



**МИНОБРНАУКИ РОССИИ**  
**федеральное государственное бюджетное образовательное**  
**учреждение высшего образования**  
**«Московский государственный технологический университет**  
**«СТАНКИН»**  
**(ФГБОУ ВО МГТУ «СТАНКИН»)**

---

**Институт**

информационных систем и технологий

**Кафедра**

информационных систем

**Отчёт по самостоятельной работе**  
**по дисциплине «Управление данными»**  
**на тему: Проектирование базы данных поликлиники**

**Студент**

группа ИДБ-16-07

\_\_\_\_\_ Махмудов Б.Н.

подпись

**Руководитель**

Старший преподаватель

\_\_\_\_\_ Быстрикова В.А.

подпись

Москва 2018

# Оглавление

<b>1</b>	<b>Анализ предметной области</b>	<b>4</b>
1.1	Определение анализа предметной области . . . . .	4
1.1.1	Предметная область . . . . .	4
1.1.2	Информационный анализ предметной области . . . . .	4
1.2	Поликлиника: описание предметной области . . . . .	5
1.3	Существующие продукты решающие проблему автоматизации .	7
1.3.1	1С : Медицина. Поликлиника . . . . .	7
1.3.2	Сайт частных поликлиник «СМ-Клиника» . . . . .	10
1.3.3	Сравнение «1С:Медицина. Поликлиника» и сайта поли- клиники «СМ-Клиника» . . . . .	12
1.4	Функции, планируемые для реализации в курсовом проекте . .	13
<b>2</b>	<b>Концептуальное проектирование</b>	<b>14</b>
2.1	Определение концептуального проектирования . . . . .	14
2.2	Концептуальная модель базы данных поликлиники . . . . .	15
<b>3</b>	<b>Логическое проектирование</b>	<b>18</b>
3.1	Определение логического проектирования . . . . .	18
3.2	ER-модель базы данных поликлиники . . . . .	19
3.3	Преобразование ER-модели в реляционную . . . . .	22
<b>4</b>	<b>Физическое проектирование</b>	<b>27</b>
4.1	Выбор СУБД . . . . .	27

4.2	Схема БД поликлиники . . . . .	28
	<b>Заключение</b>	<b>32</b>
	<b>Литература</b>	<b>32</b>
	<b>Приложение А</b>	<b>34</b>
	SQL операторы создания таблиц . . . . .	34
	Примеры заполнения таблиц . . . . .	37

# **Глава 1    Анализ предметной области**

## **1.1    Определение анализа предметной области**

### **1.1.1    Предметная область**

Предметная область — множество всех предметов, свойства которых и отношения между которыми рассматриваются в научной теории. В логике — подразумеваемая область возможных значений предметных переменных логического языка.

Предметная область — часть реального мира, рассматриваемая в пределах данного контекста. Под контекстом здесь может пониматься, например, область исследования или область, которая является объектом некоторой деятельности.[1]

### **1.1.2    Информационный анализ предметной области**

Деятельность, направленная на выявление реальных потребностей заказчика, а также на выяснения смысла высказанных требований, называется анализом предметной области. Одна из первых задач, с решением которых сталкивается разработчик программной системы - это изучение, осмысление и анализ предметной области. Дело в том, что предметная область сильно влияет на все аспекты проекта: требования к системе, взаимодействие с пользователем, модель хранения данных, реализацию и т.д. Анализ предметной области, позволяет выделить ее сущности, определить первоначальные требования к функциональности и определить границы проекта.

## 1.2 Поликлиника: описание предметной области

Поликлиника — многопрофильное или специализированное лечебно-профилактическое учреждение для оказания амбулаторной медицинской помощи больным на приёме и на дому. На территории России распределены по территориальному признаку, и являются базовым уровнем медицинского обслуживания населения. По мощности городские поликлиники делятся на 5 групп. В структуре городской поликлиники предусматриваются следующие подразделения:

1. руководство поликлиникой;
2. регистратура;
3. кабинет доврачебного приема;
4. отделение профилактики;
5. лечебно-профилактические подразделения:
6. терапевтические отделения;
7. отделение восстановительного лечения;
8. отделения по оказанию специализированных видов медицинской помощи (хирургическое, гинекологическое) с кабинетами соответствующих специалистов (кардиологический, ревматологический, неврологический, урологический, офтальмологический, оториноларингологический);
9. параклинические службы (физиотерапевтический и рентгеновский кабинеты, лаборатории, кабинет функциональной диагностики, УЗИ-кабинет);
10. дневной стационар и стационар на дому;
11. административно-хозяйственная часть;
12. врачебные и фельдшерские здравпункты на прикрепленных предприятиях.

Число отделений и кабинетов, их потенциальные возможности определяются мощностью поликлиники и количеством штатных должностей, которые зависят от численности закрепленного за поликлиникой населения. Структура поликлиники (открытие тех или иных отделений, кабинетов и т. п.) зависит от обращаемости населения в это учреждение, от способности поликлиники предоставить больным необходимую медицинскую помощь.[2]

На сегодняшний день автоматизации подвержено подавляющее большинство сфер деятельности человека, включая здравоохранение. Автоматизация здравоохранения особенно актуальна ввиду роста человеческого населения и бюрократизации в сфере оказания медицинских услуг, что приводит к неудобствам и затруднениям для больных в получении вышеупомянутых услуг. Но если разработать информационную систему с централизованной базой данных, позволяющую пользователям удалённо получать справки и записываться на приём к врачам, то можно уменьшить нагрузку самого учреждения и улучшить качество услуг для пациентов. Таким образом автоматизация функционирования поликлиники, в частности разработка базы данных для неё позволит пациентам сэкономить время на очередях и бюрократических формальностях, а сотрудникам сосредоточиться непосредственно на оказании медицинских услуг.

## 1.3 Существующие продукты решающие проблему автоматизации

### 1.3.1 1С : Медицина. Поликлиника

Прикладное решение «1С:Медицина. Поликлиника» предназначено для автоматизации основных процессов медицинских организаций различных организационно-правовых форм, оказывающих медицинскую помощь в амбулаторно-поликлинических условиях.

**Функциональные возможности.** Прикладное решение «1С:Медицина. Поликлиника» позволяет создать единое информационное пространство медицинской организации с разделением доступа к данным по ролевому принципу. Имеется возможность вести учет по нескольким медицинским организациям в одной информационной базе.

Программа позволяет вести несколько медицинских карт для одного пациента - амбулаторную карту, стоматологическую карту и т.д., пример карты пациента приведён ниже (Рис. 1.1). Для каждого медицинского работника указывается, к какому типу карт он имеет доступ. В программе имеются гибкие механизмы квотирования, которые позволяют устанавливать ограничения на объемы оказываемой медицинской помощи. Учет деятельности медицинского персонала ведется по медицинским услугам. Пример пользовательского интерфейса программы показан на Рис. 1.2

Предварительную запись пациентов может осуществлять как регистратура, так и врачи при выполнении назначений повторных приемов, консультаций, исследований, манипуляций. Для осуществления оперативного планирования врачебному медицинскому персоналу и кабинетам задаются графики работы, нормы загрузки, перечень выполняемых услуг. Оперативное планирование деятельности кабинетов осуществляется по данным предварительной записи пациентов.[3]





В целом процесс работы программного продукта «1С:Медицина. Поликлиника» можно описать с помощью диаграммы (Рис. 1.3).

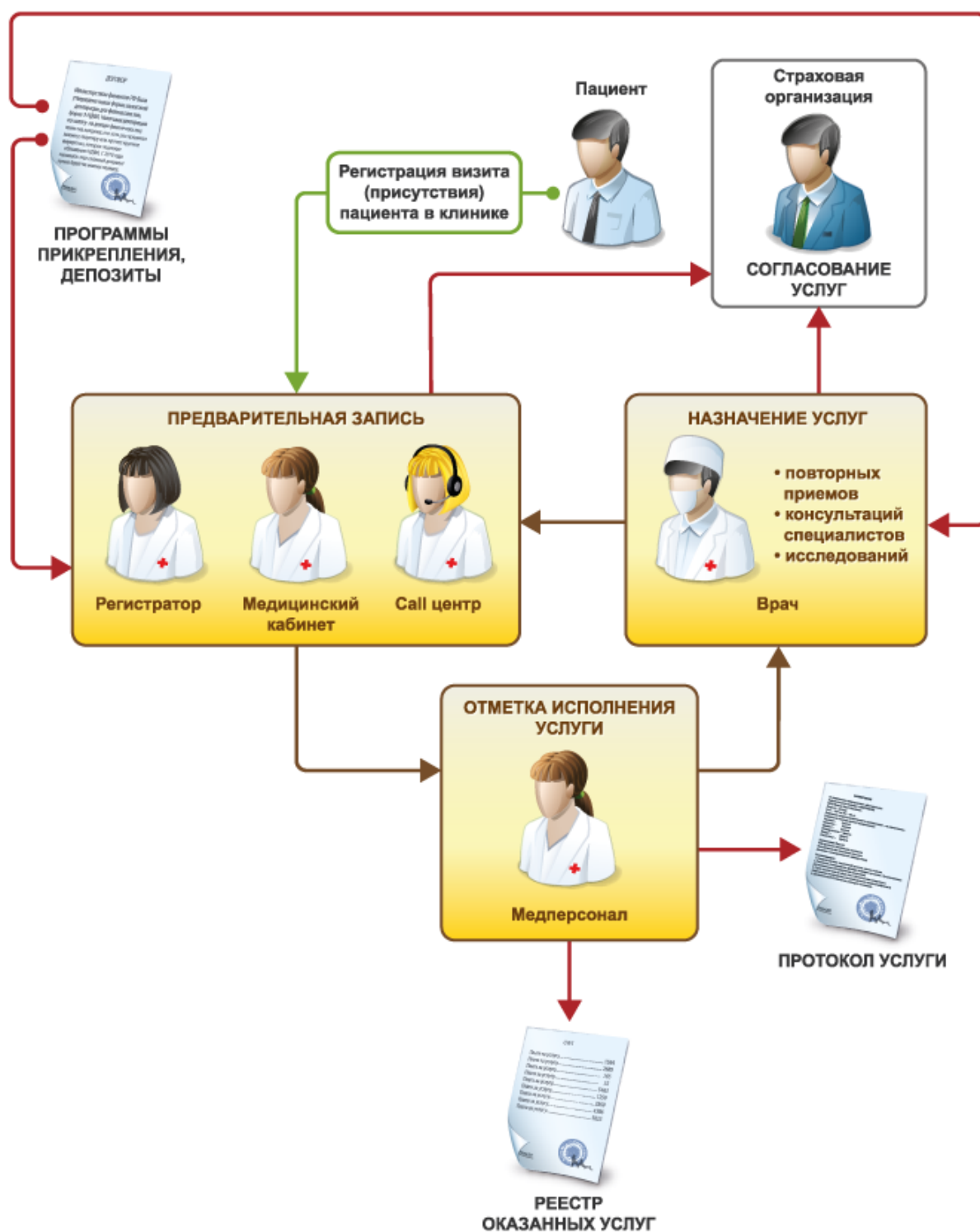


Рис. 1.3: Диаграмма функционирования «1С:Медицина. Поликлиника»

### 1.3.2 Сайт частных поликлиник «СМ-Клиника»

Многопрофильный медицинский холдинг «СМ-Клиника» - это сеть многопрофильных медицинских центров для взрослых и детей, основанной в 2002 году. Сайт компании доступен по адресу <http://www.smclinic.ru/>. Скриншот сайта приведён на Рис. 1.4.

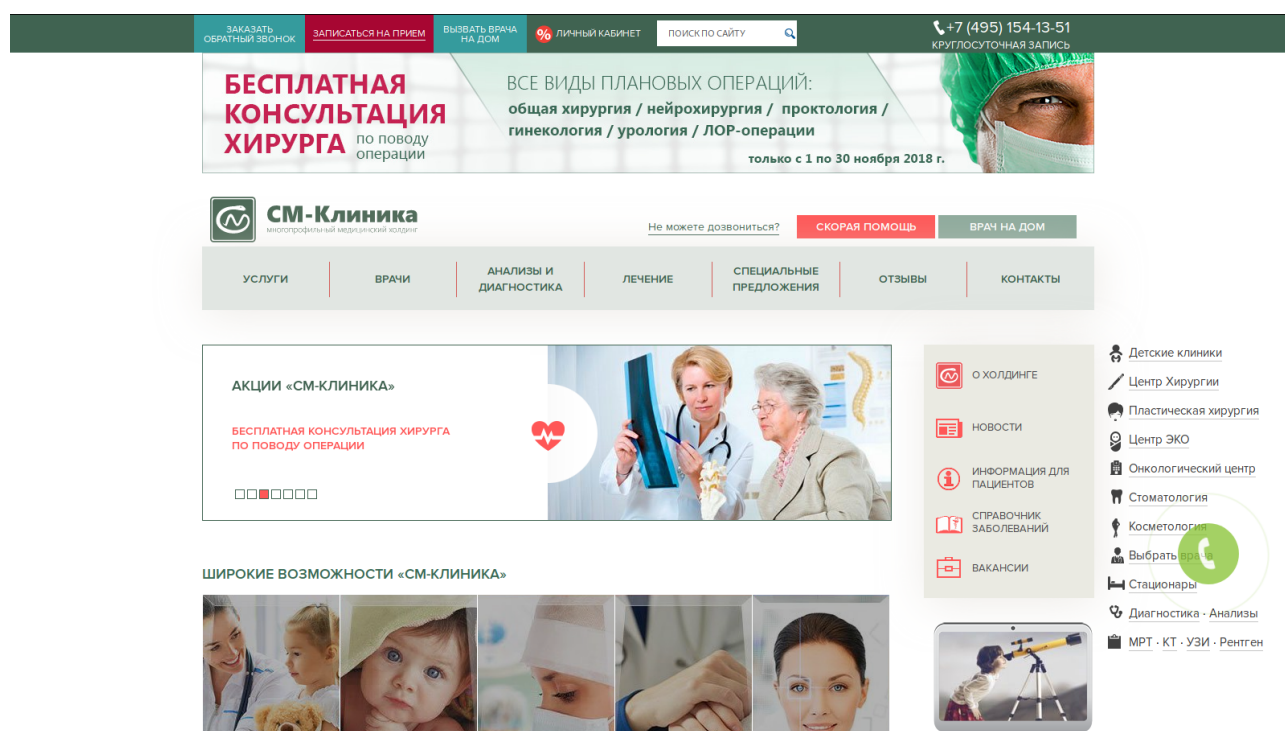


Рис. 1.4: Внешний вид сайта «СМ-Клиника»

Интерес здесь представляют функции «Записаться на приём» и «Личный кабинет».

1-ое позволяет предварительно записаться на приём к лечащему врачу посредством заполнения со стороны пользователя соответствующей формы (Рис. 1.5).

ЗАКАЗАТЬ ОБРАТНЫЙ ЗВОНОК   ЗАПИСАТЬСЯ НА ПРИЕМ   ВЫЗВАТЬ ВРАЧА НА ДОМ   Личный кабинет   ПОИСК ПО САЙТУ   +7 (495) 154-13-51   КРУГЛОСУТОЧНАЯ ЗАПИСЬ

## БЕСПЛАТНАЯ КОНСУЛЬТАЦИЯ ХИРУРГА

### ВСЕ ВИДЫ ПЛАНОВЫХ ОПЕРАЦИЙ:

общая хирургия / нейрохирургия / проктология /

### ЗАПИСЬ К ВРАЧУ

ВАШЕ ИМЯ

ТЕЛЕФОН\*

ВЫБЕРИТЕ СПЕЦИАЛИСТА

ЖЕЛАТЕЛЬНАЯ ДАТА ПРИЕМА

ВРЕМЯ ПРИЕМА...

Возраст пациента  
☒ Взрослый   ☐ Ребенок (до 18 лет)

**Запись через сайт является предварительной.**  
 Наш сотрудник свяжется с Вами для подтверждения записи к специалисту.

**ОТПРАВИТЬ ЗАЯВКУ**

Нажимая на кнопку, вы даете согласие на обработку своих персональных данных

ВЫЗОВ ВРАЧА НА ДОМ

Если состояние вашего здоровья не позволяет посетить клинику, вы можете вызвать врача на дом. При необходимости на дом придет терапевт, узкий специалист или медицинская сестра. Возможно также взятие анализов, проведение диагностики и различных медицинских процедур на дому.

НАШИ УСЛУГИ

Консультация врача  
 Вызов врача на дом  
 Диагностика  
 Комплексная лабораторная диагностика  
 Анализы  
 Стационары

Терапевтические стационары

- Кардиология
- Пульмонология
- Нефрология
- Онкология

Хирургические стационары

- Дневные стационары

Операционные

Вакцинация

Медицинские справки

Комплексные программы для физических лиц

- Программы реабилитации

КОНСУЛЬТАЦИЯ ВРАЧА

приедут к вам на дом.

Детские клиники  
 Центр Хирургии  
 Пластическая хирургия  
 Центр ЭКО  
 Онкологический центр  
 Стоматология  
 Косметология  
 Выбрать специалиста  
 Стационары  
 Диагностика · Анализы  
 МРТ · КТ · УЗИ · Рентген

Рис. 1.5: Форма записи на приём к врачу в «СМ-Клиника»

## СМ-Клиника

многопрофильный медицинский холдинг

### ВХОД В ЛИЧНЫЙ КАБИНЕТ

Личный кабинет для смартфона

Загрузите в App Store

доступно в Google Play

**ВОЙТИ**

Забыли пароль?

Акция!

При записи через личный кабинет вы получаете скидку в 7% на консультацию специалиста.

Что такое личный кабинет?

**Личный кабинет** – это система, с которой следить за здоровьем стало проще. У вас будет постоянный доступ онлайн к своей медицинской карте в любое время с вашего домашнего компьютера или мобильного телефона.

**В личном кабинете Вы сможете:**

- получить доступ к своей медицинской карте – увидеть детальную историю всех посещений наших клиник, фамилии лечащих докторов, их специализации, точные даты посещений и другую полезную информацию

Подробнее

Как получить пароль?

Для получения доступа к личному кабинету или восстановления пароля обратитесь на регистратуру любой из клиник холдинга.

Рис. 1.6: Форма входа в личный кабинет пользователя

Функция «Личный кабинет» (Рис. 1.6) предоставляет широкий перечень возможностей для зарегистрированных пользователей:[4]

1. получить доступ к своей медицинской карте - увидеть детальную историю всех посещений клиники, фамилии лечащих докторов, их специализации, точные даты посещений и другую полезную информацию;
2. посмотреть назначенную схему лечения, рекомендации лечащих врачей, назначенные обследования и др.;
3. ознакомиться с результатами анализов и обследований, сохранить их на локальный компьютер или сразу распечатать;
4. увидеть, когда лечащий врач пациента работает и какое время для приема на текущий момент у него свободно;
5. самостоятельно записаться к врачу в удобное для пользователя время;
6. посмотреть текущую скидку пользователя, актуальные акции и предложения клиник;
7. оставить отзыв о враче или клинике.

### **1.3.3 Сравнение «1С:Медицина. Поликлиника» и сайта поликлиники «СМ-Клиника»**

Прямое сравнение данных продуктов не представляется возможным, так как они ориентированы на две совершенно разные группы пользователей, сотрудников поликлиники и пациентов соответственно. Как результат оба продукта обладают не поддающимся непосредственному сравнению возможностями. Данный факт наталкивает на идею создания информационной экосистемы где вышеупомянутые группы пользователей имели бы возможность взаимодействовать на базе одной единой платформы. Таким образом описание системы совмещающей в себе функции двух программных продуктов, которые были рассмотрены в предыдущих подразделах, и является темой следующего раздела данной курсовой работы.

## 1.4 Функции, планируемые для реализации в курсовом проекте

Как говорилось раньше, в рамках данной курсовой работы планируется создание единой платформы для взаимодействия пациентов с сотрудниками поликлиники. Отсюда вытекает что по сути существует две различные категории потенциальных пользователей системы, а т.к. и потребности этих категорий вообще говоря различны, то и при проектировании возможностей системы логично будет придерживаться данного разделения. Основные функции для реализации, таким образом можно подразделить на 2 группы:

### 1. Функции для пациентов:

- 1.1. бронирование услуг предоставляемых поликлиникой;
- 1.2. запись на приём к врачу;
- 1.3. просмотр личных медицинских карт;
- 1.4. просмотр результатов анализов;
- 1.5. ознакомление с графиком работы врачей.
- 1.6. прямая связь с сотрудником поликлиники
- 1.7. оставить отзыв о работе врачей и поликлиники в целом;

### 2. Функции для сотрудников:

- 2.1. просмотр и/или редактировании медицинских карт;
- 2.2. просмотр и/или редактировании результатов анализов;
- 2.3. запись на приём к врачу;
- 2.4. просмотр и/или редактирование записей к врачам;
- 2.5. связь с пациентами и другим персоналом;
- 2.6. регистрация нового пользователя;
- 2.7. доступ к личному календарю.

## Глава 2 Концептуальное проектирование

### 2.1 Определение концептуального проектирования

Концептуальное проектирование технических систем — начальная стадия проектирования, на которой принимаются решения определяющие последующий облик, и проводится исследование и согласование параметров созданных технических решений с возможной их организацией. Термин «концепция» применяется для описания принципа действия не только в технических системах, но и в научных, художественных и прочих видах деятельности. «Концепт» (лат.) — содержание понятия, смысл. Таким образом, проектирование на концептуальном уровне — на уровне смысла или содержания понятия систем. В частности концептуальное (инфологическое) проектирование базы данных — построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных. Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является как образом реальности, так и образом проектируемой базы данных для этой реальности. Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные ER-диаграммам.

Диаграмма вариантов использования (ДВИ) — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе. [5]

## **2.2 Концептуальная модель базы данных поликлиники**

В ходе выполнения анализа предметной области можно выделить следующие задачи, подлежащие решению для имплементации информационной системы описанной в предыдущей главе.

1. создание единой базы данных (БД) для всех пользователей ИС;
2. разграничение доступа к БД как между отдельными группами пользователей так и внутри самих групп;
3. реализация механизма поиска необходимой информации по БД;
4. реализации методов предоставления соответствующих функций в зависимости от роли и прав пользователя;
5. разработка подсистемы связи между пользователями с хранением истории непосредственно в самой базе данных.

Если же рассматривать данную ИС с точки зрения возможных вариантов использования, то можно выделить два действующих лица, сотрудник и пациент. Такое небольшое количество действующих лиц объясняется тем, что

функции, которые будет предоставлять система, подразделены на две основные категории, в то время как возможности пользователей внутри своей группы будут разграничены при помощи вышестоящего логического уровня рассматриваемой ИС.

Так как функции которые будут предоставляться пользователям были перечислены в конце предыдущей главы, то далее представлена диаграмма вариантов использования (Рис. 2.1).

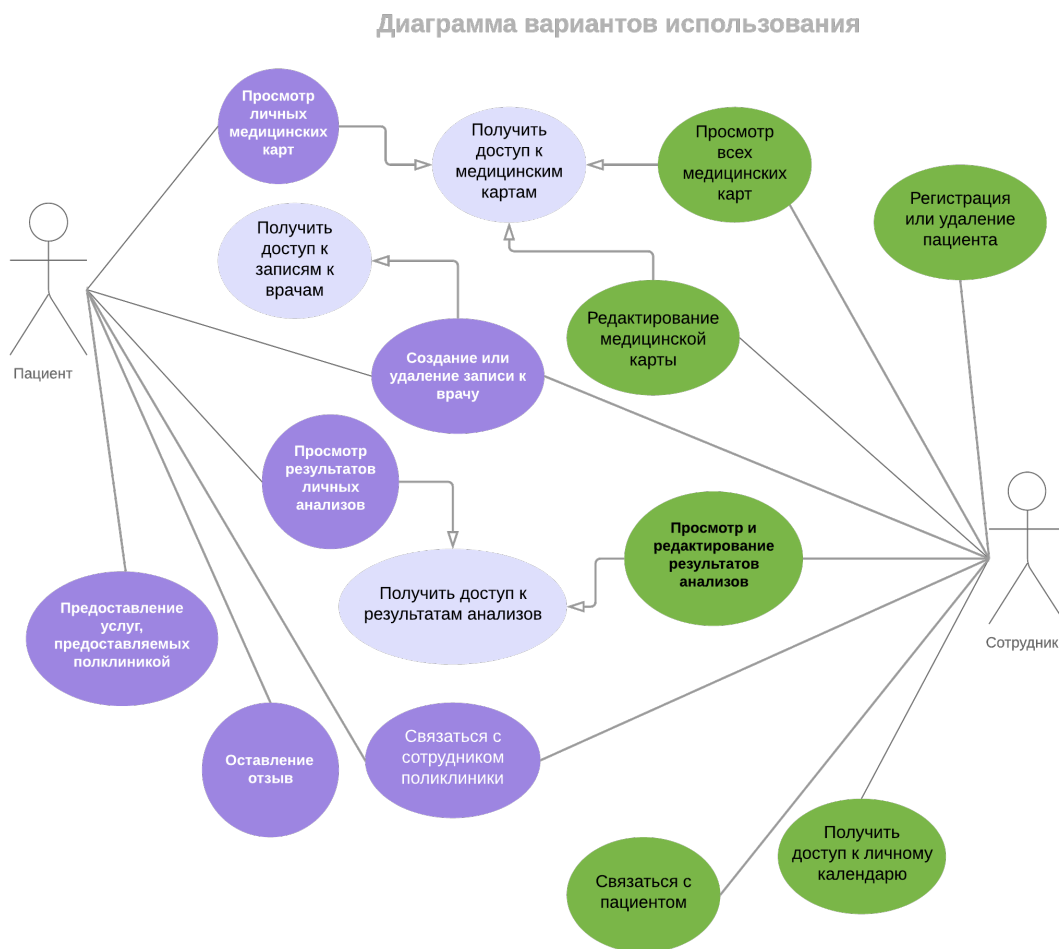


Рис. 2.1: Диаграмма вариантов использования



Приняв во внимание всё вышесказанное, можно выделить данные которые будут храниться в проектируемой базе данных.

1. информация о предоставляемых услугах;
2. информация о пациентах;
3. информация о сотрудниках поликлиники;
4. информация о медицинских картах, т.е. их цифровые копии;
5. информация о записях пациентов;
6. информация об анализах;
7. история связи между пользователями системы;

Перечисленные данные можно подразделить на две группы:

- условно-постоянные:

1. информация о предоставляемых услугах;
2. информация о пациентах;
3. информация о сотрудниках поликлиники;

- изменяемые:

1. информация о медицинских картах, т.е. их цифровые копии;
2. информация о записях пациентов;
3. информация об анализах;
4. история связи между пользователями системы;

## Глава 3 Логическое проектирование

### 3.1 Определение логического проектирования

Логическое (дatalogическое) проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных дatalogическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

ER-модель (от англ. entity-relationship model, модель «сущность — связь») — модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями. Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).[5]

Основные преимущества ER-моделей:

- наглядность;
- модели позволяют проектировать базы данных с большим количеством объектов и атрибутов;

Основные элементы ER-моделей:

- объекты (сущности);
- атрибуты объектов;
- связи между объектами.

Сущность — объект предметной области, имеющий атрибуты.

Связь между сущностями характеризуется:

- типом связи (1:1, 1:N, N:M);
- классом принадлежности. Класс может быть обязательным и необязательным. Если каждый экземпляр сущности участвует в связи, то класс принадлежности — обязательный, иначе — необязательный.

## 3.2 ER-модель базы данных поликлиники

На основе приведенных ранее данных можно составить диаграммы сущность-связь проектируемой базы данных.



Рис. 3.1: Запись пациента к сотруднику (врачу)

Тип связи “Запись” - M:N.

Класс принадлежности необязательный для обоих экземпляров сущностей.

Обоснование: Пациент может записаться к нулю или более сотрудникам (врачам), сотрудник может принять ноль или более пациентов.

---

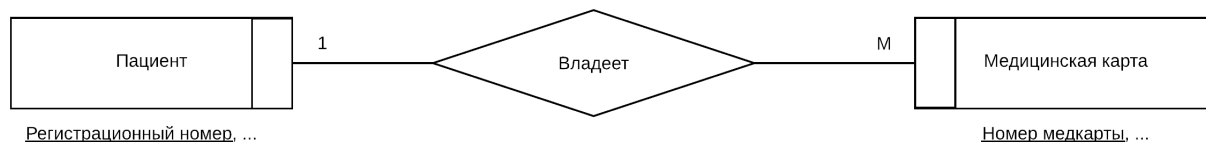


Рис. 3.2: Принадлежность медицинской карты пациенту

Тип связи “Владеет” - 1:M.

Класс принадлежности обязательный для обоих экземпляров сущностей.

Обоснование: Пациент может владеть одной или более медкартами, медкарта должна принадлежать только одному пациенту.

---



Рис. 3.3: Прикрепление результатов анализов к медицинской карте

Тип связи “Прикрепление” - 1:M.

Класс принадлежности обязательный только лишь для экземпляра сущности “Результат анализов”.

Обоснование: Результат анализов должен быть прикреплен только к одной медицинской карте, медицинская карта может содержать нуль или более результатов анализов.

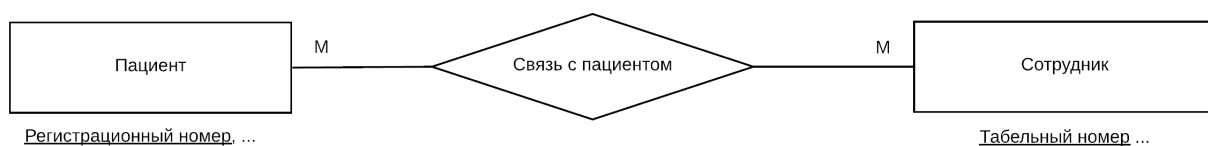


Рис. 3.4: Связь пациента с сотрудником поликлиники

Тип связи “Связывается с пациентом” - M:N.

Класс принадлежности необязательный для обоих экземпляров сущностей.

Обоснование: Пациент может связаться с нуль или более сотрудниками, сотрудник может связаться с нуль или более пациентами.

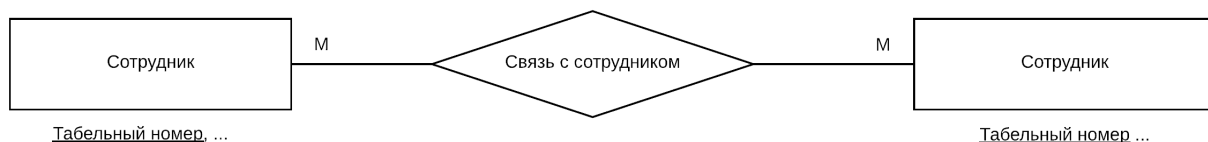


Рис. 3.5: Связь сотрудника с другим сотрудником поликлиники

Тип связи “Связывается с сотрудником” - M:N.

Класс принадлежности необязательный для обоих экземпляров сущностей.

Обоснование: Сотрудник может связаться с нуль или более сотрудниками.



Рис. 3.6: Назначение лечения пациенту по его медицинской карте

Тип связи “Лечение” - M:N.

Класс принадлежности необязательный для обоих экземпляров сущностей.

Обоснование: Сотрудник может лечить нуль или более пациентов по их медицинским картам, пациента по его медицинским картам могут лечить нуль или более сотрудников.

---



Рис. 3.7: Бронирование услуги пациентом

Тип связи “Бронирует” - М:N.

Класс принадлежности необязательный для обоих экземпляров сущностей.

Обоснование: Пациент может забронировать нуль или более услуг, услуга может быть оказана нулю или более пациентам.

### 3.3 Преобразование ER-модели в реляционную

Для того чтобы преобразовать вышеизложенную ER-модель в реляционную существует перечень правил:

1. Если связь типа 1:1 и класс принадлежности обеих сущностей является обязательным, то необходима только одна таблица. Первичным ключом этой таблицы может быть первичный ключ любой из двух сущностей.
2. Если связь типа 1:1 и класс принадлежности одной сущности является обязательным, а другой — необязательным, то необходимо построить таблицу для каждой сущности. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Первичный ключ сущности, для которой класс принадлежности является необязательным, добавляется как атрибут в таблицу для сущности с обязательным классом принадлежности.

3. Если связь типа 1:1 и класс принадлежности обеих сущностей является необязательным, то необходимо построить три таблицы — по одной для каждой сущности и одну для связи. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Таблица для связи среди своих атрибутов должна иметь ключи обеих сущностей.
4. Если связь типа 1:M и класс принадлежности сущности на стороне M является обязательным, то необходимо построить таблицу для каждой сущности. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Первичный ключ сущности на стороне 1 добавляется как атрибут в таблицу для сущности на стороне M.
5. Если связь типа 1:M и класс принадлежности сущности на стороне M является необязательным, то необходимо построить три таблицы — по одной для каждой сущности и одну для связи. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Таблица для связи среди своих атрибутов должна иметь ключи обеих сущностей.
6. Если связь типа M:N, то необходимо построить три таблицы — по одной для каждой сущности и одну для связи. Первичный ключ сущности должен быть первичным ключом соответствующей таблицы. Таблица для связи среди своих атрибутов должна иметь ключи обеих сущностей.[6]

Ссылаясь на эти правила можно построить реляционную модель проектируемой БД.

1. Связь “Запись” удовлетворяет условиям правила 6, откуда получаются следующие таблицы
  - Пациент(Регистрационный номер, ...);
  - Сотрудник(Табельный номер, ...);

- Запись(Табельный номер, Регистрационный номер, ...).
2. Связь “Владеет” удовлетворяет условиям правила 4, откуда получаются следующие таблицы
- Пациент(Регистрационный номер, ...);
  - Медкарта(Номер медкарты, Регистрационный номер, ...).
3. Связь “Прикрепление” удовлетворяет условиям правила 4, откуда получаются следующие таблицы
- Медкарта(Номер медкарты, ...);
  - Результат анализов(Номер анализа, Номер медкарты, ...).
4. Связь “Связывается с пациентом” удовлетворяет условиям правила 6, откуда получаются следующие таблицы
- Пациент(Регистрационный номер, ...);
  - Сотрудник(Табельный номер, ...);
  - СвязьСПациентом(Табельный номер, Регистрационный номер, ...).
5. Связь “Связывается с сотрудником” удовлетворяет условиям правила 6, откуда получаются следующие таблицы
- Сотрудник(Табельный номер, ...);
  - СвязьССотрудником(Табельный номер, Табельный номер, ...).
6. Связь “Лечение” удовлетворяет условиям правила 6, откуда получаются следующие таблицы
- Сотрудник(Табельный номер, ...);
  - Медкарта(Номер медкарты, ...);



- Лечение(Табельный номер, номер медкарты, ...).

7. Связь “Бронирует” удовлетворяет условиям правила 6, откуда получаются следующие таблицы

- Пациент(Регистрационный номер, ...);
- Бронь(Номер услуги, Регистрационный номер, ...);
- Услуга(Номер услуги, ...).

В итоге получается 10 таблиц, которые приведены ниже с перечнем всех относящихся к ним атрибутов.

1. Пациент(Регистрационный номер, ФИО, дата рождения, пол, адрес проживания, контактный телефон, пароль от ЛК);
2. Сотрудник(Табельный номер, ФИО, должность, стаж, дата рождения, пол, адрес проживания, контактный телефон, пароль от ЛК);
3. Запись(Табельный номер, Регистрационный номер, дата и время записи).
4. Медкарта(Номер медкарты, Регистрационный номер, дата, заведения, тип).
5. Результат анализов(Номер анализа, Номер медкарты, вид анализа, дата сдачи, результат).
6. СвязьССотрудником(Табельный номер, Табельный номер, дата создания, история связи).
7. СвязьСПациентом(Табельный номер, Регистрационный номер, дата создания, история связи).
8. Лечение(Табельный номер, номер медкарты, дата назначения, заболевание, назначенное лечение).

9. Услуга(Номер услуги, название, описание, стоимость).
10. Бронь(Номер услуги, Регистрационный номер, дата бронирования, дата и время оказания, статус);

## Глава 4 Физическое проектирование

Физическое проектирование — создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.[5]

### 4.1 Выбор СУБД

В данной курсовой работе для реализации базы данных поликлиники выбор пришёлся на реляционную СУБД SQLite. Такой выбор обоснован спецификой предметной области, а именно относительно небольшим объёмом хранимых данных и средней частотой обращения к базе данных. SQLite является встраиваемой, т.е. вместо использования привычной парадигмы клиент-сервер, SQLite представляет из себя библиотеку с интерфейсами для многих языков программирования. Этот факт делает SQLite чрезвычайно компактным и быстрым, т.к. СУБД встраивается напрямую в разрабатываемое приложение. Несмотря на свою простоту данная СУБД может обслуживать базы данных размером до 140 терабайт и поддерживает параллельный доступ к БД несколькими процессами. Помимо этого SQLite выполняет подавляющее большинство требований изложенных в стандарте SQL, а также соблюдает

ACID-требования (Atomicity, Consistency, Isolation, Durability).[7]

## 4.2 Схема БД поликлиники

На основании информации, полученной в конце предыдущей главы, касательно таблиц и их атрибутов, далее приведены окончательные структуры таблиц, какими они будут в СУБД SQLite. Поля перечислены в том же порядке, который был описан ранее. SQL операторы использованные для создания таблиц приведены в приложении А, там же приведены примеры их заполнения.

Столбец	Тип данных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
regid	integer	not null	первичный			
fio	text	not null				
birthdate	date	not null			< date('now')	
gender	char(1)	not null			like 'М' or like 'Ж'	
address	text					
pnumber	text					
password	text	not null		abs(random())		

Таблица 4.1: Структура таблицы «Пациент» Листинги 5.1,5.11

Столбец	Тип данных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
tabid	integer	not null	первичный			
fio	text	not null				
position	text	not null				
experien	integer	not null		0	< 100	
birthdate	date	not null			< date('now')	
gender	char(1)	not null			like 'М' or like 'Ж'	
address	text					
pnumber	text					
password	text	not null		abs(random())		

Таблица 4.2: Структура таблицы «Сотрудник» Листинги 5.2, 5.12

Столбец	Тип данных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
tabid	integer	not null	первичный, внешний			tabid в «Сотрудник»
regid	integer	not null	первичный, внешний			regid в «Пациент»
recdatetime	datetime	not null				
ограничения на таблицу: unique(tabid, recdatetime), unique(regid, recdatetime)						

Таблица 4.3: Структура таблицы «Запись» Листинги 5.3, 5.13

Столбец	Тип данных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
cardid	integer	not null	первичный			
regid	integer	not null	внешний			regid в «Пациент»
crdate	date	not null		date('now')		
type	text					
ограничения на таблицу: unique(regid, type)						

Таблица 4.4: Структура таблицы «Медкарта» Листинги 5.4 5.14

Столбец	Тип данных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
resid	integer	not null	первичный			
cardid	integer	not null	внешний			cardid в «Медкарта»
type	text	not null				
passdate	date	not null		date('now')		
result	text					

Таблица 4.5: Структура таблицы «Результат анализов» Листинги 5.5, 5.15

Столбец	Тип данных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
stabid	integer	not null	первичный, внешний			tabid в «Сотрудник»
dtabid	integer	not null	первичный, внешний			tabid в «Сотрудник»
crttime	datetime	not null		datetime('now')		
lhist	text					

Таблица 4.6: Структура таблицы «СвязьССотрудником», Листинги 5.7, 5.17

Столбец	Тип данных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
tabid	integer	not null	первичный, внешний			tabid в «Сотрудник»
regid	integer	not null	первичный, внешний			regid в «Пациент»
crttime	datetime	not null		datetime('now')		
lhst	text					

Таблица 4.7: Структура таблицы «СвязьСПациентом» Листинги 5.6, 5.16

Столбец	Тип данных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
tabid	integer	not null	первичный, внешний			tabid в «Сотрудник»
cardid	integer	not null	первичный, внешний			cardid в «Медкарта»
trdate	date	not null		date('now')		
illness	text					
treatment	text					

Таблица 4.8: Структура таблицы «Лечение» Листинги 5.8, 5.18

Столбец	Тип данных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
servid	integer	not null	первичный			
name	text	not null			unique	
descrip	text	not null				
cost	integer	not null		0		

Таблица 4.9: Структура таблицы «Услуга» Листинги 5.9, 5.19

Столбец	Тип данных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
servid	integer	not null	первичный, внешний			servid в «Услуга»
regid	integer	not null	первичный, внешний			regid в «Пациент»
rdate	date	not null		date('now')		
pdate	datetime	not null				
status	text	not null				

Таблица 4.10: Структура таблицы «Бронь» Листинги 5.10, 5.20

На Рис. 4.1 можно ознакомиться с общей схемой БД. Визуализация была произведена средствами программного продукта DataGrip версия 2017.2.

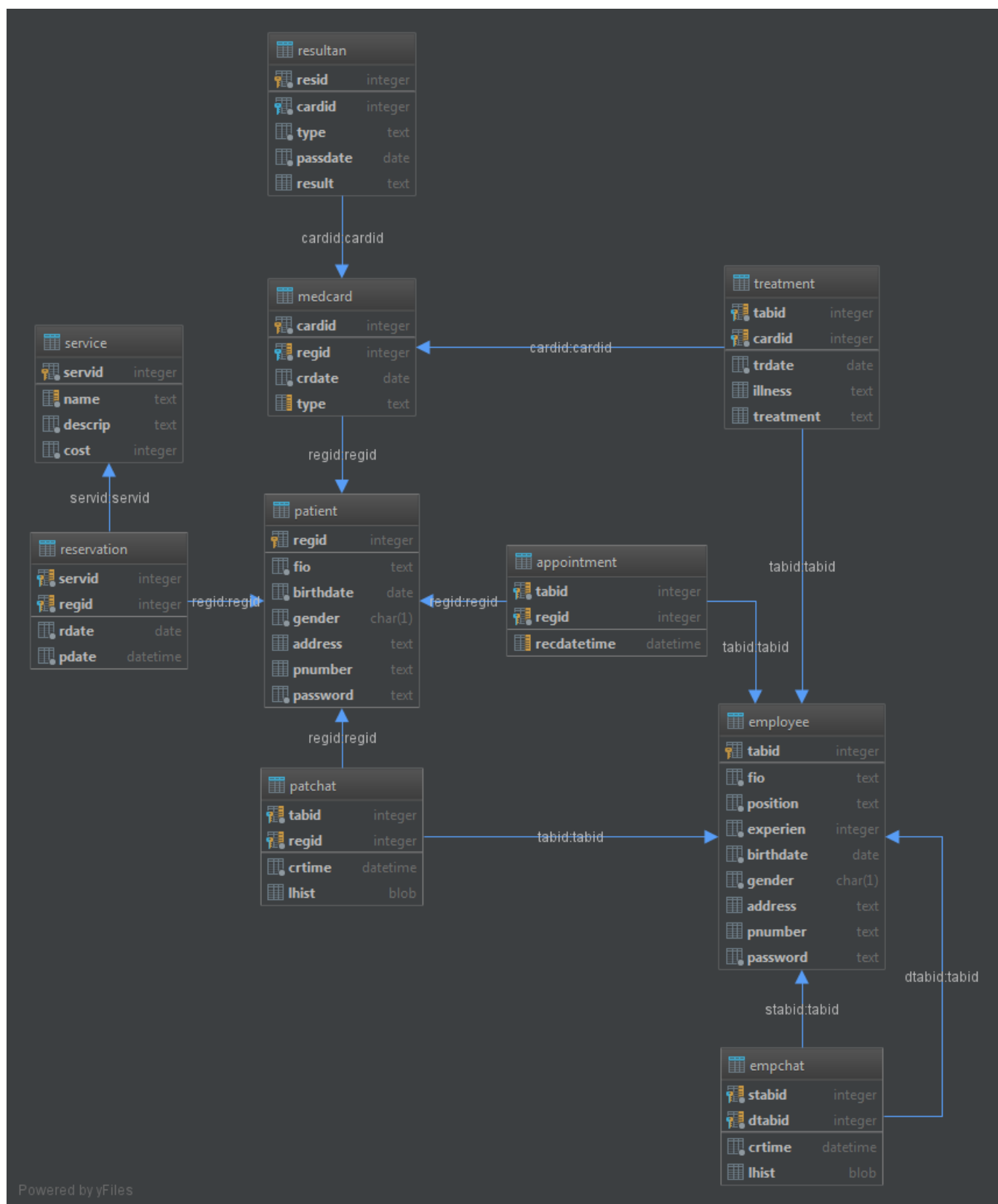


Рис. 4.1: Схема базы данных поликлиники

## Заключение

Целью данной курсовой работы было проектирование базы данных для поликлиники, эта задача была решена в 4 этапа:

1. Была проанализирована предметная область поликлиника и её особенности, также были приведены аргументы в пользу необходимости в автоматизации. Были рассмотрены 2 существующих продукта, которые в той или иной мере решают проблему автоматизации, предоставляемые ими функции и их специфика. На основании анализа предметной области был выделен перечень функций для будущей реализации.
2. Следующим этапом являлось концептуальное проектирование, в ходе которого были выделены 2 действующих лица (сотрудник, пациент), для которых была составлена диаграмма вариантов использования.
3. Далее в ходе логического проектирования была создана ER-модель базы данных, где были определены основные сущности и их взаимодействия. Для визуализации ER-модели были использованы ER-диаграммы. В завершении данного этапа были определены таблицы БД путём преобразования ER-модели в реляционную.
4. Завершающим этапом стало физическое проектирование, в ходе которого в СУБД SQLite была создана база данных поликлиники со всеми таблицами, определенных на предыдущем этапе.



## Литература

- [1] Предметная область: Wikipedia - свободная энциклопедия. — [https://ru.wikipedia.org/wiki/Domain\\_knowledge](https://ru.wikipedia.org/wiki/Domain_knowledge) (дата обращения: 09.11.2018).
- [2] *Ольга Жидкова*. Медицинская статистика: конспект лекций. — Eksmo education, 2009. — С. 180.
- [3] 1С:Медицина. Поликлиника. — <https://solutions.1c.ru/catalog/clinic/features> (дата обращения: 11.11.2018).
- [4] Авторизация СМ-Клиника. — <https://lk.smclinic.ru/> (дата обращения: 11.11.2018).
- [5] Проектирование баз данных: Wikipedia - свободная энциклопедия. — [https://ru.wikipedia.org/wiki/Database\\_design](https://ru.wikipedia.org/wiki/Database_design) (дата обращения: 11.11.2018).
- [6] Преобразование ER-модели в реляционную модель данных. — [https://life-prog.ru/1\\_22432](https://life-prog.ru/1_22432) (дата обращения: 13.11.2018).
- [7] *Owens Mike*. The Definitive Guide to SQLite. — 1-ое изд. — Apress, 2006. — С. 440.

# Приложение А

## SQL операторы создания таблиц

Листинг 5.1: Создание таблицы «Пациент»

```
CREATE TABLE patient (  
    regid integer PRIMARY KEY,  
    fio text NOT NULL,  
    birthdate date NOT NULL,  
    gender char(1) NOT NULL CHECK(gender LIKE 'М'  
        OR gender LIKE 'Ж'),  
    address text,  
    pnumber text,  
    password text NOT NULL DEFAULT (abs(random()))  
);
```

Листинг 5.2: Создание таблицы «Сотрудник»

```
CREATE TABLE employee (  
    tabid integer PRIMARY KEY,  
    fio text NOT NULL,  
    position text NOT NULL,  
    experien integer NOT NULL DEFAULT 0,  
    birthdate date NOT NULL,  
    gender char(1) NOT NULL CHECK(gender LIKE 'М'  
        OR gender LIKE 'Ж'),  
    address text,  
    pnumber text,  
    password text NOT NULL DEFAULT (abs(random()))  
);
```

### Листинг 5.3: Создание таблицы «Запись»

```
CREATE TABLE appointment (  
    tabid integer NOT NULL REFERENCES employee(tabid),  
    regid integer NOT NULL REFERENCES patient(regid),  
    recdatetime datetime,  
    PRIMARY KEY(tabid, regid),  
    UNIQUE(tabid, recdatetime),  
    UNIQUE(regid, recdatetime)  
);
```

### Листинг 5.4: Создание таблицы «Медкарта»

```
CREATE TABLE medcard (  
    cardid integer NOT NULL PRIMARY KEY,  
    regid integer NOT NULL REFERENCES patient(regid),  
    crdate date NOT NULL DEFAULT (date('now')),  
    type text,  
    UNIQUE(regid, type)  
);
```

### Листинг 5.5: Создание таблицы «Результат анализов»

```
CREATE TABLE resultan (  
    resid integer NOT NULL PRIMARY KEY,  
    cardid integer NOT NULL REFERENCES medcard(cardid),  
    type text NOT NULL,  
    passdate date NOT NULL DEFAULT (date('now')),  
    result text  
);
```

Листинг 5.6: Создание таблицы «СвязьСПациентом»

```
CREATE TABLE patchat (  
    tabid integer NOT NULL REFERENCES employee(tabid),  
    regid integer NOT NULL REFERENCES patient(regid),  
    crtime datetime NOT NULL DEFAULT (datetime('now')),  
    lhist text,  
    PRIMARY KEY(tabid, regid)  
);
```

Листинг 5.7: Создание таблицы «СвязьССотрудником»

```
CREATE TABLE empchat (  
    stabid integer NOT NULL REFERENCES employee(tabid),  
    dtabid integer NOT NULL REFERENCES employee(tabid),  
    crtime datetime NOT NULL DEFAULT (datetime('now')),  
    lhist text,  
    PRIMARY KEY(tabid, regid)  
);
```

Листинг 5.8: Создание таблицы «Лечение»

```
CREATE TABLE treatment (  
    tabid integer NOT NULL REFERENCES employee(tabid),  
    cardid integer NOT NULL REFERENCES medcard(cardid),  
    trdate date NOT NULL DEFAULT (date('now')),  
    illness text,  
    treatment text,  
    PRIMARY KEY(tabid, cardid)  
);
```

### Листинг 5.9: Создание таблицы «Услуга»

```
CREATE TABLE service (  
    servid integer NOT NULL PRIMARY KEY,  
    name text NOT NULL unique,  
    describ text NOT NULL,  
    cost integer NOT NULL DEFAULT 0  
);
```

### Листинг 5.10: Создание таблицы «Бронь»

```
CREATE TABLE reservation (  
    servid integer NOT NULL REFERENCES service(servid),  
    regid integer NOT NULL REFERENCES patient(regid),  
    rdate date NOT NULL DEFAULT (date('now')),  
    pdate datetime NOT NULL,  
    status text NOT NULL,  
    PRIMARY KEY(servid, regid)  
);
```

## Примеры заполнения таблиц

### Листинг 5.11: Заполнение таблицы «Пациент»

```
INSERT INTO patient  
(fio, birthdate, gender, address, pnumber, password)  
VALUES  
(  
    'Махаров КузьмаВикторович ', '1983-11-28', 'М',  
    'г. Москва, Тихоходскийпроездд .5 кв.56', '+79314149204', 'moscow_831'  
),  
(  
    'Никифорова ВалерияГригорьевна ', '1995-10-01', 'Ж',  
    'г. Москва, ул. Новослободскаяд .9', '+79652132309', 'tututochkaru'  
);
```

### Листинг 5.12: Заполнение таблицы «Сотрудник»

```
INSERT INTO employee
(fio, position, experien, birthdate, gender, address, pnumber, password
)
VALUES
(
    'Варшавин АлександрМихайлович ', 'Главный врач', 23, '1978-04-20',
    'М', 'г. Москва, Варламовпереулокд .1 кв.13', '+79962429419', 'alex23'
),
(
    'Прокофьева ЗинаидаДмитриевна ', 'Врач терапевт', 12, '1981-01-31',
    'Ж', 'г. Москва, ул. Тимурад .31 с.1 кв.21', '+79664568381', '03
    mazina03'
);
```

### Листинг 5.13: Заполнение таблицы «Запись»

```
INSERT INTO appointment
(tabid, regid, recdatetime)
VALUES
(1, 2, '2018-01-01'),
(2, 2, '2018-10-11');
```

### Листинг 5.14: Заполнение таблицы «Медкарта»

```
INSERT INTO medcard
(regid, crdate, type)
VALUES
(1, date('now'), 'Амбулаторная'),
(2, date('now'), 'Стационарная');
```

Листинг 5.15: Заполнение таблицы «Результат анализов»

```
INSERT INTO resultan
(resid, cardid, type, passdate, result)
VALUES
(1, 1, 'Гемоглобин', '2018-08-09', 'Уровень гемоглобина 9'),
(2, 1, 'Грибок', '2018-09-01', 'Отрицательный');
```

Листинг 5.16: Заполнение таблицы «СвязьСПациентом»

```
INSERT INTO patchat
(stabid, regid, crtime, lhist)
VALUES
(
  1, 2, date('now'), 'epchats/1-2.chat'
)
(
  2, 1, date('now'), 'epchats/2-1.chat'
);
```

Листинг 5.17: Заполнение таблицы «СвязьССотрудником»

```
INSERT INTO empchat
(stabid, dtabid, crtime, lhist)
VALUES
(
  1, 2, date('now'), 'eechats/1-2.chat'
)
(
  2, 1, date('now'), 'eechats/2-1.chat'
);
```

Листинг 5.18: Заполнение таблицы «Лечение»

```
INSERT INTO treatment
(tabid, cardid, trdate, illness, treatment)
VALUES
(2, 1, date('now'), 'Воспаление\ седалищного\ нерва', 'Электротерапия, 10
ампул\ комбипилена');
(1, 1, '2018-10-10', 'Гайморит', 'Лазер, Ринурус\ в\ течении\ месяца');
```

Листинг 5.19: Заполнение таблицы «Услуга»

```
INSERT INTO service
(name, descrip, cost)
VALUES
('Лазер', 'servdesc/laser.dsc', 2200),
('Электрофорез', 'servdesc/electroforez.dsc', 3200);
```

Листинг 5.20: Заполнение таблицы «Бронь»

```
INSERT INTO reservation
(servid, regid, rdate, pdate)
VALUES
(1, 2, '2018-10-31', '2018-11-14', 'Исполнено'),
(2, 2, '2018-08-12', '2018-09-01', 'Исполнено');
```