**Бюджетное учреждение высшего образования**

**Ханты-Мансийского автономного округа – Югры**

**«СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Политехнический институт

Кафедра АСОИУ

**Курсовой проект по дисциплине «Информационные технологии»**

На тему: «Разработка автоматизированной информационной системы»

**Выполнил:**

студент группы 606-11з

Хорина Вера Юрьевна

**Проверил:**

Старший преподаватель

Горбунов Дмитрий Владимирович

Сургут 2022

**ОГЛАВЛЕНИЕ**

[Введение 4](#_Toc106118848)

[1 Описание и анализ предметной области 5](#_Toc106118849)

[2 Моделирование предметной области «Поликлиника» 8](#_Toc106118850)

[3 Выбор СУБД 12](#_Toc106118851)

[4 Разработка базы данных «Поликлиника» 15](#_Toc106118852)

[4.1 Создание таблиц и схемы данных 15](#_Toc106118853)

[4.2 Создание форм 17](#_Toc106118854)

[4.3 Разработка запросов 21](#_Toc106118855)

[4.4 Создание отчётов 27](#_Toc106118856)

[5 Разработка интерфейса базы данных 33](#_Toc106118857)

[Заключение 37](#_Toc106118858)

[Список использованных источников 38](#_Toc106118859)

# Введение

Основные идеи современной информационной технологии базируются на концепции, согласно которой данные должны быть организованы в базы данных с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей. Эти базы данных создаются и функционируют под управлением специальных программных комплексов, называемых системами управления базами данных (СУБД). Как правило, базы данных являются основой построения автоматизированных информационных систем (АИС).

Моделирование и проектирование баз данных – одна из наиболее сложных и ответственных задач, связанных с созданием автоматизированных системы обработки данных.

**Цель работы -** изучение возможностей использование баз данных при разработке информационных систем.

**Актуальность темы работы -** заключается в том, что разработка базы данных ПОЛИКЛИНИКА позволит на её основе создать информационную систему для автоматизации работы сотрудников поликлиники.

**Задачи работы**–ознакомиться с методами концептуального, логического и физического моделирования, проектирования баз данных и информационных систем с использованием современных CASE-средств и СУБД MS Access.

# 1 Описание и анализ предметной области

В поликлинике ведётся учёт приёма и лечения пациентов в журнале, содержание которого можно представить в виде следующей таблице.

.

Таблица 1.1 Перечень входных (первичных) документов.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ФИО**  **пациента** | **Страховойполис** | **Паспорт** | **Палата** | **Отделение** | **ФИО   лечащего врача** | **Диагноз** | **Симптом** | **Дата**  **поступления** | **Дата**  **выписки** | **Аллергия к препаратам** | **Назначенные препараты** |
| **Иванов Иван Иванович** | **6401932071947388** | **63 03**  **435508** | **8** | **терапевтическое** | **Петрова**  **Надежда**  **Ивановна** | **острая**  **очаговая**  **пневмония** | **аскультатив-новлажныемелко-пузырчатыехрипы** | **1/04/07** | **10/04/07** | **NULL** | **антибиотики,**  **бронхолити-**  **ки** |
| **Петров Иван Сергеевич** | **6401932071727388** | **63 05**  **457108** | **5** | **терапевтическое** | **Иванова Елена   Петровна** | **язвенная болезнь** | **боли      при пальпации** | **10/05/07** | **20/04/08** | **NULL** | **кокарбо-**  **ксилаза** |
| **Семенов**  **Сергей**  **Петрович** | **6401925701727388** | **63 02**  **408515** | **5** | **терапевтическое** | **Иванова Елена   Петровна** | **язвенная болезнь** | **тошнота,**  **рвота,**  **ночные**  **голодные**  **боли** | **10/05/07** | **24/04/08** | **NULL** | **антациды** |
| **ЗубковМихаил  Кирилович** | **6401925701727396** | **63 99**  **478565** | **6** | **терапевтическое** | **Павлова**  **Марья**  **Ивановна** | **бронхиальная астма** | **приступы удушья** | **05/05/07** | **12/05/07** | **антибиотики пенициллинового ряда, поллиноз** | **преднизолон, ингаляторы** |

Таблица содержит информацию о пациентах, которые приходили в поликлинику на прием к разным врачам. Врачи бывают разных специальностей. По одной специальности могут работать несколько докторов. Прием пациента происходит в конкретной палате. Необходимо создать базу данных для хранения выше представленной информации.

Ограничения предметной области, которые необходимо учитывать при проектировании.

* Врачи бывают разных специальностей. По одной специальности может работать несколько докторов;
* Полис пациента имеет формат: пример (ЧБ1234Г);
* В одной палате может работать только 1 врач.
* Один пациент в один день может находиться только в одной палате во время пребывания в больнице.
* Пациент может иметь аллергию на несколько лекарств.
* Одному пациенту могут быть назначены несколько лекарств.

Исходное отношение (таблица) находится в первой нормальной формы (1NF), так как удовлетворяет условию атомарности атрибутов отношения.

Отношение находится во второй нормальной форме (2NF) тогда и только тогда, когда отношение находится в первой нормальной форме и каждый его не ключевой атрибут полностью зависит от первичного ключа или, что тоже справедливо, отношение, находящееся во второй нормальной форме, не содержит атрибутов, зависящих от части ключа.

Данное условие второй нормальной форме (2NF) не выполняется, так как, например, от фамилии пациента зависит его паспорт, страховой полис, аллергия.

Для приведения исходного отношения к 2NF необходимо провести разбиение (декомпозиции) на несколько отношений, удовлетворяющих соответствующим требованиям нормализации. Произведём декомпозицию исходного отношения на четыре отношения: «Пациент», «Врач», «Лекарство» и «Лечение»:

ВРАЧ (Код врача, ФИО, Специализация).

ПАЦИЕНТ (Код пациента, ФИО, Паспорт, Полис).

ЛЕКАРСТВО (Код лекарства, Наименование, Цена)

ЛЕЧЕНИЕ (Код лечения, Код пациента, Код врача, Жалобы, Диагноз, Дата поступления, Дата выписки, Палата, Отделение)

Ключом отношения «Врач» является Код врач

Ключом отношения «Пациент» является Код пациента.

Ключом отношения «Лекарство» является Код лекарства.

Ключом отношения «Лечение» является ключ Код лечения.

После декомпозиции все четыре отношения находятся в третьей нормальной форме (3NF), так как каждый не ключевой атрибут не транзитивно зависит от первичного ключа, т. е. среди атрибутов отношения нет атрибутов, транзитивно зависящих от ключа (среди его не ключевых атрибутов нет зависящих от другого не ключевого атрибута).

Таким образом, после декомпозиции исходное отношение состоит из четырёх взаимосвязанных отношений (таблиц), которые находятся в третьей нормальной форме (3NF).

Состав описательных атрибутов таблиц будет определён при дальнейшем проектировании.

Для этого необходимо выполнить следующие действия:

1. построить модели данных задачи (концептуальную, логическую и физическую);
2. определить структуру таблиц нормативно-справочной и оперативно-учетной информации;
3. выделить ключевые поля и определить связи между таблицами;
4. реализовать базу данных в среде Access;
5. создать таблицы и схему данных;
6. создать форма для ввода данных в таблицы;
7. заполнить таблицы тестовыми данными;
8. разработать запросы;
9. создать отчёты;
10. разработать интерфейс.

Для ввода, редактирования и просмотра информации в базе данных должны быть разработаны экранные формы для всех информационных объектов.

При разработке ИС должны быть созданы запросы и отчёты для работы с информацией в базе данных, которые обеспечивают основные требования сотрудников поликлиники.

# 2 Моделирование предметной области «Поликлиника»

В результате анализа предметной области [2-5, 7-9] были выявить следующие сущности*:* «Пациент», «Врач», «Лекарство» и «Лечение»

При этом сущности «Врач», «Лекарство» хранят нормативно-справочную информацию, которая является условно-постоянной. Сущности Пациент и Лечение хранят оперативно-учётную информацию о текущей работе поликлиники.

Определим типы связей для концептуальной модели, существующие между выделенными нами сущностями [7-9]. Для этого снова анализируем требования к БД. Результат анализа представлен в таблице 2.1.

Таблица 2.1 - Типы связей между сущностями

|  |  |  |  |
| --- | --- | --- | --- |
| Главный ИО | Связь | Подчинённый ИО | Тип  связи |
| ПАЦИЕНТ  ПАЦИЕНТ  ВРАЧ  ЛЕЧЕНИЕ | лечится  имеет аллергию  лечит  принимаются | ЛЕЧЕНИЕ  ЛЕКАРСТВО  ЛЕЧЕНИЕ  ЛЕКАРСТВО | 1:М  М:М  1:М  М:М |

Для построения моделей данных воспользуемся возможностями CASE –средства PowerDesigner версии 16.6 [5].

Создадим новый проект, выполнив пункт главного меню *File–New Project*. (рисунок 2.1). Укажем имя проекта «Поликлиника» и путь к папке проекта.

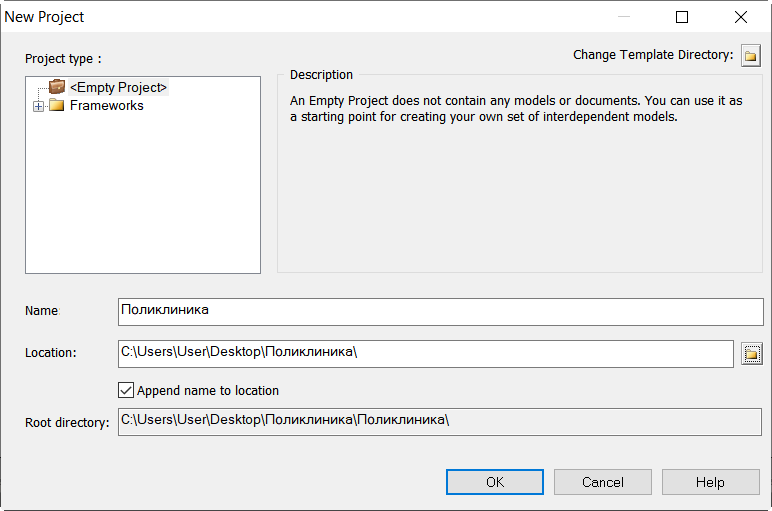


Рисунок 2.1. Новый проект

Сохраним проект, выполнив пункт меню *File–Save All*. Проект сохраняется в файле с расширением \*.prj.

Далее создадим концептуальную модель, выполнив пункт главного меню *File–New Model*. В открывшемся окне выберем значок *ConceptualData* и зададим имя модели «Поликлиника». Сохраним модель, выполнив пункт меню *File–Save All*. Концептуальная модель сохранится в файле с расширением \*.cdm.

Воспользуемся возможностями Палитры (Polette) и создадим сущности и связи между ними. В результате получим концептуальную модель для нашей БД, которая представлена на рисунке 2.2 в нотации Баркера.

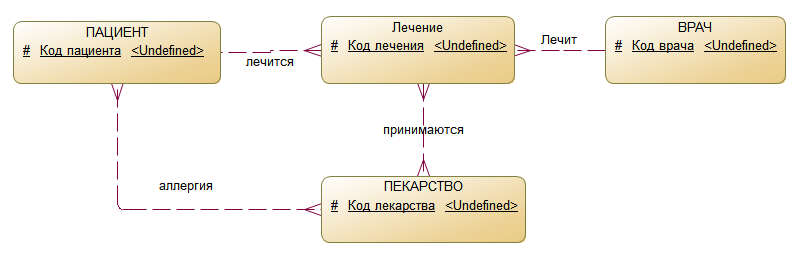


Рисунок 2.2. Концептуальная модель базы данных

На концептуальной схеме отсутствуют внешние ключи и не определены форматы данных атрибутов. Для преобразования концептуальной модели в логическую выполним команду Tools / Cenerate Loqical Data Model [5]. В результате получим логическую модель данных, которая содержит только первичные и вторичные ключи (рисунок 2.3).

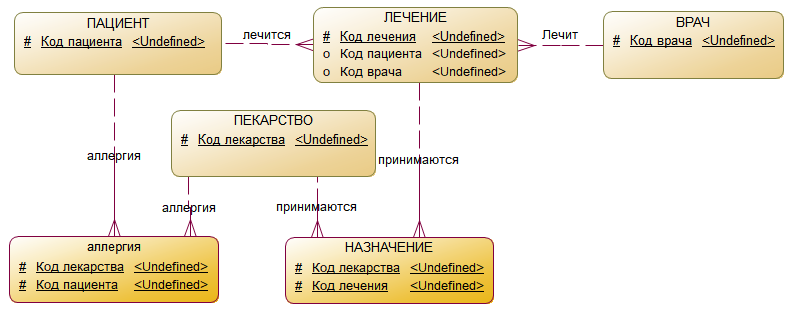


Рисунок 2.3. Логическая модель базы данных

Добавим логическую модель остальными атрибутами сущностей и зададим для них тип данных.

Логическая модель базы данных в PowerDesigner со всеми атрибутами приведена на рисунке 2.4.

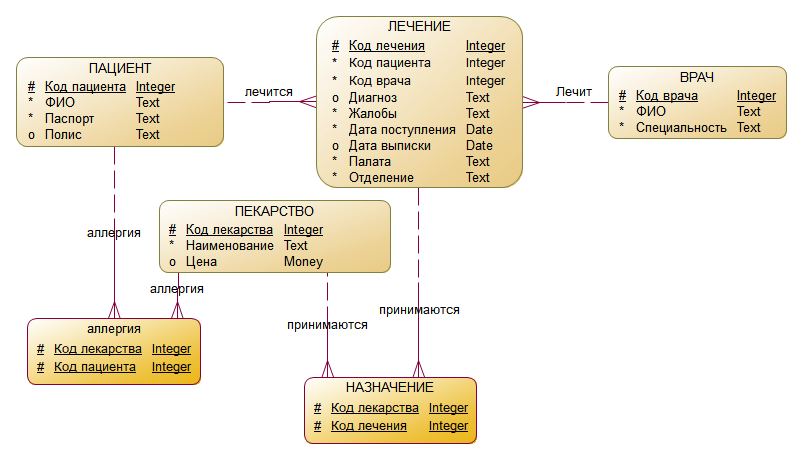


Рисунок 2.4. Логическая модель базы данных со всеми атрибутами

Логическая модель удовлетворяет требованиям 3-ей нормальной форме, так как каждый неключевой атрибут нетранзитивно зависит от первичного ключа, т. е. среди атрибутов отношения нет атрибутов, транзитивно зависящих от ключа (среди его неключевых атрибутов нет зависящих от другого неключевого атрибута).

Для преобразования логической модели в физическую выполним команду Tools / Cenerate Physical Data Model [5]. В результате получим физическую модель данных в среде PowerDesigner (рисунок 2.5).

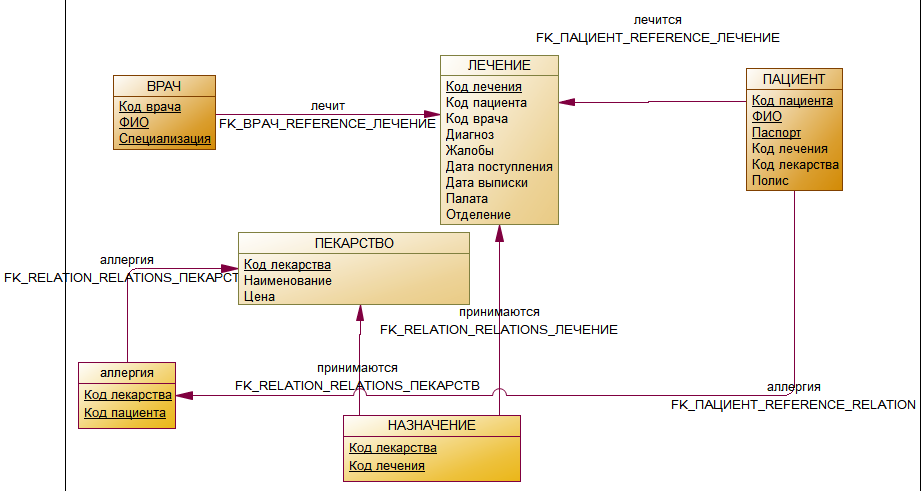


Рисунок 2.5. Физическая модель базы данных

Физическая модель базы данных в дальнеёшем будет рееализована в среде выбранной СУБД.

# 3 Выбор СУБД

Выбор СУБД является одним из важнейших моментов в разработке проекта БД, так как он принципиальным образом влияет на весь процесс проектирования БД и реализации информационной системы.

Теоретически при осуществлении этого выбора нужно принимать во внимание десятки факторов. Но на практике разработчики руководствуются лишь собственной интуицией и несколькими наиболее важными критериями, к которым, в частности, относятся:

* тип модели данных, которую поддерживает данная СУБД: реляционная;
* запас функциональных возможностей для дальнейшего развития информационной системы: из возможностей системы буду использовать только хранение данных в таблицах, выборку по запросам и индексирование;
* степень оснащенности СУБД инструментарием для персонала администрирования данными: система оснащена визуальной офисной системой управления данными, а также внутренним «провайдером» для доступа из сторонних приложений;
* удобство и надежность СУБД в эксплуатации: система имеет возможность создания резервных копий;
* стоимость СУБД и дополнительного программного обеспечения.

Доя малых и средних организаций при выборе СУБД де факто стандартом является MS Access.

MS Access – это система управления базами данных под управлением Windows. MS Access входит в пакет Microsoft Office [1-6].

СУБД Access является системой управления базами данных реляционного типа. Данные хранятся в такой базе в виде таблиц, строки (записи) которых состоят из наборов полей определенных типов. С каждой таблицей могут быть связаны индексы (ключи), задающие нужные пользователю порядки на множестве строк. Таблицы могут иметь однотипные поля (столбцы), и это позволяет устанавливать между ними связи, выполнять операции реляционной алгебры. Типичными операциями над базами данных являются определение, создание и удаление таблиц, модификация определений (структур, схем) существующих таблиц, поиск данных в таблицах по определенным критериям (выполнение запросов), создание отчетов о содержимом базы данных [8].

Основные объекты Microsoft Access.

Таблица – объект, который определяется и используется для хранения данных. Для каждой таблицы можно определить *первичный ключ* и один или несколько *индексов,* помогающих ускорить доступ к данным.

Запрос– объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц.

Форма – объект, предназначенный в основном для ввода данных, отображения их на экране или управления работой приложения {7]. С помощью формы можно в ответ на некоторое событие  запустить *макрос* или *процедуру* VBA.

Отчет – объект, предназначенный для создания документа, который впоследствии может быть распечатан или включен в документ другого приложения.

Макрос – объект, представляющий собой структурированное описание одного или нескольких действий, которые должен выполнить Access в ответ на определенное событие.

Модуль – объект, содержащий программы, написанные на языке Visual Basic для приложений.

Microsoft Access является настольной СУБД реляционного типа. Преимуществом Access является то, что она имеет очень простой графический интерфейс, который позволяет не только создавать собственную базу данных, но и разрабатывать приложения, используя встроенные средства [10].

В Microsoft Access предусмотрены все необходимые средства для определения и обработки данных, а также для управления с большими объемами информации. Microsoft Access спроектирован так, что его можно использовать как самостоятельную СУБД на отдельном персональном компьютере, так и в сети в режиме клиент-сервер, следовательно, данная СУБД имеет надежные средства защиты информации [2,3].

Основные достоинства СУБД Access:

* хранение данных в виде таблиц.
* легко связывать данные, хранящиеся в разных таблицах.
* дает возможность работать с другими БД, такими как Paradox, FoxPro.
* позволяет устанавливать связь с другими программами из пакета Microsoft Office.
* возможность интеграции с Visual Basic Application или Delphi.
* для выполнения почти всех основных операций Access предлагает большое количество Мастеров (Wizards), которые делают основную работу за пользователя при работе с данными и разработке приложений.

Итак, в качестве среды разработки целесообразно выбрать MS Access, так как сотрудники поликлиники имеют навыки работы с офисными программами. Кроме того, возможности СУБД MS Access, по объёму хранимой информации и быстродействию, более чем удовлетворяют требованиям разрабатываемой базы данных.

# 4 Разработка базы данных «Поликлиника»

## 4.1 Создание таблиц и схемы данных

Рассмотрим реализацию модели базы данных «ПОЛИКЛИНИКА» в среде MS Access.

Для создания таблиц и схемы данных БД «ПОЛИКЛИНИКА» вновь воспользуемся возможностями PowerDesigner [5]. Для этого выполним генерацию физической модели базы данных в СУБД Access с помощью команды меню «DataBase» – «Generate DataBase». Затем откроем в среде MS Access созданную БД и проверим её соответствие физической модели.

Базы данных в соответствии с разработанной моделью данных, состоит из шести таблиц.

Таблица Врач в режиме Конструктор представлена на рисунке 4.1.

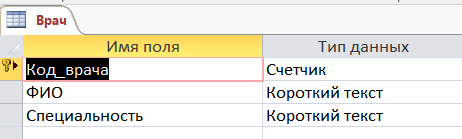


Рисунок 4.1. Таблица Врач в режиме Конструктор

4.

Эта же таблица с введёнными данными в режиме Таблица приведена на 4.2.

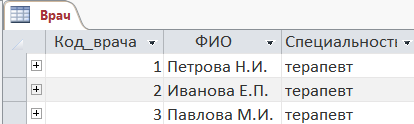


Рисунок 4.2. Таблица Врач в режиме Таблица

Ниже приведена таблица Лечение в режиме Конструктор (рисунок 4.3).

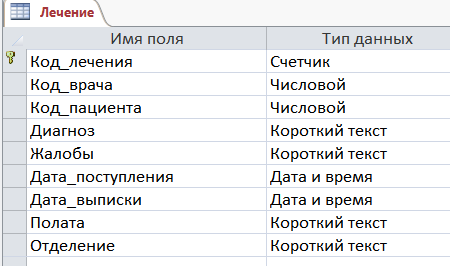


Рисунок 4.3. Таблица Лечение в режиме Конструктор

Для ввода Кода\_врача, , Кода\_пациента с помощью Мастера подстановок значения берутся из главных таблиц. Для ввода дат используется маска ввода краткой даты.

Данная таблица с введёнными данными в режиме Таблица приведена на 4.4.

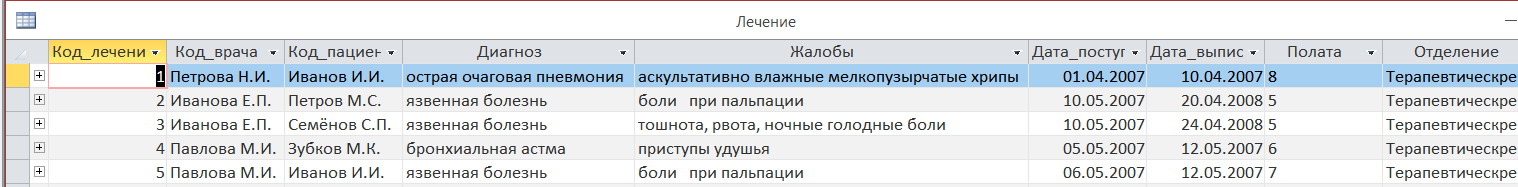


Рисунок 4.4. Таблица Лечение в режиме Таблица

В соответствии с моделью данных созданы остальные 4 таблицы: с атрибутами сущностей, определёнными ранее.

Выполнив команду Работа с базой данных – Схема данных, можно вывести Схему данных в Access (рисунок 4.5).

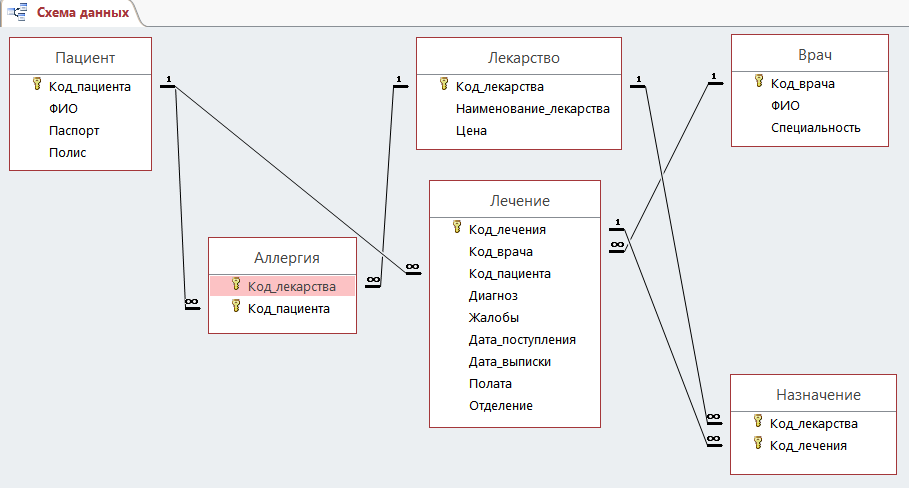


Рисунок 4.5. Схема базы данных

Схема данных имеет три уровня.

Заполним все таблицы данными, вводя их для обеспечения целостности сначала в главные таблицы, а потом в подчинённые. Ввод можно выполнять как в режиме ТАБЛИЦА, так и использую соответствующие формы ввода данных, создание которых рассмотрено в следующем подразделе.

## 4.2 Создание форм

Для удобства ввода данных в таблицы можно для каждой из них создать форму тремя различными способами [6-11]:

* при помощи Автоформына основе таблицы или запроса,
* при помощи Мастерана основе одной\нескольких таблиц или запросов,
* вручную в режиме Конструктор. Сначала создается базовая форма, которая затем изменяется в соответствии с требованиями в режиме конструктора.

Удобнее всего сначала создать базовую форму в Мастере, которая затем может быть изменена в соответствии с требованиями в режиме Конструктор.

По внешнему виду формы, создаваемые в Мастере, бывают: в один столбец, ленточные, табличные и выровненные.

Перейдём на вкладку Создание и нажмём кнопку Мастер форм. В появившемся окне выберем таблицу Врач и все доступные поля (рисунок 4.6) и нажмём Далее.

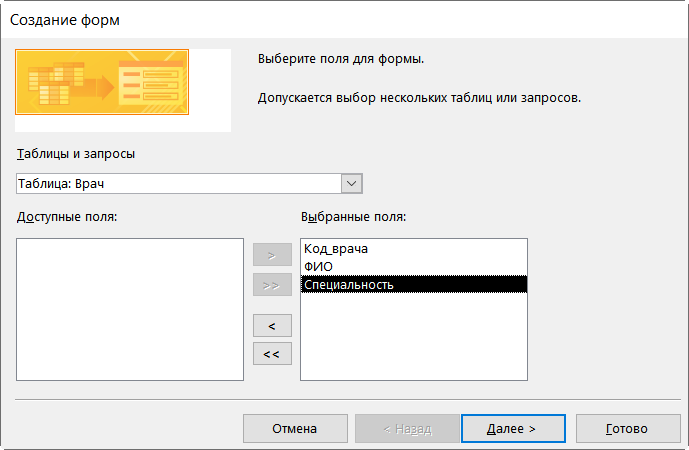


Рисунок 4.6. Диалоговое окно Создание форм

В следующем окне выберем внешний вид формы В один столбец и нажмём Готово. В результате получим форму Врач (рисунок. 4.7), в которую значительно удобнее вводить данные, чем в таблицу.

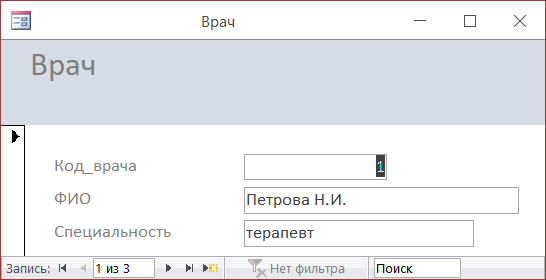


Рисунок 4.7. Форма Врач

При необходимости в режиме Конструктор форм её можно откорректировать [2-4].

На рисунке 4.8 приведена Ленточная форма Пациент.

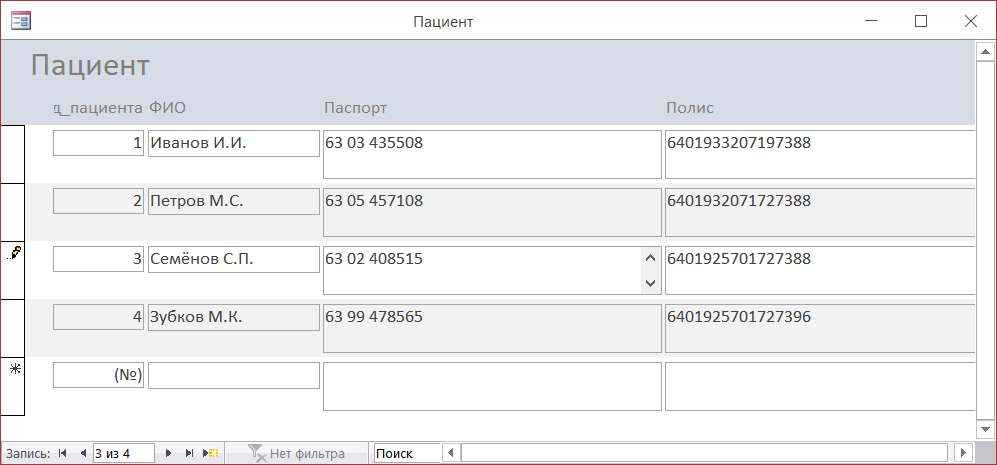


Рисунок 4.8. Ленточная форма Пациент.

Создадим форму в другом стиле оформления: на рисунке 4.9 представлена табличная форма Лекарство.

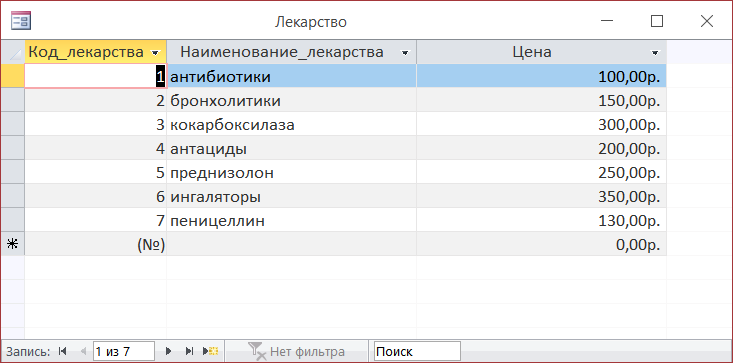


Рисунок 4.9. Табличная форма Лекарство

Все созданные формы могут быть отнесены к простым формам, которые созданы на основании одной таблицы. Однако более удобно работать со сложно-подчинёнными формами, которые созданы на основании главной и подчинённой таблицей. Пример такой сложной формы Пациент и Лечение представлен на рисунке 4.10

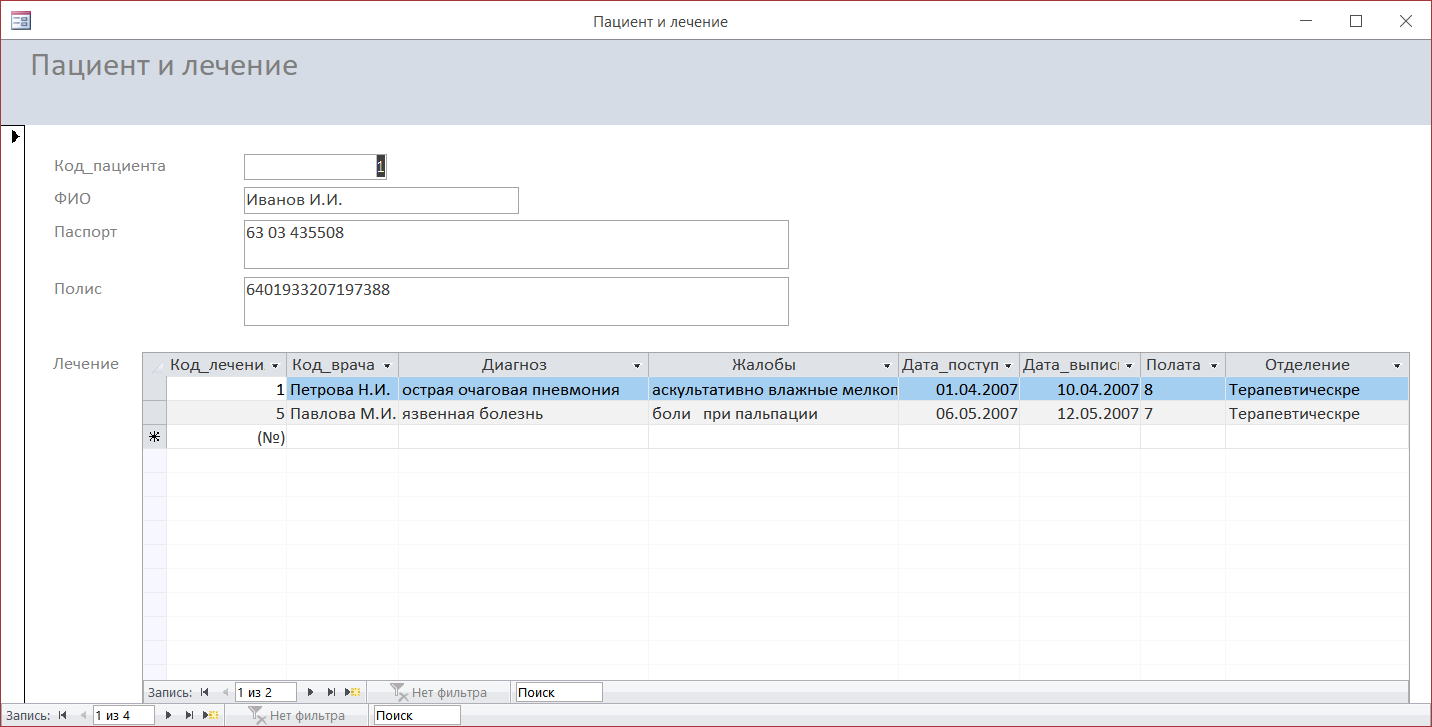


Рисунок 4.10. Сложноподчинённая форма Пациент и Лечение

Аналогичным образом создадим формы для всех остальных таблиц базы данных.

Использую созданные формы, заполним данными все таблицы. В качестве примера таблица Пациент с введёнными данными представлена на рисунке 4.11.

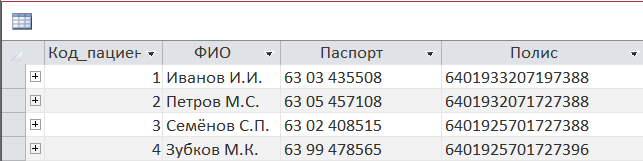


Рисунок 4.11. Таблица Пациент с данными в режиме Таблица

Данные по остальным таблицам можно просмотреть в соответствующих таблицах или формах. База данных готова для разработки на её основе информационной системы.

## 4.3 Разработка запросов

**Запросы** – это объекты базы данных, предназначенные для просмотра, анализа, ввода и изменения информации в БД [1, 2, 6-10]. Кроме того, они используются в качестве источника данных для форм и отчетов.

Запросы бывают следующих типов:

* запросы на выборку,
* запросы с вычислениями,
* запросы с параметрами,
* запросы с групповыми вычислениями (итоговые),
* перекрёстные запросы,
* запросы действия (удаления, добавления, обновления и создания таблиц).

В СУБД MS Access существует три способа построения запросов: с помощью Мастера запросов, с помощью Конструктора запросов, в режиме SQL.

Удобнее всего воспользоваться Конструктором запросов, который находится на вкладке Создание [2-4]. Любой запрос, создаваемый с помощью конструктора, по умолчанию, является запросом на выборку. В появившемся диалоговое окно надо выбрать таблицы для добавления.

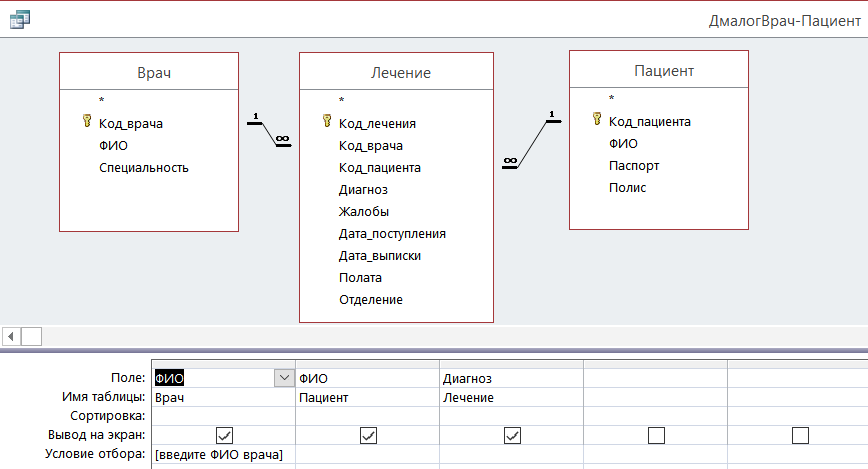
Кроме того, задаём остальные параметры в бланке запроса. Для запроса о Враче и пациенте в диалоге, вид запроса представлен на рисунке 4.12

Рисунок 4.12. Диалоговый запрос о Враче и пациенте

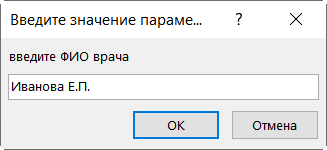
Этот же запрос в виде SQL-предложения.

SELECT Врач.ФИО, Пациент.ФИО, Лечение.Диагноз

FROM Пациент INNER JOIN (Врач INNER JOIN Лечение ON Врач.Код\_врача = Лечение.Код\_врача) ON Пациент.Код\_пациента = Лечение.Код\_пациента

WHERE (((Врач.ФИО)=[введите ФИО врача]));

При его запуске появляется диалоговое окно, в которое необходимо ввести ФИО врача. Диалоговое окно и результат выполнения запроса о Враче Иванова Е.П. и пациенте представлен на рисунке 4.13.



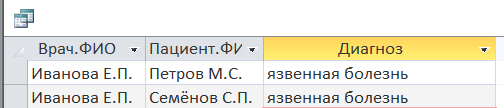


Рисунок 4.13. Запрос о враче Иванова Е.П.

Разработаем Итоговый запрос о количестве лекарств, назначенных каждому пациенту врачом (рисунок 4.14).

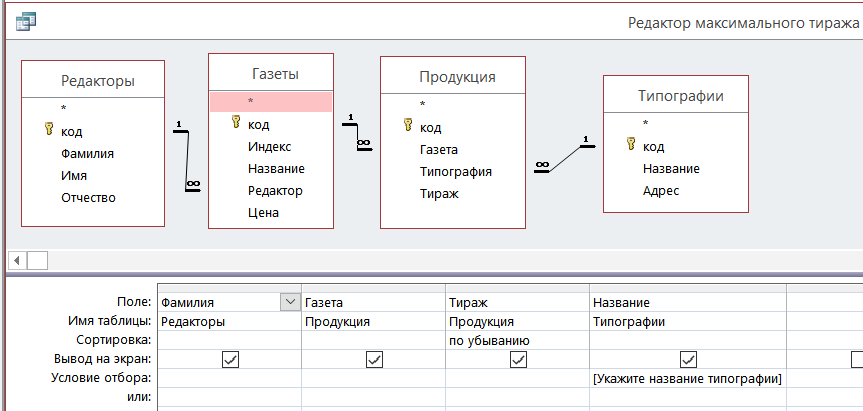
. 

Рисунок 4.14. Итоговый запрос о количестве лекарств, назначенных каждому пациенту в режиме Конструктор

Этот же запрос в виде SQL-предложения.

SELECT Пациент.ФИО, Count(Назначение.Код\_лекарства) AS [Count-Код\_лекарства]

FROM Пациент INNER JOIN (Лечение INNER JOIN Назначение ON Лечение.Код\_лечения = Назначение.Код\_лечения) ON Пациент.Код\_пациента = Лечение.Код\_пациента

GROUP BY Пациент.ФИО;

Результат его выполнения показан на рисунке 4.15

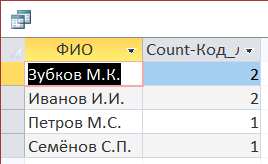


Рисунок 4.15. Результат выполнения

Создадим запрос - Вывести палаты, в которых лежат пациенты не более количества дней, заданного пользователем. (рисунок4.16)

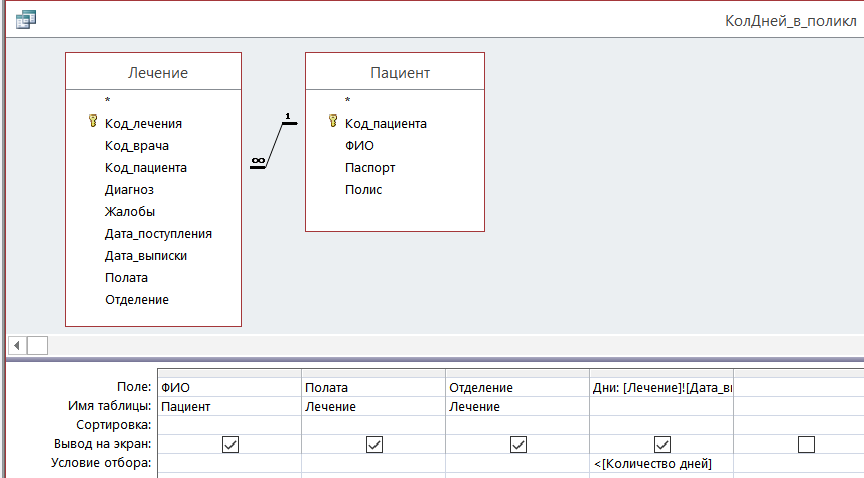


Рисунок 4.16. Запрос в режиме Конструктор

Этот же запрос в виде SQL-предложения.

SELECT Пациент.ФИО, Лечение.Полата, Лечение.Отделение, [Лечение]![Дата\_выписки]-[Лечение]![Дата\_поступления] AS Дни

FROM Пациент INNER JOIN Лечение ON Пациент.Код\_пациента = Лечение.Код\_пациента

WHERE ((([Лечение]![Дата\_выписки]-[Лечение]![Дата\_поступления])<[Количество дней]));

Результат его выполнения для количества дней 10 (рисунок 4.17).

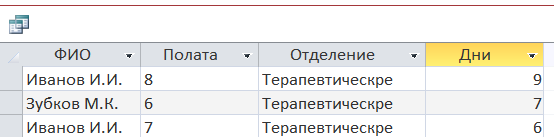


Рисунок 4.17. Результат выполнения запроса

Вывести пациентов, которые обращаются к врачам, чьи должности заданы пользователем (Рисунок 4.18)

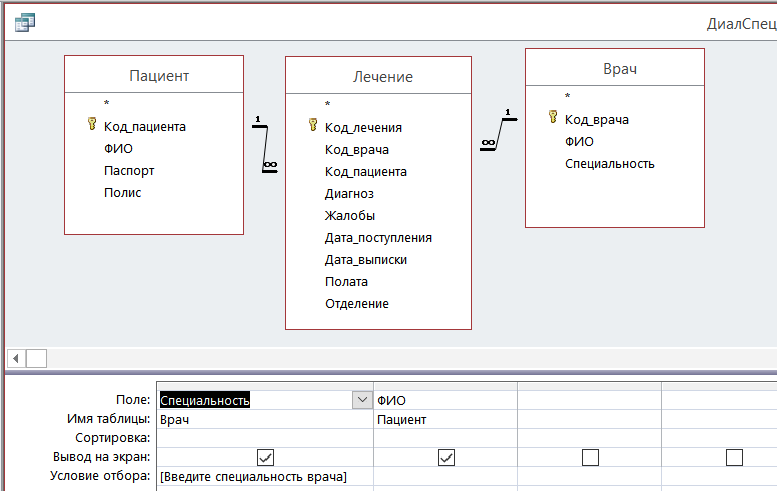


Рисунок 4.18. Запрос в режиме Конструктор

Этот же запрос в виде SQL-предложения.

SELECT TOP 10 Врач.Специальность, Пациент.ФИО

FROM Пациент INNER JOIN (Врач INNER JOIN Лечение ON Врач.Код\_врача = Лечение.Код\_врача) ON Пациент.Код\_пациента = Лечение.Код\_пациента

WHERE (((Врач.Специальность)=[Введите специальность врача]));

Результат его выполнения для специальности терапевт показан на рисунке 4.19.

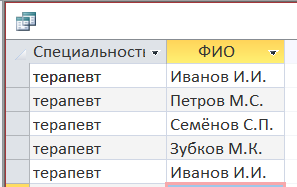


Рисунок 4.19. Результат выполнения запроса

Запрос - Вывести среднее количество пациентов по каждому врачу с жалобой, содержащей ключевое слово, заданное пользователем. (рисунок 4.20)

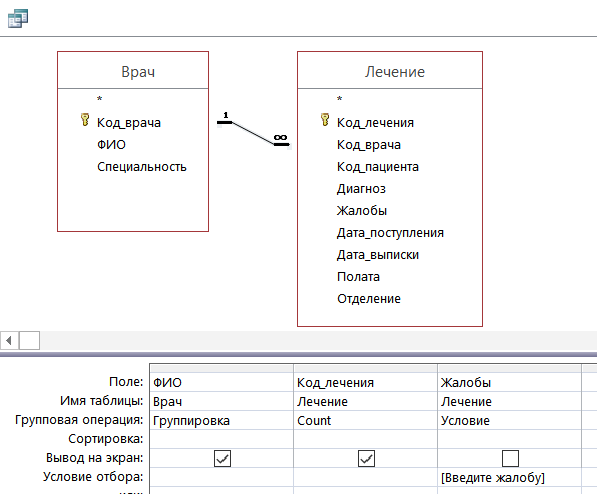


Рисунок 4.20. Запрос в режиме Конструктор

Этот же запрос в виде SQL-предложения.

SELECT Врач.ФИО, Count(Лечение.Код\_лечения) AS [Count-Код\_лечения]

FROM Врач INNER JOIN Лечение ON Врач.Код\_врача = Лечение.Код\_врача

WHERE (((Лечение.Жалобы)=[Введите жалобу]))

GROUP BY Врач.ФИО;

Результат его выполнения для жалобы Боли при пальпации показан на рисунке 4.21.

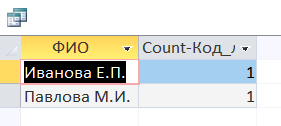


Рисунок 4.21. Результат выполнения запроса

Также разработан диалоговый запрос Больничный лист для формирования соответствующего отчёта.

Разработанные запросы могут использоваться непосредственно, либо на их основе можно создать отчёты для вывода информации на печать.

## 4.4 Создание отчётов

Отчет — это гибкое и эффективное средство для организации данных при выводе на печать [1,2, 10, 11].

Основные данные для формирования отчета берутся из базовой таблицы, запроса или инструкции SQL, являющихся источниками данных для отчета. Другие сведения (заголовки, примечания отчетов, количество страниц и другая сопроводительная информация) вводятся при разработке отчета.

Пользователь имеет возможность разработать отчет самостоятельно или создать отчет с помощью мастера. Мастер по разработке отчетов MS Access выполняет всю рутинную работу и позволяет быстро разработать макет отчета. После создания основной части отчета разработчик может переключиться в режим Конструктор и внести изменения в стандартный макет.

Рассмотрим создание отчёта о лекарствах. Создание отчётов выполним с помощью Мастера отчётов, который находится на вкладке Создание [2-4,10,11].

При нажатии кнопки Мастер отчётов появляется диалоговое окно, в котором выбираем источники (таблицы) и поля (рисунок 4.21).

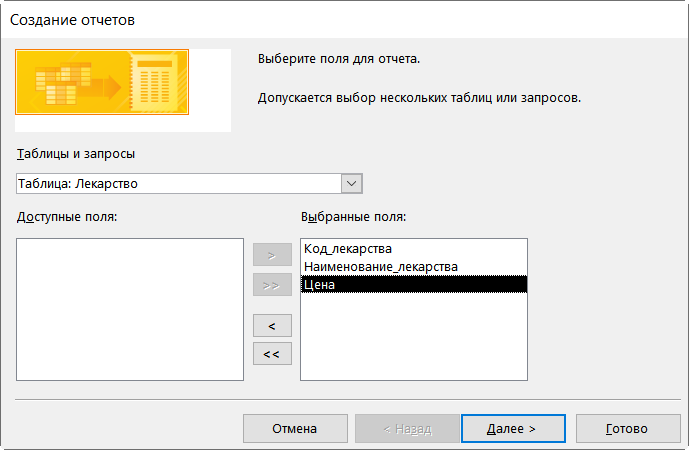


Рисунок 4.21. Диалоговое окно Мастера отчётов

Далее соглашаемся со структурой диалогового окна Создание отчёта (рисунок 4.22), нажимаем Далее.

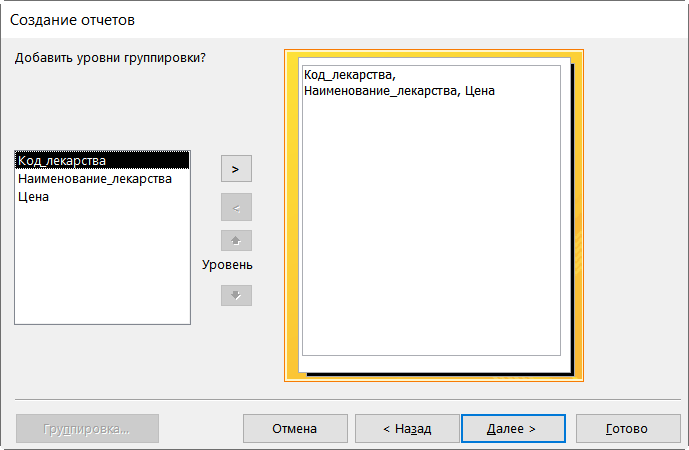


Рисунок 4.22. Диалоговое окно Создание отчёта Вид представления

Далее задаём сортировку в диалоговом окне Сортировка и альбомную ориентацию в окне Макет отчёта (рисунок 4.23).

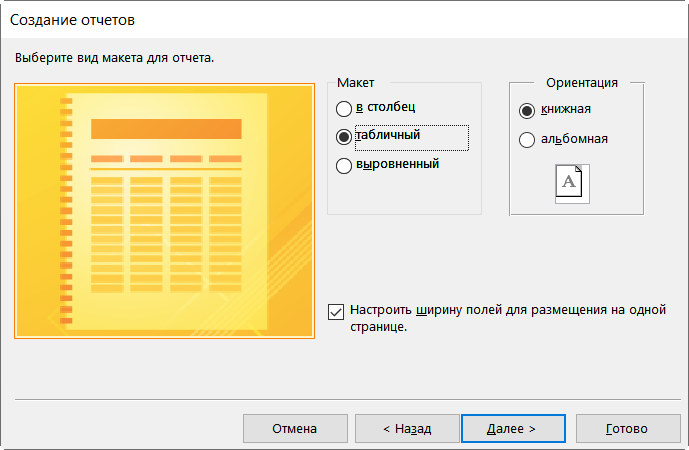


Рисунок 4.23. Диалоговое окно Макет отчёта

Щёлкаем Готово и отчёт создан. Далее открываем его в Конструкторе отчетов, где осуществляем корректировку.

Полученный отчёт представлен ниже на рисунке 4.24.

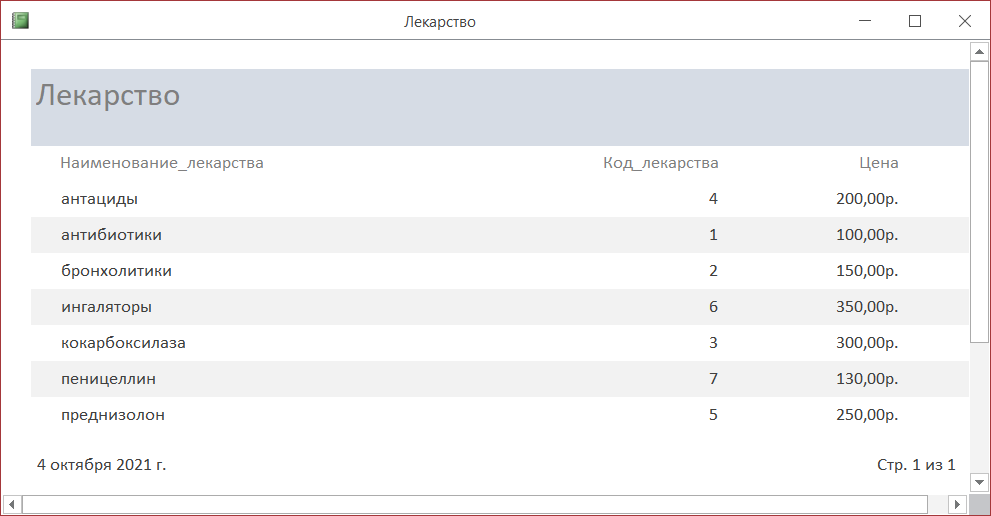


Рисунок 4.24. Отчёт о лекарствах

Разработаем сложноподчинённый отчёт Пациент и лечение помощью Мастера. В появившихся диалоговых окнах (рисунок 4.25 – 4.28) зададим необходимые параметры, последовательно нажимая Далее.

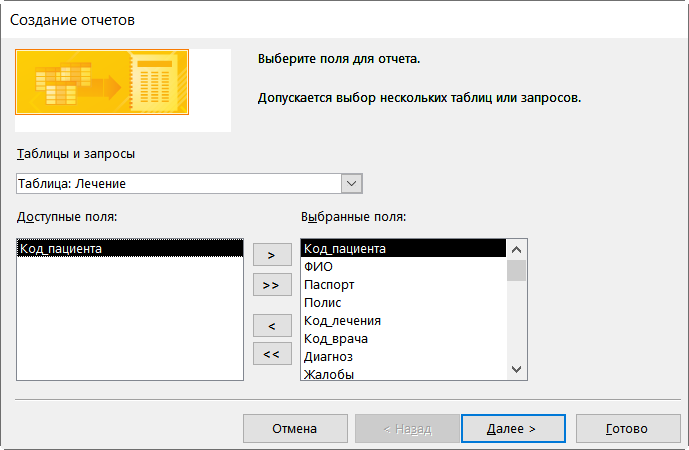


Рисунок 4.25. Диалоговое окно Создание отчёта Выбор полей

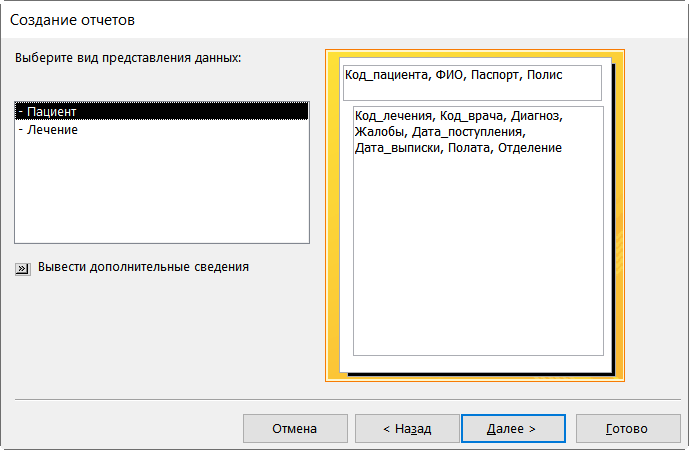


Рисунок 4.26. Диалоговое окно Создание отчёта Представление данных

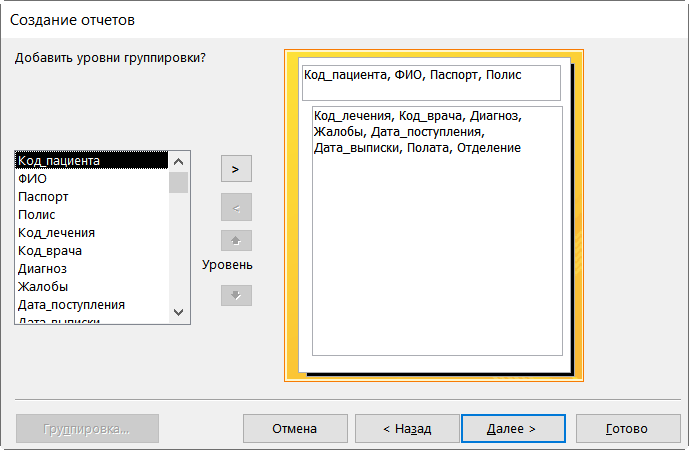


Рисунок 4.27. Диалоговое окно Создание отчёта Уровни группировки

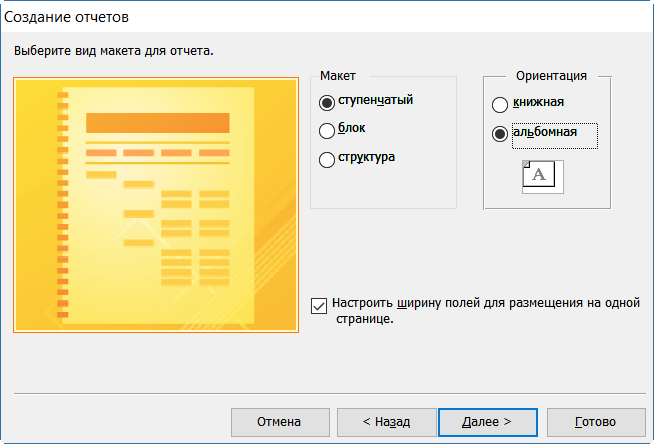


Рисунок 4.28. Диалоговое окно Создание отчёта Макет отчёта

В результате получим отчёт, который после корректировке в Конструкторе отчётов представлен на рисунке 4.29.

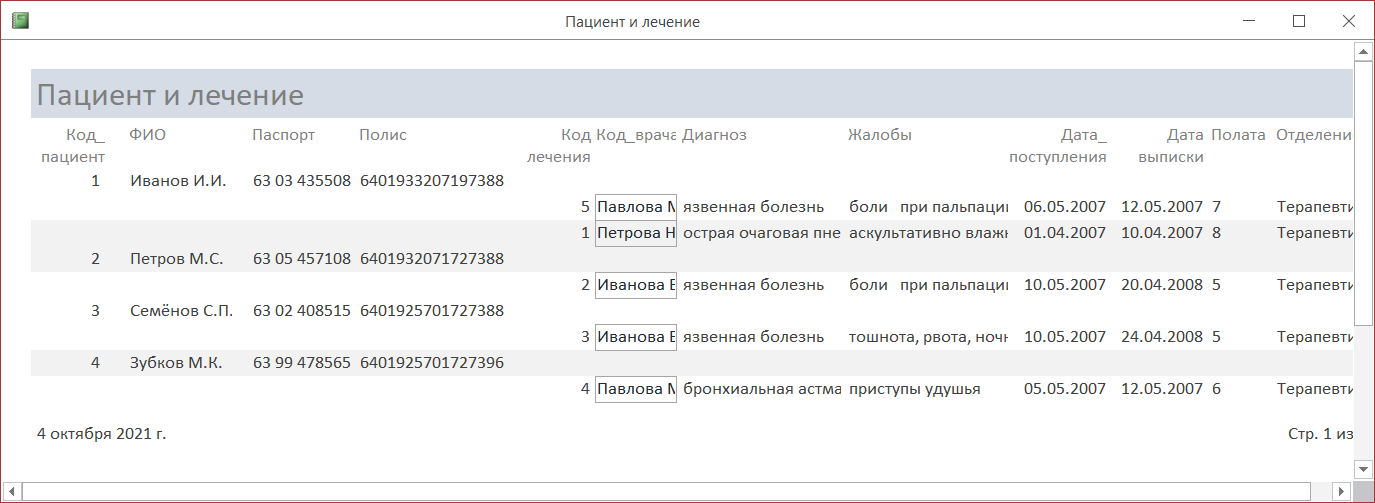


Рисунок 4.29. Сложноподчинённый отчёт Пациент и лечение

Разработаем отчёт о палатах, в которых лежат пациенты не более количества дней, заданного пользователем. Отчёт, созданный на основе запроса, по рассмотренной ранее технологии представлен на рисунке 4.30

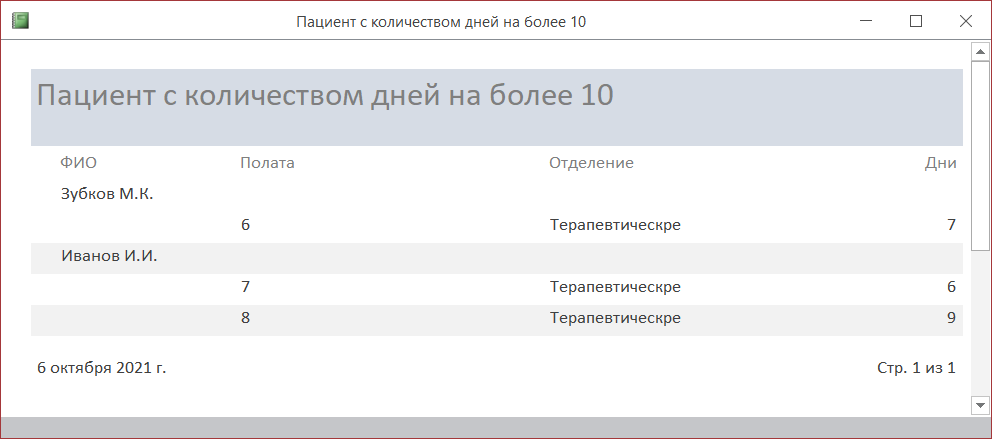


Рисунок 4.30. Отчёт о палатах, в которых лежат пациенты не более количества дней, заданного пользователем, в данном случае 10 дней.

Кроме того, по рассмотренной методике разработан отчёт о больничных листах пациентах, на основании диалогового запроса.

# 5 Разработка интерфейса базы данных

Для разработки кнопочного интерфейса базы данных воспользуемся возможностями среды Access [2-5]. Для этого в конструкторе форм создадим Главную кнопочную форму (ГКФ), на которую поместим кнопки открытия подчинённых кнопочных форм Ввод данных, Запросы и Отчёты, а также кнопку Выход из БД (рисунок 4.1).

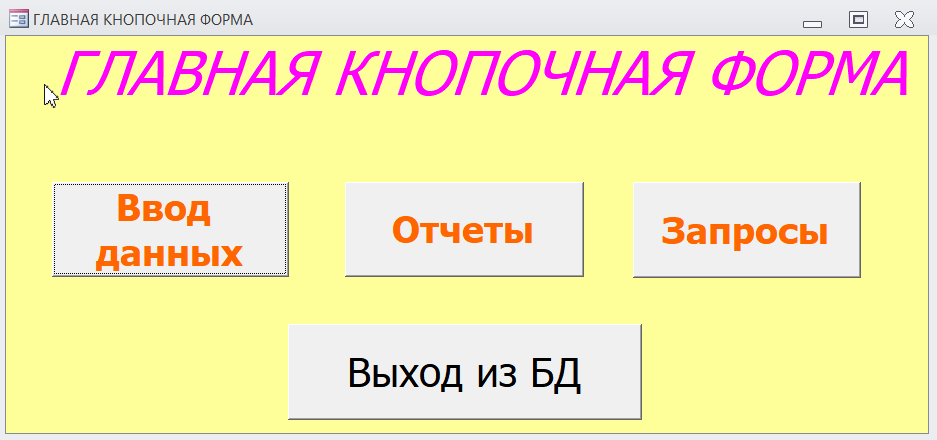


Рисунок 4.1. Главная кнопочная форма-

Далее создадим три подчинённые формы Ввод данных (рисунок 4.2), Запросы (рисунок 4.3) и Отчёты (рисунок 4.4), содержащие кнопки для открытия соответствующих объектов, которые были созданы ранее, и кнопку возврата в ГКФ. В качестве объектов (форм, запросов, отчетов) поместим на подчинённые формы основные из них, которые наиболее часто используются при работе библиотеки. Доступ к объектам, обращение к которым происходит достаточно редко, как правило, один раз в несколько лет, будет доступно только квалифицированным пользователям из окна базы данных.

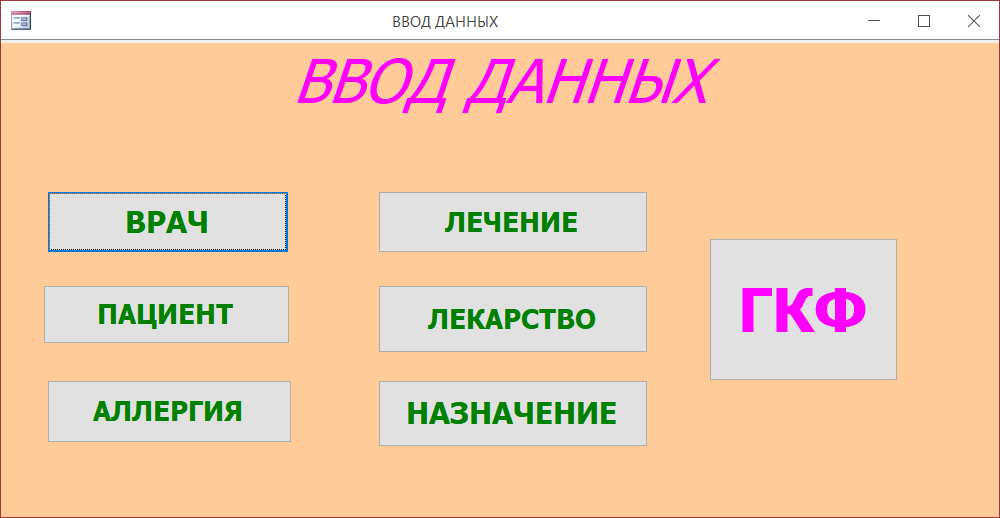


Рисунок 4.2. Форма-Ввод данных

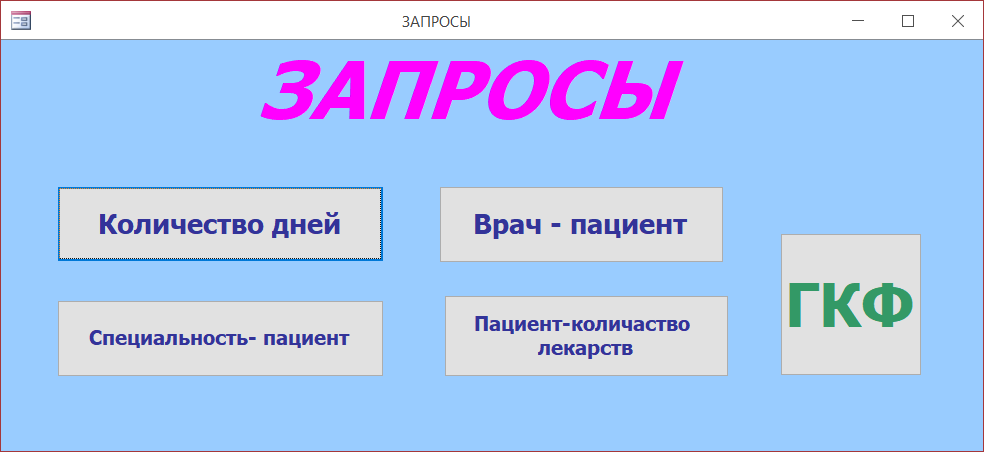


Рисунок 4.3. Форма Запросы

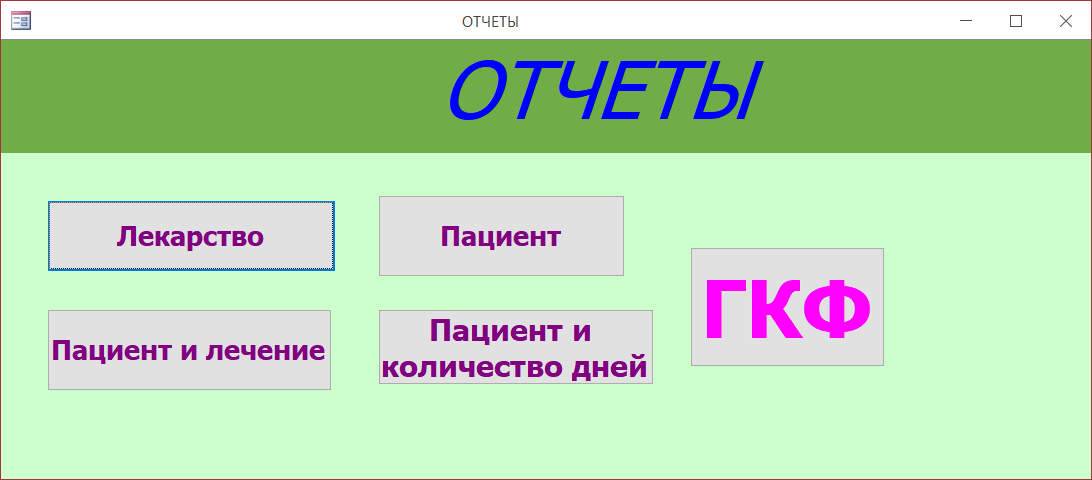


Рисунок 4.4. Форма Отчёты

Для задания реакции на однократное нажатие кнопка в свойствах всех кнопок зададим процедуру обработки события (рисунок 4.5) на языке VBA.

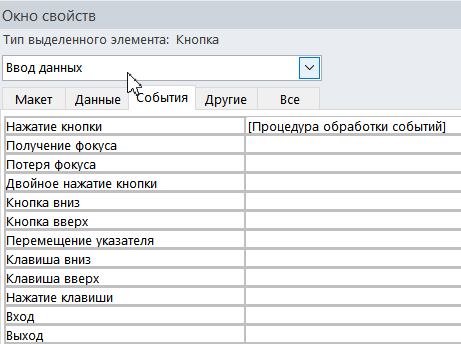


Рисунок 4.5. Окно свойств кнопки

На рисунке 4.6 приведена процедуру обработки события для кнопки Ввод данных, в результате выполнения которой открывается подчинённая форма Ввод данных.

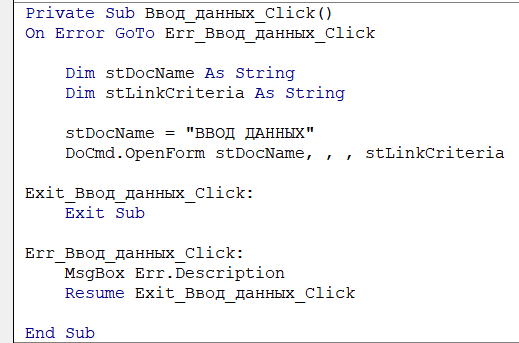


Рисунок 4.6. Процедура обработки события кнопки Ввод данных

Аналогичные процедуры обработки событий созданы для всех кнопок Главной формы (Приложение А) и для кнопок подчинённых форм: Ввод данных (Приложение Б), Запросы и Отчёты.

Созданный кнопочный интерфейс интуитивно понятен и позволяет пользователю любой квалификации легко в нём работать: вводить данные, а также получать выходную информацию в виде запросов и отчётов в соответствии с заданными критериями.

Для прямого доступа к ГКФ при открытии базы данных выполним команду Файл – Параметра и на вкладке Текущая база данных для опции Форма просмотра выберем - ГКФ. Для работы в обычном режиме удерживайте клавишу Shift при открытии базы данных.

**Заключение**

В ходе выполнения данной работы была создана база данных и АИС «Поликлиника» для автоматизации работы сотрудников поликлиники.

Разработка выполнялась в среде PowerDesigner и Microsoft Access.

Разработаны концептуальная, логическая и физическая модели данных.

Реализация базы данных выполнена в СУБД Microsoft Access.

На основе разработанной модели данных реализация базы данных выполнена в СУБД Microsoft Access.

В базе данных сведения из разных источников сохранены в отдельных таблицах.

Созданы шесть таблиц.

Для работы с данными из нескольких таблиц установлены связи между таблицами, которые отражаются в Схеме данных Access.

Для удобства работы с базой данных созданы формы для ввода данных.

Разработаны 7 запросов в режиме Конструктор и SQL.

Созданы 5 отчётов, которые откорректированы в Конструкторе отчётов.

Для удобства работы с базой данных разработан кнопочный интерфейс в среде Access с использованием VBA.

Разработанная база данных и АИС не претендует на полноту учёта всех особенностей предметной области. База данных представляет собой учебную модель, которая может быть использована в качестве основы для разработки реального проекта информационной системы «Поликлиника».

# Список использованных источников

1. Бекаревич Ю. Б. Самоучитель Access 2010 / Ю. Б. Бекаревич, Н. В. Пушкина. – СПб.: БХВ-Петербург, 2011. – 432 с.
2. Советов, Б.Я. Базы данных: теория и практика: учебник для бакалавров / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. - 2-е изд. - М.:Юрайт, 2012. - 464 с.
3. Дейт К. Дж. Введение в системы баз данных : пер. с англ. / К. Дж. Дейт. — 6-е изд. — Киев ; М. : Диалектика, 1998. — 784 с.: ил.
4. Саймон Алан Р. Стратегические технологии баз данных: менеджмент на 2000 год : пер. с англ. / Алан Р. Саймон ; ред. М. Р. Когаловский. — М. : Финансы и статистика, 1999. — 480 с.: ил.
5. Нартова А. Power Designer 15. Моделирование данных. М.: Издательство «Лори», 2012. – 480 с.
6. Кузин А. В. Базы данных: учебное пособие для студ. высш. учеб. Заведений.— 5-е изд., испр. — М. : Издательский центр «Академия», 2012
7. Харитонова, И. А. MS Office Access 2007 / И. А. Харитонова, Л. В. Рудикова. – Санкт-Петербург: БХВ, 2008. – 1280 с.
8. Сурядный А.С. Microsoft Access 2010. Лучший самоучитель. – 3-е изд., доп. и перераб. / А. С. Сурядный – М.: Астрель, Владимир. ВКТ. 2012. – 448 с.
9. Автоматизация проектирования баз данных в среде Sybase PowerDesigner: Методические указания к лабораторным работам. / Рязан. гос. радиотехн. универ.; Сост. А.В. Благодаров, Р.В. Тишкин, Рязань, 2010. 35 с. [Электронный ресурс] URL : <https://studfiles.net/preview/3073506/> (дата обращения: 12.04.2023).
10. Кузнецов С. Д. Основы современных баз данных [Электронный ресурс] // Информационно-аналитические материалы Центра Информационных технологий. — URL : http://citforum.ru/database/osbd/contents.shtml (да-та обращения: 12.04.2023).
11. Кириллов В. В. Основы проектирования реляционных баз данных: учеб. пособие [Электронный ресурс] / В. В. Кириллов. — URL: http://citforum.ru/database/dbguide/index.shtml (дата обращения: 12.04.2023).