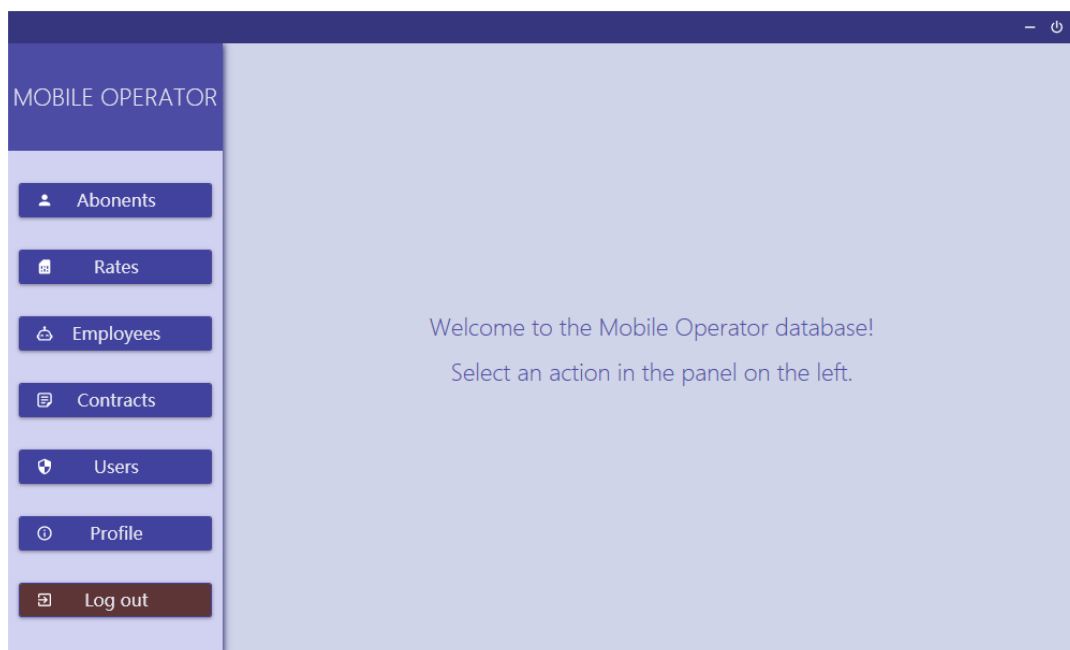


Рисунок 3.2 – Главное окно приложения

### 3.1.2 Окно «Abonents»

Если нажать на кнопку «Abonents» на главном окне приложения, то появится страница с таблицей, в которой отобразится информация о всех абонентах (Рисунок 3.3). Можно выбрать одного или нескольких абонентов и если нажать на кнопку «Delete data», то информация о них полностью удалится из базы данных.

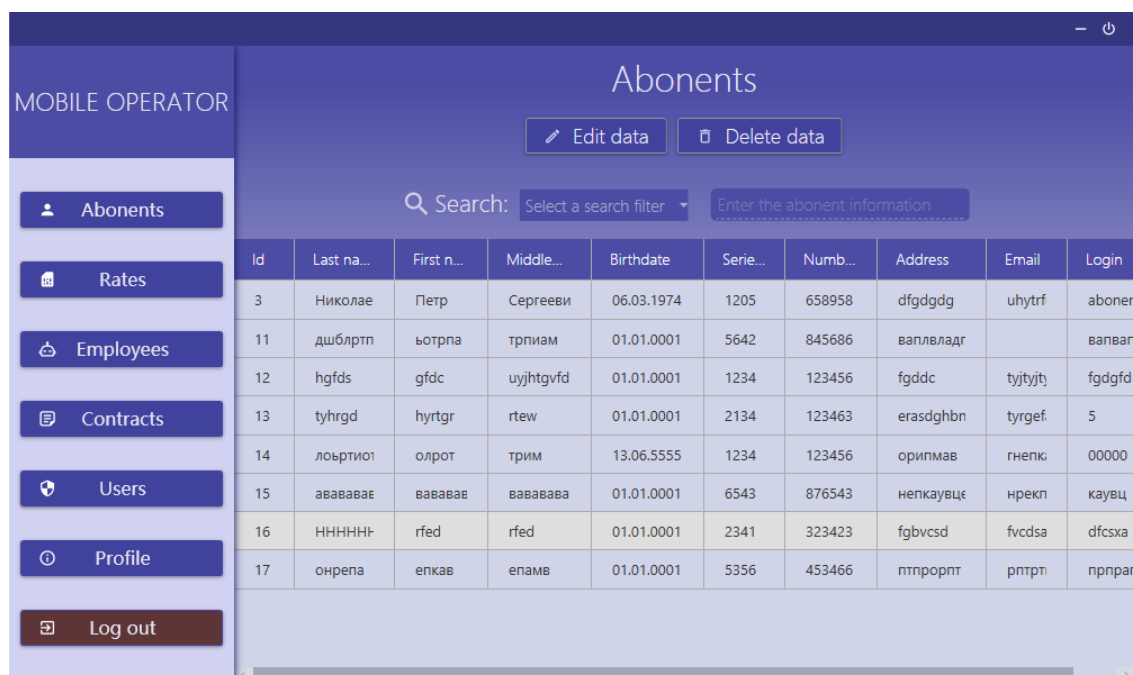


Рисунок 3.3 – Страница «Abonents»



С помощью кнопки «Edit data», можно отредактировать информацию о выбранном из таблицы абоненте (Рисунок 3.4). Также на странице с абонентами можно воспользоваться удобным поиском по выбранному фильтру (Рисунок 3.5).

**MOBILE OPERATOR**

### Edit abonent

Last name:

First name:

Middle name:

Birthdate:

Gender:

Passport series:

Passport number:

Address:

Email:

Рисунок 3.4 – Страница редактирования информации об абонентах

**MOBILE OPERATOR**

### Abonents

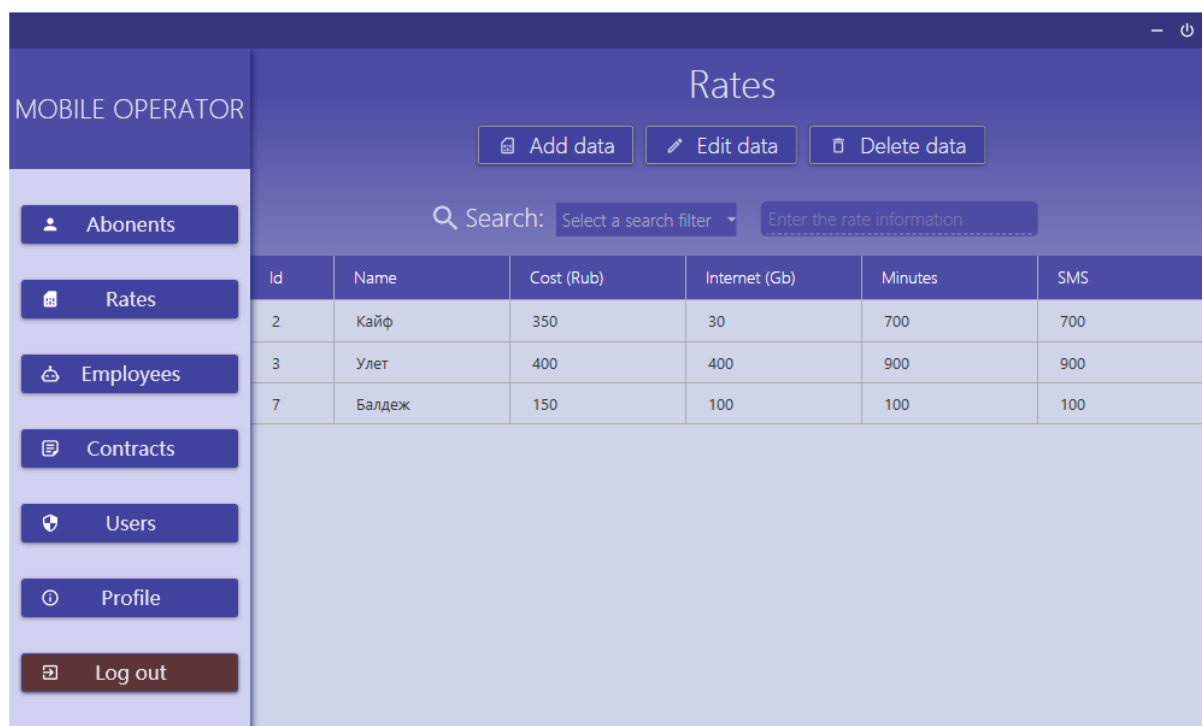
Search:

Id	Last na...	First n...	Middle...	Birthdate	Serie...	Numb...	Address	Email	Login
3	Николаев	Петр	Сергеевич	06.03.1974	1205	658958	dfgdgdg	uhytrf	abor
13	tyhrgd	hyrtgr	rtew	01.01.0001	2134	123463	erasdghbn	tyrgef	5

Рисунок 3.5 – Поиск по таблице на странице абонентов

### 3.1.3 Окно «Rates»

Если нажать на кнопку «Rates» на главном окне приложения, то появится страница с таблицей, в которой отобразится информация о всех тарифах (Рисунок 3.6). Можно выбрать одни или несколько тарифов и если нажать на кнопку «Delete data», то информация о них полностью удалится из базы данных.



Id	Name	Cost (Rub)	Internet (Gb)	Minutes	SMS
2	Кайф	350	30	700	700
3	Улет	400	400	900	900
7	Балдеж	150	100	100	100

Рисунок 3.6 – Страница «Rates»

Если нажать на кнопку «Add data», то откроется страница с полями ввода данных для нового тарифа. С помощью кнопки «Edit data», можно отредактировать информацию о выбранном из таблицы тарифе (Рисунок 3.7). Также на странице с тарифами можно воспользоваться удобным поиском по выбранному фильтру.

MOBILE OPERATOR

Edit rate

Abonents

Rates

Employees

Contracts

Users

Profile

Log out

Name

Enter the rate's name  
Кайф

Cost

Enter the rate's cost  
350

Internet

Enter the rate's internet  
30

Minutes

Enter the rate's minutes  
700

SMS

Enter the rate's SMS  
700

Back

Save

Рисунок 3.7 – Страница редактирования информации о тарифах

### 3.1.4 Окно «Employees»

Если нажать на кнопку «Employees» на главном окне приложения, то появится страница с таблицей, в которой отобразится информация о всех сотрудниках (Рисунок 3.8). Можно выбрать одного или несколько сотрудников и если нажать на кнопку «Delete data», то информация о них полностью удалится из базы данных.

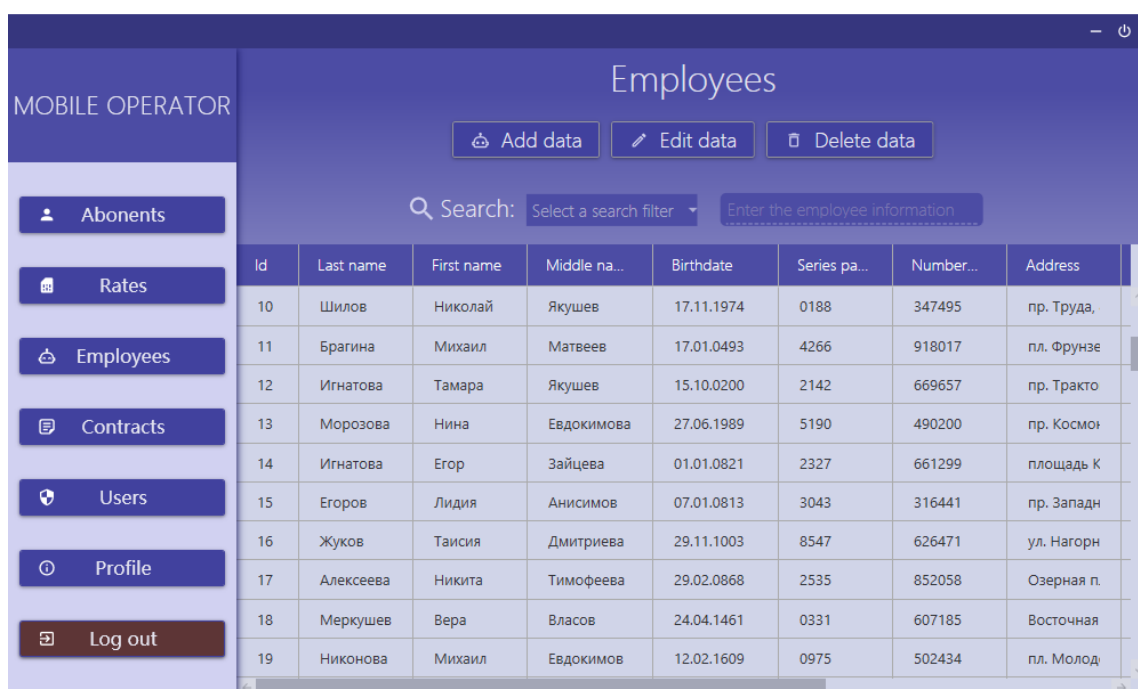


Рисунок 3.8 – Страница «Employees»

Если нажать на кнопку «Add data», то откроется страница с полями ввода данных для нового сотрудника. С помощью кнопки «Edit data», можно отредактировать информацию о выбранном из таблицы сотруднике (Рисунок 3.9). Также на странице с сотрудниками можно воспользоваться удобным поиском по выбранному фильтру.

**Edit employee**

Last name:

First name:

Middle name:

Birthdate:

Gender:

Passport series:

Passport number:

Address:

Email:

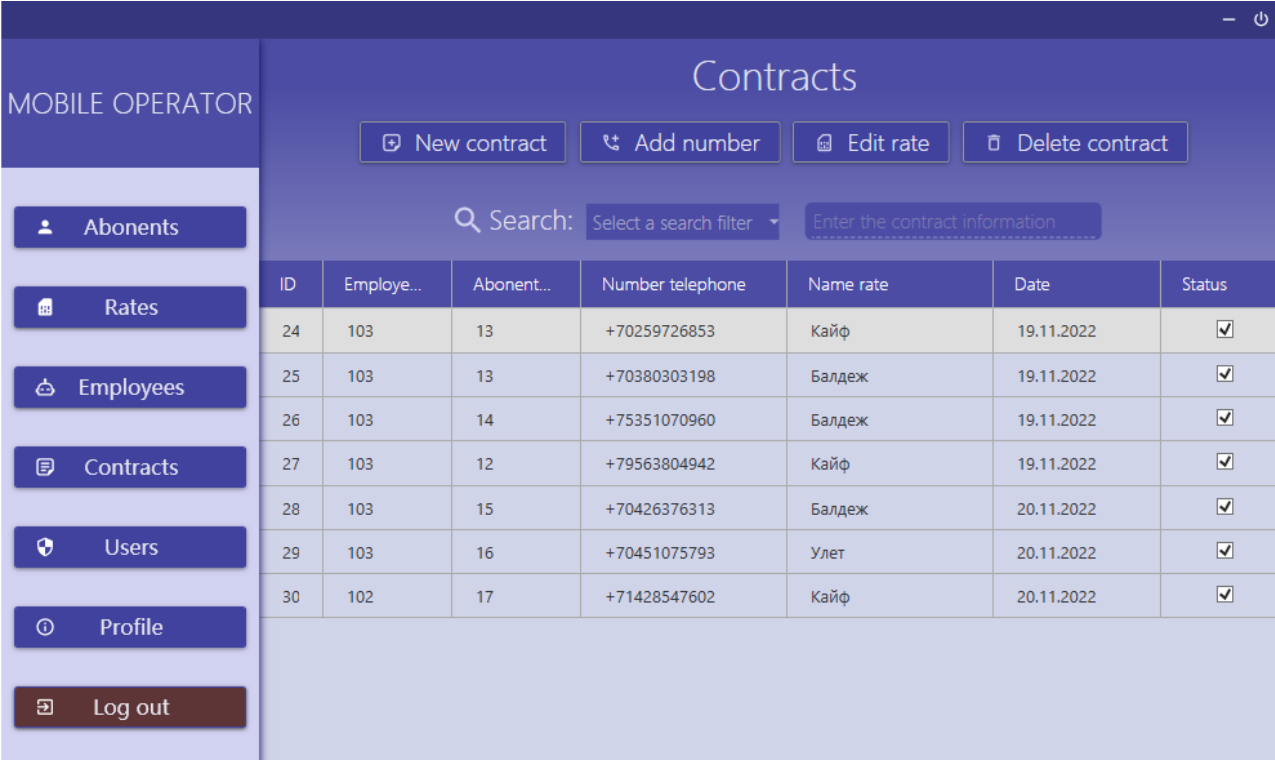
Salary:

Post:

Рисунок 3.9 – Страница редактирования информации о сотрудниках

### 3.1.5 Окно «Contracts»

Если нажать на кнопку «Contracts» на главном окне приложения, то появится страница с таблицей, в которой отобразится информация о всех договорах (Рисунок 3.10). Можно выбрать один или несколько договоров и если нажать на кнопку «Delete contract», то информация о них полностью удалится из базы данных.



ID	Employee...	Abonent...	Number telephone	Name rate	Date	Status
24	103	13	+70259726853	Кайф	19.11.2022	<input checked="" type="checkbox"/>
25	103	13	+70380303198	Балдеж	19.11.2022	<input checked="" type="checkbox"/>
26	103	14	+75351070960	Балдеж	19.11.2022	<input checked="" type="checkbox"/>
27	103	12	+79563804942	Кайф	19.11.2022	<input checked="" type="checkbox"/>
28	103	15	+70426376313	Балдеж	20.11.2022	<input checked="" type="checkbox"/>
29	103	16	+70451075793	Улет	20.11.2022	<input checked="" type="checkbox"/>
30	102	17	+71428547602	Кайф	20.11.2022	<input checked="" type="checkbox"/>

Рисунок 3.10 – Страница «Contracts»

Если нажать на кнопку «New contract», то откроется страница с полями для ввода данных и таблицей свободных номеров для нового абонента (Рисунок 3.11). После заполнения данных можно выбрать тариф для выбранного номера на следующей странице, которая появляется после нажатия на кнопку «Save» (Рисунок 3.12). С помощью кнопки «Edit rate» на открывшейся странице можно сменить тариф выбранному из таблицы номеру (Рисунок 3.12).

**MOBILE OPERATOR**

**New contract**

Abonents Rates Employees Contracts Users Profile Log out

Last name: Enter the abonent's last name  
Иванов

First name: Enter the abonent's first name  
Иван

Middle name: Enter the abonent's middle name  
Иванович

Birthdate: Enter the abonent's birthdate  
05.03.1998

Gender: Мужской

Passport series: Enter the abonent's series passport  
6543

Passport number: Enter the abonent's number passport  
765432

Address: Enter the abonent's address  
Томск, улица Федора Лыткина, дом 103

Email: Enter the abonent's email

Login: Enter the abonent's login  
ivanov

Password: Enter the abonent's password  
ivanov

Number telephone: Number telephone  
+76295305823  
+76460853223  
+76699737575

Back Save

Рисунок 3.11 – Страница добавления нового абонента

**MOBILE OPERATOR**

**Rates**

Search: Select a search filter Enter the rate information

Id	Name	Cost (Rub)	Internet (Gb)	Minutes	SMS
2	Кайф	350	30	700	700
3	Улет	400	400	900	900
7	Балдеж	150	100	100	100

No rate Save

Рисунок 3.12 – Страница подключения тарифов

С помощью кнопки «Add number» можно подключить новый номер уже существующему абоненту. Для этого на появившейся странице нужно выбрать

абонента, которому нужно подключить новый номер, и свободный номер телефона (Рисунок 3.13). После этого можно выбрать для него тариф.

Также на странице с контрактами можно воспользоваться удобным поиском по выбранному фильтру.

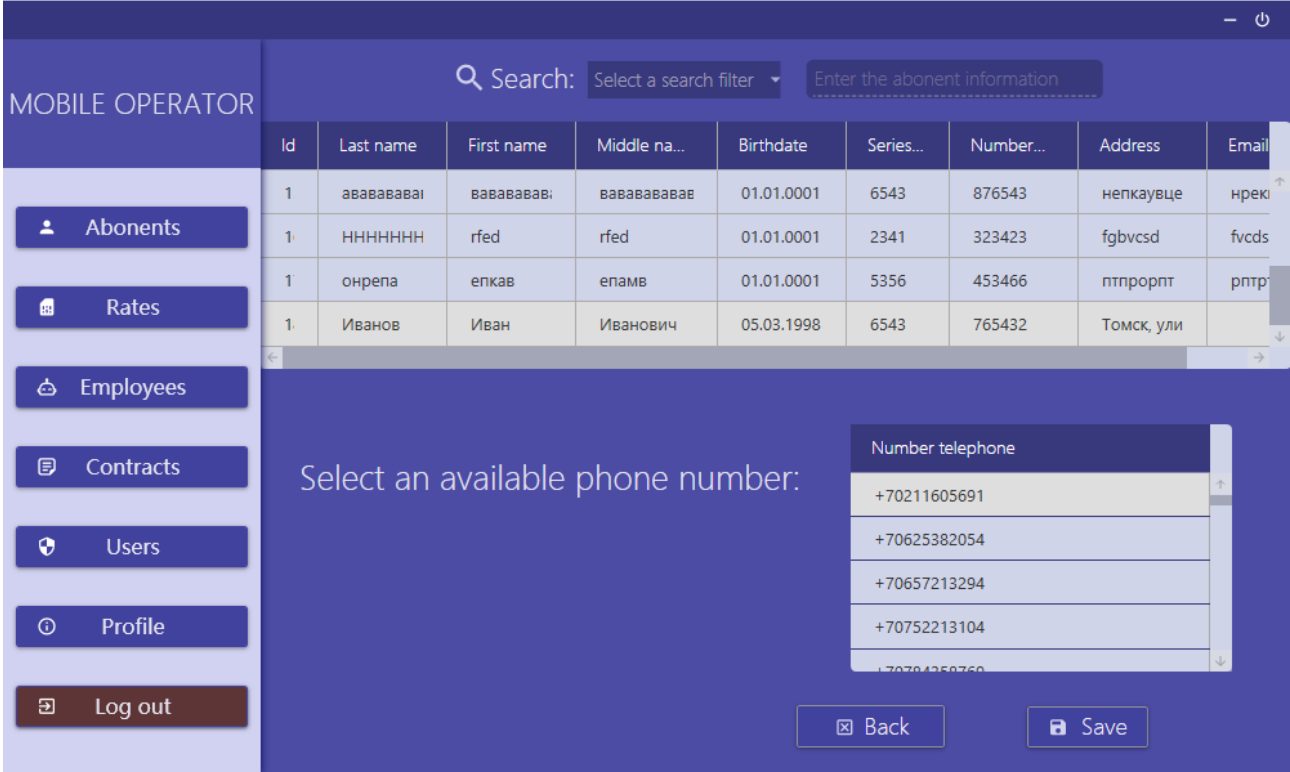


Рисунок 3.13 – Страница подключения нового номера абоненту

### 3.1.6 Окно «Users»

Если нажать на кнопку «Users» на главном окне приложения, то появится страница с таблицей, в которой отобразятся логины, роли, пароли в хэшированном виде и «соль» всех пользователей базы данных (Рисунок 3.14).

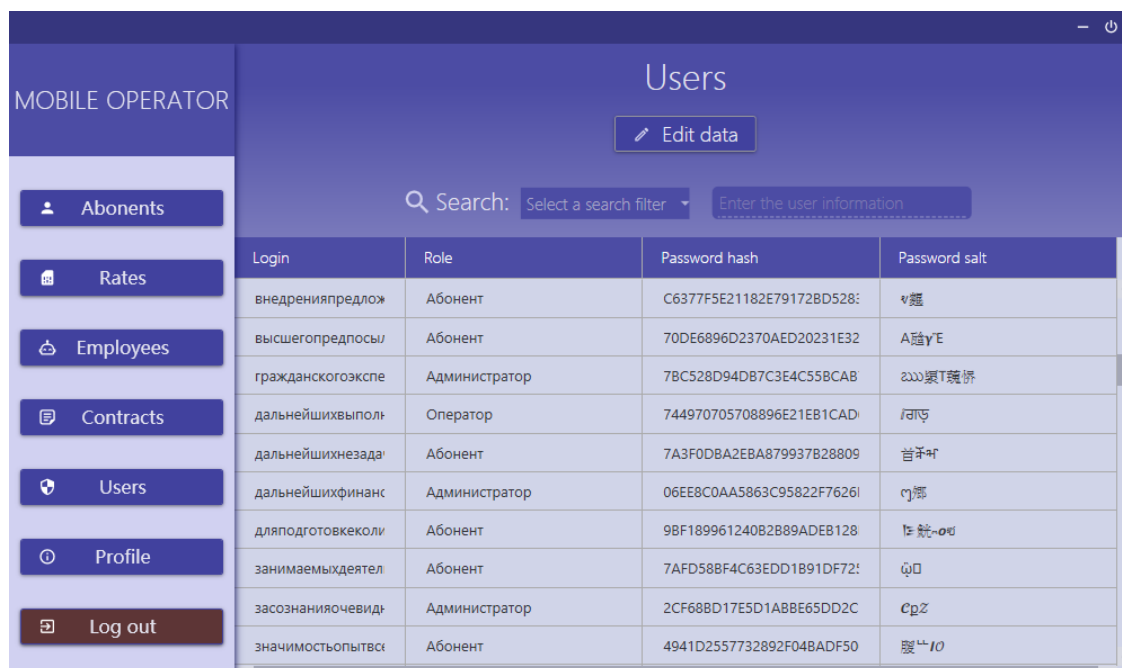


Рисунок 3.14 – Страница «Users»

Если нажать на кнопку «Edit data», то откроется страница с полями ввода данных для нового сотрудника. С помощью кнопки «Edit data», можно отредактировать информацию о выбранном из таблицы пользователе базы данных (Рисунок 3.15).

MOBILE OPERATOR

Edit user

Login: Enter the user's login  
2

Role: Администратор

Password: Enter the user's pass

Back Save

Рисунок 3.15 – Страница редактирования информации о пользователе



### 3.1.7 Окно «Profile»

Если нажать на кнопку «Profile» на главном окне приложения, то появится страница с информацией о текущем пользователе (Рисунок 3.16). На данной странице пользователь может также сменить пароль, нажав на кнопку «Change password» (Рисунок 3.17).

MOBILE OPERATOR	
Abonents	
Rates	
Employees	
Contracts	
Users	
Profile	
Log out	

Profile			
Last name:	Измаилов	Address:	Московская область, город Москва, Подмосковский район, улица Подмосковская дом 56
First name:	Владимир		
Middle name:	Игнатьевич	Email:	izmailoffvladig@mail.ru
Birthdate:	25.09.1960	Login:	3
Gender:	Мужской	Salary:	50000
Passport series:	6475	Post:	Менеджер
Passport number:	543467		

Change password

Рисунок 3.16 – Страница «Profile»

MOBILE OPERATOR	
Abonents	
Rates	
Employees	
Contracts	
Users	
Profile	
Log out	

Profile			
Last name:	Измаилов	Address:	Московская область, город Москва, Подмосковский район, улица Подмосковская дом 56
First name:	Владимир		
Middle name:	Игнатьевич	Email:	izmailoffvladig@mail.ru
Birthdate:	25.09.1960	Login:	3
Gender:	Мужской	Salary:	50000
Passport series:	6475	Post:	Менеджер
Passport number:	543467	New pass:	Enter the password

Change password

Рисунок 3.17 – Страница «Profile» смена пароля

## 3.2 Руководства администратора

Пользователям для работы с платформой необходимы базовые знания ПК.

Администратору необходимы базовые знания администрирования системы.

Для эксплуатации разрабатываемой информационной системы необходимы следующие условия:

- компьютер под управлением операционной системы Windows 10 и выше;
- питание устройства от сети или батареи;
- наличие дисплея или монитора.

Доступ к работе с интерфейсом системы имеют только авторизованные пользователи.

Администраторы систем не имеют технической возможности узнать пользовательские пароли (пароли хранятся в системе в зашифрованном виде).

Для функционирования системы необходимо техническое обеспечение с характеристиками, описанными в таблице 3.1.

Таблица 3.1 – Минимальные требования к техническому обеспечению клиентской части

Компонент	Требования
Процессор	Тактовая частота 1 ГГц и выше.
Память	ОЗУ 128 Мб.
Свободное пространство	30 Мб.
Дисплей и периферийные устройства	Разрешение экрана не менее 1280x720.

Для функционирования системы на сервере должно быть установлено следующее программное обеспечение: СУБД Microsoft SQL Server и библиотеки, представленные на рисунке 3.18.

Bogus.dll	28.03.2022 3:01	Расширение при...	2 364 КБ
EntityFramework.dll	23.10.2017 12:15	Расширение при...	5 103 КБ
EntityFramework.SqlServer.dll	23.10.2017 12:15	Расширение при...	608 КБ
EntityFramework.SqlServer.xml	23.10.2017 12:15	Документ XML	155 КБ
EntityFramework.xml	23.10.2017 12:15	Документ XML	3 594 КБ
MaterialDesignColors.dll	28.11.2021 12:05	Расширение при...	295 КБ
MaterialDesignColors.pdb	28.11.2021 12:05	Program Debug D...	18 КБ
MaterialDesignThemes.Wpf.dll	28.11.2021 12:05	Расширение при...	8 577 КБ
MaterialDesignThemes.Wpf.pdb	28.11.2021 12:05	Program Debug D...	100 КБ
MaterialDesignThemes.Wpf.xml	28.11.2021 12:05	Документ XML	101 КБ
MobileOperator.exe	17.12.2022 15:06	Приложение	502 КБ
MobileOperator.exe.config	19.11.2022 0:55	XML Configuratio...	2 КБ
MobileOperator.pdb	17.12.2022 15:06	Program Debug D...	382 КБ

Рисунок 3.18 – Библиотеки для приложения

### 3.3 Руководство программиста

Проект состоит из классов, представленных на рисунке 3.19. Для более удобного просмотра структуры проекта все класса размещены по папкам. Листинг кода авторизации представлен в приложении Б.

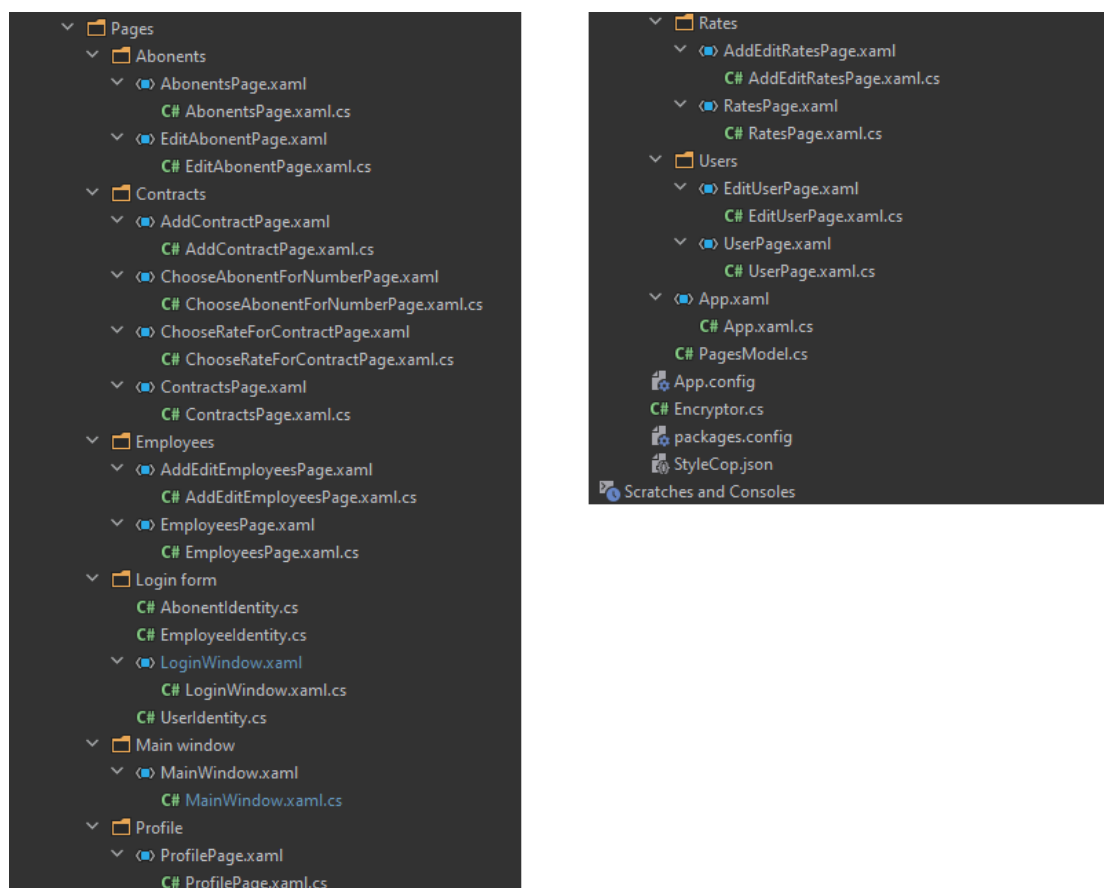


Рисунок 3.19 – Структура проекта

## Заключение

В результате выполнения курсовой работы, были получены навыки проектирования и разработки автоматизированных систем, а также:

- неформально описано ПО с использованием естественного языка;
- описаны бизнес-процессы ПО в методологии функционального моделирования IDEF0 до и после внедрения автоматизированной информационной системы;
- построена концептуальная информационная модель данных для ПО;
- определены типы пользователей и их ограничения по работе с объектами ПО;
- определены отношения и связи между отношениями на основе концептуальной информационной модели и выделены первичные и внешние ключи;
- составлена логическая модель данных в методологии IDEF1x;
- определены типы данных атрибутов отношений;
- составлена физическая модель данных в методологии IDEF1x;
- составлена физическая модель данных на языке SQL;
- разработан программный комплекс для работы с БД для мобильного оператора;
- составлены руководства администратора, пользователя и программиста для программного комплекса.

## **Список использованных источников**

1 ОС ТУСУР 01-2021. Библиографическая ссылка. РАБОТЫ СТУДЕНЧЕСКИЕ ПО НАПРАВЛЕНИЯМ ПОДГОТОВКИ И СПЕЦИАЛЬНОСТЯМ ТЕХНИЧЕСКОГО ПРОФИЛЯ. Общие требования и правила оформления. – Томск: Томский гос. ун-т систем упр. и радиоэлектроники, 2021. – 52с.

2 Курс: Безопасность систем баз данных (1 и 2 семестры). [Электронный ресурс]: Система дистанционного образования ТУСУР. URL: <https://sdo.tusur.ru/course/view.php?id=2121> (дата обращения 03.12.2022).

# Приложение А

(обязательное)

## Физическая модель данных на языке SQL

```
USE [MobileOperator2022]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Abonents](
    [abonent_ID] [int] IDENTITY(0,1) NOT NULL,
    [person_ID] [int] NOT NULL,
    [user_login] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Abonents] PRIMARY KEY CLUSTERED
(
    [abonent_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Contracts](
    [contract_ID] [int] IDENTITY(0,1) NOT NULL,
    [abonent_ID] [int] NOT NULL,
    [employee_ID] [int] NULL,
    [Number_telephone] [nvarchar](12) NOT NULL,
    [Date] [date] NOT NULL,
    [Status] [bit] NOT NULL,
    CONSTRAINT [PK_Contracts] PRIMARY KEY CLUSTERED
(
    [contract_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Employees](
    [employee_ID] [int] IDENTITY(0,1) NOT NULL,
    [person_ID] [int] NOT NULL,
    [user_login] [nvarchar](50) NOT NULL,
    [Salary] [int] NOT NULL,
    [Post] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_Employees] PRIMARY KEY CLUSTERED
(
    [employee_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Numbers](
    [Number_telephone] [nvarchar](12) NOT NULL,
    [rate_ID] [smallint] NULL,
    CONSTRAINT [PK_Numbers] PRIMARY KEY CLUSTERED
(
    [Number_telephone] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```

CREATE TABLE [dbo].[Persons](
    [person_ID] [int] IDENTITY(0,1) NOT NULL,
    [Last_name] [nvarchar](50) NOT NULL,
    [First_name] [nvarchar](50) NOT NULL,
    [Middle_name] [nvarchar](50) NULL,
    [Birthdate] [date] NOT NULL,
    [Gender] [nvarchar](50) NOT NULL,
    [Series_passport] [nvarchar](4) NOT NULL,
    [Number_passport] [nvarchar](6) NOT NULL,
    [Address] [nvarchar](300) NOT NULL,
    [Email] [nvarchar](100) NULL,
    CONSTRAINT [PK_Persons] PRIMARY KEY CLUSTERED
(
    [person_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Rates](
    [Rate_ID] [smallint] IDENTITY(0,1) NOT NULL,
    [Name_rate] [nvarchar](100) NOT NULL,
    [Cost] [smallint] NOT NULL,
    [Internet] [smallint] NOT NULL,
    [Minutes] [smallint] NOT NULL,
    [SMS] [smallint] NOT NULL,
    CONSTRAINT [PK_Rates] PRIMARY KEY CLUSTERED
(
    [Rate_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Users](
    [Login] [nvarchar](50) NOT NULL,
    [Role] [nvarchar](50) NOT NULL,
    [PasswordHash] [nvarchar](44) NOT NULL,
    [PasswordSalt] [nvarchar](10) NOT NULL,
    CONSTRAINT [PK_Users] PRIMARY KEY CLUSTERED
(
    [Login] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Contracts] ADD CONSTRAINT [DF_Contracts_Status] DEFAULT ((1)) FOR [Status]
GO
ALTER TABLE [dbo].[Rates] ADD CONSTRAINT [DF_Rates_Cost] DEFAULT ((0)) FOR [Cost]
GO
ALTER TABLE [dbo].[Rates] ADD CONSTRAINT [DF_Rates_Internet] DEFAULT ((0)) FOR [Internet]
GO
ALTER TABLE [dbo].[Rates] ADD CONSTRAINT [DF_Rates_Minutes] DEFAULT ((0)) FOR [Minutes]
GO
ALTER TABLE [dbo].[Rates] ADD CONSTRAINT [DF_Rates_SMS] DEFAULT ((0)) FOR [SMS]
GO
ALTER TABLE [dbo].[Abonents] WITH CHECK ADD CONSTRAINT [FK_Abonents_Persons] FOREIGN KEY([person_ID])
REFERENCES [dbo].[Persons] ([person_ID])
GO
ALTER TABLE [dbo].[Abonents] CHECK CONSTRAINT [FK_Abonents_Persons]
GO
ALTER TABLE [dbo].[Abonents] WITH CHECK ADD CONSTRAINT [FK_Abonents_Users] FOREIGN KEY([user_login])
REFERENCES [dbo].[Users] ([Login])
GO
ALTER TABLE [dbo].[Abonents] CHECK CONSTRAINT [FK_Abonents_Users]
GO
ALTER TABLE [dbo].[Contracts] WITH CHECK ADD CONSTRAINT [FK_Contracts_Abonents] FOREIGN KEY([abonent_ID])
REFERENCES [dbo].[Abonents] ([abonent_ID])
GO
ALTER TABLE [dbo].[Contracts] CHECK CONSTRAINT [FK_Contracts_Abonents]
GO
ALTER TABLE [dbo].[Contracts] WITH CHECK ADD CONSTRAINT [FK_Contracts_Employees] FOREIGN KEY([employee_ID])
REFERENCES [dbo].[Employees] ([employee_ID])

```

```

GO
ALTER TABLE [dbo].[Contracts] CHECK CONSTRAINT [FK_Contracts_Employees]
GO
ALTER TABLE [dbo].[Contracts] WITH CHECK ADD CONSTRAINT [FK_Contracts_Numbers] FOREIGN KEY([Number_telephone])
REFERENCES [dbo].[Numbers] ([Number_telephone])
GO
ALTER TABLE [dbo].[Contracts] CHECK CONSTRAINT [FK_Contracts_Numbers]
GO
ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [FK_Employees_Persons] FOREIGN KEY([person_ID])
REFERENCES [dbo].[Persons] ([person_ID])
GO
ALTER TABLE [dbo].[Employees] CHECK CONSTRAINT [FK_Employees_Persons]
GO
ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [FK_Employees_Users] FOREIGN KEY([user_login])
REFERENCES [dbo].[Users] ([Login])
GO
ALTER TABLE [dbo].[Employees] CHECK CONSTRAINT [FK_Employees_Users]
GO
ALTER TABLE [dbo].[Numbers] WITH CHECK ADD CONSTRAINT [FK_Numbers_Rates] FOREIGN KEY([rate_ID])
REFERENCES [dbo].[Rates] ([Rate_ID])
GO
ALTER TABLE [dbo].[Numbers] CHECK CONSTRAINT [FK_Numbers_Rates]
GO
ALTER TABLE [dbo].[Abonents] WITH CHECK ADD CONSTRAINT [CK_abonent_ID] CHECK (([abonent_ID]>=(0)))
GO
ALTER TABLE [dbo].[Abonents] CHECK CONSTRAINT [CK_abonent_ID]
GO
ALTER TABLE [dbo].[Abonents] WITH CHECK ADD CONSTRAINT [CK_Abonents_person_ID] CHECK (([person_ID]>=(0)))
GO
ALTER TABLE [dbo].[Abonents] CHECK CONSTRAINT [CK_Abonents_person_ID]
GO
ALTER TABLE [dbo].[Contracts] WITH CHECK ADD CONSTRAINT [CK_contract_ID] CHECK (([contract_ID]>=(0)))
GO
ALTER TABLE [dbo].[Contracts] CHECK CONSTRAINT [CK_contract_ID]
GO
ALTER TABLE [dbo].[Contracts] WITH CHECK ADD CONSTRAINT [CK_Contracts_abonent_ID] CHECK (([abonent_ID]>=(0)))
GO
ALTER TABLE [dbo].[Contracts] CHECK CONSTRAINT [CK_Contracts_abonent_ID]
GO
ALTER TABLE [dbo].[Contracts] WITH CHECK ADD CONSTRAINT [CK_Contracts_employee_ID] CHECK (([employee_ID]>=(0)))
GO
ALTER TABLE [dbo].[Contracts] CHECK CONSTRAINT [CK_Contracts_employee_ID]
GO
ALTER TABLE [dbo].[Contracts] WITH CHECK ADD CONSTRAINT [CK_Contracts_Number_telephone] CHECK
((len([Number_telephone])=12)))
GO
ALTER TABLE [dbo].[Contracts] CHECK CONSTRAINT [CK_Contracts_Number_telephone]
GO
ALTER TABLE [dbo].[Contracts] WITH CHECK ADD CONSTRAINT [CK_Status] CHECK (([Status]=(1) OR [Status]=(0)))
GO
ALTER TABLE [dbo].[Contracts] CHECK CONSTRAINT [CK_Status]
GO
ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [CK_employee_ID] CHECK (([employee_ID]>=(0)))
GO
ALTER TABLE [dbo].[Employees] CHECK CONSTRAINT [CK_employee_ID]
GO
ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [CK_Persons_person_ID] CHECK (([person_ID]>=(0)))
GO
ALTER TABLE [dbo].[Employees] CHECK CONSTRAINT [CK_Persons_person_ID]
GO
ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [CK_Post] CHECK (([Post]='Онлайн консультант' OR
[Post]='Администратор' OR [Post]='Оператор' OR [Post]='Менеджер'))
GO
ALTER TABLE [dbo].[Employees] CHECK CONSTRAINT [CK_Post]
GO
ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [CK_Users_user_login] CHECK ((len([user_login])>(0)))
GO
ALTER TABLE [dbo].[Employees] CHECK CONSTRAINT [CK_Users_user_login]
GO
ALTER TABLE [dbo].[Numbers] WITH CHECK ADD CONSTRAINT [CK_rate_ID] CHECK (([rate_ID]>=(-1)))
GO
ALTER TABLE [dbo].[Numbers] CHECK CONSTRAINT [CK_rate_ID]
GO
ALTER TABLE [dbo].[Persons] WITHCHECK ADD CONSTRAINT [CK_Gender] CHECK (([Gender]='Мужской' OR [Gender]='Женский'))
GO
ALTER TABLE [dbo].[Persons] CHECK CONSTRAINT [CK_Gender]
GO
ALTER TABLE [dbo].[Persons] WITH CHECK ADD CONSTRAINT [CK_Number_passport] CHECK ((len([Number_passport])=(6)))
GO

```



```

ALTER TABLE [dbo].[Persons] CHECK CONSTRAINT [CK_Number_passport]
GO
ALTER TABLE [dbo].[Persons] WITH CHECK ADD CONSTRAINT [CK_person_ID] CHECK (([person_ID]>=(0)))
GO
ALTER TABLE [dbo].[Persons] CHECK CONSTRAINT [CK_person_ID]
GO
ALTER TABLE [dbo].[Persons] WITH CHECK ADD CONSTRAINT [CK_Series_passport] CHECK ((len([Series_passport])=(4)))
GO
ALTER TABLE [dbo].[Persons] CHECK CONSTRAINT [CK_Series_passport]
GO
ALTER TABLE [dbo].[Rates] WITH CHECK ADD CONSTRAINT [CK_Cost] CHECK (([Cost]>=(0)))
GO
ALTER TABLE [dbo].[Rates] CHECK CONSTRAINT [CK_Cost]
GO
ALTER TABLE [dbo].[Rates] WITH CHECK ADD CONSTRAINT [CK_Internet] CHECK (([Internet]>=(0)))
GO
ALTER TABLE [dbo].[Rates] CHECK CONSTRAINT [CK_Internet]
GO
ALTER TABLE [dbo].[Rates] WITH CHECK ADD CONSTRAINT [CK_Minutes] CHECK (([Minutes]>=(0)))
GO
ALTER TABLE [dbo].[Rates] CHECK CONSTRAINT [CK_Minutes]
GO
ALTER TABLE [dbo].[Rates] WITH CHECK ADD CONSTRAINT [CK_Rates] CHECK (([Rate_ID]>=(0)))
GO
ALTER TABLE [dbo].[Rates] CHECK CONSTRAINT [CK_Rates]
GO
ALTER TABLE [dbo].[Rates] WITH CHECK ADD CONSTRAINT [CK_SMS] CHECK (([SMS]>=(0)))
GO
ALTER TABLE [dbo].[Rates] CHECK CONSTRAINT [CK_SMS]
GO
ALTER TABLE [dbo].[Users] WITHCHECK ADD CONSTRAINT [CK_Role] CHECK (([Role]='Стажер' OR [Role]='Администратор' OR
[Role]='Оператор' OR [Role]='Абонент'))
GO
ALTER TABLE [dbo].[Users] CHECK CONSTRAINT [CK_Role]
GO

```

## Приложение Б

(обязательное)

### Листинг кода окна авторизации

```
using System.Linq;
using System.Windows;
using System.Windows.Input;

namespace MobileOperator
{
    public partial class LoginWindow : Window
    {
        public LoginWindow()
        {
            InitializeComponent();
        }

        private void BtnExitClick(object sender, RoutedEventArgs e)
        {
            Application.Current.Shutdown();
        }

        private void GridTopRowMouseDown(object sender, MouseButtonEventArgs e)
        {
            if (e.ChangedButton == MouseButton.Left)
            {
                DragMove();
            }
        }

        private void BtnHideClick(object sender, RoutedEventArgs e)
        {
            Application.Current.MainWindow.WindowState = WindowState.Minimized;
        }

        private void BtnAuthClick(object sender, RoutedEventArgs e)
        {
            var login = TxtBoxLogin.Text;
            var pass = TxtBoxPass.Password;

            IQueryable<User> identifiableUser = Context.Get().Users
                .Where(user => user.Login == login);

            if (!identifiableUser.Any())
            {
                MessageBox.Show("Invalid login entered!");
                return;
            }

            var identifiedUser = identifiableUser.Single();
            var encryptor = new Encryptor();
            var enteredPassHash = encryptor.PasswordEncrypt(pass, identifiedUser.PasswordSalt);

            if (enteredPassHash != identifiedUser.PasswordHash)
            {
                MessageBox.Show("Invalid pass entered!");
                return;
            }

            TxtBoxLogin.Text = "";
            TxtBoxPass.Password = "";

            AuthorizeUser(identifiedUser);
        }

        private void AuthorizeUser(User authenticatedUser)
        {
            switch (authenticatedUser.Role)
            {
                case "Администратор":
                {
                    InitPagesForAdmin(authenticatedUser);
                }
            }
        }
    }
}
```

```

        break;
    }

    case "Оператор":
    {
        InitPagesForOperator(authenticatedUser);

        break;
    }

    case "Абонент":
    {
        InitPagesForAbonent(authenticatedUser);

        break;
    }

    default:
    {
        MessageBox.Show("Please contact your administrator with this error");

        return;
    }
}

private void InitPagesForAbonent(User authenticatedUser)
{
    AuthorizeAbonent(authenticatedUser);

    var pages = new PagesModel
    {
        Contracts = new ContractsPage(),
        Rates = new RatesPage(),
        Profile = new ProfilePage(),
        LoginWindow = this,
    };
    pages.Rates.GridRatePage.Visibility = Visibility.Collapsed;
    pages.Rates.RowDefinitionButton.Height = new GridLength(1.0, GridUnitType.Star);
    pages.Contracts.BdrDelContract.Visibility = Visibility.Collapsed;
    pages.Contracts.BdrAddNumber.Visibility = Visibility.Collapsed;
    pages.Contracts.BdrAddContract.Visibility = Visibility.Collapsed;
    var mainWindow = new MainWindow(pages);
    mainWindow.BtnAbonents.Visibility = Visibility.Collapsed;
    mainWindow.BtnEmployees.Visibility = Visibility.Collapsed;
    mainWindow.BtnUsers.Visibility = Visibility.Collapsed;
    Hide();
    mainWindow.Show();
}

private void InitPagesForOperator(User authenticatedUser)
{
    AuthorizeEmployee(authenticatedUser);
    var pages = new PagesModel
    {
        Abonents = new AbonentsPage(),
        Contracts = new ContractsPage(),
        Rates = new RatesPage(),
        Profile = new ProfilePage(),
        LoginWindow = this,
    };
    pages.Rates.GridRatePage.Visibility = Visibility.Collapsed;
    pages.Rates.RowDefinitionButton.Height = new GridLength(1.0, GridUnitType.Star);
    var mainWindow = new MainWindow(pages);
    mainWindow.BtnEmployees.Visibility = Visibility.Collapsed;
    mainWindow.BtnUsers.Visibility = Visibility.Collapsed;
    Hide();
    mainWindow.Show();
}

private void InitPagesForAdmin(User authenticatedUser)
{
    AuthorizeEmployee(authenticatedUser);
    var pages = new PagesModel
    {
        Abonents = new AbonentsPage(),
        Contracts = new ContractsPage(),
        Employees = new EmployeesPage(),
    };
    mainWindow = new MainWindow(pages);
    mainWindow.Show();
}

```

```

        Rates = new RatesPage(),
        User = new UserPage(),
        Profile = new ProfilePage(),
        LoginWindow = this,
    };
    var mainWindow = new MainWindow(pages);
    Hide();
    mainWindow.Show();
}

private void AuthorizeEmployee(User authenticatedUser)
{
    var authenticatedEmployee = Context.Get().Employees
        .Single(employee => employee.user_login == authenticatedUser.Login);
    var authenticatedPerson = Context.Get().Persons
        .Single(person => person.person_ID == authenticatedEmployee.person_ID);
    EmployeeIdentity.employee_ID = authenticatedEmployee.employee_ID;
    EmployeeIdentity.Salary = authenticatedEmployee.Salary;
    EmployeeIdentity.Post = authenticatedEmployee.Post;

    SetUserIdentity(authenticatedPerson, authenticatedUser);
}

private void AuthorizeAbonent(User authenticatedUser)
{
    var authenticatedAbonent = Context.Get().Abonents
        .Single(abonent => abonent.user_login == authenticatedUser.Login);
    var authenticatedPerson = Context.Get().Persons
        .Single(person => person.person_ID == authenticatedAbonent.person_ID);

    AbonentIdentity.abonent_ID = authenticatedAbonent.abonent_ID;
    SetUserIdentity(authenticatedPerson, authenticatedUser);
}

private static void SetUserIdentity(Person person, User authenticatedUser)
{
    UserIdentity.Login = authenticatedUser.Login;
    UserIdentity.Role = authenticatedUser.Role;
    UserIdentity.PasswordSalt = authenticatedUser.PasswordSalt;
    UserIdentity.person_ID = person.person_ID;
    UserIdentity.Last_name = person.Last_name;
    UserIdentity.First_name = person.First_name;
    UserIdentity.Middle_name = person.Middle_name;
    UserIdentity.Birthdate = person.Birthdate;
    UserIdentity.Gender = person.Gender;
    UserIdentity.Series_passport = person.Series_passport;
    UserIdentity.Number_passport = person.Number_passport;
    UserIdentity.Address = person.Address;
    UserIdentity.Email = person.Email;
}
}
}

```