**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №11**

**«Создание приложения»**

**по дисциплине «Базы данных»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8383 |  | Киреев К.А. |
| Преподаватель |  | Фомичева Т.Г. |

Санкт-Петербург

2020

## Содержание.

[Содержание. 2](#_Toc60033501)

[Цель работы. 4](#_Toc60033502)

[Задание. 4](#_Toc60033503)

[Выполнение работы. 6](#_Toc60033504)

[1. Этап проектирования БД методом: «сущность - связь» (ER-методом). 6](#_Toc60033505)

[2. Генерация предварительных отношений и проверка их на соответствие НФБК. 7](#_Toc60033506)

[3. Проектирование базы данных в MS Access 14](#_Toc60033507)

[4. Разработка запросов 17](#_Toc60033508)

[5. Разработка интерфейса пользователя 28](#_Toc60033509)

[5.1. Основная информация 28](#_Toc60033510)

[5.2. Форма авторизации 30](#_Toc60033511)

[5.3. Форма «АвторизацияРегистратура» 31](#_Toc60033512)

[5.4. Форма «АвторизацияГлавныйВрач» 33](#_Toc60033513)

[5.5. Форма «АвторизацияВрач» 35](#_Toc60033514)

[5.6. Форма «1 Информация о больном» 35](#_Toc60033515)

[5.7. Форма «2 ФИО лечащего врача больного» 36](#_Toc60033516)

[5.8. Форма «3 Дни приёма врача» 37](#_Toc60033517)

[5.9. Форма «Расписание подчиненная форма1» 38](#_Toc60033518)

[5.10. Форма «4 Больные на лечении» 38](#_Toc60033519)

[5.11. Форма «5 Назначения при заболевании» 39](#_Toc60033520)

[5.12. Форма «6 Врач в кабинете» 40](#_Toc60033521)

[5.13. Форма «7 Посещения за последний месяц» 42](#_Toc60033522)

[5.14. Форма «8 Количество пациентов за прошедший месяц» 43](#_Toc60033523)

[5.15. Форма «Добавить\_кабинет» 44](#_Toc60033524)

[5.16. Форма «Добавить\_врача» 44](#_Toc60033525)

[5.17. Форма «Новое\_посещение» 45](#_Toc60033526)

[5.18. Форма «Подчиненная\_адрес\_врач» 46](#_Toc60033527)

[5.19. Форма «Новая\_карточка» 47](#_Toc60033528)

[5.20. Форма «Уволить\_врача» 48](#_Toc60033529)

[5.21. Форма «Выдать\_больничный\_лист» 49](#_Toc60033530)

[5.22. Отчет «Расписание\_работы» 49](#_Toc60033531)

[5.23. Отчет «Справка» 51](#_Toc60033532)

[Выводы. 52](#_Toc60033533)

## Цель работы.

Создание небольшой автоматизированной системы, объединяющей таблицы, запросы, экранные формы и отчеты, и предназначенной для решения задачи из некоторой предметной области**.**

## Задание.

Пусть требуется создать программную систему, предназначенную для врачей и работников регистратуры поликлиники. Такая система должна хранить сведения об участках, которые относятся к поликлинике, о расписании работы участковых врачей, информацию о врачах, а также карточки пациентов. Карточка имеет номер, в нее заносятся сведения о каждом посещении поликлиники пациентом: дата посещения, жалобы, предварительный диагноз, назначения, выписан или нет больничный лист, и, если выписан, то на какой срок, имя врача. В карточке на первой странице указаны также фамилия, имя, отчество пациента, его домашний адрес, пол и возраст, номер страхового полиса, дата заполнения карточки. В расписании работы врачей указывается, на каком участке работает врач, дни и часы приема, номер кабинета. Врач может обслуживать более одного участка. В случае увольнения врача его участок(участки)передается другим врачам. Данные о враче, которые хранятся в БД, — это фамилия, имя отчество, категория, стаж работы, дата рождения. В карточку больного при каждом его посещении поликлиники врачом заносится очередная запись. Работники регистратуры регистрируют пациента, заполняя первую страницу его карточки. Уволить врача имеет право только заведующий поликлиникой. Он удаляет из базы сведения о враче и передает его больных другому врачу.

Работникам поликлиники могут потребоваться следующие сведения:

* Адрес данного больного, дата последнего посещения поликлиники и диагноз?
* Фамилия и инициалы лечащего врача данного больного?
* Номер кабинета, дни и часы приема данного врача?
* Больные, находящиеся в данный момент на лечении у данного врача (не истек срок больничного листа);
* Назначения врачей при указанном заболевании?
* Кто работает в данный момент в указанном кабинете?
* Сколько раз за прошедший месяц обращался в поликлинику указанный больной?
* Какое количество больных обслужил за прошедший месяц каждый из врачей поликлиники?

Необходимо предусмотреть возможность выдачи пациенту больничного листа и справки установленного образца, в которой указано, когда, и по какому поводу пациент обращался в поликлинику в текущем году. Расписание работы врачей поликлиники, которое можно повесить в холле, также должно формироваться автоматически.

## Выполнение работы.

## Этап проектирования БД методом: «сущность - связь» (ER-методом).

Была спроектирована ER-модель, представленная на рис. 1.1.

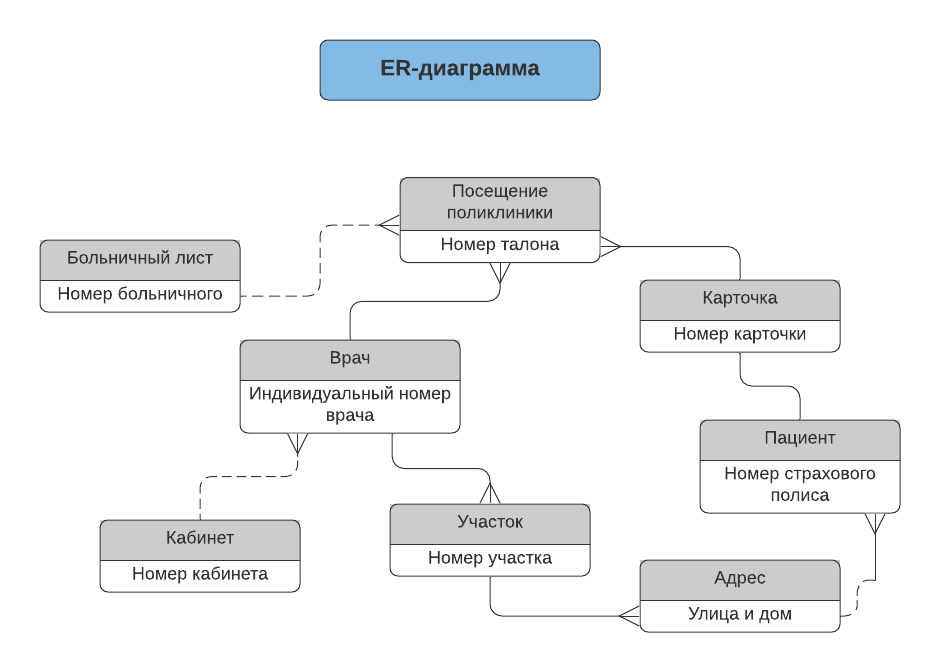


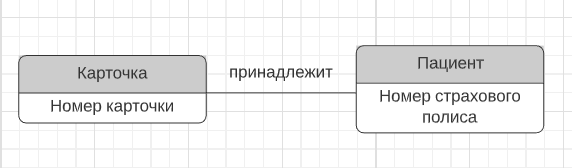
Рисунок 1.1 – ER-модель

ER-модель содержит следующие сущности:

1. Сущность «Посещение поликлиники» содержит основную информацию о посещении пациентом поликлиники. Ключевое поле – Номер талона.
2. Сущность «Больничный лист» содержит данные о больничном листе пациента. Ключевое поле – Номер больничного.
3. Сущность «Карточка» содержит информацию о пациентах, прикрепленных к поликлинике, требуемую для лечения. Ключевое поле ­– Номер карточки
4. Сущность «Пациент» содержит личную информацию о пациентах, прикрепленных к поликлинике. Ключевое поле – Номер страхового полиса.
5. Сущность «Врач» содержит информацию о врачах, работающих в поликлинике. Ключевое поле – Индивидуальный номер врача.
6. Сущность «Кабинет» содержит информацию о кабинетах поликлиники. Ключевое поле – Номер кабинета.
7. Сущность «Участок» содержит информацию об участках, относящихся к поликлинике. Ключевое поле – Номер участка.
8. Сущность «Адрес» содержит информацию о прикреплении пациента к участку поликлиники. Ключевое поле – Улица и дом.

## Генерация предварительных отношений и проверка их на соответствие НФБК.

Рассмотрим правила генерации отношений:

****

Связь между сущностью «Карточка» и сущностью «Пациент» со стороны сущности «Карточка» можно обозначить «карточка принадлежит пациенту», а со стороны сущности «Пациент»: «пациент владеет карточкой». Карточка заводится только на одного пациента, и каждый пациент имеет в регистратуре только одну карточку. Связь обязательна с обеих сторон, так как больной обязательно должен иметь карточку в регистратуре, а карточка заводится на конкретного больного. Следовательно, связь между отношениями должна быть 1:1.

По правилу 1 генерируется одно отношение:

**Карточка (Номер карточки**, Фамилия пациента, Имя пациента, Отчество пациента, Улица и дом, Номер квартиры, Пол, Возраст, Номер страхового полиса, Дата заполнения карточки, Номер паспорта)

Что же вы всё в одну кучу? Сначала разберитесь с предварительными отношениями, которые генерируются по правилам, а потом уже распределяйте оставшиеся атрибуты. Улица и дом появятся за счёт связи Пациента и Адреса, пока ещё рано.

****

Связь между сущностью «Больничный лист» и сущностью «Посещение поликлиники» со стороны сущности «Больничный лист» можно обозначить «БЛ может быть выписан в результате посещения», а со стороны сущности «Посещение поликлиники»: «посещение поликлиники может изменить больничный лист». Больничный лист может быть выписан в результате нескольких посещений больным врача, а посещение поликлиники не всегда связано с выдачей, продлением или закрытием больничного листа. Следовательно, связь между отношениями должна быть 1:n.

По правилу 5 генерируется три отношения:

**Больничный лист (Номер больничного**, Срок истечения)

**Изменение больничного листа (**Номер больничного, Номер талона, Действие)

**Посещение поликлиники (Номер талона**, Дата посещения, Жалобы, ИН врача, Номер карточки, Диагноз, Назначение)

Опять всё в куче.

****

Связь между сущностью «Посещение поликлиники» и сущностью «Карточка» со стороны сущности «Посещение поликлиники» можно обозначить «посещение заносится в карточку», а со стороны сущности «Карточка»: «карточка заполняется посещениями». Посещение записывается в одну конкретную карточку, а карточка может заполняться несколькими посещениями. Следовательно, связь между отношениями должна быть 1:n. По правилу 4 генерируется два отношения, ключ сущности «Карточка» был добавлен как атрибут в отношение «Посещение поликлиники»:

**Посещение поликлиники (Номер талона**, Дата посещения, Жалобы, ИН врача, Номер карточки, Диагноз, Назначение)

**Карточка (Номер карточки**, Фамилия пациента, Имя пациента, Отчество пациента, Улица и дом, Номер квартиры, Пол, Возраст, Номер страхового полиса, Дата заполнения карточки, Номер паспорта)

****

Связь между сущностью «Врач» и сущностью «Посещение поликлиники» со стороны сущности «Врач» можно обозначить «врач осматривает пациента во время посещения», а со стороны сущности «Посещение поликлиники»: «посещение пациентом поликлиники обслуживается врачом». Осмотр во время посещения проводит только один врач, а врач может осматривать разных пациентов в разные посещения. Следовательно, связь между отношениями должна быть 1:n.

По правилу 4 генерируется два отношения, ключ сущности «Врач» был добавлен как атрибут в отношение «Посещение поликлиники»:

**Посещение поликлиники (Номер талона**, Дата посещения, Жалобы, ИН врача, Номер карточки, Диагноз, Назначение) Да вы его уже и раньше добавили, не известно почему.

**Врач (Индивидуальный номер врача**, Фамилия, Имя, Отчество, Категория, Стаж работы, Дата рождения)

****

Связь между сущностью «Врач» и сущностью «Кабинет» со стороны сущности «Врач» можно обозначить «врач принимает в кабинете», а со стороны сущности «Кабинет»: «кабинет может быть занят врачом». Врач принимает в одном и том же кабинете, но в другую смену или в другие дни в этом кабинете может работать другой, приписанный к этому кабинету врач. Следовательно, связь между отношениями должна быть 1:n. Не пояснили, почему не обязательная связь со стороны врача.

По правилу 5 генерируется три отношения:

**Врач (Индивидуальный номер врача**, Фамилия, Имя, Отчество, Категория, Стаж работы, Дата рождения)

**Расписание (**ИН кабинета, ИН врача, Дни, Смена)

**Кабинет (ИН кабинета**, Корпус, Этаж)

Почему вдруг **ИН кабинета,** если был номер кабинета?

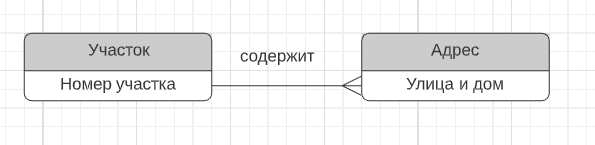
****

Связь между сущностью «Врач» и сущностью «Участок» со стороны сущности «Врач» можно обозначить «врач обслуживает участок», а со стороны сущности «Участок»: «участок обслуживается врачом». На участке один врач, но допускается обслуживание одним врачом нескольких участков. Следовательно, связь между отношениями должна быть 1:n. Что с обязательностью?

По правилу 4 генерируется два отношения, ключ сущности «Врач» был добавлен как атрибут в отношение «Участок»:

**Участок (Номер участка**, ИН врача)

**Врач (Индивидуальный номер врача**, Фамилия, Имя, Отчество, Категория, Стаж работы, Дата рождения)

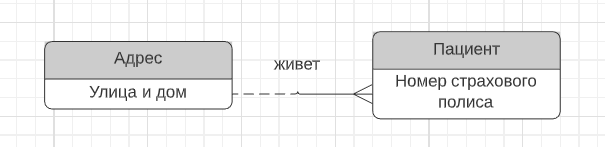
****

Связь между сущностью «Адрес» и сущностью «Участок» со стороны сущности «Адрес» можно обозначить «адрес входит в участок», а со стороны сущности «Участок»: «участок содержит адреса». Участок может содержать множество адресов, а адрес может входить только в один участок. Следовательно, связь между отношениями должна быть 1:n. Обязательная?

По правилу 4 генерируется два отношения, ключ сущности «Участок» был добавлен как атрибут в отношение «Адрес»:

**Адрес (Улица и дом**, Номер участка)

**Участок (Номер участка**, ИН врача)

****

Связь между сущностью «Адрес» и сущностью «Пациент» со стороны сущности «Адрес» можно обозначить «по адресу проживает пациент», а со стороны сущности «Пациент»: «пациент имеет адрес проживания». По одному адресу живет много пациентов, но каждый из них имеет только один адрес. Связь обязательна только со стороны пациента: пациент обязан иметь место жительства, но пациенты не обязаны проживать по каждому адресу. Следовательно, связь между отношениями должна быть 1:n.

По правилу 4 генерируется два отношения, ключ сущности «Адрес» был добавлен как атрибут в отношение «Карточка». Стоит отметить, что в предыдущих шагах сущности «Пациент» и «Карточка» были объединены в одно отношение «Карточка»: И что же с этим делать?

**Карточка (Номер карточки**, Фамилия пациента, Имя пациента, Отчество пациента, Улица и дом, Номер квартиры, Пол, Возраст, Номер страхового полиса, Дата заполнения карточки, Номер паспорта)

Т.е. подменили отношение. А зачем вообще делали сразу 2 сущности? Все атрибуты пациента указываются обязательно в карточке при её заполнении.

**Адрес (Улица и дом**, Номер участка)

В результате были получены следующие предварительные отношения: В результате чего? Вот вы дальше ещё какие-то атрибуты добавляете, корпус и этаж, например.

**Посещение поликлиники (Номер талона**, Дата посещения, Жалобы, ИН врача, Номер карточки, Диагноз, Назначение)

**Врач (Индивидуальный номер врача**, Фамилия, Имя, Отчество, Категория, Стаж работы, Дата рождения)

**Карточка (Номер карточки**, Фамилия пациента, Имя пациента, Отчество пациента, Улица и дом, Номер квартиры, Пол, Возраст, Номер страхового полиса, Дата заполнения карточки, Номер паспорта)

**Адрес (Улица и дом**, Номер участка)

**Участок (Номер участка**, ИН врача)

**Расписание (**ИН кабинета, ИН врача, Дни, Смена)

**Кабинет (ИН кабинета**, Корпус, Этаж)

**Больничный лист (Номер больничного**, Срок истечения)

**Изменение больничного листа (**Номер больничного, Номер талона, Действие)

Определим функциональную зависимость м/у атрибутами каждого отношения и проверим соответствие ее НФБК. У каждого отношения несколько ФЗ Вот понимаете, по каким-то мелочам сразу виден источник, с которого вы копируете.

1. В отношении «Врач» функциональные зависимости выглядят следующим образом:

Индивидуальный номер врача Фамилия, Имя, Отчество, Категория, Стаж работы, Дата рождения

Детерминант является ключом отношения, других зависимостей нет, поэтому отношение находится в НФБК. Это требует пояснения.

1. В отношении «Кабинет» функциональные зависимости выглядят следующим образом:

ИН кабинета Корпус, Этаж

Детерминант является ключом отношения, других зависимостей нет, поэтому отношение находится в НФБК.

1. В отношении «Расписание» функциональных зависимостей нет, как и ключей, так что отношение находится в НФБК. Как это нет?
2. В отношении «Участок» функциональные зависимости выглядят следующим образом:

Номер участка ИН врача

Детерминант является ключом, других зависимостей нет, поэтому отношение «Участок» находится в НФБК. Надо пояснить, почему нет обратной.

1. В отношении «Адрес» функциональные зависимости выглядят следующим образом:

Улица и дом Номер участка

Детерминант является ключом, других зависимостей нет, поэтому отношение «Адрес» находится в НФБК.

1. В отношении «Карточка» функциональные зависимости выглядят следующим образом:

Номер карточки Фамилия пациента, Имя пациента, Отчество пациента, Улица и дом, Номер квартиры, Пол, Возраст, Номер страхового полиса, Дата заполнения карточки, Номер паспорта

Номер паспорта Фамилия пациента, Имя пациента, Отчество пациента, Улица и дом, Номер квартиры, Пол, Возраст

Номер страхового полиса Фамилия пациента, Имя пациента, Отчество пациента, Пол, Возраст, Номер паспорта

Это разве всё?

Все детерминанты являются возможными ключами, других зависимостей нет, поэтому отношение «Карточка» находится в НФБК.

1. В отношении «Посещение поликлиники» функциональные зависимости выглядят следующим образом:

Номер талона Дата посещения, Жалобы, ИН врача, Номер карточки, Диагноз, Назначение

Детерминант является ключом, других зависимостей нет, поэтому отношение «Посещение поликлиники» находится в НФБК. Есть и другие.

1. В отношении «Больничный лист» функциональные зависимости выглядят следующим образом:

Номер больничного Срок истечения

Детерминант является ключом, других зависимостей нет, поэтому отношение «Больничный лист» находится в НФБК.

1. В отношении «Изменение больничного листа» функциональных зависимостей нет, как и ключей, так что отношение находится в НФБК. Есть, конечно.

Все отношения находятся в НФБК.

Проект не закончен.

## Проектирование базы данных в MS Access Это уже реализация, а не проектирование.

В MS Access была спроектирована база данных, основанная на вышеперечисленных отношениях. Схема данных представлена на рис. 1.2.

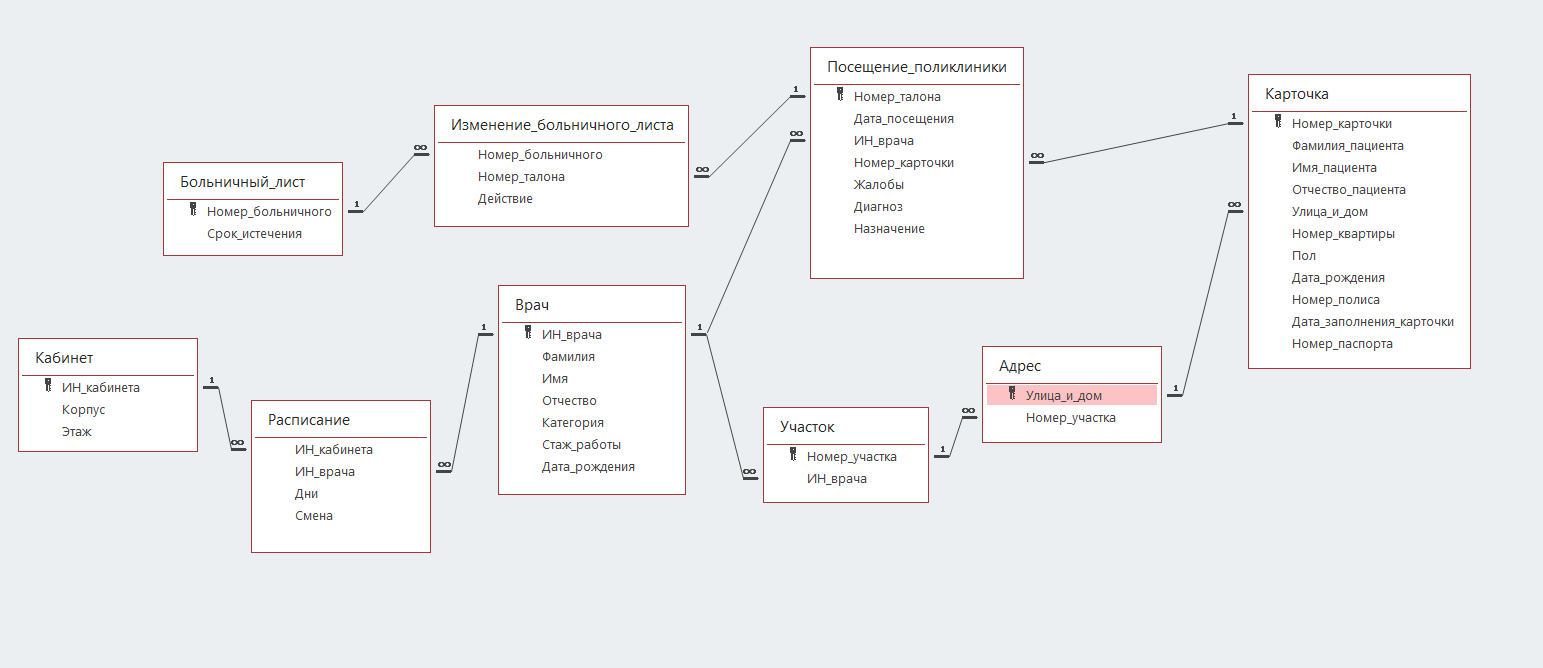


Рисунок 1.2 – Схема данных в MS Access

Прежде, чем заполнять таблицы данными, неплохо бы создать их структуру.

Далее было выполнено заполнение таблиц данными.

Таблица «Кабинет» представлена на рис. 2.1.

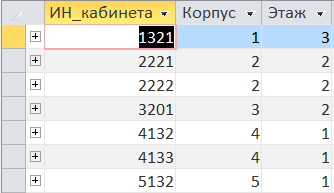


Рисунок 2.1 – Таблица «Кабинет»

Таблица «Врач» представлена на рис. 2.2.

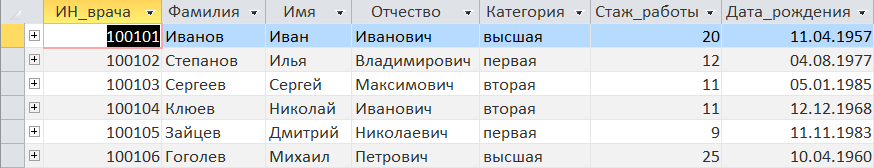


Рисунок 2.2 – Таблица «Врач»

Таблица «Участок» представлена на рис. 2.3.

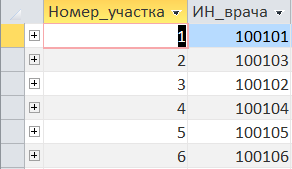


Рисунок 2.3 – Таблица «Участок»

Таблица «Адрес» представлена на рис. 2.4.

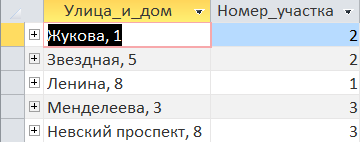


Рисунок 2.4 – Таблица «Адрес»

Таблица «Больничный лист» представлена на рис. 2.5.

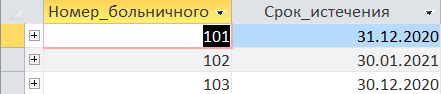


Рисунок 2.5 – Таблица «Больничный лист»

Таблица «Изменение больничного листа» представлена на рис. 2.6.

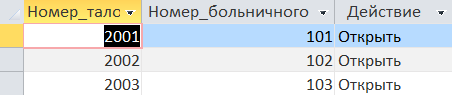


Рисунок 2.6 – Таблица «Изменение больничного листа»

Таблица «Расписание» представлена на рис. 2.7.

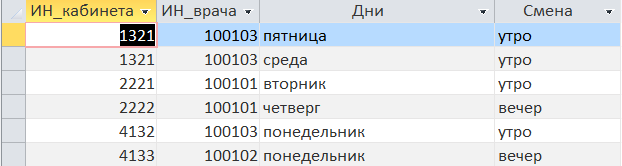


Рисунок 2.7 – Таблица «Расписание»

Таблица «Карточка» представлена на рис. 2.8.

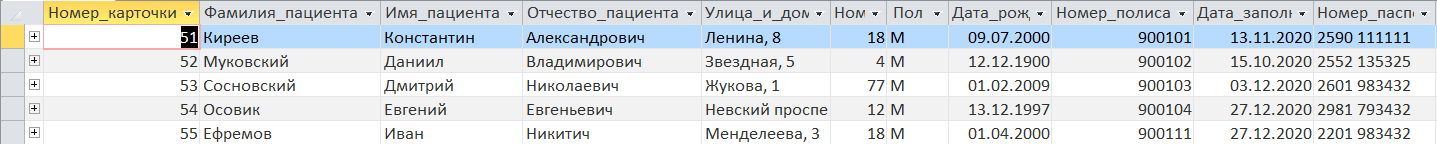


Рисунок 2.8 – Таблица «Карточка»

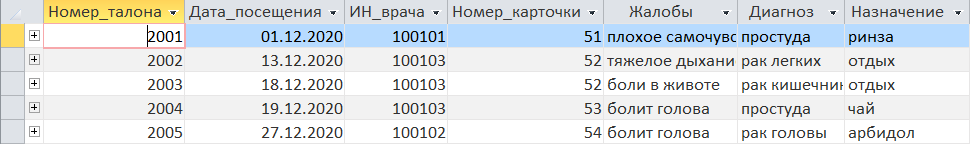
Таблица «Посещение поликлиники» представлена на рис. 2.9.

Рисунок 2.9 – Таблица «Посещение поликлиники»

Для полей были настроены их свойства, например, поле номер паспорта в таблице «Карточка» имеет условие на значение «Like "#### ######"», что обеспечивает возможность корректного ввода номера паспорта.

## Разработка запросов

Для наполнения интерфейса пользователя и обеспечения его действий был разработан ряд запросов.

* Запрос «1 Адрес, дата и диагноз» необходим для получения информации об адресе больного, дате его последнего посещения поликлиники и поставленном диагнозе. SQL-код запроса представлен ниже:

SELECT Карточка.Фамилия\_пациента, Карточка.Улица\_и\_дом, Карточка.Номер\_квартиры, Max(Посещение\_поликлиники.Дата\_посещения) AS [Max-Дата\_посещения], Last(Посещение\_поликлиники.Диагноз) AS [Last-Диагноз]

FROM Карточка INNER JOIN Посещение\_поликлиники ON Карточка.Номер\_карточки = Посещение\_поликлиники.Номер\_карточки

GROUP BY Карточка.Фамилия\_пациента, Карточка.Улица\_и\_дом, Карточка.Номер\_квартиры, Карточка.Номер\_полиса

ORDER BY Карточка.Фамилия\_пациента, Max(Посещение\_поликлиники.Дата\_посещения) DESC;

Результатом запроса является таблица, в которой содержатся необходимые сведения о пациентах. Таблица представлена на рис. 3.

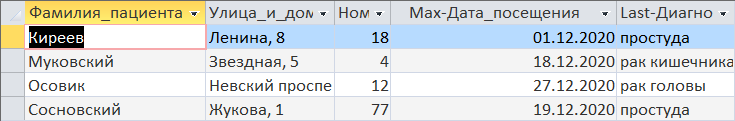


Рисунок 3 – Результат запроса «1 Адрес, дата и диагноз»

* Запрос «2 ФИО лечащего врача» необходим для получения списка пациентов с указанием их лечащих врачей. SQL-код запроса представлен ниже:

SELECT Карточка.Фамилия\_пациента, [Врач].[Фамилия] & " " & Left([Врач].[Имя],1) & "." & Left([Врач].[Отчество],1) & "." AS Выражение1

FROM Карточка INNER JOIN (Врач INNER JOIN Посещение\_поликлиники ON Врач.ИН\_врача = Посещение\_поликлиники.ИН\_врача) ON Карточка.Номер\_карточки = Посещение\_поликлиники.Номер\_карточки

GROUP BY Карточка.Фамилия\_пациента, [Врач].[Фамилия] & " " & Left([Врач].[Имя],1) & "." & Left([Врач].[Отчество],1) & ".", Карточка.Номер\_полиса;

Результатом является таблица с двумя столбцами, где указаны фамилии пациентов и ФИО лечащих врачей. Таблица представлена на рис. 4.

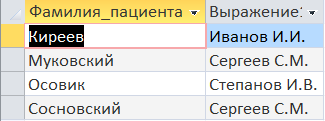


Рисунок 4 – Результат запроса «2 ФИО лечащего врача»

* Запрос «3 Расписание врача» необходим для получения информации о расписании каждого врача, с указанием кабинета, дней и смен. SQL-код запроса представлен ниже:

SELECT Врач.Фамилия, Кабинет.ИН\_кабинета, Расписание.Дни, Расписание.Смена, Кабинет.Корпус, Кабинет.Этаж

FROM Кабинет INNER JOIN (Врач INNER JOIN Расписание ON Врач.ИН\_врача = Расписание.ИН\_врача) ON Кабинет.ИН\_кабинета = Расписание.ИН\_кабинета

ORDER BY Врач.Фамилия, Кабинет.ИН\_кабинета, Расписание.Дни;

Результат выполнения запроса представлен на рис. 5.

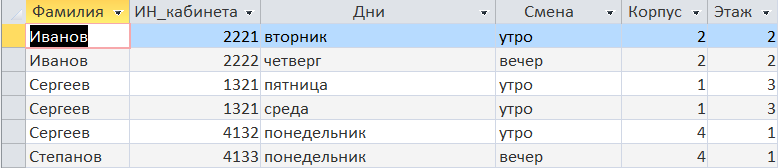


Рисунок 5 – Результат запроса «3 Расписание врача»

* Запрос «4 Текущие больные» необходим для определения пациентов, у которых ещё не истек срок больничного листа. SQL-код запроса представлен ниже:

SELECT Больничный\_лист.Срок\_истечения, [Больные у врача].Фамилия\_пациента, [Больные у врача].Имя\_пациента, [Больные у врача].Отчество\_пациента, [Больные у врача].Фамилия

FROM Больничный\_лист INNER JOIN ([Больные у врача] INNER JOIN Изменение\_больничного\_листа ON [Больные у врача].Номер\_талона = Изменение\_больничного\_листа.Номер\_талона) ON Больничный\_лист.Номер\_больничного = Изменение\_больничного\_листа.Номер\_больничного

GROUP BY Больничный\_лист.Срок\_истечения, [Больные у врача].Фамилия\_пациента, [Больные у врача].Имя\_пациента, [Больные у врача].Отчество\_пациента, [Больные у врача].Фамилия

HAVING (((Больничный\_лист.Срок\_истечения)>Date()));

Результат выполнения запроса представлен на рис. 6.

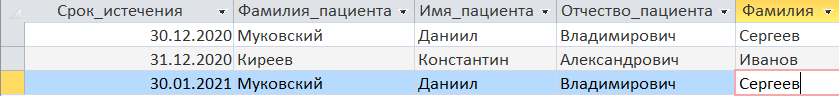


Рисунок 6 – Результат запроса «4 Текущие больные»

* Запрос «5 Назначения при заболевании» необходим для получения списка заболеваний и назначений по ним, с указанием фамилии врача, назначившего лечение. SQL-код представлен ниже:

SELECT Посещение\_поликлиники.Назначение, Врач.Фамилия, Посещение\_поликлиники.Диагноз

FROM Врач INNER JOIN Посещение\_поликлиники ON Врач.ИН\_врача = Посещение\_поликлиники.ИН\_врача;

Результат выполнения запроса представлен на рис. 7.

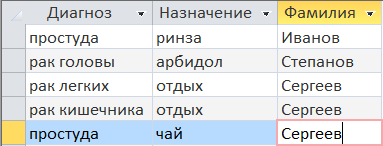


Рисунок 7 – Результат запроса «5 Назначения при заболевании»

* Запрос «6\_1 Врач в кабинете» необходим для получения номера и ФИО врача, который работает в указанном кабинете в указанное время. SQL-код запроса представлен ниже:

SELECT Врач.ИН\_врача, Врач.Фамилия, Врач.Имя, Врач.Отчество, Кабинет.ИН\_кабинета, Расписание.Дни, Расписание.Смена

FROM Кабинет INNER JOIN (Врач INNER JOIN Расписание ON Врач.ИН\_врача = Расписание.ИН\_врача) ON Кабинет.ИН\_кабинета = Расписание.ИН\_кабинета;

Результат выполнения запроса представлен на рис. 8.

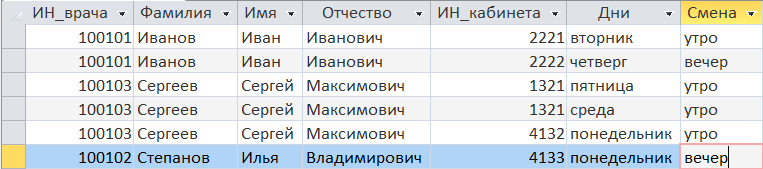


Рисунок 8 – Результат запроса «6\_1 Врач в кабинете»

* Запрос «7 Посещения за последний месяц» необходим для получения списка пациентов с указанием количества посещений за последний месяц. SQL-код запроса представлен ниже:

SELECT Карточка.Фамилия\_пациента, Карточка.Имя\_пациента, Карточка.Отчество\_пациента, Count(Mid([Дата\_посещения],4,2)) AS Выражение1

FROM Карточка INNER JOIN Посещение\_поликлиники ON Карточка.Номер\_карточки = Посещение\_поликлиники.Номер\_карточки

GROUP BY Карточка.Фамилия\_пациента, Карточка.Имя\_пациента, Карточка.Отчество\_пациента, Mid([Дата\_посещения],4,2), Карточка.Номер\_карточки

HAVING (((Mid([Дата\_посещения],4,2))=Month(Date())));

Результат выполнения запроса представлен на рис. 9.

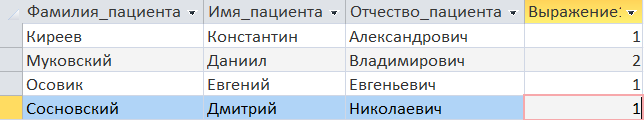


Рисунок 9 – Результат запроса «7 Посещения за последний месяц»

* Запрос «8 Количество пациентов за прошедший месяц» необходим для получения списка врачей с указанием количества пациентов за последний месяц. SQL-код запроса представлен ниже:

SELECT Врач.ИН\_врача, Врач.Фамилия, Врач.Имя, Врач.Отчество, Count(Mid([Дата\_посещения],4,2)) AS Выражение1

FROM Врач INNER JOIN Посещение\_поликлиники ON Врач.ИН\_врача = Посещение\_поликлиники.ИН\_врача

GROUP BY Врач.ИН\_врача, Врач.Фамилия, Врач.Имя, Врач.Отчество, Mid([Дата\_посещения],4,2)

HAVING (((Mid([Дата\_посещения],4,2))=Month(Date())))

ORDER BY Врач.ИН\_врача;

Результат выполнения запроса представлен на рис. 10.

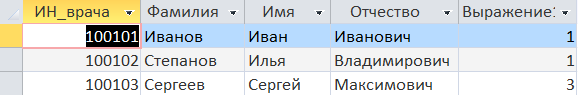


Рисунок 10 – Результат запроса «8 Количество пациентов за прошедший месяц»

* Запрос «Больные у врача» необходим для получения списка пациентов поликлиники с указанием их врачей и посещений. SQL-код запроса представлен ниже:

SELECT Карточка.Номер\_карточки, Карточка.Фамилия\_пациента, Карточка.Имя\_пациента, Карточка.Отчество\_пациента, Посещение\_поликлиники.Номер\_талона, Врач.Фамилия

FROM Карточка INNER JOIN (Врач INNER JOIN Посещение\_поликлиники ON Врач.ИН\_врача = Посещение\_поликлиники.ИН\_врача) ON Карточка.Номер\_карточки = Посещение\_поликлиники.Номер\_карточки

GROUP BY Карточка.Номер\_карточки, Карточка.Фамилия\_пациента, Карточка.Имя\_пациента, Карточка.Отчество\_пациента, Посещение\_поликлиники.Номер\_талона, Врач.Фамилия;

Результат выполнения запроса представлен на рис. 11.

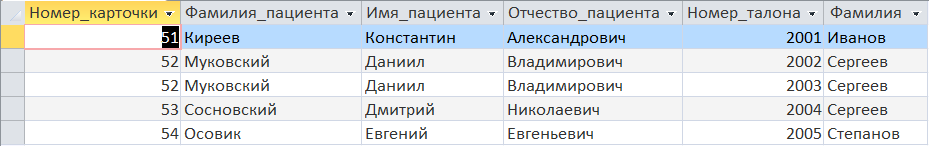


Рисунок 11 – Результат запроса «Больные у врача»

* Запрос «Больные у врача» необходим для получения списка пациентов поликлиники с указанием их врачей и посещений. SQL-код запроса представлен ниже:

SELECT Карточка.Номер\_карточки, Карточка.Фамилия\_пациента, Карточка.Имя\_пациента, Карточка.Отчество\_пациента, Посещение\_поликлиники.Номер\_талона, Врач.Фамилия

FROM Карточка INNER JOIN (Врач INNER JOIN Посещение\_поликлиники ON Врач.ИН\_врача = Посещение\_поликлиники.ИН\_врача) ON Карточка.Номер\_карточки = Посещение\_поликлиники.Номер\_карточки

GROUP BY Карточка.Номер\_карточки, Карточка.Фамилия\_пациента, Карточка.Имя\_пациента, Карточка.Отчество\_пациента, Посещение\_поликлиники.Номер\_талона, Врач.Фамилия;

Результат выполнения запроса представлен на рис. 11.

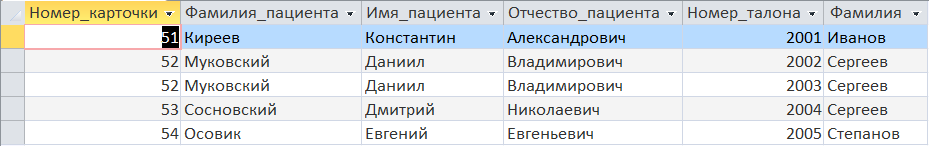


Рисунок 11 – Результат запроса «Больные у врача»

* Запрос «Врач в данный момент» необходим для получения врача, который работает в указанном кабинете в текущий момент времени. SQL-код запроса представлен ниже:

SELECT Врач.ИН\_врача, Врач.Фамилия, Врач.Имя, Врач.Отчество, Расписание.Дни, Расписание.Смена, Кабинет.ИН\_кабинета

FROM Кабинет INNER JOIN (Врач INNER JOIN Расписание ON Врач.ИН\_врача = Расписание.ИН\_врача) ON Кабинет.ИН\_кабинета = Расписание.ИН\_кабинета

WHERE (((Расписание.Дни)=WeekdayName(Weekday(Date(),2))) AND ((Расписание.Смена)=IIf(Left(Time(),2)<12,"утро","вечер")) AND ((Кабинет.ИН\_кабинета)=[Номер кабинета:]));

Результат выполнения запроса представлен на рис. 12.



Рисунок 12 – Результат запроса «Врач в данный момент»

* Запрос «Выборка\_справка» необходим для выдачи справки. Запрос предоставляет основную информацию о пациенте, даты посещений поликлиники и диагнозы. SQL-код запроса представлен ниже:

SELECT Карточка.Фамилия\_пациента, Карточка.Имя\_пациента, Карточка.Отчество\_пациента, Карточка.Дата\_рождения, Карточка.Номер\_полиса, Посещение\_поликлиники.Дата\_посещения, Посещение\_поликлиники.Диагноз

FROM Карточка INNER JOIN Посещение\_поликлиники ON Карточка.Номер\_карточки = Посещение\_поликлиники.Номер\_карточки

WHERE (((Карточка.Номер\_полиса)=[Формы]![АвторизацияВрач]![Полис]));

Результат выполнения запроса представлен на рис. 13.

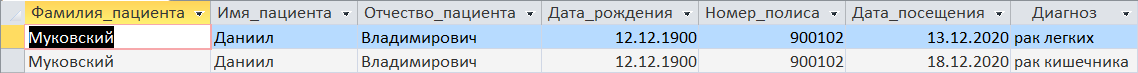


Рисунок 13 – Результат запроса «Выборка\_справка»

* Запрос «Расписание работы» необходим для создания расписания работы врачей. Запрос предоставляет информацию об участке, кабинете и смене врачей. SQL-код запроса представлен ниже:

SELECT Участок.Номер\_участка, Врач.Фамилия, Врач.Имя, Врач.Отчество, Расписание.ИН\_кабинета, Расписание.Дни, Расписание.Смена

FROM (Кабинет INNER JOIN (Врач INNER JOIN Расписание ON Врач.ИН\_врача = Расписание.ИН\_врача) ON Кабинет.ИН\_кабинета = Расписание.ИН\_кабинета) INNER JOIN Участок ON Врач.ИН\_врача = Участок.ИН\_врача

ORDER BY Участок.Номер\_участка, Врач.Фамилия, Расписание.ИН\_кабинета, Расписание.Дни, Расписание.Смена DESC;

Результат выполнения запроса представлен на рис. 14.

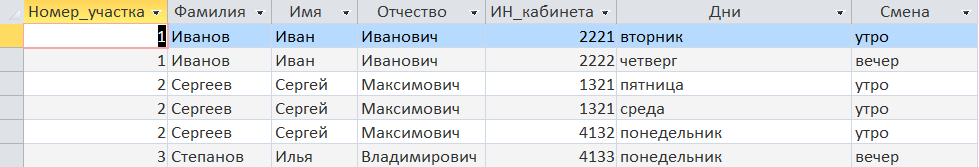


Рисунок 14 – Результат запроса «Расписание работы»

* Запрос «Новая\_карточка» - запрос на добавление новой карточки пациента. SQL-код запроса представлен ниже:

INSERT INTO Карточка ( Номер\_карточки, Фамилия\_пациента, Имя\_пациента, Отчество\_пациента, Улица\_и\_дом, Номер\_квартиры, Пол, Дата\_рождения, Номер\_полиса, Номер\_паспорта, Дата\_заполнения\_карточки )

SELECT Max([Карточка]![Номер\_карточки])+1 AS Выражение1, [Формы]![Новая\_карточка]![Фамилия] AS Выражение2, [Формы]![Новая\_карточка]![Имя] AS Выражение3, [Формы]![Новая\_карточка]![Отчество] AS Выражение4, [Формы]![Новая\_карточка]![УлицаДом] AS Выражение5, [Формы]![Новая\_карточка]![Квартира] AS Выражение6, [Формы]![Новая\_карточка]![Пол] AS Выражение7, [Формы]![Новая\_карточка]![ДатаРождения] AS Выражение8, [Формы]![Новая\_карточка]![Полис] AS Выражение9, [Формы]![Новая\_карточка]![Паспорт] AS Выражение10, [Формы]![Новая\_карточка]![ДатаЗаполнения] AS Выражение11

FROM Карточка;

* Запрос «Новая\_посещение» - запрос на добавление новой записи о посещении поликлиники. SQL-код запроса представлен ниже:

INSERT INTO Посещение\_поликлиники ( Номер\_талона, Дата\_посещения, ИН\_врача, Номер\_карточки, Жалобы, Диагноз, Назначение )

SELECT Max([Посещение\_поликлиники]![Номер\_талона])+1 AS Выражение1, [Формы]![Новое\_посещение]![Дата] AS Выражение2, [Формы]![Новое\_посещение]![Врач] AS Выражение3, [Формы]![Новое\_посещение]![Карточка] AS Выражение4, [Формы]![Новое\_посещение]![Жалобы] AS Выражение5, [Формы]![Новое\_посещение]![Диагноз] AS Выражение6, [Формы]![Новое\_посещение]![Назначение] AS Выражение7

FROM Посещение\_поликлиники;

* Запрос «Добавить\_больничный\_лист» - запрос на добавление нового больничного листа. SQL-код запроса представлен ниже:

INSERT INTO Больничный\_лист ( Номер\_больничного, Срок\_истечения )

SELECT Max([Больничный\_лист]![Номер\_больничного])+1 AS Выражение1, [Формы]![Выдать\_больничный\_лист]![Срок] AS Выражение2

FROM Больничный\_лист;

* Запрос «Добавить\_действие\_с\_больничным» - запрос на добавление нового больничного листа. SQL-код запроса представлен ниже:

INSERT INTO Изменение\_больничного\_листа ( Номер\_талона, Действие, Номер\_больничного )

SELECT [Формы]![Выдать\_больничный\_лист]![Комбинированная0] AS Выражение1, "Открыть" AS Выражение2, Max([Изменение\_больничного\_листа]![Номер\_больничного])+1 AS Выражение3

FROM Изменение\_больничного\_листа;

* Запрос «Новый\_адрес» - запрос на добавление нового адреса. SQL-код запроса представлен ниже:

INSERT INTO Адрес ( Улица\_и\_дом, Номер\_участка )

SELECT [Формы]![Новая\_карточка]![УлицаДом] AS Выражение1, [Формы]![Новая\_карточка]![Участок] AS Выражение2;

* Запрос «Новый\_врач» - запрос на добавление нового адреса. SQL-код запроса представлен ниже:

INSERT INTO Врач ( ИН\_врача, Фамилия, Имя, Отчество, Категория, Стаж\_работы, Дата\_рождения )

SELECT Max([Врач]![ИН\_врача])+1 AS Выражение1, [Формы]![Добавить\_врача]![Фамилия] AS Выражение2, [Формы]![Добавить\_врача]![Имя] AS Выражение3, [Формы]![Добавить\_врача]![Отчество] AS Выражение4, [Формы]![Добавить\_врача]![Категория] AS Выражение5, [Формы]![Добавить\_врача]![Стаж] AS Выражение6, [Формы]![Добавить\_врача]![ДатаРождения] AS Выражение7

FROM Врач;

* Запрос «Новый\_кабинет» - запрос на добавление нового адреса. SQL-код запроса представлен ниже:

INSERT INTO Кабинет ( ИН\_кабинета, Корпус, Этаж )

SELECT [Формы]![Добавить\_кабинет]![Номер] AS Выражение1, [Формы]![Добавить\_кабинет]![Корпус] AS Выражение2, [Формы]![Добавить\_кабинет]![Этаж] AS Выражение3;

* Запрос «Обновить\_посещение» - запрос на передачу больных другому врачу. SQL-код запроса представлен ниже:

UPDATE Посещение\_поликлиники SET Посещение\_поликлиники.ИН\_врача = [Формы]![Уволить\_врача]![Кому]

WHERE (((Посещение\_поликлиники.ИН\_врача)=[Формы]![Уволить\_врача]![Кого]));

* Запрос «Обновить\_участок» - запрос на передачу участков другому врачу. SQL-код запроса представлен ниже:

UPDATE Врач INNER JOIN Участок ON Врач.ИН\_врача = Участок.ИН\_врача SET Участок.ИН\_врача = [Формы]![Уволить\_врача]![Кому]

WHERE (((Врач.ИН\_врача)=[Формы]![Уволить\_врача]![Кого]));

* Запрос «Удалить\_врача» - удаление записи о враче из таблицы «Врач». SQL-код запроса представлен ниже:

DELETE Врач.ИН\_врача

FROM Врач

WHERE (((Врач.ИН\_врача)=[Формы]![Уволить\_врача]![Кого]));

* Запрос «Удалить\_кабинет» - удаление записи о расписании врача из таблицы «Расписание». SQL-код запроса представлен ниже:

DELETE Расписание.ИН\_врача

FROM Расписание

WHERE (((Расписание.ИН\_врача)=[Формы]![Уволить\_врача]![Кого]));

## Разработка интерфейса пользователя

## Основная информация

Схематичное представление интерфейса пользователя представлено на рис. 15. Стрелочками обозначается возможность перехода между формами.

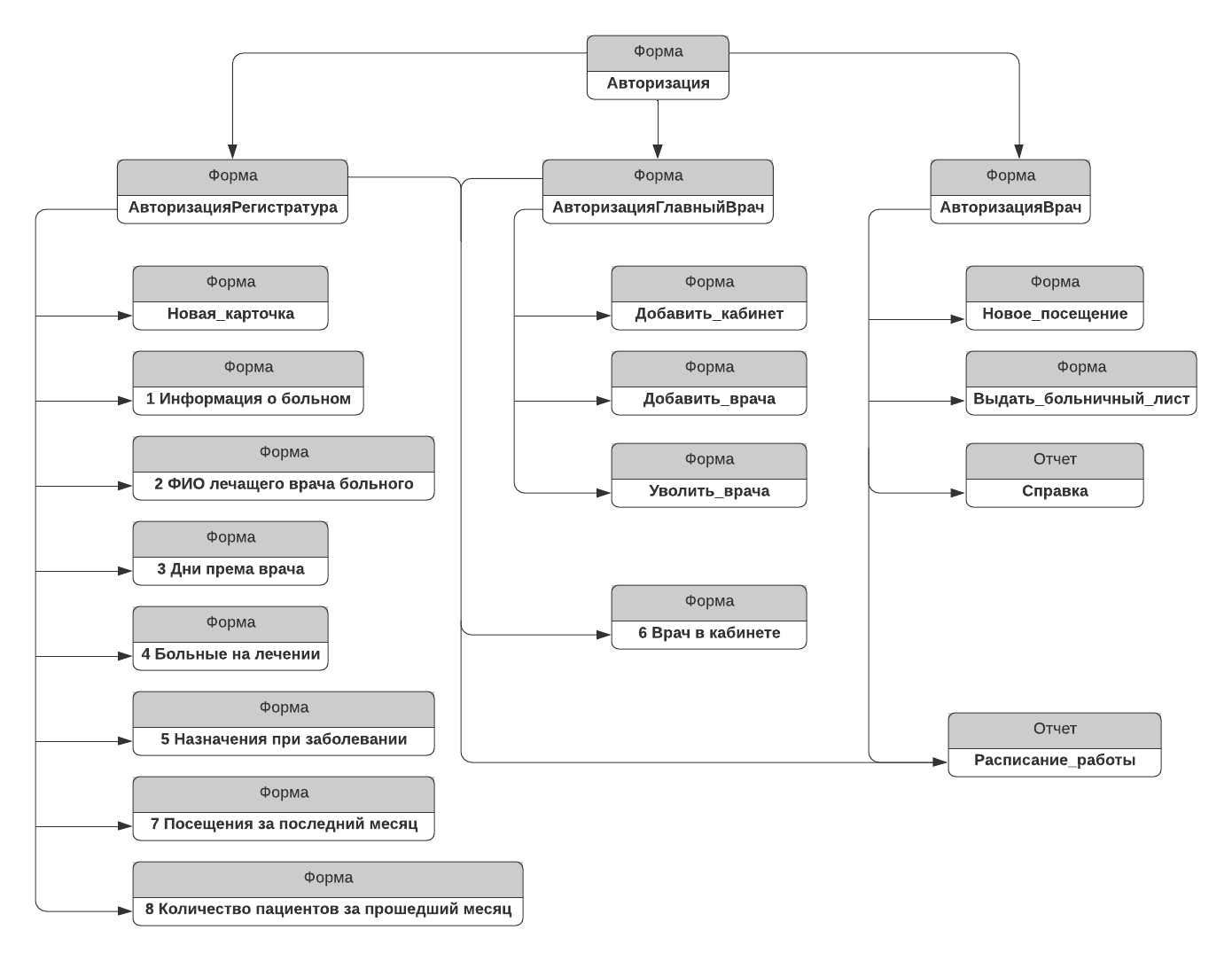


Рисунок 15 – Схема интерфейса

Странно, что в схеме нет запросов и макросов

Основные задачи разработки приложения и формы, реализующие их, представлены ниже.

1. Уволить или добавить нового врача;
   1. Форма «Добавить\_врача» позволяет добавлять врача (при добавлении можно указать основные данные)
   2. Форма «Удалить\_врача» позволяет удалить врача поликлиники и передать его пациентов другому врачу
2. Добавить новый кабинет;
   1. Форма «Добавить\_кабинет» позволяет добавлять кабинет (при добавлении можно указать характеристики)
3. Зарегистрировать нового пациента и заполнить его карточку – Форма «Новая\_карточка»;
4. Внести запись о новом посещении в карточку пациента – Форма «Новое\_посещение»;
5. Выписать больничный лист при посещении – Форма «Выдать\_больничный\_лист»;
6. Узнать требуемую работникам поликлиники информацию о пациенте;
   1. Форма «1 Информация о больном» позволяет узнать об адресе, последнем посещении и диагнозе пациента
   2. Форма «2 ФИО лечащего врача больного» позволяет узнать ФИО лечащего врача пациента
   3. Форма «7 Посещения за последний месяц» позволяет узнать количество посещений поликлиники пациентом за последний месяц
7. Узнать требуемую работникам поликлиники информацию о враче;
   1. Форма «3 Дни приема врача» позволяет узнать о расписании приема врача
   2. Форма «4 Больные на лечении» позволяет узнать пациентов, находящихся в данный момент на лечении у врача
   3. Форма «6 Врач в кабинете» позволяет узнать какой врач находится в указанном кабинете в указанное время
   4. Форма «8 Количество пациентов за прошедший месяц» позволяет узнать количество пациентов, осмотренных врачом
8. Узнать назначения врачей для конкретного заболевания – Форма «5 Назначения при заболевании»
9. Выдать справку, в которой указано, когда, и по какому поводу пациент обращался в поликлинику в текущем году;
   1. Отчет «Справка»
10. Создать расписание, которое можно повесить в холле поликлиники;

## Форма авторизации

При запуске приложения открывается форма авторизации. На форме при вводе логина и пароля и нажатия на кнопку «Войти» происходит аутентификация пользователя, который может авторизоваться как работник регистратуры, участковый врач или главный врач. При нажатии кнопки «Закрыть» происходит выход из текущей базы данных. Вид формы авторизации представлен на рис. 16.1. Макрос кнопки «Войти» представлен на рис. 16.2.

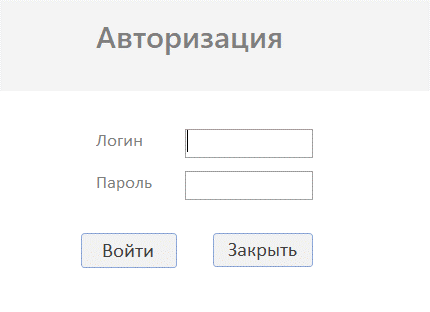


Рисунок 16.1 – Форма авторизации

При соответствующей авторизации осуществляется переход на другие формы:

* АвторизацияРегистратура – логин: “reg”, пароль: “reg”
* АвторизацияВрач – логин: “doc”, пароль: “doc”
* АвторизацияГлавныйВрач – логин: “admin”, пароль: “admin”

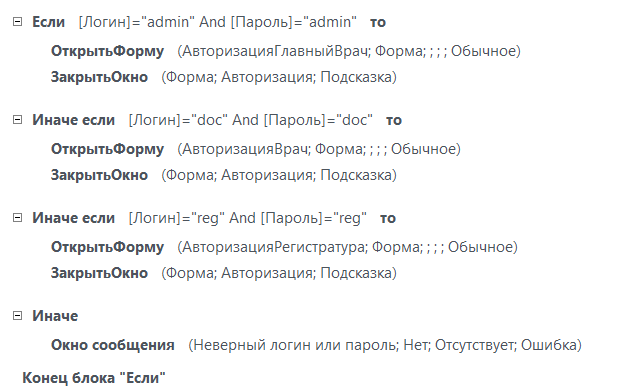


Рисунок 16.2 – Макрос на нажатие кнопки «Войти»

## Форма «АвторизацияРегистратура»

Интерфейс формы для работников регистратуры представлен на рис. 17 и 18. В форме имеется множество кнопок для перехода на другие формы или отчеты.

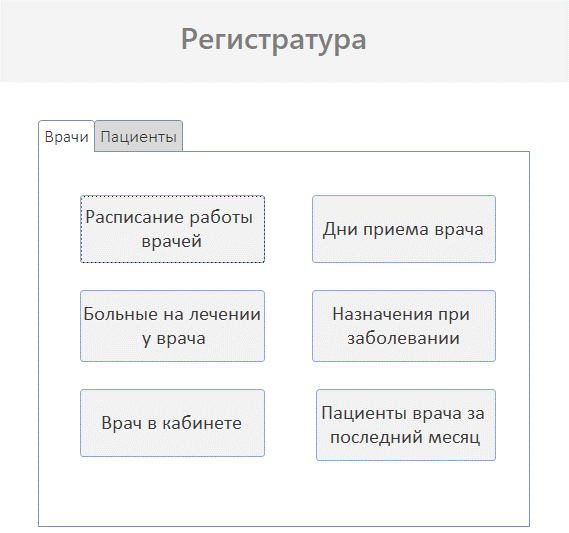


Рисунок 17 – Интерфейс формы «АвторизацияРегистратура», вкладка «Врачи»

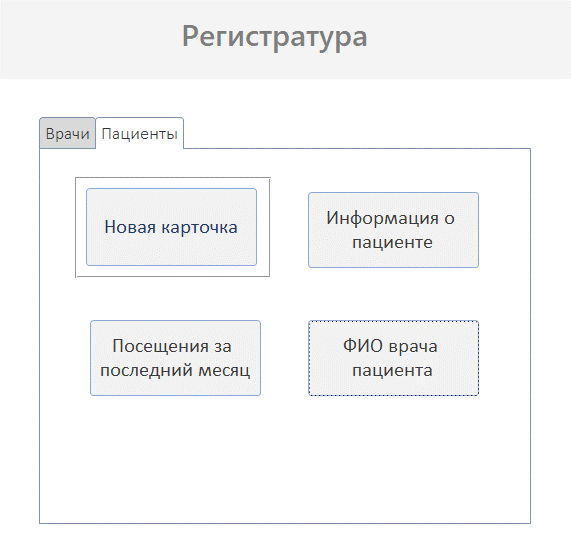


Рисунок 18 – Интерфейс формы «АвторизацияРегистратура», вкладка «Пациенты»

Вкладка «Врачи»:

* При нажатии на кнопку «Расписание работы врачей» открывается отчёт «Расписание\_работы»
* При нажатии на кнопку «Дни приема врача» открывается форма «3 Дни приема врача»
* При нажатии на кнопку «Больные на лечении у врача» открывается форма «4 Больные на лечении»
* При нажатии на кнопку «Назначения при заболевании» открывается форма «5 Назначения при заболевании»
* При нажатии на кнопку «Врач в кабинете» открывается форма «6 Врач в кабинете»
* При нажатии на кнопку «Пациенты врача за последний месяц» открывается форма «8 Количество пациентов за прошедший месяц»

Вкладка «Пациенты»:

* При нажатии на кнопку «Новая карточка» открывается форма «Новая\_карточка»
* При нажатии на кнопку «Информация о пациенте» открывается форма «1 Информация о больном»
* При нажатии на кнопку «Посещения за последний месяц» открывается форма «7 Посещения за последний месяц»
* При нажатии на кнопку «ФИО врача пациента» открывается форма «2 ФИО лечащего врача больного»

## Форма «АвторизацияГлавныйВрач»

Интерфейс формы для главного врача поликлиники представлен на рис. 19 и 20. В форме имеются кнопки для перехода на другие формы или отчеты.

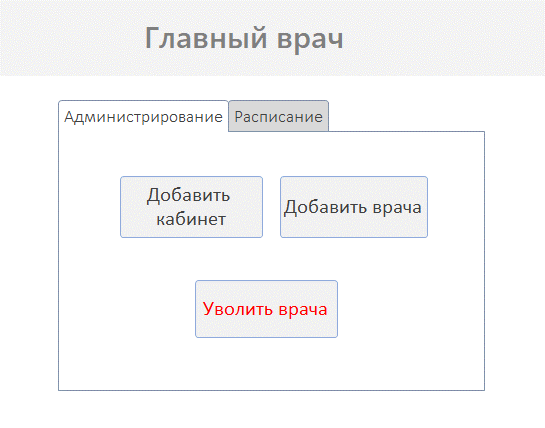


Рисунок 19 – Интерфейс формы «АвторизацияГлавныйВрач», вкладка «Администрирование»

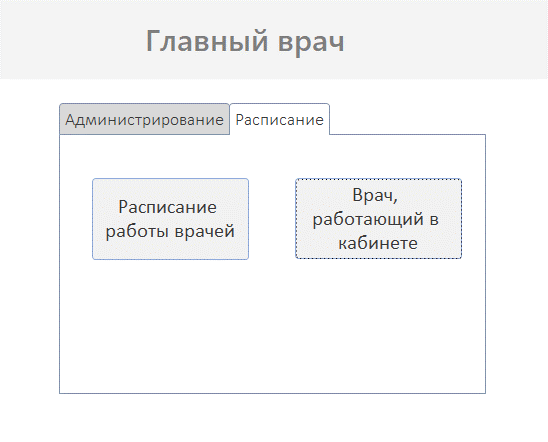


Рисунок 20 – Интерфейс формы «АвторизацияГлавныйВрач», вкладка «Расписание»

Вкладка «Администрирование»:

* При нажатии на кнопку «Добавить кабинет» открывается форма «Доавить\_кабинет»
* При нажатии на кнопку «Добавить врача» открывается форма «Добавить\_врача»
* При нажатии на кнопку «Уволить врача» открывается форма «Уволить\_врача»

Вкладка «Расписание»:

* При нажатии на кнопку «Расписание» открывается отчет «Расписание\_работы»
* При нажатии на кнопку «Врач, работающий в кабинете» открывается форма «6 Врач в кабинете»

## Форма «АвторизацияВрач»

Интерфейс формы для участкового врача поликлиники представлен на рис. 21. В форме имеются кнопки для перехода на другие формы или отчеты.

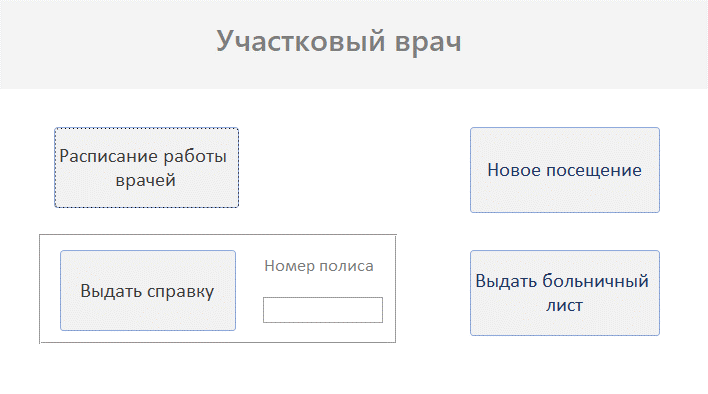


Рисунок 21 – Интерфейс формы «АвторизацияВрач»

* При нажатии на кнопку «Расписание работы врачей» открывается отчет «Расписание\_работы»
* При нажатии на кнопку «Новое посещение» открывается форма «Новое\_посещение»
* При нажатии на кнопку «Выдать больничный лист» открывается форма «Выдать\_больничный\_лист»
* Для того, чтобы выдать справку о посещениях поликлиники пациентом за последний год нужно ввести его полис в поле «Номер полиса» и затем при нажатии кнопки «Выдать справку» открывается отчет «Справка» с нужными данными для указанного пациента

## Форма «1 Информация о больном»

Форма «1 Информация о больном» показывает краткую информацию о пациенте поликлиники. На форме можно получить информацию о каждом больном – его адрес, когда он в последний раз посещал поликлинику и какой диагноз ему поставлен. По выбору фамилии пациента из поля со списком происходит вывод нужных данных. SQL-код запроса для формы был представлен ранее в одноименном запросе.

Источник строк поля со списком «Выбор пациента»:

SELECT [1 Адрес, дата и диагноз].[Фамилия\_пациента] FROM [1 Адрес, дата и диагноз];

Вид формы представлен на рис. 22.

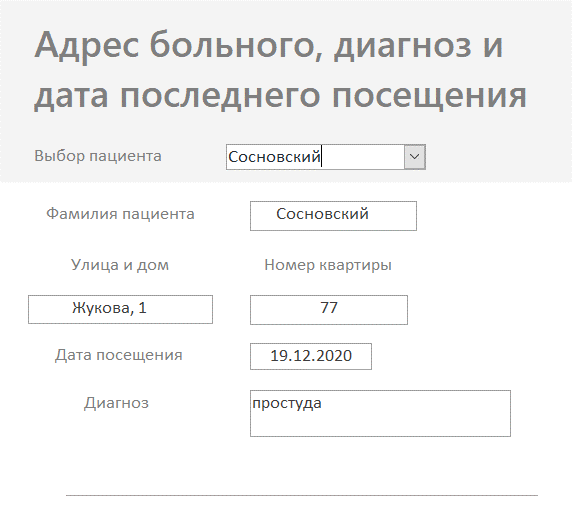


Рисунок 22 – Форма «1 Информация о больном»

## Форма «2 ФИО лечащего врача больного»

Форма «2 ФИО лечащего врача больного» представляет собой два поля, в первом указывается фамилия пациента из списка больных поликлиники и во втором соответственно выводятся фамилия и инициалы лечащего врача пациента. SQL-код запроса для формы был представлен ранее в одноименном запросе.

Источник строк поля со списком «Фамилия пациента»:

SELECT [2 ФИО лечащего врача].[Фамилия\_пациента] FROM [2 ФИО лечащего врача];

Вид формы представлен на рис. 23.

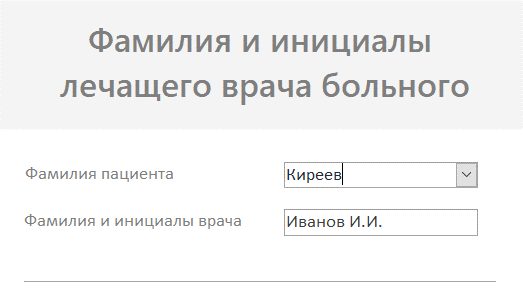


Рисунок 23 – Форма «2 ФИО лечащего врача больного»

## Форма «3 Дни приёма врача»

В этой форме выводится информация о номере кабинета, днях приёма, смене врача, который работает в данное время в этом кабинете, а также дополнительная информация о самом кабинете, такая как корпус, в котором он находится и этаж. По выбору фамилии врача из поля со списком происходит вывод нужных данных в подчиненной форме «Расписание подчиненная форма1» (поле связи – «ИН врача»). SQL-код запроса для формы был представлен ранее в одноименном запросе. Вид формы представлен на рис. 24.

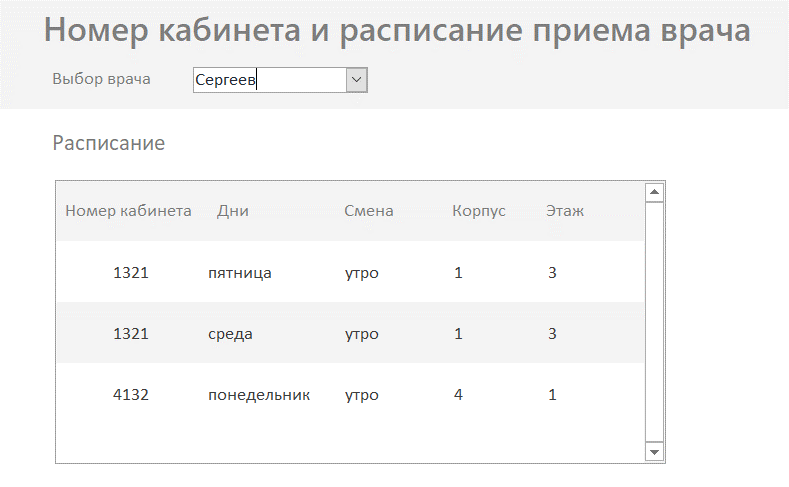


Рисунок 24 – Форма «3 Дни приёма врача»

Источник строк поля со списком «Выбор врача»:

SELECT [Врач].[ИН\_врача], [Врач].[Фамилия] FROM Врач;

## Форма «Расписание подчиненная форма1»

Данная форма является подчинённой формой для формы «3 Дни приёма врача». В этой форме выводится информация о номере кабинета, днях приёма, смене врача, который работает в данное время в этом кабинете, а также корпус, в котором находится кабинет и этаж. Вид формы представлен на рис. 25.

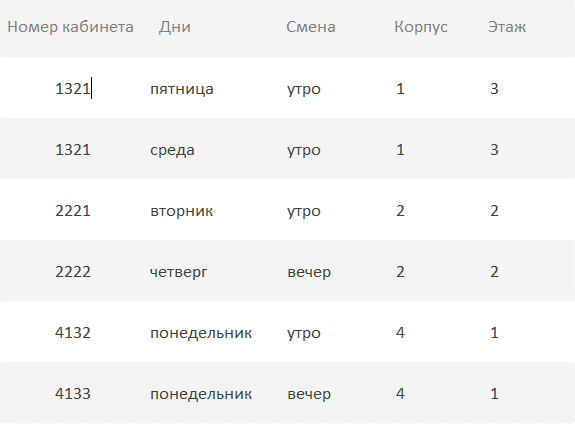


Рисунок 25 – Форма «Расписание подчиненная форма1»

## Форма «4 Больные на лечении»

На форме можно получить ФИО больных, у которых еще не истек срок действия больничного листа. Вид формы представлен на рис. 26.1. По выбору ФИО из поля со списком происходит фильтрация записей по ФИО лечащего врача. Вид формы после выбора врача из поля со списком представлен на рис. 26.2. SQL-код запроса для формы был представлен ранее в одноименном запросе.

Источник строк поля со списком «Выбор врача»:

SELECT DISTINCT Врач.Фамилия FROM Врач;

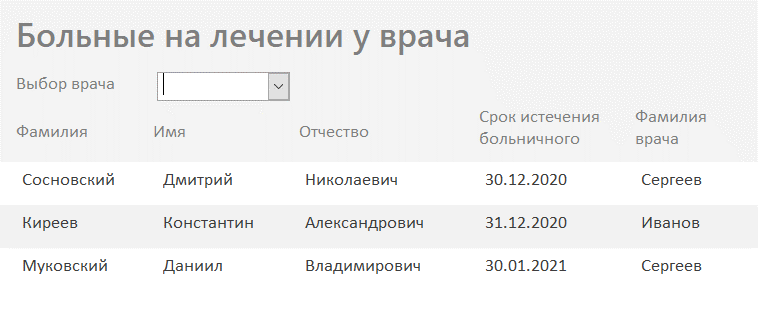


Рисунок 26.1 – Форма «4 Больные на лечении»

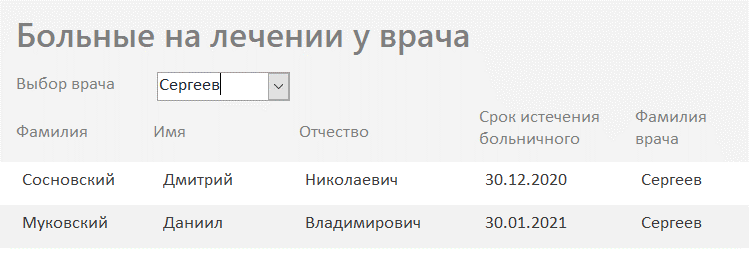


Рисунок 26.2 – Форма «4 Больные на лечении» после выбора врача

## Форма «5 Назначения при заболевании»

В данной форме выводятся диагнозы, которые ставили врачи, и соответствующие им назначения, а также фамилия врача, который поставил этот диагноз. Вид формы представлен на рис. 27.1. По выбору диагноза из поля со списком происходит фильтрация записей по диагнозу. Вид формы после выбора диагноза из поля со списком представлен на рис. 27.2. SQL-код запроса для формы был представлен ранее в одноименном запросе.

Источник строк поля со списком «Выбор диагноза»:

SELECT DISTINCT Посещение\_поликлиники.Диагноз FROM Посещение\_поликлиники;

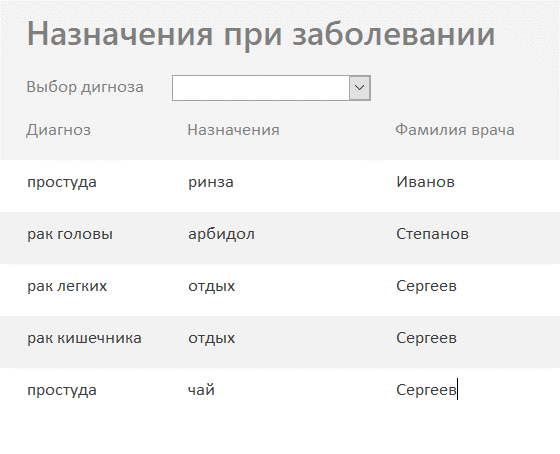


Рисунок 27.1 – Форма «5 Назначения при заболевании»

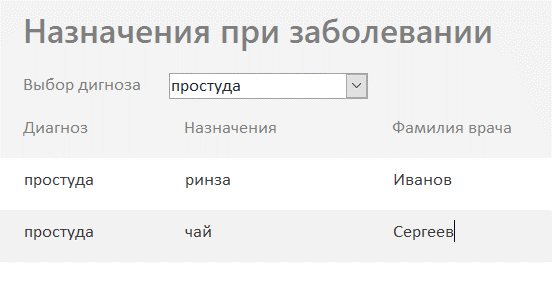


Рисунок 27.2 – Форма «5 Назначения при заболевании» после выбора диагноза

## Форма «6 Врач в кабинете»

В данной форме отображаются фамилия, имя, отчество и ИН врача для заданного кабинета и времени. Необходимо выбрать из полей со списком номер кабинета, день приёма и смену и нажать кнопку со знаком фильтрации. После этого происходит фильтрация записей по нужным полям. Вид формы представлен на рис. 28.1. Вид формы после выбора нужных атрибутов из полей со списком представлен на рис. 28.2. Также на форме имеется дополнительная кнопка для сброса всех фильтров, которая возвращает записи в исходное состояние. SQL-код запроса для формы был представлен ранее в одноименном запросе.

Источник строк поля со списком «Поиск1»:

SELECT DISTINCT Кабинет.ИН\_кабинета FROM Кабинет ORDER BY Кабинет.ИН\_кабинета;

Источник строк поля со списком «Поиск2»:

SELECT DISTINCT Расписание.Дни FROM Расписание ORDER BY Расписание.Дни;

Источник строк поля со списком «Поиск3»:

SELECT DISTINCT Расписание.Смена FROM Расписание ORDER BY Расписание.Смена;

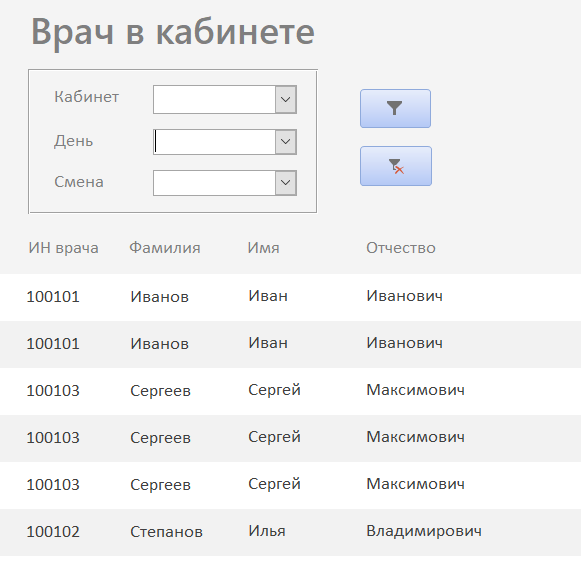


Рисунок 28.1 – Форма «6 Врач в кабинете»

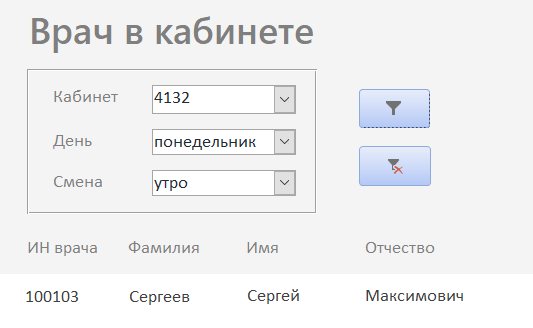


Рисунок 28.2 – Форма «6 Врач в кабинете» после фильтрации

Внедренный макрос, который активируется при нажатии кнопки фильтрации представлен на рис. 28.3.

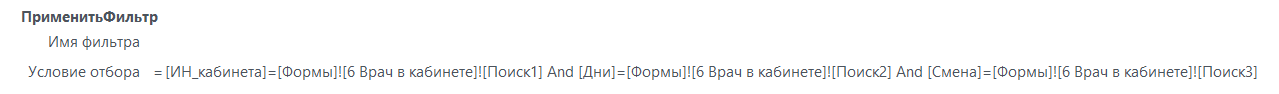


Рисунок 28.3 – Макрос на кнопке фильтрации

## Форма «7 Посещения за последний месяц»

В данной форме выводится количество посещений поликлиники за последний месяц и ФИО больного. Выводится информация только о тех больных, которые посещали поликлинику за последний месяц хотя бы один раз. По выбору фамилии пациента из поля со списком происходит вывод нужных данных. SQL-код запроса для формы был представлен ранее в одноименном запросе. Вид формы представлен на рис. 29.

Источник строк поля со списком «Выбор пациента»:

SELECT [7 Посещения за последний месяц].[Фамилия\_пациента] FROM [7 Посещения за последний месяц];

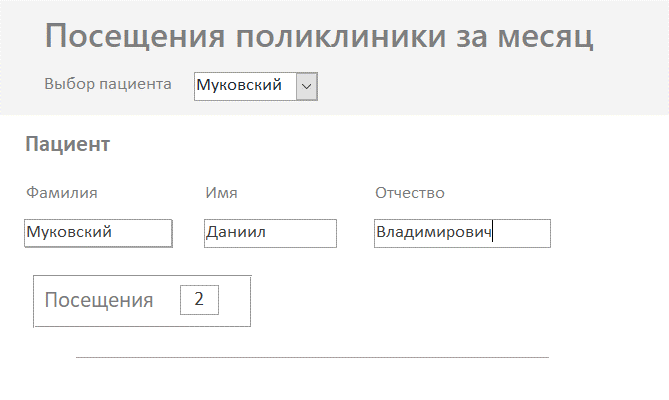


Рисунок 29 – Форма «7 Посещения за последний месяц»

## Форма «8 Количество пациентов за прошедший месяц»

На форме выводится ФИО врача с указанием количества пациентов, обслуженных им за последний месяц. По выбору фамилии врача из поля со списком происходит вывод нужных данных. SQL-код запроса для формы был представлен ранее в одноименном запросе. Вид формы представлен на рис. 30.

Источник строк поля со списком «Выбор врача»:

SELECT [8 Количество пациентов за прошедший месяц].[Фамилия] FROM [8 Количество пациентов за прошедший месяц];

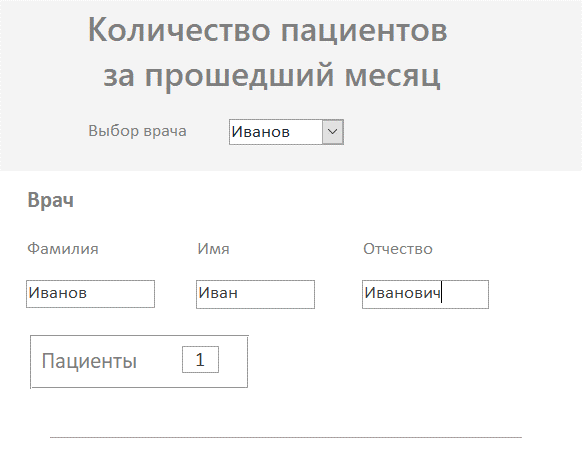


Рисунок 30 – Форма «8 Количество пациентов за прошедший месяц»

## Форма «Добавить\_кабинет»

С помощью данной формы главный врач может добавить новый кабинет в базу поликлиники. Сначала заполняются поля номера, корпуса и этажа кабинета и далее по кнопке «Добавить запись» выполняется запрос «Новый кабинет». SQL-код запроса для формы был представлен ранее. Вид формы представлен на рис. 31.

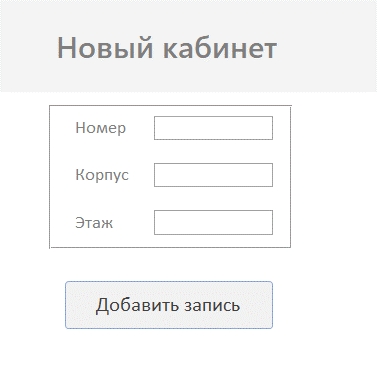


Рисунок 31 – Форма «Добавить\_кабинет»

## Форма «Добавить\_врача»

С помощью данной формы главный врач может добавить нового врача в базу поликлиники. Сначала заполняются поля фамилии, имя и отчества врача, затем выбирается категория врача из списка: высшая, первая или вторая, также заполняются поля стаж и дата рождения. Далее по кнопке «Добавить» выполняется запрос «Новый врач». SQL-код запроса для формы был представлен ранее. Вид формы представлен на рис. 32.

Источник строк поля со списком «Категория»:

"высшая";"первая";"вторая"

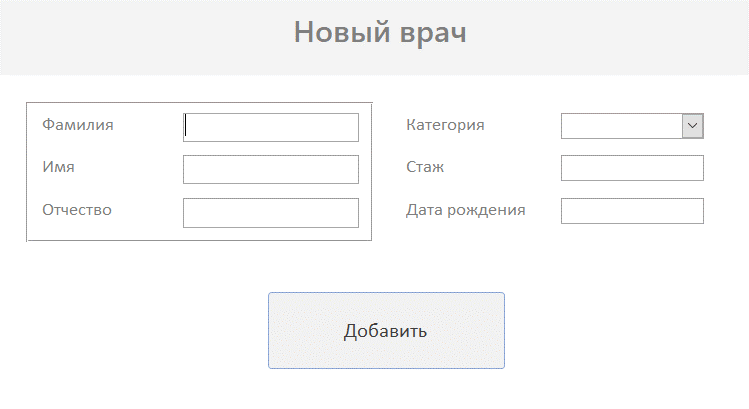


Рисунок 32 – Форма «Добавить\_врача»

## Форма «Новое\_посещение»

Для того, чтобы создать новую запись о посещении пациента в карточке, врачу необходимо заполнить поля Номер карточки, Жалобы, Диагноз, Назначение, Номер врача и нажать кнопку «Добавить». Дата заполнения устанавливается сегодняшним днем. Номер врача заполняется вручную, так как есть возможность того, что пациент посетил поликлинику, когда его участковый врач не работает. Если пациент пришел в правильное время, то с помощью подчиненной формы «Подчиненная\_адрес\_врач» (поле связи – «Номер карточки») можно узнать номер лечащего врача пациента и вписать его, при приеме пациента другим врачом он выбирается из списка. По клику на кнопку «Добавить» происходит выполнение запроса «Новое\_посещение».

Источник строк поля со списком «Номер карточки»:

*SELECT DISTINCT Карточка.Номер\_карточки FROM Карточка;*

Источник строк поля со списком «Номер врача»:

*SELECT DISTINCT Врач.ИН\_врача FROM Врач ORDER BY Врач.ИН\_врача;*

SQL-код запроса для формы был представлен ранее. Вид формы представлен на рис. 33.

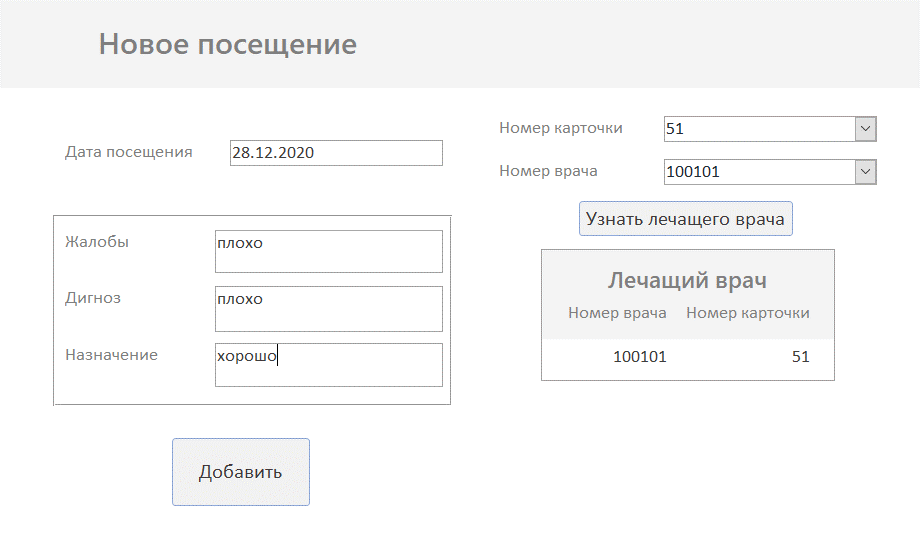


Рисунок 33 – Форма «Новое\_посещение»

## Форма «Подчиненная\_адрес\_врач»

Данная форма является подчиненной для формы «Новая\_посещение». На данной форме отображается номер участкового врача и номер карточки пациента. Вид формы представлен на рис. 34.

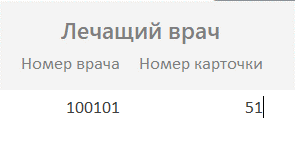


Рисунок 34 – Форма «Подчиненная\_адрес\_врач»

## Форма «Новая\_карточка»

На форме необходимо ввести данные нового пациента и на него будет заведена карта по кнопке «Добавить». Нужно заполнить поля ФИО, Номер паспорта, Номер полиса, Пол, Дата рождения, Улица и дом, Квартира, Участок. Дата заполнения ставится сегодняшняя. По клику на кнопку «Принять» выполняется запрос на добавление «Новая\_карточка». SQL-код запроса для формы был представлен ранее. Вид формы представлен на рис. 35.

Поле со списком «Номер участка» имеет следующий источник строк:

*SELECT DISTINCT Участок.Номер\_участка FROM Участок ORDER BY Участок.Номер\_участка;* Поле со списком «Пол» имеет следующий источник строк:

*"М";"Ж"*

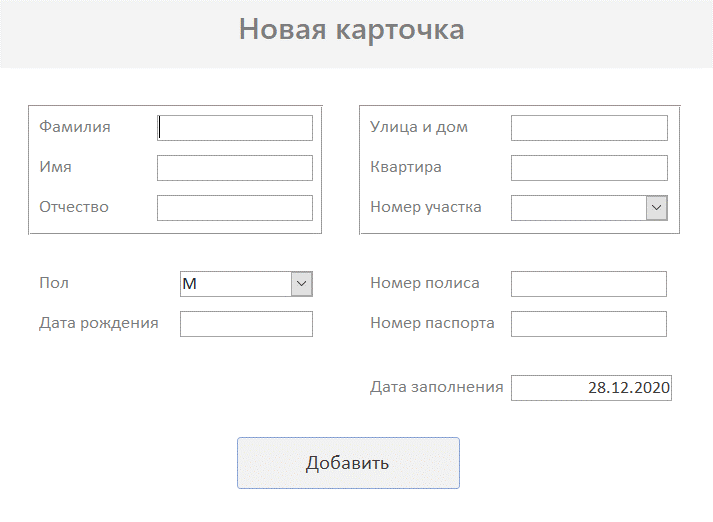


Рисунок 35 – Форма «Новая\_карточка»

## Форма «Уволить\_врача»

В списке «Кого уволить» нужно выбрать ИН врача, информация о котором будет удаляться из БД. В списке «Кому передать» нужно выбрать ИН врача, который будет в дальнейшем обслуживать участки удаляемого врача и его пациентов. По кнопке запускается макрос «Увольнение», который в свою очередь запускает по очереди четыре запроса: «Обновить\_посещение», «Обновить\_участок», «Удалить\_врача», «Удалить\_кабинет». Уволить врача может только Главный врач. SQL-код запросов для формы был представлен ранее. Вид формы представлен на рис. 36. Макрос «Увольнение» представлен на рис. 37.

Поля со списком «Кого уволить» и «Кому передать» имеет следующий источник строк:

*SELECT DISTINCT Врач.ИН\_врача FROM Врач ORDER BY Врач.ИН\_врача;*

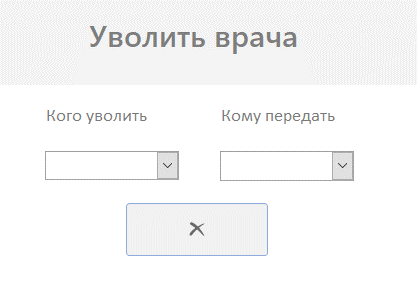


Рисунок 36 – Форма «Уволить\_врача»

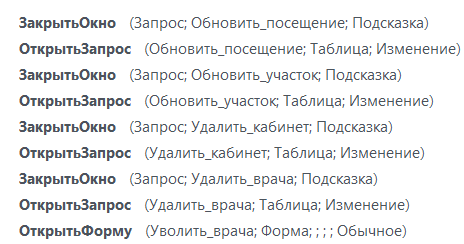


Рисунок 37 – Макрос «Увольнение»

## Форма «Выдать\_больничный\_лист»

С помощью данной формы врач может выдать больничный лист пациенту. Сначала заполняются поля номера талона и срока больничного и далее по кнопке «Выписать» выполняются запросы «Добавить\_действие\_с\_больничным» и «Добавить\_больничный\_лист». SQL-код запросов для формы был представлен ранее. Срок больничного первоначально устанавливается в значение даты посещения плюс тридцать дней, но его можно изменить. Вид формы представлен на рис. 38.

Источник строк поля со списком «Номер талона»:

*SELECT DISTINCT Посещение\_поликлиники.Номер\_талона FROM Посещение\_поликлиники WHERE (((Посещение\_поликлиники.Дата\_посещения)=Date())) ORDER BY Посещение\_поликлиники.Номер\_талона;*

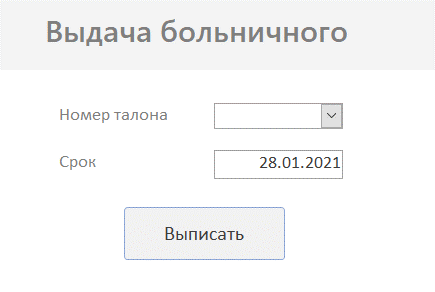


Рисунок 38 – Форма «Выдать\_больничный\_лист»

## Отчет «Расписание\_работы»

Выводит информацию о врачах, участках, на которых они работают, кабинет, дни приёма и смены. Расписание можно повесить в холле.

SQL-код запроса данных для отчета представлен ниже:

SELECT Участок.Номер\_участка, Врач.Фамилия, Врач.Имя, Врач.Отчество, Расписание.ИН\_кабинета, Расписание.Дни, Расписание.Смена

FROM Кабинет INNER JOIN ((Врач INNER JOIN Расписание ON Врач.ИН\_врача = Расписание.ИН\_врача) INNER JOIN Участок ON Врач.ИН\_врача = Участок.ИН\_врача) ON Кабинет.ИН\_кабинета = Расписание.ИН\_кабинета

ORDER BY Участок.Номер\_участка, Врач.Фамилия, Расписание.ИН\_кабинета, Расписание.Дни, Расписание.Смена DESC;

Данный отчет имеет следующую структуру:

* Номер участка, Фамилия, Имя, Отчество
  + Номер кабинета
  + Дни приема
  + Смена

Вид отчета представлен на рис. 39.

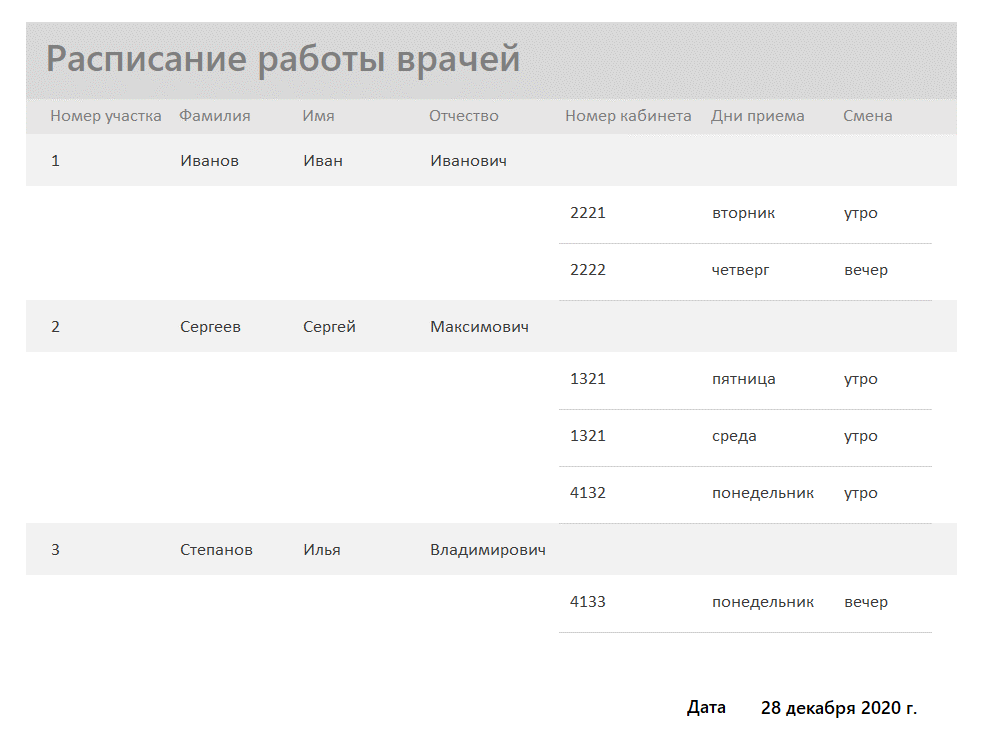


Рисунок 39 – Отчет о расписании работы врачей

## Отчет «Справка»

Выводит информацию о посещениях поликлиники за последний год с указанием даты посещения и диагноза в виде справки установленного образца.

SQL-код запроса данных для справки представлен ниже:

SELECT Карточка.Фамилия\_пациента, Карточка.Имя\_пациента, Карточка.Отчество\_пациента, Карточка.Дата\_рождения, Карточка.Номер\_полиса, Посещение\_поликлиники.Дата\_посещения, Посещение\_поликлиники.Диагноз

FROM Карточка INNER JOIN Посещение\_поликлиники ON Карточка.Номер\_карточки = Посещение\_поликлиники.Номер\_карточки

WHERE (((Карточка.Номер\_полиса)=[Формы]![АвторизацияВрач]![Полис]));

Вид справки представлен на рис. 40.

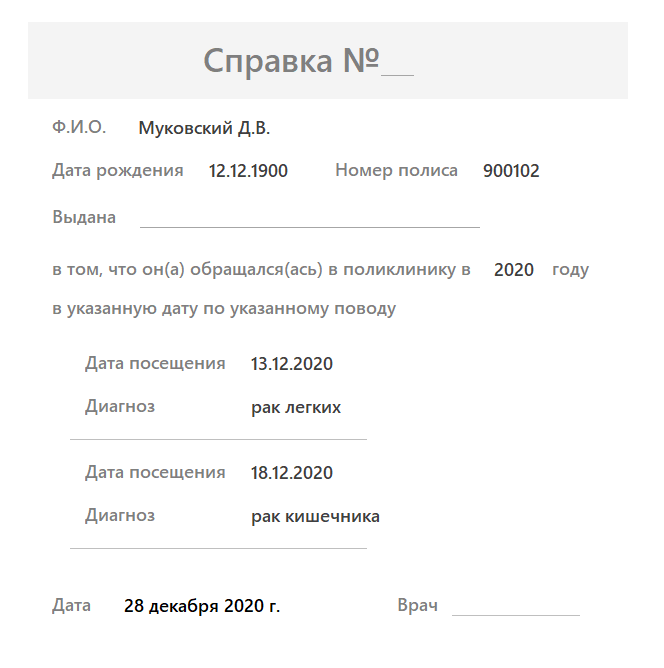


Рисунок 40 – Справка установленного образца

## Выводы.

В результате выполнения лабораторной работы были получены навыки проектирования баз данных, построения ER диаграмм, реляционной модели, реализации пользовательского интерфейса в MS Access в соответствии с задачами, которые должны быть реализуемы пользователем. В результате была разработана база данных и программа к ней, которые позволяют администрировать работу поликлиники, просматривать в различных форматах данные о врачах и пациентах, изменять их, формировать отчет и справки.