



**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>3</b>
1.1	Определение анализа предметной области . . . . .	3
1.2	Поликлиника: описание предметной области . . . . .	3
1.3	Существующие продукты решающие проблему автоматизации .	5
1.3.1	1С : Медицина. Поликлиника . . . . .	5
1.3.2	Сайт частных поликлиник «СМ-Клиника» . . . . .	7
1.3.3	Сравнение «1С:Медицина. Поликлиника» и сайта поли- клиники «СМ-Клиника» . . . . .	10
<b>2</b>	<b>Концептуальное проектирование</b>	<b>11</b>
2.1	Определение концептуального проектирования . . . . .	11
2.2	Концептуальная модель базы данных поликлиники . . . . .	11
<b>3</b>	<b>Логическое проектирование</b>	<b>14</b>
3.1	Определение логического проектирования . . . . .	14
3.2	ER-модель базы данных поликлиники . . . . .	14
3.3	Преобразование ER-модели в реляционную . . . . .	17
<b>4</b>	<b>Физическое проектирование</b>	<b>21</b>
4.1	Выбор СУБД . . . . .	21
4.2	Схема БД поликлиники . . . . .	22
	<b>Заключение</b>	<b>29</b>
	<b>Литература</b>	<b>29</b>
	<b>Приложение А</b>	<b>31</b>
	<b>Приложение Б</b>	<b>34</b>

# Глава 1 Анализ предметной области

## 1.1 Определение анализа предметной области

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]

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.

Предметной областью данной работы является поликлиника, далее следует анализ деятельности поликлиники, выявление требований к разрабатываемой системе, а также определение функций, которые данная система должна будет предоставлять пользователям.

## 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 различные подразделения.

1. регистратура,
2. лечебно-профилактические подразделения,
3. терапевтические отделения,
4. отделение восстановительного лечения,
5. отделения по оказанию специализированных видов медицинской помощи (хирургическое, гинекологическое) с кабинетами соответствующих специалистов (кардиологический, ревматологический, неврологический, урологический, офтальмологический, оториноларингологический).

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 бюро-

кратических 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.

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 на (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

Создать карту на основании   Отправить в архив   Печать   Смена			
<<нет комментария>>			
№ карты:	66	от 13.08.2013, Амбулаторная, Действующая	
Место хранения:	Регистратура		
Представители:			
Диспанс. группа:			
<<нет комментария>>			
ФИО	Мишин	Петр	Иванович
Дата рождения	10.05.1980	Возраст	33 года
Пол	М	СНИЛС	
Место рождения			
Документ:	Паспорт гражданина РФ, серия: 45 54, № 456987		
Группа инвалидности:			
Полисы:	Полис ОМС. Серия: , №: 4987546587854, ООО «РГС-Медицина» (Росгосстрах)		
Адрес:	Регистрации	Москва г, Карачаровская 1-я ул, дом № 2, кв.1	
	Фактический	Москва г, Карачаровская 1-я ул, дом № 2, кв.1	
Контакты:	Электронная почта	Mishin@mail.ru	
	Мобильный телефон	89054658965	
Соц.статус:	Работающий	ЗАО ТеплоИндустрия	

Рис. 1.1 Пример карты пациента

Рис. 1.2 Пользовательский интерфейс «1С:Медицина. Поликлиника»

работы, нормы загрузки, перечень выполняемых услуг. Оперативное планирование деятельности кабинетов осуществляется по данным предварительной

записи пациентов.[3] Можно выделить основные функциональные возможности «1С:Медицина. Поликлиника»:

1. Регистратура
2. Листки нетрудоспособности (больничные)
3. Договорной отдел
4. Контроль исполнения медицинских услуг персоналом
5. Руководитель и аналитическая (статистическая) служба
6. Электронные медицинские карты
7. Профосмотры
8. Интернет запись на прием и обмен данными с сайтами

### **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. Услуги поликлиник предоставляются на коммерческой основе. Сайт компании доступен по адресу «<http://www.smclinic.ru/>». Скриншот главной страницы сайта приведён на рис. 1.3. На главной странице сайта можно выделить следующие функции:

- записаться на приём,
- личный кабинет,
- услуги,
- анализы и диагностика.

Функция «Записаться на приём» позволяет предварительно записаться на приём к лечащему врачу посредством заполнения со стороны пользователя соответствующей формы (рис. 1.4).

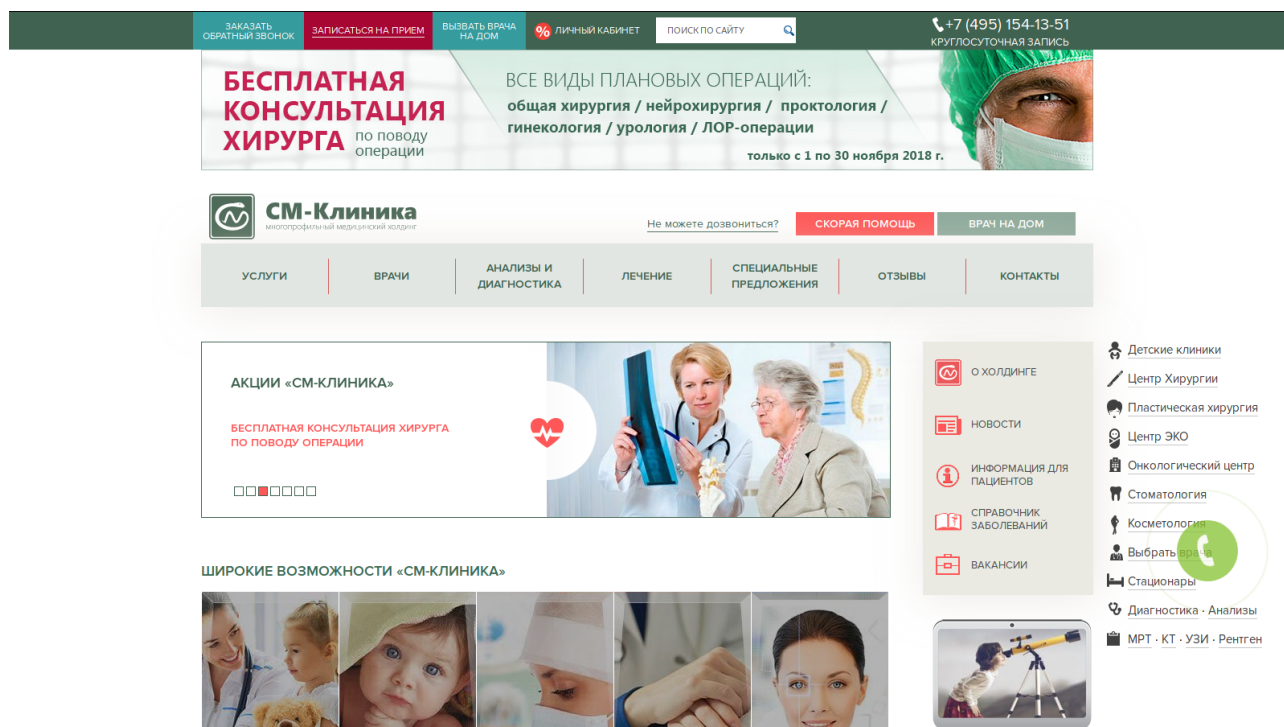


Рис. 1.3 Внешний вид сайта «СМ-Клиника»

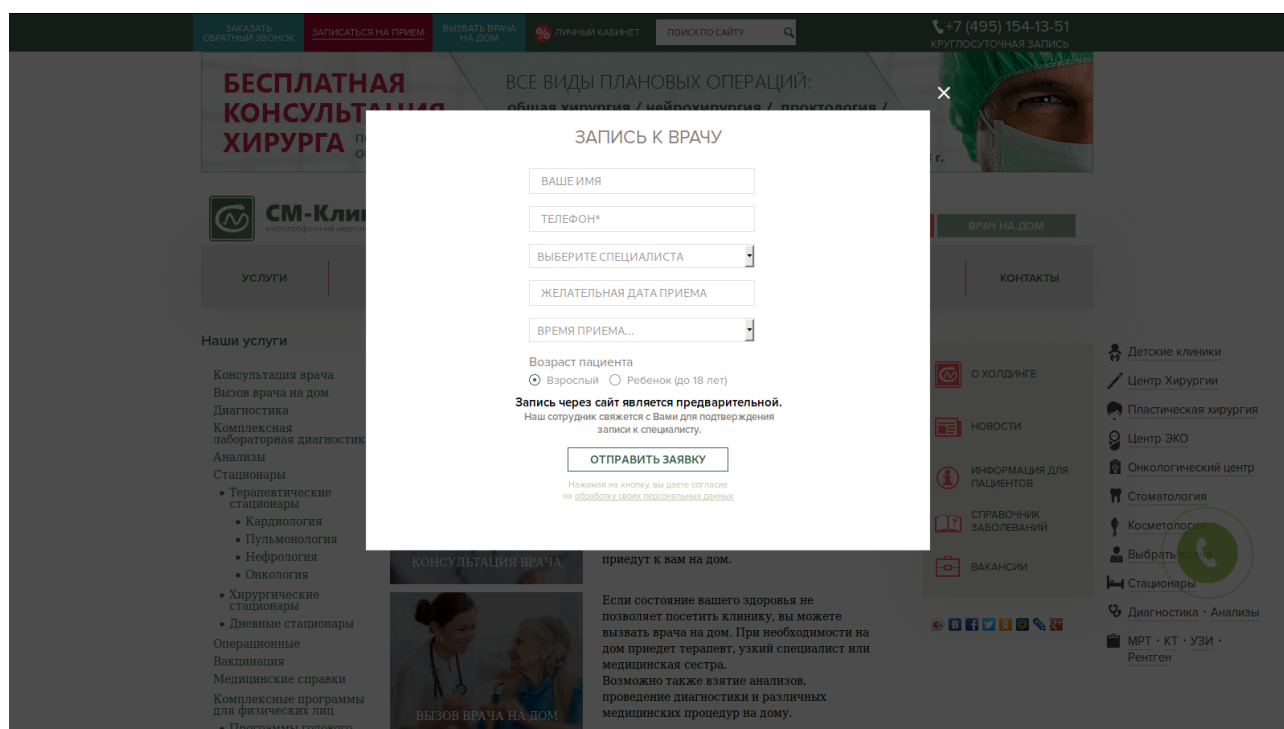


Рис. 1.4 Форма записи на приём к врачу в «СМ-Клиника»





Рис. 1.5 Страница с перечнем предоставляемых услуг

Функция «Личный кабинет» предоставляет широкий перечень возможностей для зарегистрированных пользователей:

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.[4]

Функция «Услуги» позволяет подробно ознакомиться с перечнем услуг предоставляемых поликлиникой (рис. 1.5).

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

Оба рассмотренных продукта, хотя и ориентированы на разные категории пользователей, имеют перечень общих функций:

- доступ к электронной медицинской карте,
- запись на прием к врачу,
- ознакомиться с графиком лечащего врача,
- доступ к перечню предоставляемых поликлиникой услуг.

В рамках данной работы планируется создание единой платформы для взаимодействия пациентов с сотрудниками поликлиники. Проанализировав программные продукты решающие схожие задачи в данной предметной области, можно выделить ряд функций которые необходимо реализовать в процессе разработки:

1. просмотр и/или редактирование медицинских карт,
2. просмотр и/или редактирование результатов анализов,
3. запись на приём к врачу,
4. просмотр и/или редактирование записей к врачам,
5. создание листов нетрудоспособности,
6. просмотр и/или редактирование результатов профосмотров,
7. поиск всей информации касательно пациента,
8. доступ к графику работы врачей,
9. ознакомление с перечнем услуг, предоставляемых поликлиникой.

## Глава 2 Концептуальное проектирование

### 2.1 Определение концептуального проектирования

Концептуальное (инфологическое) проектирование — 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. Чаще всего концептуальная модель базы данных включает в себя описание информационных объектов или понятий предметной области и связей между ними. Для визуализации концептуальной модели часто используется диаграмма вариантов использования.

Диаграмма вариантов использования (ДВИ) — диаграмма, отражающая отношения между действующими лицами и вариантами использования разрабатываемой системы, и позволяющей описать систему на концептуальном уровне. Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

Действующее лицо - внешняя по отношению к ИС сущность, которая может взаимодействовать с системой. Действующим лицом могут быть как люди, так и внешние системы или устройства.

Вариант использования — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. [5]

### 2.2 Концептуальная модель базы данных поликлиники

В завершении анализа предметной области был выделен перечень функций которые необходимо реализовать в рамках данной работы, и рас-

сматривая данную ИС с точки зрения возможных вариантов использования, можно разделить эти функции между действующими лицами.

### **Действующие лица:**

1. сотрудник регистратуры,
2. врач,
3. пациент.

Далее приведены варианты использования ИС каждым из действующих лиц. Сама же диаграмма показана на рис. 2.1.

- **Сотрудник регистратуры:**

1. поиск по медицинским картам пациентов,
2. выдача информации по графику работы врачей,
3. выдача информации информации о предоставляемых услугах,
4. создание медицинской карты пациента,
5. запись пациента к врачу.

- **Врач:**

1. просмотр истории болезней пациента.
2. назначение лечения,
3. назначение анализов,
4. выписка рецептов пациентам,
5. выписка листков нетрудоспособности.

- **Пациент:**

1. ознакомление с графиком работы врачей,
2. запись на приём к врачу,
3. просмотр личной медицинской карты,
4. получение информации о предоставляемых услугах.

Приняв во внимание всё вышесказанное, можно выделить данные, которые необходимо хранить в проектируемой базе данных.

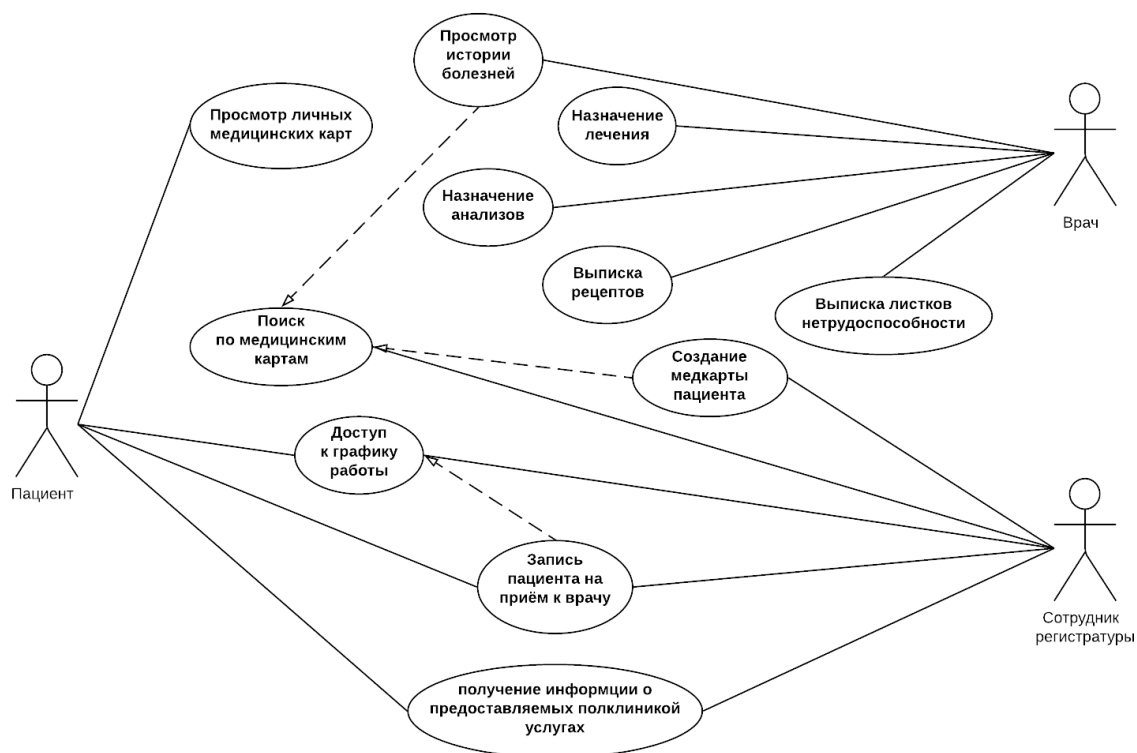


Рис. 2.1 Диаграмма вариантов использования

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 графике приема пациентов врачами,
6. информация о записях пациентов,
7. информация об анализах,
8. информация о выписанных рецептах,
9. информация о выданных листках нетрудоспособности,
10. информация о назначенных лечениях.

Перечисленные данные можно подразделить на две группы:

- условно-постоянные(предоставляемые услуги, пациенты)
- оперативно-обновляемые(медицинские карты, график приема пациентов врачами, информация об анализах и т.д.).

## Глава 3 Логическое проектирование

### 3.1 Определение логического проектирования

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

Опираясь на концептуальную модель, проектируется логическую модель. Часто для выполнения этой задачи используются различные модели данных, например ER-модель. 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.[5]

### 3.2 ER-модель базы данных поликлиники

На основе концептуальной модели поликлиники, описанной в конце предыдущей главы, можно спроектировать ER-модель данной предметной области, и представить полученную модель с помощью стандартной графической нотации ER-диаграммы (диаграммы сущность-связь).

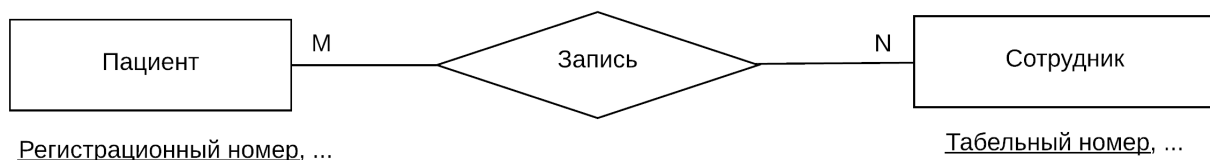


Рис. 3.1 Запись пациента к сотруднику (врачу)

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 Посещения врача пациентом по медкарте

Тип связи "Посещения" - M:N.

Класс принадлежности необязательный для обоих экземпляров сущностей. Обоснование: Сотрудник может принимать нуль или более пациентов по медкарте, медкарта может содержать информацию о нуль или более посещениях пациентам.

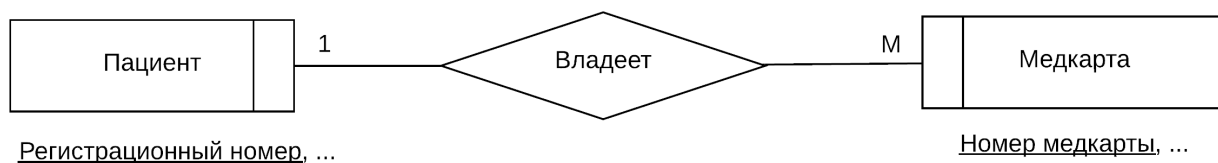


Рис. 3.3 Принадлежность медицинской карты пациенту

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.

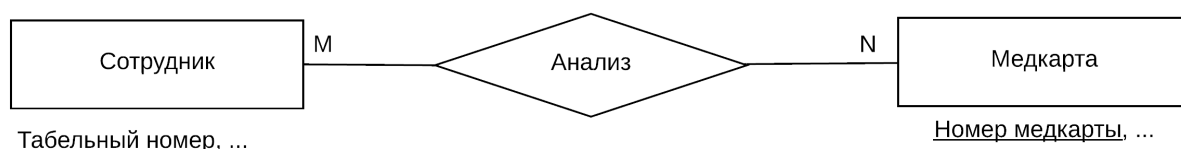


Рис. 3.4 Назначение анализов пациенту по медкарте

Тип связи “Анализ” - M:N.

Класс принадлежности необязательный для обоих экземпляров сущностей.  
Обоснование: Сотрудник может назначить анализ нуль или более пациентам по медкарте, по медкарте анализ могут назначить нуль или более сотрудников.



Рис. 3.5 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 медицинской карте, по медицинской карте пациента могут лечить нуль или более сотрудников. .

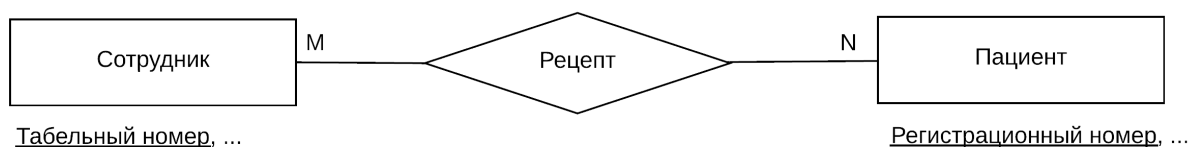


Рис. 3.6 Выписывание врачом рецепта пациенту

Тип связи “Рецепт” - M:N.

Класс принадлежности необязательный для обоих экземпляров сущностей.  
Обоснование: Сотрудник может выписать рецепты нуль или более пациентам,



пациент могут получить рецепты от нуля или более врачей.



Рис. 3.7 Оказание услуги пациенту по медкарте

Тип связи “Оказывается” - M:N.

Класс принадлежности необязательный для обоих экземпляров сущностей.

Обоснование: Услуга может быть оказана нулю или более пациентам, пациент может получить нуль или более услуг.

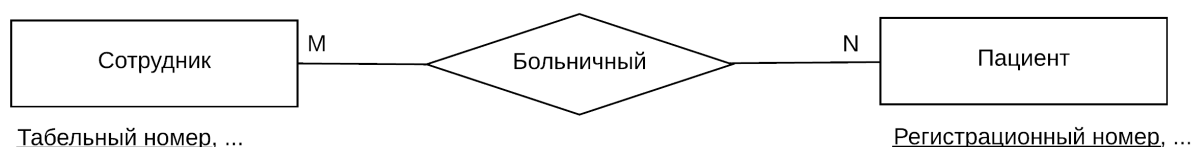


Рис. 3.8 Выписка больничного врачом пациенту

Тип связи “Больничный” - M:N.

Класс принадлежности необязательный для обоих экземпляров сущностей.

Обоснование: Сотрудник может выписать больничные нуль или более пациентам, пациент может получить больничные от нуля или более сотрудников.

### 3.3 Преобразование ER-модели в реляционную

1. Связь “Запись” удовлетворяет условиям правила 6, откуда получаются следующие таблицы

- Пациент(Регистрационный номер, ...);
- Сотрудник(Табельный номер, ...);
- Запись(Табельный номер, Регистрационный номер, ...).

2. Связь “Посещения” удовлетворяет условиям правила 6, откуда получаются следующие таблицы
- Пациент(Регистрационный номер, ...);
  - Сотрудник(Табельный номер, ...);
  - Посещения(Табельный номер, Регистрационный номер, ...).
3. Связь “Владеет” удовлетворяет условиям правила 4, откуда получаются следующие таблицы
- Пациент(Регистрационный номер, ...);
  - Медкарта(Номер медкарты, Регистрационный номер, ...).
4. Связь “Анализ” удовлетворяет условиям правила 6, откуда получаются следующие таблицы
- Сотрудник(Табельный номер, ...);
  - Медкарта(Номер медкарты, ...).
  - Анализ(Табельный номер, номер медкарты, ...).
5. Связь “Лечение” удовлетворяет условиям правила 6, откуда получаются следующие таблицы
- Сотрудник(Табельный номер, ...);
  - Медкарта(Номер медкарты, ...);
  - Лечение(Табельный номер, номер медкарты, ...).
6. Связь “Рецепт” удовлетворяет условиям правила 6, откуда получаются следующие таблицы
- Сотрудник(Табельный номер, ...);
  - Пациент(Регистрационный номер, ...);

- Рецепт(Табельный номер, Регистрационный номер, ...).
7. Связь “Оказывается” удовлетворяет условиям правила 6, откуда получают следующие таблицы
- Услуга(Номер услуги, ...).
  - Медкарта(Номер медкарты, ...);
  - Оказание(Название, Номер медкарты, ...);
8. Связь “Больничный” удовлетворяет условиям правила 6, откуда получают следующие таблицы
- Пациент(Регистрационный номер, ...);
  - Сотрудник(Табельный номер, ...);
  - Больничный(Табельный номер, Регистрационный номер, ...);

В итоге получается 11 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. Посещения(Табельный номер, Регистрационный номер, дата посещения, цель визита).
5. Медкарта(Номер медкарты, Регистрационный номер, дата заведения, тип).

6. Анализ(Табельный номер, номер медкарты, дата сдачи, вид анализа, результат).
7. Лечение(Табельный номер, номер медкарты, дата назначения, заболевание, назначенное лечение).
8. Услуга(Номер услуги, название, описание, стоимость).
9. Оказание(Название, Номер медкарты, дата оказания);
10. Рецепт(Табельный номер, Регистрационный номер, медикаменты, дата выдачи).
11. Больничный(Табельный номер, Регистрационный номер, дата выдачи, период освобождения от деятельности, место работы/обучения);

## 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 работе для реализации базы данных поликлиники была выбрана реляционная 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).[6]

## 4.2 Схема БД поликлиники

На основании ER-модели, полученной в конце предыдущей главы, где были описаны таблицы и их атрибуты, 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, в приложении Б приведены примеры их заполнения.

Таблица 4.1

Структура таблицы «Пациент»

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			‘М’ или ‘Ж’	
address	text					
pnumber	text					
password	text	not null		abs(random())		

Таблица 4.2

Структура таблицы «Сотрудник»

Столбец	Тип данных	Нуль?	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			‘М’ или ‘Ж’	
address	text					
pnumber	text					
password	text	not null		abs(random())		

Таблица 4.3

Структура таблицы «Запись»

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

Таблица 4.4

Структура таблицы «Медкарта»

Столбец	Тип дан-ных	Нуль?	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 4.5

Структура таблицы «Результат анализов»

Столбец	Тип дан-ных	Нуль?	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 4.6

Структура таблицы «Результат анализов»

Столбец	Тип дан-ных	Нуль?	Ключ	По умолч.	Огранич.	Ссылка
stabid	integer	not null	первичный, внешний			tabid в «Со-труд-ник»
dtabid	integer	not null	первичный, внешний			tabid в «Со-труд-ник»
crttime	datetime	not null		datetime('now')		
lhst	text					

Таблица 4.7

Структура таблицы «Результат анализов»

Столбец	Тип дан-ных	Нуль?	Ключ	По умолч.	Огранич.	Ссылка
tabid	integer	not null	первичный, внешний			tabid в «Со-труд-ник»
regid	integer	not null	первичный, внешний			regid в «Па-ци-ент»
crttime	datetime	not null		datetime('now')		
lhst	text					

Таблица 4.8

Структура таблицы «Результат анализов»

Столбец	Тип дан-ных	Нуль?	Ключ	По умолч.	Огранич.	Ссылка
tabid	integer	not null	первичный, внешний			tabid в «Со-труд-ник»
cardid	integer	not null	первичный, внешний			cardid в «Мед-карта»
trdate	date	not null		date('now')		
illness	text					
treatment	text					

Таблица 4.9

Структура таблицы «Результат анализов»

Столбец	Тип дан-ных	Нуль?	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 4.10

Структура таблицы «Результат анализов»

Столбец	Тип дан- ных	Ноль?	Ключ	По умолч.	Огранич.	Ссылка
servid	integer	not null	первичный, внешний			servid в «Услу- га»
regid	integer	not null	первичный, внешний			regid в «Паци- ент»
rdate	date	not null		date('now')		
pdate	datetime	not null				
status	text	not null				

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

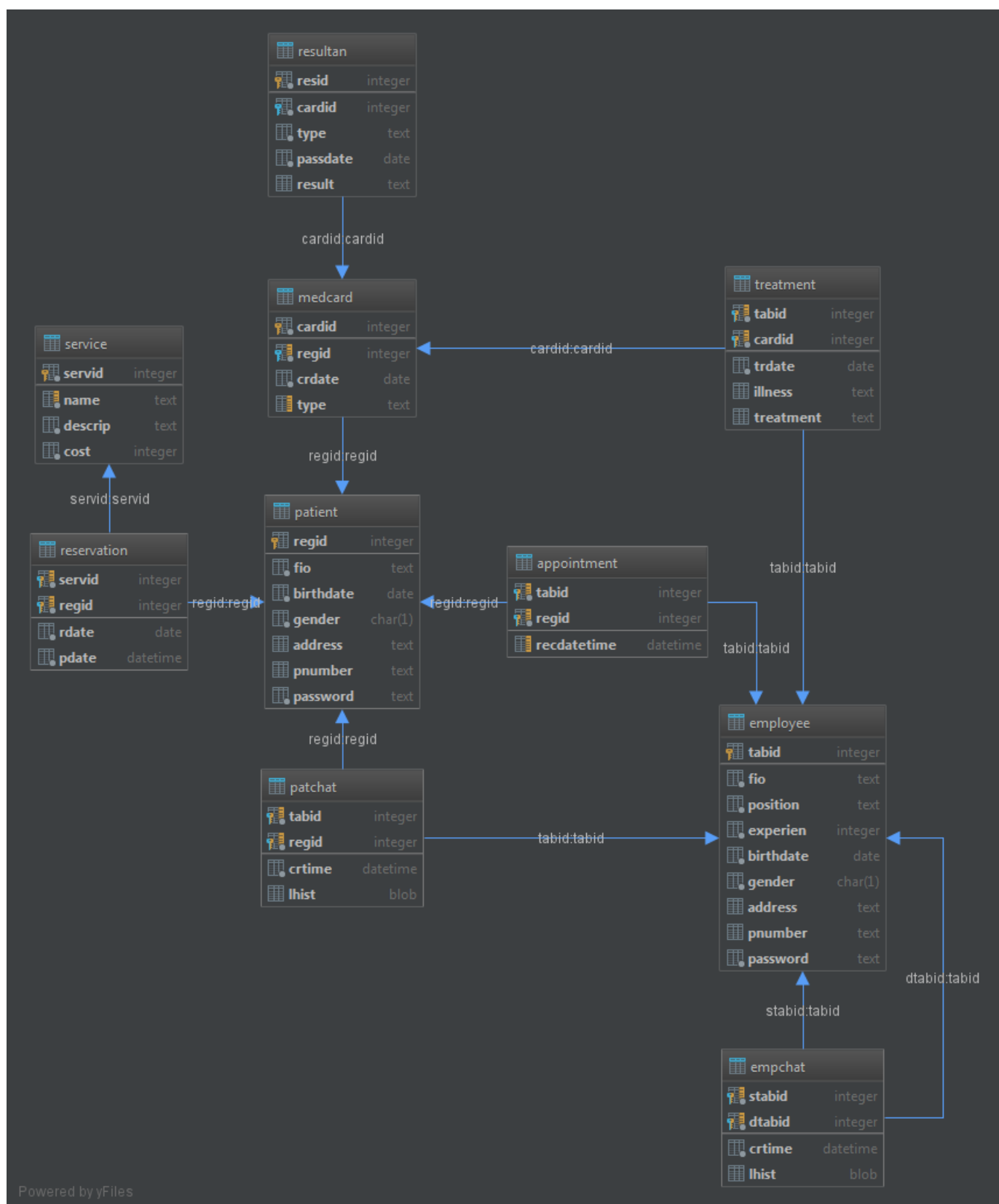


Рис. 4.1 Схема базы данных поликлиники

# 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] *Owens Mike*. The Definitive Guide to SQLite. — 1-ое изд. — Apress, 2006. — С. 440.

# Приложение А

Листинг 4.1: 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 4.2: 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 4.3: 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 4.4: 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 4.5: 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 4.6: Создание таблицы «Связь СПациентом»

```

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 4.7: Создание таблицы «Связь ССотрудником»

```

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 4.8: 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 4.9: Создание таблицы «Услуга»

```

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

```

#### Листинг 4.10: 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)
);

```

## Приложение Б