

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт педагогики, психологии и социологии
Кафедра современных образовательных технологий

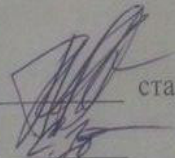
УТВЕРЖДАЮ
Заведующий кафедрой
И. А. Ковалевич
« 16 » 06 2016 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03.19 Прикладная информатика в социальных коммуникациях

**РАЗРАБОТКА БАЗЫ ДАННЫХ «КЛИЕНТЫ КГБУ СО
ПАНСИОНАТА ДЛЯ ГРАЖДАН ПОЖИЛОГО ВОЗРАСТА И
ИНВАЛИДОВ «СОЛНЕЧНЫЙ»»**

Научный руководитель


старший преподаватель В. А. Помазан

Выпускник

С. В. Заболотский

Красноярск 2016

20 июня 2016 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме: Разработка базы данных «Клиенты КГБУ СО Пансионата для граждан пожилого возраста и инвалидов «Солнечный»» содержит 59 страниц текстового документа, 21 источник литературы, 29 рисунков.

БАЗА ДАННЫХ, СУБД, MICROSOFT ACCESS, ПРОГРАММНЫЕ СРЕДСТВА СОЗДАНИЯ БАЗ БАННЫХ

Цель бакалаврской работы: разработка базы данных для учета клиентов КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»

Апробация БД: КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный» г. Красноярск.

Бакалаврская работа состоит из двух основных глав.

В первой части бакалаврской работы рассматриваются программные средства создания баз данных имеющих системы управления базами данных, структуру баз данных и основные понятия. Описываются основные особенности работы с СУБД Microsoft Access, термины и возможности программного средства.

Во второй части описан процесс создания базы данных по текущему учету клиентов. Описание предметной области и результатов апробации по текущему учету клиентов и медицинского инвентаря пансионата. Рассматриваются этапы создания таблиц, определение связей между ними, создание отчетов, запросов и форм, а также макросов управления.

Далее следует заключение, где содержатся общие выводы и положения по бакалаврской работе, список использованных источников.

СОДЕРЖАНИЕ

Введение.....	4
1 Програмные средства создания базы данных.....	8
1.1 Система управления базами данных	8
1.2 Система управления базами данных Microsoft Access	16
2 Создание Базы Данных «Клиенты КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»».....	26
2.1 Описание предметной области базы данных.....	26
2.2 Проектирование базы данных	29
2.3 Описание эталонов создания базы данных.....	40
2.4 Описание работы базы даных	54
2.5 Апробация базы данных	57
Заключение.....	58
Список использованных источников.....	59

ВВЕДЕНИЕ

В наши дни информационные технологии внедрены практически во все сферы человеческой деятельности. Использование ПК в качестве инструмента обработки повышает информационную культуру общества, способствует легкому переходу к информационному обществу, где информация является важнейшим ресурсом наравне с материальными, энергетическими, людскими и др. ресурсами [2]. От вовремя полученной, правильно обработанной и четко представленной информации часто зависит эффективность принимаемых решений и их результат. Система обслуживания различных сфер деятельности, комплексы для научных исследований, системы автоматизации проектирования и производства и т.д. Сейчас очень часто проводятся разработки и внедрение типовых информационно-вычислительных систем. Это обеспечивает создание и широкое использование систем обработки данных разнообразного назначения.

Социальная сфера не является исключением, внедрение информационных технологий в ее формировании играет большую роль. Любая социальная сфера нуждается в оперативном доступе к информации. Ценность информации на данный момент очень высока. Роль распорядителей информации чаще всего выполняют базы данных. Базы данных обеспечивают надежное хранение информации, в структурированном виде и своевременный доступ к ней. Практически любая современная организация нуждается в базе данных удовлетворяющей потребностям по хранению, управлению и администрированию данных [15].

Разработка БД выполняется для определенной предметной области, в нашем случае это КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный». С первых шагов проводится изучение предметной области и разрабатывается ее формализованное описание. На основании

полученных результатов, проводится даталогическое проектирование. Затем производится разработка и генерация форм, запросов и отчетов. База данных позволяет оперативно, своевременно удалять или вносить изменения в базу данных. Позволяет получить достоверную и полную информацию об искомом объекте [7].

Данная База Данных создается в первую очередь для руководства пансионата. Которым в свою очередь всегда необходимо быть в курсе поступления новых клиентов и наличия необходимого медицинского оборудования. Для этого нужна общая база данных, включающая всю необходимую информацию. Универсальность базы данных определена возможностью ее постоянного пополнения новой информацией, причем в большом количестве по мере поступления и выписки клиентов пансионата [12]. Таким образом, создание базы данных, обладающей такими свойствами является очень полезной и применение её в КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный» считается необходимым.

Современное развитие компьютерной техники помогает справляться с огромными объемами информации. Компьютерные технологии, такие как Система Управления Базами Данных (СУБД) позволяют перейти на совершенно новую технологию хранения, обмена и обработки информации. СУБД представляет собой комплекс языков, программ, а также лингвистических средств обеспечивающих создания, ведения и совместного использования БД многими пользователями [2]. Практически во всех социальных сферах используются СУБД, которые в свою очередь позволяют хранить и обрабатывать информацию.

СУБД Access востребована при автоматизации работы КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный». Программа является очень актуальной на сегодняшний день, она автоматизирует работу с базой данных и предоставляет администратору понятный интерфейс с помощью которого он может как посмотреть

интересующую его информацию, так и внести данные. Access это система учета информации, которая экономит очень много времени при ее использовании [7].

Главной целью успешного функционирования и развития КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный», безусловно, является повышение качества обслуживания клиентов. Современная технология тесно связана с вопросом автоматизации управления санаторием, внедрением ИТ-технологии влияет на повышение качества услуг. Не секрет, что автоматизация производства способствует существенному уменьшению затрат на труд, а результат становится более эффективным при использовании ресурсов пансионата [2]. Постоянно растущий спрос населения, требования органов контроля и управления здравоохранением к качеству медицинских услуг и обслуживания пациентов актуализирует разработку БД.

Задачи, которые необходимо решать при использовании любой БД:

- 1 повышение надежности и качества получаемых сведений;
- 2 снижение затрат на обработку и систематизацию информации;
- 3 увеличить эффективность обработки и информативность.

На примере нашей БД: это организация учета клиентов, предоставление по мере необходимости услуг по восстановлению и лечению, организация учета мягкого инвентаря и медицинского оборудования и т.д.

Необходимо также отметить такой эффект, как снижение временных затрат на принятие управленческого решения ввиду следующих факторов:

- 1 Сокращение большого числа бумажных потоков;
- 2 Повышение качества управленческих решений с позиции точности анализа управленческого явления, процесса или субъекта;
- 3 Наличие информирующей обратной связи.

Цель бакалаврской работы: разработать базу данных «Клиенты КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»».

Для достижения цели бакалаврской работы необходимо выполнить ряд задач:

- 1 Провести обзор СУБД;
- 2 Разработать модель базы данных «Клиенты КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»»;
- 3 Создать Базу данных «Клиенты КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»» в MS Access;
- 4 Описать алгоритм заполнения и работы Базы Данных.

Бакалаврская работа состоит из двух основных глав.

В первой части рассматриваются программные средства создания баз данных имеющих системы управления базами данных, структуру баз данных и основные понятия. Описываются основные особенности работы с СУБД Microsoft Access, термины и возможности программного средства.

Во второй части описан процесс создания базы данных по текущему учету клиентов. Описание предметной области результатов апробации базы данных по текущему учету клиентов . Рассматриваются этапы создания БД, а именно таблиц, определение связей между ними, создание отчетов, запросов и форм, а также макросов управления. В конце рассматривается работа с системой в режиме наполнения и просмотра.

Далее следует заключение, где содержатся общие выводы и положения по бакалаврской работе, список использованных источников.

1 Программные средства создания базы данных

1.1 Система управления базами данных

На сегодняшний день компьютер занял прочную позицию в нашей повседневной жизни. Современный человек на сегодняшний день не представляет себе ни одно предприятие без компьютеризации и автоматизации процесса. С помощью компьютера можно управлять большими потоками данных и создавать масштабные базы данных.

База данных — это упорядоченная совокупность данных, предназначенных для хранения, накопления и обработки с помощью ЭВМ, которая предполагает какой либо метод сохранения информации на диске и возможности доступа и манипуляции с нею, а так же набор программных продуктов, предоставляющий пользователю все допустимые средства работы с данными. Для создания и ведения базы данных (обновления, обеспечения доступа к ним по запросам и выдачи их пользователю) используется набор языковых и программных средств, называемых системой управления базы данных (СУБД). [2].

Можно выделить три основных типа:

- табличные (реляционные);
- сетевые;
- иерархические.

Наиболее распространенными в практике являются реляционные базы данных. Реляционная база данных представляет собой множество взаимосвязанных таблиц, каждая из которых содержит данные об объектах определенного типа. Каждая строка таблицы включает данные об одном объекте, а столбцы таблицы содержат различные характеристики этих объектов — атрибуты. Строки таблицы называются записями; все записи имеют одинаковую структуру — они состоят из полей, в которых хранятся

атрибуты объекта. Каждое поле записи содержит одну характеристику объекта и имеет строго определенный тип данных. Все записи имеют одни и те же поля, только в них содержатся разные значения атрибутов [2].

В реляционные базы данных удобны в том, что для получения ответов на различные запросы существует разработанный математический аппарат, который называется «реляционная алгебра» [2].

Иерархическая модель данных - совокупность элементов, связанных между собой по определенным правилам. Иерархическая модель данных строится по принципу иерархии типов объектов, то есть один тип объекта является главным, а остальные, находящиеся на низших уровнях иерархии - подчиненными. Между главным и подчиненными объектами устанавливается взаимосвязь «один ко многим», т.е. для главного типа объекта существует несколько подчиненных типов объектов. В то же время для каждого экземпляра главного объекта может быть несколько экземпляров подчиненных типов [6].

Узлы и ветви образуют иерархическую древовидную структуру. Наивысший в иерархии узел называется корневым (это главный тип объекта). Корневой узел находится на первом уровне. Зависимые узлы находятся на втором, третьем и др. уровнях [2].

Достоинства иерархической модели данных:

- эффективное использование памяти и удовлетворительные показатели временных затрат на выполнение операций;
- пригодны для формирования БД с теми данными, которые сами по себе имеют иерархическую структуру.

Соответственно возможностям организации реляционных, иерархических и сетевых информационных структур, существуют и аналогичные виды баз данных. В них данные представлены в формах, адекватных соответствующим структурам. Однако иерархические и сетевые базы данных гораздо менее распространены, чем реляционные и не могут

быть реализованы с помощью наиболее популярных СУБД, входящих в состав программного обеспечения ЭВМ [5].

Для работы с данными используются системы управления базами данных. Основные функции СУБД — это определение данных, обработка данных и управление данными [2].

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

Атрибут - это свойство объекта, его характеристика. Для баз данных атрибутом является поле – элемент данных в кортеже [1].

Сущность – это совокупность объектов, обладающих одинаковым набором свойств (атрибутов) [2].

Данные – это формализованное представление информации, на каком-либо носителе, доступное для обработки и обмена между людьми [3].

Система управления базами данных (СУБД) — представляет собой комплекс языковых и программных средств, которые обеспечивают управление созданием и использованием баз данных.

Основные функций СУБД:

- 1 управление данными во внешней памяти (на дисках);
- 2 управление данными в оперативной памяти с использованием дискового кэша;
- 3 журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
- 4 поддержка языков БД (язык определения данных, язык манипулирования данными) [2].

Компоненты СУБД:

- ядро, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию;

- процессор языка базы данных, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода;

- подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД, а также сервисные программы (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы [2].

Первоначально СУБД применялись только в крупных организациях с мощной аппаратной поддержкой, необходимой для работы с большими объемами данных [2].

Первые СУБД были очень большими и дорогими системами работающими только на мощных компьютерах. Это все следует из того, что для хранения большого объема данных нужны были компьютерные системы [2]. На сегодняшний день с разработкой современных компьютеров с большим объемом памяти все информация помещается в на одном диске, и этого достаточно чтобы работать с базой данных на компьютере [1].

Индекс - объект базы данных, создаваемый с целью повышения производительности поиска данных. Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и их поиск по заданному критерию путем последовательного просмотра таблицы строка за строкой может занимать много времени [7]. Индекс формируется из значений одного или нескольких столбцов таблицы и указателей на соответствующие строки таблицы и, таким образом, позволяет искать строки, удовлетворяющие критерию поиска. Ускорение работы с использованием индексов достигается в первую очередь за счёт того, что индекс имеет структуру, оптимизированную под поиск — например, сбалансированного дерева [3].

Некоторые СУБД расширяют возможности индексов введением возможности создания индексов по столбцам представлений или индексов по выражениям. Кроме того, индексы могут быть объявлены как уникальные и как не уникальные. Уникальный индекс реализует ограничение целостности на таблице, исключая возможность вставки повторяющихся значений [2].

Запросы - важнейший инструмент для извлечения информации из одной или нескольких таблиц БД. С помощью запроса можно вносить изменения в саму базу данных. Запрос может служить источником данных для форм, отчетов и страниц доступа к данным. Запрос позволяет решать многие задачи не прибегая к программированию [7].

По способу организации взаимодействия с базой данных через сеть, СУБД можно разделить на:

- СУБД с централизованной архитектурой;
- СУБД с архитектурой файл-сервер;
- СУБД с архитектурой клиент-сервер;
- СУБД с трехуровневой архитектурой: «тонкий клиент» - сервер приложений – сервер базы данных [2].

В СУБД с централизованной архитектурой, СУБД и сама база данных размещается и функционирует на центральном миникомпьютере, а пользователи получают доступ к базе данных при помощи обычных терминалов – компьютер рассматривается просто как устройство ввода и отображения информации: на мэйнфрейм передаются нажатия клавиш, в обратном направлении передаются данные, отображаемые непосредственно на мониторе пользователя [7].

В СУБД с архитектурой файл-сервер база данных хранится на сервере, а копии СУБД устанавливаются на компьютерах пользователей. Файл базы данных находящийся на сервере, совместно используется всеми пользователями одновременно, при помощи сетевого программного обеспечения и самой операционной системы. Ярким примером такой

архитектуры является СУБД MS Access: копии СУБД установлены на компьютере каждого пользователя, а сам файл базы данных находится на сервере в сетевой папке [8].

Архитектура файл-сервер позволяет добиться приемлемой производительности, т.к. в распоряжении каждой копии СУБД находятся все ресурсы компьютера пользователя. С другой стороны, производительность такой схемы для каждого пользователя, напрямую зависит от характеристик компьютера пользователя. Кроме того, такая схема работы значительно загружает сеть [17].

При архитектуре клиент-сервер база данных хранится на сервере, а СУБД подразделяется на две части: клиентскую и серверную. Клиентская часть СУБД выполняется на стороне клиента и обеспечивает интерактивное взаимодействие с пользователем и формирование запросов к базе данных (на языке SQL) [2].

Большинство современных СУБД реализованы по архитектуре клиент-сервер: Oracle, MS SQL Server, PostgreSQL, MySQL, Informix, DB2 и др.

Однако и архитектура клиент-сервер не лишена недостатков.

Если деловая логика взаимодействия с базой данных (логика, определяющая порядок работы предприятия: какие таблицы и в каком порядке заполнять, что делать при добавлении нового сотрудника и т.д.) изменяется, то приходится заново переписывать клиентские программы (водить новые формы, менять порядок их заполнения и т.д.) [19].

Если изменения происходят слишком часто, а количество рабочих мест велико, то постоянная переустановка программного обеспечения становится серьезной проблемой [7].

В таких случаях следует переходить к трехуровневой архитектуре: «тонкий клиент» - сервер приложений – сервер базы данных.

При трехуровневой архитектуре в функции клиентской части («тонкий клиент») входит только интерактивное взаимодействие с пользователем, а

вся деловая логика вынесена на сервер приложений, который собственно и обеспечивает формирование запросов к базе данных, передаваемых на выполнение серверу базы данных [9].

«Тонкий клиент» находится на компьютере пользователя и чаще всего представляет собой Web-браузер (например, Internet Explorer) с применением соответствующей HTML-странице апплетов Java или компонентов ActiveX.

Сервер приложений находится на сервере и может являться специализированной программой или обычным Web-сервером, вызывающим для обработки HTTP-запроса внешнюю программу через интерфейс CGI [2].

Система управления базами данных обладают как преимуществами по сравнению с файловыми системами, так и недостатками.

Преимущества:

- Контроль за избыточностью данных. Попытки освободить избыточное количество занимаемого места что бы не хранить несколько копий одного и того же элемента;
- Непротиворечивость данных. Сократить риск противоречивых состояний;
- Совместное использование данных. База данных принадлежит группам пользователей которым предоставлен доступ. Это упрощает обработку информации которой теперь может заниматься не один человек, а группа пользователей которые в свою очередь могут работать с большим количеством информации;
- Поддержка целостности данных. Корректность и непротиворечивость хранимой в базе данных информации;
- Повышенная безопасность. Защита данных от несанкционированного доступа со стороны;
- Применение стандартов. Интеграция позволяет администратору баз данных определять и применять необходимые стандарты;
- Повышение эффективности с ростом масштабов системы;

- Нахождения компромисса для противоречивых требований. У одних людей потребности могут не совпадать с потребностями других. База данных контролируется администратором который принимает взвешенное решение о наилучшем использовании базы данных в зависимости от определенной ситуации;

- Доступности данных и их готовности к работе. Данные в результате интеграции доступны конечным пользователям;

- Улучшение показателей производительности.

- Упрощение сопровождения системы за счет независимости от данных. В СУБД описания сведений отделены от приложений, а потому приложения защищены от изменений;

- Улучшенное управление параллельностью. Существует возможность параллельного доступа к данным. В ситуации когда несколько пользователей получают одновременный доступ к одному и тому же файлу;

- Развитая служба резервного копирования и восстановления. В современных СУБД предусмотрены средства от возникновения различных сбоев и несанкционированного доступа посторонних лиц к информации в базе данных[2].

Недостатки:

- Сложность. Функционал СУБД сопровождается ее значительным усложнением. Каждый разработчик СУБД должен понимать принцип работы системы.

- Размер. Занимает много места на диске и необходимо большой объем оперативной памяти;

- Стоимость СУБД. Цена за СУБД может варьироваться в очень широких пределах;

- Дополнительные затраты на аппаратное обеспечение. Для нормальной работы СУБД может потребоваться приобрести дополнительные устройства хранения информации;

- Затраты на преобразование приложений. Затраты на приобретение существующих приложений и обучение персонала;

- Более серьезные последствия при выходе системы из строя. Выход из строя одного из ее компонентов СУБД может привести к полному или частичному прекращению работы предприятия [10];

1.2 Система управления базами данных Microsoft Access

Сбор информации, ее обработка, создание форм для просмотра и распечатки данных - все эти функции обеспечивает система управления базами данных Microsoft Access.

Microsoft Access - это система управления базами данных (СУБД), которая позволяет не только хранить большие массивы данных в определенном формате, но и обрабатывать их, представляя в удобном для пользователей виде [2].

Все составляющие базы в Access хранятся в едином дисковом файле, который имеет расширение .mdb.

Основным структурным компонентом базы данных является таблица. В таблицах хранятся вводимые данные. Каждая таблица состоит из столбцов, называемых полями, и строк, называемых записями. Каждая запись таблицы содержит всю необходимую информацию об отдельном элементе базы данных.

При разработке структуры таблицы, прежде всего, необходимо задать поля, определив их свойства.

Таблицы - это основные объекты любой базы данных. Именно в них хранятся, во-первых, все данные, имеющиеся в базе, а, во-вторых, структура самой базы (поля, их типы и свойства) [2].

По внешнему виду таблица базы данных сходна с электронной таблицей, в которой данные располагаются в строках и столбцах. Поэтому

электронные таблицы обычно легко импортируются в таблицы базы данных. Основное различие между хранением данных в электронной таблице и в базе данных — способ организации данных [7].

Чтобы обеспечить наибольшую гибкость базы данных, необходимо распределить данные по таблицам так, чтобы избежать их избыточности. Например, если в базе хранятся сведения о сотрудниках, каждого из них следует один раз внести в таблицу, которая предназначена исключительно для хранения данных о сотрудниках. Данные о продуктах будут храниться в отдельной таблице, а данные о филиалах — в другой. Эта процедура называется нормализацией.

Каждую строку в таблице называют записью. Запись — это место хранения отдельного элемента информации. Каждая запись состоит из одного или нескольких полей. Поля соответствуют столбцам таблицы. Поля должны быть определены как конкретный тип данных: текст, дата или время, число или какой-либо иной тип [6].

Чтобы понять, что такое записи и поля, можно представить себе библиотечный каталог с карточками. Каждая карточка в ящике картотеки соответствует записи в базе данных. Каждый элемент сведений на отдельной карточке (автор, название и т. п.) соответствует полю в базе данных [4].

Запросы – это специальные структуры, предназначенные для обработки данных базы. С помощью запросов данные упорядочивают, фильтруют, отбирают, изменяют, объединяют, то есть обрабатывают.

СУБД Access позволяет создавать запросы трех типов:

- запросы выбора;
- перекрестные запросы;
- запросы действия.

Запрос выбора является наиболее часто используемым типом запроса. Он дает возможность:

- выбирать записи удовлетворяющие условиям отбора;

- включать в результирующую таблицу поля из одной или нескольких таблиц в нужном порядке;
- осуществлять вычисления над полями;
- выполнять статистические расчеты для групп записей.

Разновидностью запроса выбора является запрос с параметрами — это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести интересующее пользователя значение критерия отбора записей [7].

Перекрестный запрос представляет собой специальный запрос итогового типа. Он отображает результаты итоговых статистических расчетов над значениями некоторого поля в виде перекрестной таблицы. В ней значения одного или нескольких столбцов слева образуют заголовки строк, верхняя строка – заголовки столбцов из значений определенного поля, а на пересечении строк и столбцов – итоговые значения [2].

Запрос действия – это запрос, который вносит изменения в саму БД. Существует четыре типа запросов действия:

1 запрос на удаление - удаляет группу записей из одной таблицы или нескольких взаимосвязанных таблиц БД, для которых задано каскадное удаление связанных записей;

2 запрос на обновление - служит для изменения информации в полях таблицы БД;

3 запрос на добавление - производит добавление записей из таблицы с результатами запроса в таблицу БД;

4 запрос на создание таблицы - создает новую таблицу на основе всех или части данных из одной или нескольких таблиц БД. Этот запрос полезен в случае:

- создания таблицы для экспорта в другую БД;
- создания страниц доступа к данным, отображающих данные соответственно указанному моменту времени;

- создания резервной копии таблицы; создания архивной таблицы, содержащей старые записи [7].

СУБД Access позволяет создавать запросы с помощью Мастеров и с помощью Конструктора. Мастера используются для создания следующих запросов:

- простого запроса на выборку полей из источника запроса и подведение итогов;
- перекрестного запроса;
- запроса на поиск повторяющихся записей в таблице;
- запроса на поиск записей, не имеющих подчиненных им записей в другой таблице.

С помощью Конструктора можно создать любой запрос выбора, перекрестный запрос, запрос действия.

Формы – это объекты, с помощью которых в базу вводят новые данные или просматривают имеющиеся [7].

Формы иногда называются окнами ввода данных. Это интерфейсы, которые используются для работы с данными и часто содержат кнопки для выполнения различных команд. Базу данных можно создать без помощи форм, просто вводя в таблицу данные в режиме таблицы. Однако большинство пользователей баз данных предпочитают просматривать, вводить и редактировать данные таблиц при помощи форм [9].

Формы позволяют работать с данными в удобном формате; кроме того, в них можно добавлять функциональные элементы, например кнопки команд. Программным путем этим кнопкам можно назначить выполнение разнообразных задач, таких как определение данных, отображаемых в форме, или открытие других форм или отчетов[2]. Например, можно создать форму с именем «Форма клиента» для работы с данными клиента. В форме клиента может присутствовать кнопка, открывающая форму заказа, в которой создается новый заказ для данного клиента.

Формы также позволяют задавать условия работы других пользователей с информацией, содержащейся в базе данных. Например, можно создать форму со строго ограниченным набором отображаемых полей и разрешенных операций. Это помогает защитить данные и гарантировать правильность их ввода [13].

Отчеты – это формы «наоборот». С их помощью данные выдают на принтер в удобном и наглядном виде.

Отчеты служат для сбора и представления данных, содержащихся в таблицах. Обычно отчет позволяет ответить на определенный вопрос, например: «Сколько денег было получено от каждого клиента в этом году?», «В каких городах есть клиенты нашей компании?» Каждый отчет можно отформатировать так, чтобы представить сведения в наиболее удобном виде [2].

Отчет можно запустить в любое время, и он всегда будет отражать текущие сведения в базе данных. Обычно отчеты форматируют для печати, но их можно также просматривать на экране, экспортировать в другую программу или отправлять в виде сообщений электронной почты [7].

Макросы – это макрокоманды. Если какие-то операции с базой производятся особенно часто, имеет смысл сгруппировать несколько команд в один макрос и назначить его выделенной комбинации клавиш.

Макросы в приложении Access можно рассматривать как упрощенный язык программирования, который позволяет добавлять функциональные возможности в базу данных. Например, кнопке команды в форме можно назначить макрос, который будет запускаться при нажатии этой кнопки. Макрос содержит последовательность действий для выполнения определенной задачи, например для открытия отчета, выполнения запроса или закрытия базы данных. Большинство операций с базой данных, выполняемых вручную, можно автоматизировать с помощью макросов, которые позволяют существенно экономить время [8].

Модули – это программные процедуры, написанные на языке Visual Basic. Если стандартных средств Access не хватает для удовлетворения особо изощренных требований заказчика, программист может расширить возможности системы, написав для этого необходимые модули [12].

Модули, как и макросы, являются объектами, которые можно использовать для добавления функциональных возможностей в базу данных. В то время как макросы создаются в приложении Access путем выбора макрокоманд из списка, модули пишутся на языке программирования Visual Basic для приложений (VBA). Модуль представляет собой набор описаний, операторов и процедур, которые хранятся в одном программном блоке. Модуль может быть либо модулем класса, либо стандартным модулем. Модули класса присоединяются к формам или отчетам и обычно содержат процедуры, предназначенные для формы или отчета, к которому они относятся. Стандартные модули содержат процедуры общего назначения, не связанные ни с одним другим объектом. Стандартные модули, в отличие от модулей класса, перечисляются в группе Модули в области переходов [2].

База данных – это набор сведений, относящихся к определенной теме или задаче, такой как отслеживание заказов клиентов или хранение коллекции звукозаписей. Если база данных хранится не на компьютере или на компьютере хранятся только ее части, приходится отслеживать сведения из целого ряда других источников, которые пользователь должен скоординировать и организовать самостоятельно [9].

СУБД представляет несколько средств создания каждого из основных объектов базы. Их можно классифицировать как:

- ручные (разработка объектов в режиме Конструктора);
- автоматизированные (разработка с помощью программ-мастеров);
- автоматические – средства ускоренной разработки простейших объектов.

При разработке таблиц и запросов лучше использовать ручные средства – работать в режиме Конструктора.

При разработке учебных форм, отчетов, лучше пользоваться автоматизированными средствами, предоставляемыми мастерами.

Основные понятия реляционных БД: нормализация, связи и ключи

1 Принципы нормализации:

- в каждой таблице БД не должно быть повторяющихся полей;
- в каждой таблице должен быть уникальный идентификатор;
- Каждому значению первичного ключа должна соответствовать достаточная информация о типе сущности или об объекте таблицы;
- Изменение значений в полях таблицы не должно влиять на информацию в других полях.

2 Виды логической связи которые могут существовать между записями двух таблиц:

- один – к – одному, каждой записи из одной таблицы соответствует одна запись в другой таблице;
- один – ко – многим, каждой записи из одной таблицы соответствует несколько записей другой таблице;
- многие – к – одному, множеству записей из одной таблице соответствует одна запись в другой таблице;
- многие – ко – многим, множеству записей из одной таблицы соответствует несколько записей в другой таблице [2].

Тип отношения в создаваемой связи зависит от способа определения связываемых полей:

Отношение «один-ко-многим» создается в том случае, когда только одно из полей является полем первичного ключа или уникального индекса.

Отношение «один-к-одному» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы.

Отношение «многие-ко-многим» фактически является двумя отношениями «один-ко-многим» с третьей таблицей, первичный ключ которой состоит из полей внешнего ключа двух других таблиц [7].

3 Ключи. Ключ – это столбец (может быть несколько столбцов), добавляемый к таблице и позволяющий установить связь с записями в другой таблице. Существуют ключи двух типов: первичные и вторичные или внешние [9].

Первичный ключ – это одно или несколько полей, комбинация значений которых однозначно определяет каждую запись в таблице.

Вторичный ключ - это одно или несколько полей в таблице, содержащих ссылку на поле или поля первичного ключа в другой таблице. Внешний ключ определяет способ объединения таблиц.

Из двух логически связанных таблиц одну называют таблицей первичного ключа или главной таблицей, а другую таблицей вторичного (внешнего) ключа или подчиненной таблицей. СУБД позволяют сопоставить родственные записи из обеих таблиц и совместно вывести их в форме, отчете или запросе [2].

Существует три типа первичных ключей: ключевые поля счетчика (счетчик), простой ключ и составной ключ.

Поле счетчика (Тип данных «Счетчик»). Тип данных поля в базе данных, в котором для каждой добавляемой в таблицу записи в поле автоматически заносится уникальное числовое значение [7].

Простой ключ. Если поле содержит уникальные значения, такие как коды или инвентарные номера, то это поле можно определить как первичный ключ. В качестве ключа можно определить любое поле, содержащее данные, если это поле не содержит повторяющиеся значения или значения Null.

Составной ключ. В случаях, когда невозможно гарантировать уникальность значений каждого поля, существует возможность создать ключ, состоящий из нескольких полей. Чаще всего такая ситуация возникает

для таблицы, используемой для связывания двух таблиц многие - ко - многим [2].

Необходимо еще раз отметить, что в поле первичного ключа должны быть только уникальные значения в каждой строке таблицы, т.е. совпадение не допускается, а в поле вторичного или внешнего ключа совпадение значений в строках таблицы допускается.

Если возникают затруднения с выбором подходящего типа первичного ключа, то в качестве ключа целесообразно выбрать поле счетчика [13].

Программы, которые предназначены для структурирования информации, размещения ее в таблицах и манипулирования данными называются системами управления базами данных (СУБД). Другими словами СУБД предназначены как для создания и ведения базы данных, так и для доступа к данным. В настоящее время насчитывается более 50 типов СУБД для персональных компьютеров. К наиболее распространенным типам СУБД относятся: MS SQL Server, Oracle, Informix, Sybase, MS Access и т. д [2].

2 Создание базы данных «Клиенты КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»»

2.1 Описание предметной области базы данных

У нас имеется база данных «Клиенты КГБУ СО Пансионата для граждан пожилого возраста и инвалидов "Солнечный"». Для систематизации, обработки и хранения информации о клиентах и медицинском инвентаре создадим базу данных в Microsoft Access. С ее помощью мы в дальнейшем сможем вести учет клиентов и предоставляемых услуг.

В нашей базе данных можно выделить следующие сущности:

- Анкетные данные;
- Индивидуальная программа реабилитации;
- Мягкий инвентарь;
- Номер в реестре и дата списания медицинского инвентаря.

Атрибутами сущности «Анкетные данные» являются:

- Код;
- ФИО;
- Номер дела;
- Договор;
- Вид пенсии;
- Дата рождения;
- Дата прибытия;
- Впервые;
- На постоянном обслуживании;
- № путевки;
- № в регистре;
- Дата № ИП;
- Статус;

- Пол;
- № комнаты;
- Паспортные данные;
- Снилс;
- Мед. полюс;
- Категория;
- Дата смерти;
- Причины выбытия;
- Куда выбыл;
- Жилье/откуда прибыл;
- Фото;
- Рекомендация на ИПР;
- Код технического средства .

Атрибутами сущности «Индивидуальная программа реабилитации» являются:

- Номер;
- ФИО;
- Этаж;
- Группа инвалидности;
- Справка МСЭ №;
- Справка МСЭ дата выдачи;
- Справка МСЭ срок действия;
- ИПР №;
- ИПР дата выдачи;
- ИПР срок действия;
- Медицинская реабилитация;
- Профессиональная реабилитация;
- Социальная реабилитация;
- Наименование ТСР;

- ТСР дата получения;
- ТСР срок эксплуатации;
- ТСР дата след получения;
- Дата постановки на учет в ФСС.

Атрибутами сущности «мягкий инвентарь» являются:

- Пол;
- Этаж;
- Прим;
- Размер одежды;
- Размер обуви;
- Рост;
- № Одежда зимняя;
- № Одежда демисезонная;
- № Брюки;
- № Джемпер;
- № Спорт. костюм;
- № Футболка;
- № Платье;
- № Халат;
- № Рейтузы шерстяные;
- № Трусы;
- № Майка;
- № Сорочка ночная;
- № Носки;
- № Колготки;
- № Головной убор зимний;
- № Головной убор летний;
- № Обувь зимняя;
- № Обувь;

- № Обувь летняя;
- № Тапочки;
- № Одеяло;
- № Подушка.

Атрибутами сущности «Номер в реестре и дата списания медицинского инвентаря» являются:

- Номер;
- Срок эксплуатации;
- Кол-во;
- Дата выдачи;
- Дата списания;

Указанные таблицы позволят хранить всю необходимую информацию о клиентах, а также следить за медицинским инвентарём посредством удаления, изменения и отбора данных.

2.2 Проектирование базы данных

Исходя из предметной области, база данных «Клиенты КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»» включает в себя следующие таблицы (рисунок 1):

- Анкетные данные;
- Индивидуальная программа реабилитации;
- Мягкий инвентарь;
- Номер в реестре и дата списания медицинского инвентаря.

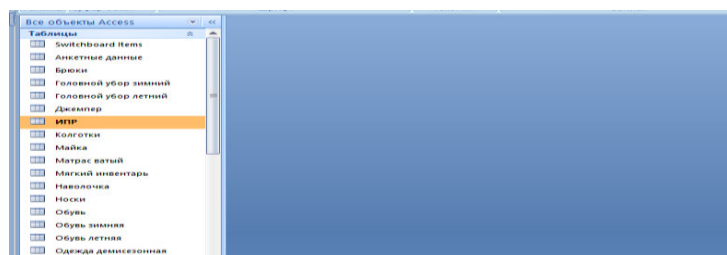


Рисунок 1 – База данных

Таблица «Анкетные данные» включает (рисунок 2):

- номер (ключевое поле), – числовой. Свойства: размер поля – длинное целое, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- ФИО – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (допускаются совпадения);

- дело – числовой. Свойства: размер поля – длинное целое, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Договор – числовой. Свойства: размер поля – длинное целое, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Вид пенсии – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (допускаются совпадения);

- Дата рождения - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как информацию, относящуюся к дате, так и информацию, относящуюся ко времени;

- Дата прибытия - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- Впервые - логический , используется для логических значений: Да/Нет, Истина/Ложь или Вкл/Выкл;

- На постоянном обслуживании - логический , используется для логических значений: Да/Нет, Истина/Ложь или Вкл/Выкл;

- № путевки – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – да, индексированное поле – да (совпадения не допускаются);

- № в регистре – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Дата № ИП - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- Статус - текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (допускаются совпадения);

- Пол - мастер подстановок. Используется для запуска мастера подстановок, позволяющего создать поле, в котором в виде раскрывающегося списка отображаются значения из другой таблицы, запроса или списка значений. Заметьте, что фактически мастер подстановок не является типом данных. "мужской \ женский";

- № комнаты – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Паспортные данные - числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Снилс - числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Мед. полюс - мастер подстановок. Используется для запуска мастера подстановок, позволяющего создать поле, в котором в виде раскрывающегося списка отображаются значения из другой таблицы, запроса или списка значений. Заметьте, что фактически мастер подстановок не является типом данных. "предоставлен \ не предоставлен";

- Категория - мастер подстановок. Используется для запуска мастера подстановок, позволяющего создать поле, в котором в виде раскрывающегося списка отображаются значения из другой таблицы, запроса или списка значений. Заметьте, что фактически мастер подстановок не является типом данных;

- Дата смерти - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- Причины выбытия - мастер подстановок. Используется для запуска мастера подстановок, позволяющего создать поле, в котором в виде раскрывающегося списка отображаются значения из другой таблицы, запроса или списка значений. Заметьте, что фактически мастер подстановок не является типом данных;

- Куда выбыл – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (допускаются совпадения);

- Жилье/откуда прибыл – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (допускаются совпадения);

- Фото - поле объекта OLE. Используется для хранения объектов OLE из других программ Microsoft Windows;

- Рекомендация на ИПР - логический , используется для логических значений: Да/Нет, Истина/Ложь или Вкл/Выкл;

- Код технического средства – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются).

Имя поля	Тип данных
Код	Счетчик
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Дело	Числовой
Договор	Числовой
дата послед_договора	Дата/время
вид пенсии	Текстовый
дата рождения	Дата/время
дата прибытия впервые	Дата/время
на пост_обслуживание	Логический
№ путевки МСП	Текстовый
№ в регистре	Текстовый

Рисунок 2 – Таблица «Анкетные данные»

Таблица «Индивидуальная программа реабилитации» включает (рисунок 3):

- Номер (ключевое поле), – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- ФИО – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (допускаются совпадения);

- Этаж тип данных – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения допускаются);

- Группа инвалидности – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Справка МСЭ № – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Справка МСЭ дата выдачи - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- Справка МСЭ срок действия – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (допускаются совпадения);

- ИПР № – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- ИПР дата выдачи - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- ИПР срок действия - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- Медицинская реабилитация - логический , используется для логических значений: Да/Нет, Истина/Ложь или Вкл/Выкл;

- Профессиональная реабилитация - логический , используется для логических значений: Да/Нет, Истина/Ложь или Вкл/Выкл;

- Социальная реабилитация - логический , используется для логических значений: Да/Нет, Истина/Ложь или Вкл/Выкл;

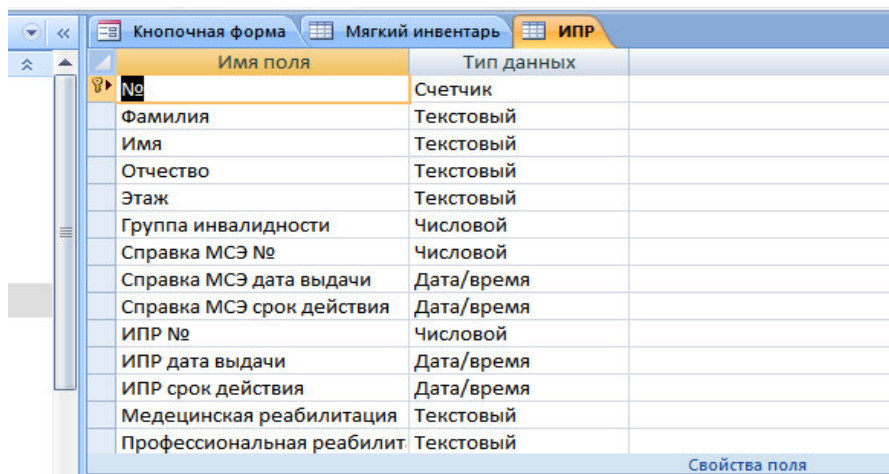
- Наименование ТСР – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (допускаются совпадения);

- ТСР дата получения - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- ТСР срок эксплуатации - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- ТСР дата след получения - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- Дата постановки на учет в ФСС - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени.



Имя поля	Тип данных
№	Счетчик
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Этаж	Текстовый
Группа инвалидности	Числовой
Справка МСЭ №	Числовой
Справка МСЭ дата выдачи	Дата/время
Справка МСЭ срок действия	Дата/время
ИПР №	Числовой
ИПР дата выдачи	Дата/время
ИПР срок действия	Дата/время
Медцинская реабилитация	Текстовый
Профессиональная реабилит.	Текстовый

Рисунок 3 – Таблица «ИПР»

Таблица «Мягкий инвентарь» включает (рисунок 4)

- Номер (ключевое поле), – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Пол – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- Этаж – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- Прим – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- Размер одежды – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- Размер обуви – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- Рост – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Одежда зимняя – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Одежда демисезонная – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Брюки – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Джемпер – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Спорт. костюм – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Футболка – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Платье – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Халат – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Рейтузы шерстяные – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Трусы – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Майка – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Сорочка ночная – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Носки – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Колготки – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Головной убор зимний – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Головной убор летний – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Обувь зимняя – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

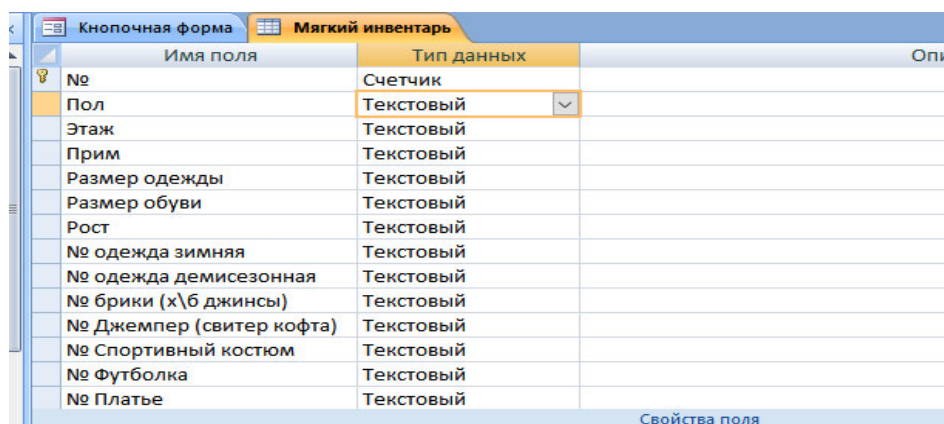
- № Обувь – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Обувь летняя – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Тапочки – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Одеяло – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);

- № Подушка – текстовый. Свойства: размер поля – 50, обязательное поле – нет, пустые строки – да, индексированное поле – да (не допускаются совпадения);



Имя поля	Тип данных
№	Счетчик
Пол	Текстовый
Этаж	Текстовый
Прим	Текстовый
Размер одежды	Текстовый
Размер обуви	Текстовый
Рост	Текстовый
№ одежда зимняя	Текстовый
№ одежда демисезонная	Текстовый
№ брики (x\б джинсы)	Текстовый
№ Джемпер (свитер кофта)	Текстовый
№ Спортивный костюм	Текстовый
№ Футболка	Текстовый
№ Платье	Текстовый

Рисунок 4 – Таблица «Мягкий инвентарь»

Таблица «Номер в реестре и дата списания медицинского инвентаря» включает (рисунок 4):

- Номер (ключевое поле), – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Срок эксплуатации - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- Кол-во – числовой. Свойства: размер поля – длинное целое, число десятичных знаков – 0, значение по умолчанию – 0, обязательное поле – нет, индексированное поле – да (совпадения не допускаются);

- Дата выдачи - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени;

- Дата списания - дата/время, используется для хранения значений дат и времени. Каждое сохраненное значение содержит как компонент, относящийся к дате, так и компонент, относящийся ко времени.

Имя поля	Тип данных
№	Числовой
срок э	Текстовый
кол-во	Числовой
дата выдачи	Дата/время
дата списания	Дата/время

Рисунок 5 – Таблица «Номер в реестре и дата списания медицинского инвентаря»

2.3 Описание этапов создания базы данных

Создание базы данных начинается с создания таблиц в режиме конструктора, где определяются поля будущей таблицы, их типы данных и свойства. Нам необходимо создать таблицы:

- Анкетные данные;
- Индивидуальная программа реабилитации;
- Мягкий инвентарь;
- Номер в реестре и дата списания медицинского инвентаря.

Далее переходим в режим таблицы, и заполняем их данными (рисунок 6 – 9).

№	Фами.	Имя	Отчество	Де	До	дата посл.	вид пенсии	дата рожде	дата прибы	вл	на	№	№ т	Д	с	пол	№ кс	паспортнык	СНИЛС
2	Иванов	Иван	Иванович	1456	98	16.04.2015	Пенсия за выслугу лет	02.04.2016	02.04.2006	да	Да	123	123	123	1	мужской	№ 12	1234 121212	89678567457
3	Петров	Петр	Петрович	127	1	31.12.1899	Пенсия по старости	17.12.2015	01.04.2016	да	Да	234	123	123	2	мужской	№ 36	8907 121323	3463463454353
7	Агиев	Павел	Павлович	111	111			14.04.2016	04.04.2016	да	Да					мужской	№ 1	8989 674563	11111
8											Да					мужской			
*	(№)										Нет								

Рисунок 6 – Таблица «Анкетные данные»

№	Фамилия	Имя	Отчество	Этаж	Группа инв.	Справка МС	Справка МС	Справка МС	ИПР №	ИПР дата вв.	ИПР срок д.	Медицинск	П
	Иванов	Иван	Иванович	№ 12	1-ая	123	27.08.2016	20.05.2017	123	13.04.2016	22.07.2016	да	д
*	(№)												

Рисунок 7 – Таблица «Индивидуальная программа реабилитации»

№	По	Этаж	Размер оде	Размер обу	Рос	№ одежда	№ одежда	№ брики (х	№ Джемпе	№ Спортив	№ Футболк	№ Платье	№ Халат	№ Рейтузы
	1 муж	1	42	43	178	1	1	1	1	1	1	1	1	1
	2 муж	1	42	43	178	1	1	1	1	1	1	1	1	1
	(№)													

Рисунок 8 – Таблица «Мягкий инвентарь»

№	срок э	кол-во	дата выдачи	дата списан	Добавить поле
	2 года	1	01.04.2016	01.04.2018	
*					

Рисунок 9 – Таблица «Номер в реестре и дата списания медицинского инвентаря»

Для связи данных таблиц необходимо создать схему данных, для чего следует выбрать все таблицы и между ними определить связи. Для

связывания полей выбирается поле в одной таблице и соединяется с соответствующим полем во второй таблице (рисунок 10).

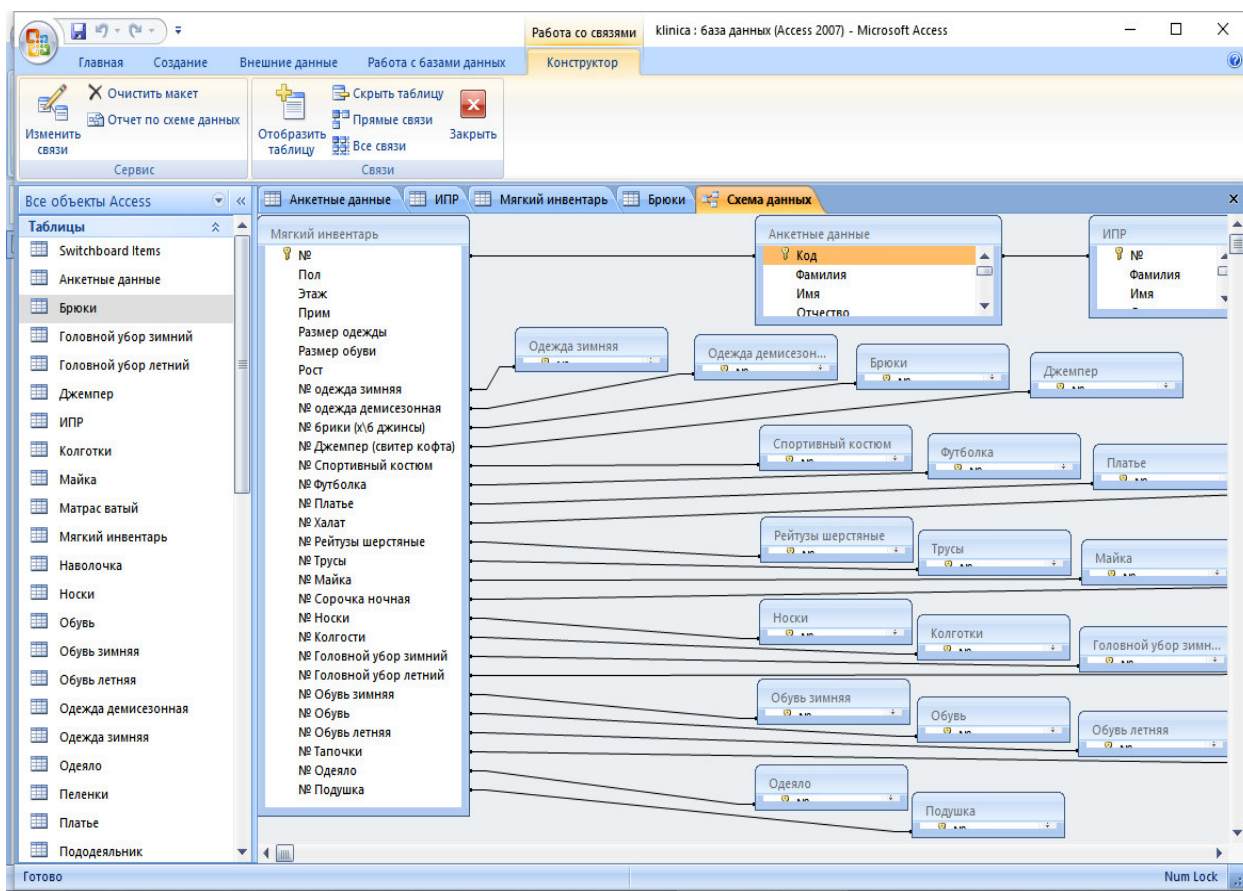


Рисунок 10 – Схема данных базы

В большинстве случаев связывают ключевое поле одной таблицы с соответствующим ему полем, которое называют полем внешнего ключа во второй таблице. Связанные поля не обязательно должны иметь одинаковые имена, но они должны иметь одинаковые типы данных и иметь содержимое одного типа.

Далее для анализа данных используются запросы. В нашей базе данных использовались запросы на выборку с условием, с вычисляемыми полями и с параметром. В общей сумме 5 запросов.

Запрос «Клиенты по алфавиту» – это запрос с параметром (параметр отбора – Фамилия) для отображения клиентов в алфавитном порядке. Он

включает поля: Фамилия (таблица «Анкетные данные»), Имя (таблица «Анкетные данные»), Отчество (таблица «Анкетные данные»), Дело (таблица «Анкетные данные»), Договор (таблица «Анкетные данные»), Вид пенсии (таблица «Анкетные данные»), Дата рождения (таблица «Анкетные данные»), Дата прибытия (таблица «Анкетные данные»), Впервые (таблица «Анкетные данные»), На постоянное обслуживание (таблица «Анкетные данные»), № путевки (таблица «Анкетные данные»), № в реестре (таблица «Анкетные данные»), дата № ИП (таблица «Анкетные данные»), Статус (таблица «Анкетные данные»), Пол (таблица «Анкетные данные»), Номер комнаты (таблица «Анкетные данные»), Снилс (таблица «Анкетные данные»), Мед. полюс (таблица «Анкетные данные»), Категория (таблица «Анкетные данные»), Причина выбытия (таблица «Анкетные данные») и Куда выбыл (таблица «Анкетные данные»).

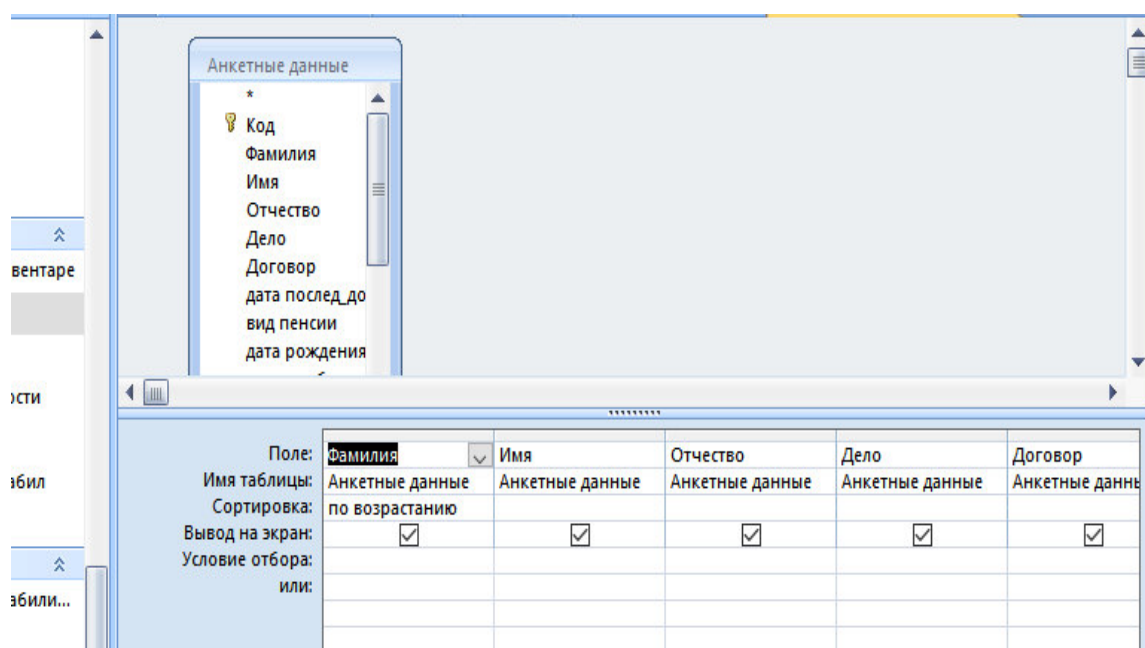


Рисунок 11 – Запрос «Клиенты по алфавиту»

Запрос «Клиенты по возрасту» – это запрос с параметром (параметр отбора – Дата рождения) для отображения клиентов от молодого и до пожилых. Он включает поля: Фамилия (таблица «Анкетные данные»), Имя

(таблица «Анкетные данные»), Отчество (таблица «Анкетные данные»), Дело (таблица «Анкетные данные»), Договор (таблица «Анкетные данные»), Вид пенсии (таблица «Анкетные данные»), Дата рождения (таблица «Анкетные данные»), Дата прибытия (таблица «Анкетные данные»), Впервые (таблица «Анкетные данные»), На постоянное обслуживание (таблица «Анкетные данные»), № путевки (таблица «Анкетные данные»), № в реестре (таблица «Анкетные данные»), дата № ИП (таблица «Анкетные данные»), Статус (таблица «Анкетные данные»), Пол (таблица «Анкетные данные»), Номер комнаты (таблица «Анкетные данные»), Снилс (таблица «Анкетные данные»), Мед. полюс (таблица «Анкетные данные»), Категория (таблица «Анкетные данные»), Причина выбытия (таблица «Анкетные данные») и Куда выбыл (таблица «Анкетные данные»).

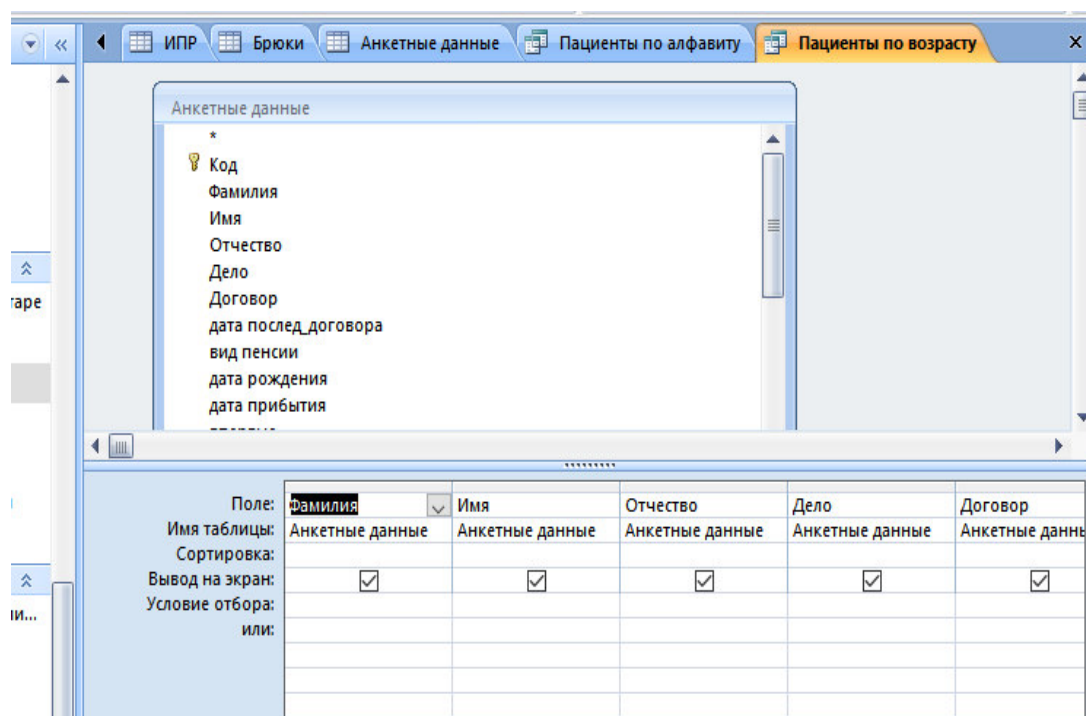


Рисунок 12 – Запрос «Клиенты по возрасту»

Запрос «Данные о уже выданном медицинском инвентаре» – это запрос для отображения медицинского инвентаря, который был выдан клиенту. Он включает поля: Фамилия (таблица «Анкетные данные»), Имя (таблица

«Анкетные данные»), Отчество (таблица «Анкетные данные»), Дело (таблица «Анкетные данные»), Номер комнаты (таблица «Анкетные данные»), № Одежда зимняя (таблица «Одежда зимняя»), № Одежда демисезонная (таблица «Одежда демисезонная»), № Брюки (таблица «Брюки»), № Джемпер (таблица «Джемпер»), № Спорт. костюм (таблица «Спорт. костюм»), № Футболка (таблица «Футболка»), № Платье (таблица «Платье»), № Халат (таблица «Халат»), № Рейтузы шерстяные (таблица «Рейтузы шерстяные»), № Трусы (таблица «Трусы»), № Майка (таблица «Майка»), № Сорочка ночная (таблица «Сорочка ночная»), № Носки (таблица «Носки»), № Колготки (таблица «Колготки»), № Головной убор зимний (таблица «Головной убор зимний»), № Головной убор летний (таблица «Головной убор летний»), № Обувь зимняя (таблица «Обувь зимняя»), № Обувь (таблица «Обувь»), № Обувь летняя (таблица «Обувь летняя»), № Тапочки (таблица «Тапочки»), № Одеяло (таблица «Одеяло») и № Подушка (таблица «Подушка»).

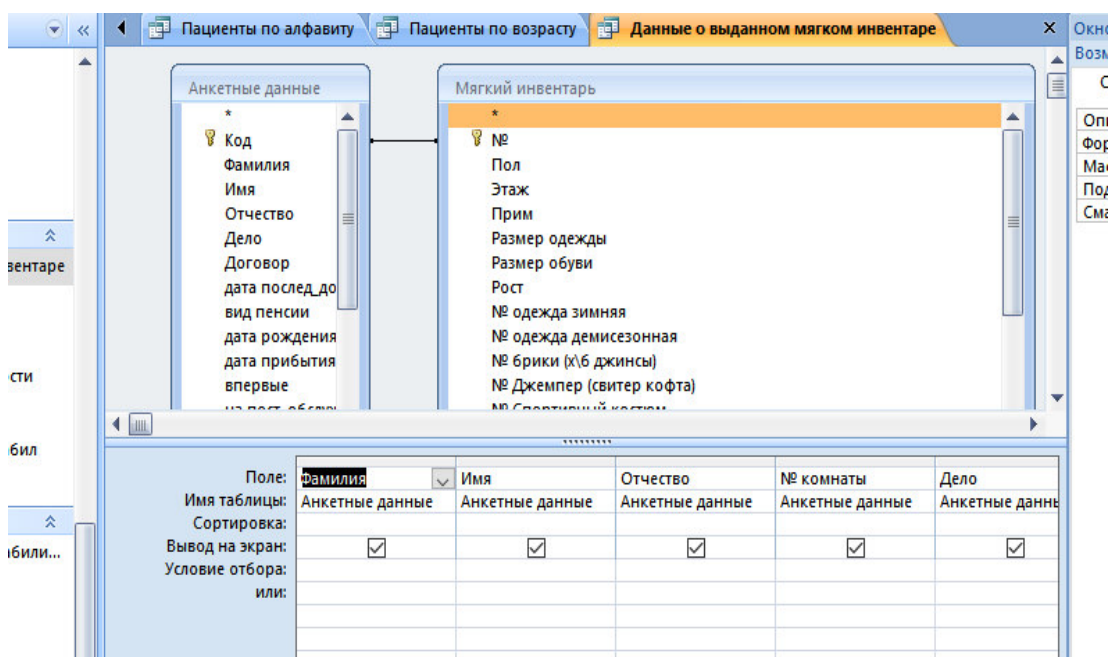


Рисунок 13 – Запрос «Данные о уже выданном медицинском инвентаре»

Запрос «Клиенты по группе инвалидности» – это запрос (параметр отбора – группа инвалидности) для отображения клиентов с определенной группой инвалидности. Он включает поля: ФИО (таблица «ИПР»), Этаж (таблица «ИПР»), Группа инвалидности (таблица «ИПР»), Справка МСЭ № (таблица «ИПР»), Справка МСЭ дата выдачи (таблица «ИПР»), Справка МСЭ (таблица «ИПР»).

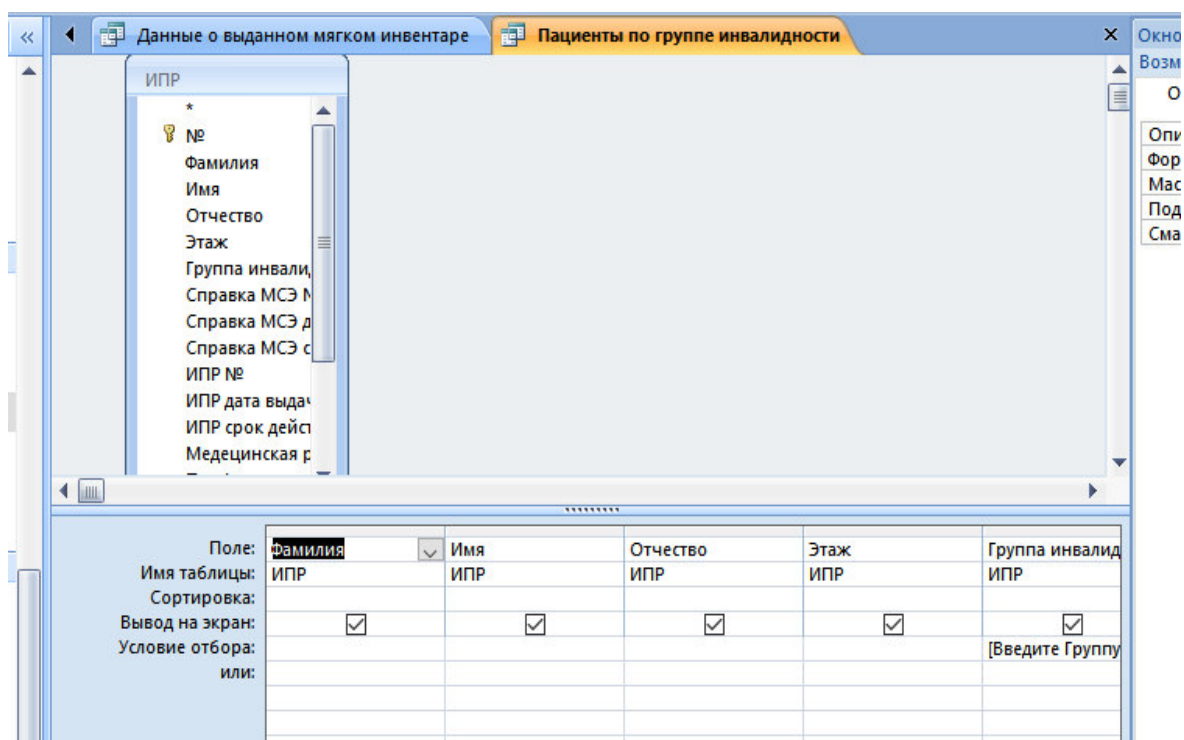


Рисунок 14 – Запрос «Клиенты по группе инвалидности»

Запрос «Рекомендации на индивидуальную программу реабилитации» – это запрос с параметром (параметр отбора – рекомендуем на инд. программ реабилитации) для отображения клиентов которые проходят лечение по индивидуальной программе реабилитации. Он включает поля: ФИО (таблица "Анкетные данные"), Дело (таблица "Анкетные данные"), Дата последующего договора (таблица "Анкетные данные"), Вид пенсии (таблица "Анкетные данные"), Рекомендуем на индивидуальную программу реабилитации (таблица "Анкетные данные").

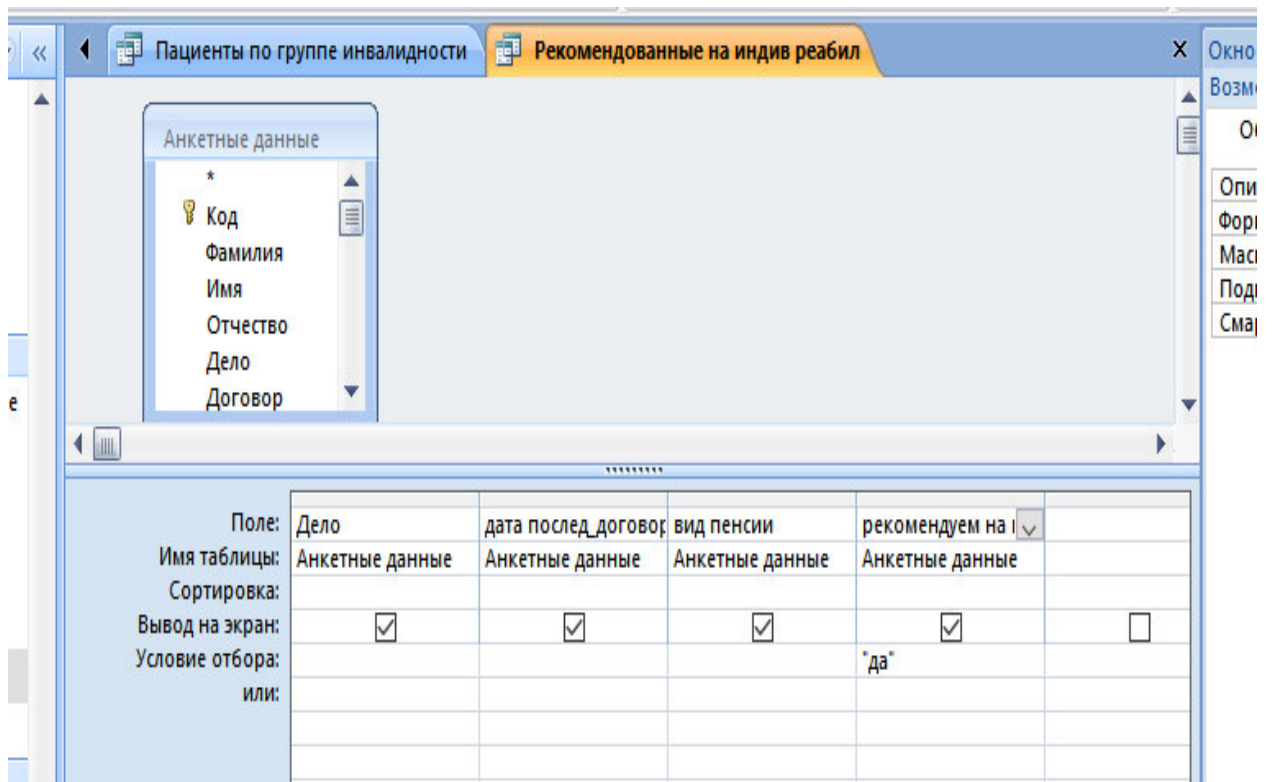


Рисунок 15 – Запрос «Рекомендации на индивидуальную программу реабилитации»

Запрос «Технические средства» – это запрос для отображения клиентов которым необходимо специальные технические средства. Он включает поля: ФИО (таблица "Анкетные данные"), Группа инвалидности (таблица "Анкетные данные"), Наименование (таблица "ИПР"), ТСР дата получения (таблица "ИПР"), ТСР срок эксплуатации (таблица "ИПР"), ТСР дата следующего получения (таблица "ИПР").

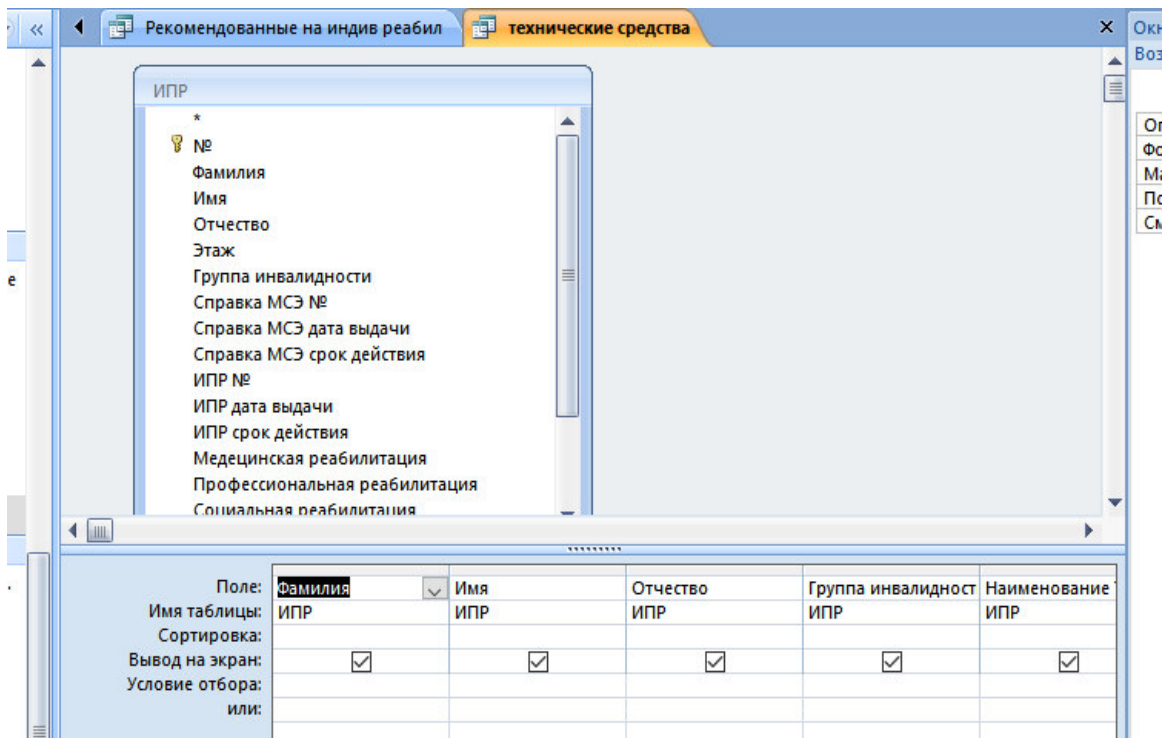


Рисунок 16 – Запрос «Технические средства»

Для печати и просмотра информации удобно использовать отчеты. В нашей базе имеется два отчета: «ИПР и технические средства», «Мягкий инвентарь выдан».

Для создания первого отчета откроем Мастер создания отчетов. Из таблицы «ИПР» добавим поля: «Номер», «ФИО», «Дело», «ИПР №», «ИПР дата выдачи», «ИПР срок действия», «Наименование ТСР», «ТСР дата получения», «ТСР срок эксплуатации», «ТСР дата получения», "ТСР дата след получения", "Медицинская реабилитация", "Профессиональная реабилитация", "Социальная реабилитация". Выберем: макет – обычный, ориентация – альбомная, , стиль – деловой. Сохраним отчет. Далее откроем его в режиме конструктора. Отредактируем стиль отчета на свое усмотрение, расположим все элементы так, чтобы они гармонично располагались на экране и не загромождали друг друга. Сохраним изменения (рисунок 16).

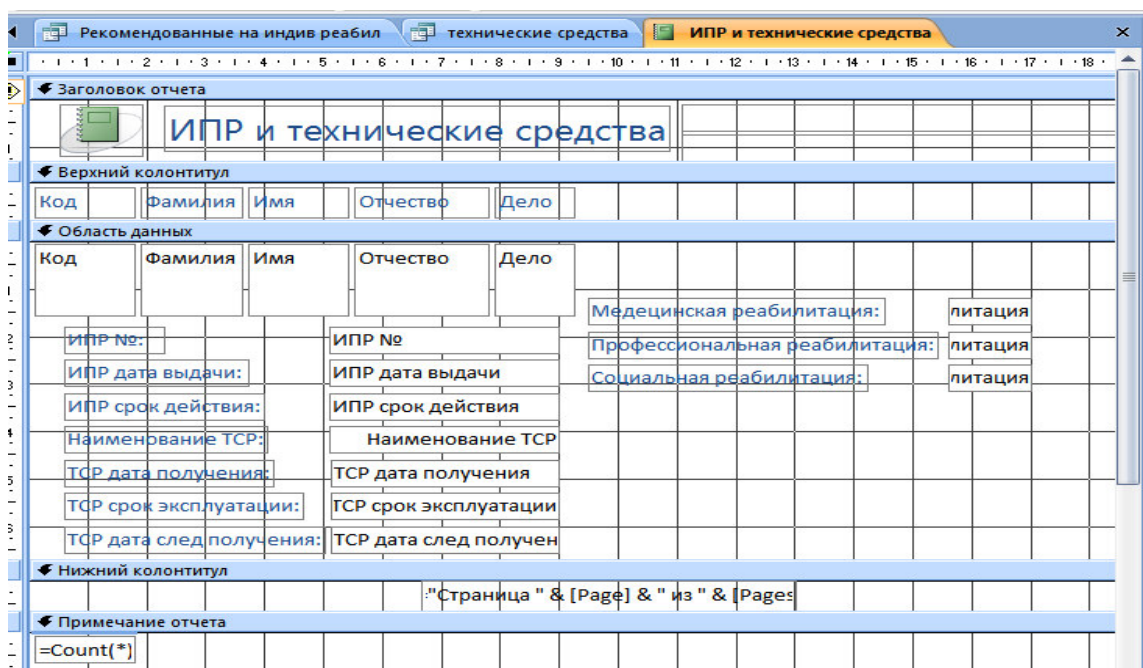


Рисунок 17 – Отчет «ИПР и технические средства»

Для создания второго отчета откроем Мастер создания отчетов. Из таблицы "Анкетные данные" и "Мягкий инвентарь" добавим поля: «ФИО», «Дело», «Пол», «Категория», «Фото», "№ Одежда зимняя", "№ Одежда демисезонная", "№ Брюки", "№ Джемпер", "№ Спорт. костюм", "№ Футболка", "№ Платье", "№ Халат", "№ Рейтузы шерстяные", "№ Трусы", "№ Майка", "№ Сорочка ночная", "№ Носки", "№ Колготки", "№ Головной убор зимний", "№ Головной убор летний", "№ Обувь зимняя", "№ Обувь", "№ Обувь летняя", "№ Тапочки", "№ Одеяло", "№ Подушка". Выберем: макет – обычный, ориентация – альбомная, стиль – деловой. Сохраним отчет. Далее откроем его в режиме конструктора. Отредактируем стиль отчета на свое усмотрение, расположим все элементы так, чтобы они гармонично располагались на экране и не загромождали друг друга. Сохраним изменения (рисунок 17).

Заголовок отчета						
Мягкий инвентарь выданный пациентам						
Верхний колонтитул						
Код	Фамилия	Имя	Отчество	Дело	пол	Категория
Область данных						
Код	Фамилия	Имя	Отчество	Дело	пол	Категория
№	№	№	№	№	№	№
№ одежда зимняя:			№ одежда зимня:			№ Сорочка ночная:
№ одежда демисезонная:			№ одежда демии:			№ Носки
№ брики (х\б джинсы):			№ брики (х\б дж			№ Колготки
№ Джемпер (свитер кофта):			№ Джемпер (сви			№ Головной убор зимний:
№ Спортивный костюм:			№ Спортивный к			№ Головной убор
№ Футболка:			№ Футболка			№ Головной убор летний:
№ Платье:			№ Платье			№ Обувь зимняя:
№ Халат:			№ Халат			№ Обувь зимняя
№ Рейтузы шерстяные:			№ Рейтузы шерс			№ Обувь:
№ Трусы:			№ Трусы			№ Обувь летняя:
№ Майка:			№ Майка			№ Тапочки:
						№ Одеяло:
						№ Подушка:
						№ Подушка

Рисунок 18 – Отчет «ИПР и технические средства»

Теперь приступим к созданию форм. Всего их будет 5: «Индивидуальная программа реабилитации», «Мягкий инвентарь», «Пациенты», «Отчеты», «Кнопочная форма».

Форму «Индивидуальная программа реабилитации» создаем с помощью мастера. Она будет содержать поля: "Номер", "ФИО", "Этаж", "Группа инвалидности", "Справка МСЭ №", "Справка МСЭ дата выдачи", "Справка МСЭ срок действия", "ИПР №", "ИПР дата выдачи", "ИПР срок действия", "Медицинская реабилитация", "Профессиональная реабилитация", "Социальная реабилитация", "Наименование ТСР", "ТСР дата получения", "ТСР срок эксплуатации", "ТСР дата след получения", "Дата постановки на учет в ФСС" из таблицы «ИПР». Кроме того, добавим кнопку для создания, сохранения и для удаления выбранной записи (создание кнопок – обработка записей – удалить запись, новая запись, сохранить запись и переход на форму) а также кнопка для перехода на главную форму (рисунок 18).

индивидуальная программа реабилитации

Фамилия: Имя: Отчество:

Этаж: Группа инвалидности:

Справка МСЭ №:

Справка МСЭ дата выдачи:

Справка МСЭ срок действия:

ИПР №:

ИПР дата выдачи:

ИПР срок действия:

Медицинская реабилитация:

Профессиональная реабилитация:

Социальная реабилитация:

Наименование ТСП:

ТСП дата получения:

ТСП срок эксплуатации:

ТСП дата след получения:

Дата постановки на учет в ФСС:

Рисунок 19 – Форма «Индивидуальная программа реабилитации»

Для редактирования таблиц «Анкетные данные» и «Мягкий инвентарь» создаем формы с соответствующими названиями с помощью мастера: внешний вид формы – в один столбец, стиль – стандартный. Затем открываем формы в режиме конструктора, редактируем размещение данных, добавляем кнопку для создания, сохранения и удаления текущей записи, а также кнопка для перехода на главную форму (создание кнопок – обработка записей – удалить запись, новая запись, сохранить запись и переход на форму) и вставляем картинку на задний фон (рисунок 19 – 20).

Пациенты

Фамилия:

Имя:

Отчество:

Дело:

Договор:

дата рождения:

дата прибытия:

впервые:

на пост_обслуживание:

пол:

№ комнаты:

паспортные данные:

СНИЛС:

медицинский полис:

Категория:




Рисунок 20 – Форма «Анкетные данные»

Рисунок 21 – Форма «Мягкий инвентарь»

Такая форма, как «Отчеты» были созданы в режиме конструктора. Для этого мы запускаем режим конструктора, сверху размещаем надпись-название, а ниже кнопки, вызывающие запросы (создание кнопок – разное – выполнить запрос), другие формы (создание кнопок – работа с формой – открыть форму), отчеты (создание кнопок – работа с отчетом – просмотр отчета) либо кнопки для распечатки отчетов (создание кнопок – работа с отчетом – печать отчета) (рисунок 21).

Рисунок 22 – Форма «Отчеты»

Для обобщения всех созданных форм создадим «Кнопочную форму» с помощью служебной программы Диспетчер кнопочных форм, создадим кнопку форму, оформив ее затем в конструкторе (рисунок 22).

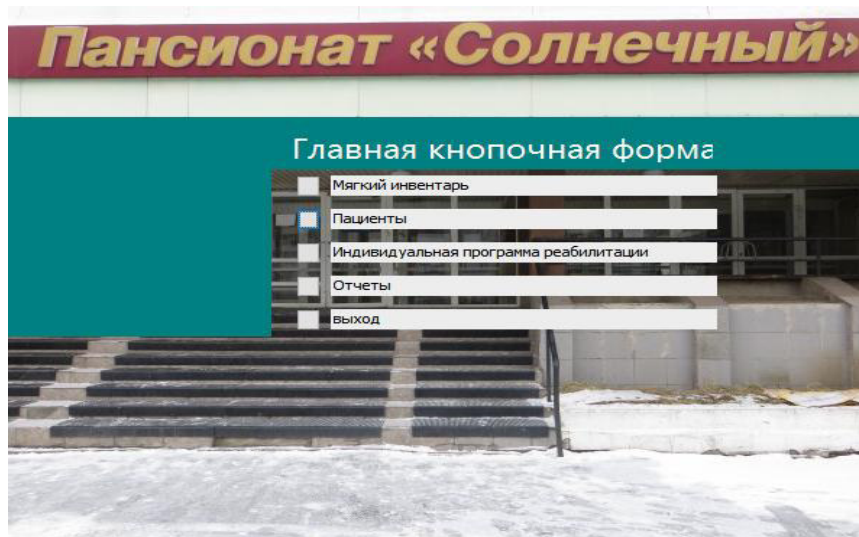


Рисунок 23 – Главная кнопочная форма

И наконец создадим макрос на открытие базы данных с помощью макрокоманды «Открыть Форму», сохранив его под именем «AutoExec» (рисунок 23).

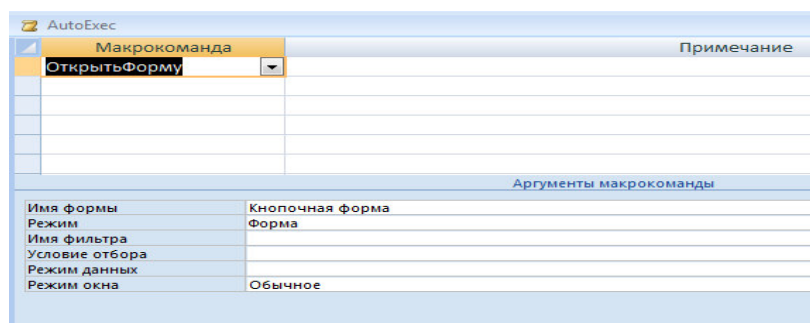


Рисунок 24 – Макрос «AutoExec»

Итак, мы рассмотрели основные этапы создания базы данных, начиная с создания таблиц и заканчивая макросами.

2.4 Описание работы базы данных

Полученная в результате выполнения проекта база данных «Клиенты КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»» позволяет работать с информацией о текущих клиентах и медицинском инвентаре.

При запуске базы данных с помощью макроса AutoExec открывается главная «Кнопочная форма», которая позволяет работать со всей базой данных (рисунок 24). Она содержит такие кнопки как: "Клиенты", "Мягкий инвентарь", "Индивидуальная программа реабилитации", "Отчеты" и "Выход".

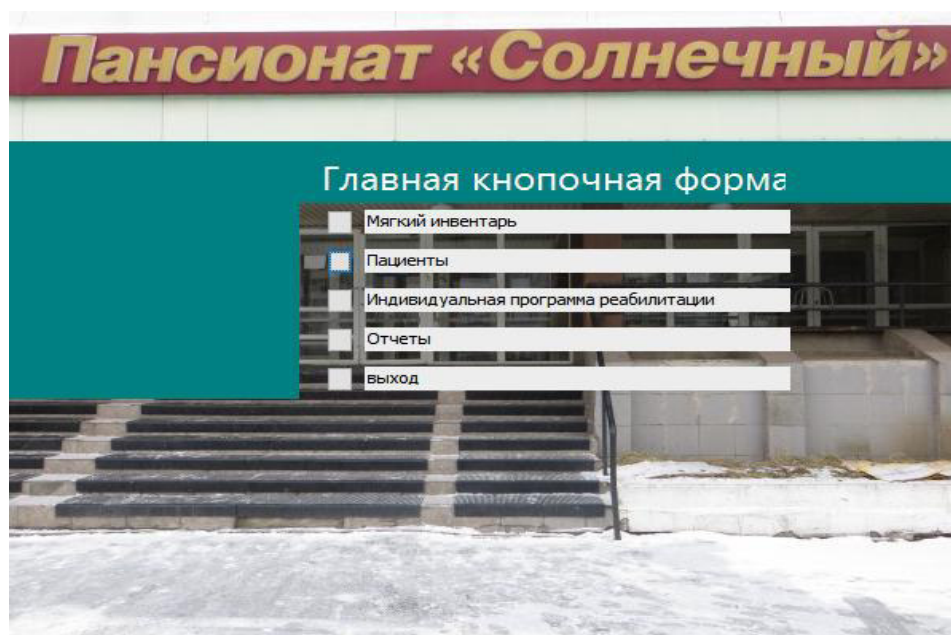


Рисунок 25 – Главная кнопочная форма

При нажатии кнопки «Клиенты» открывается соответствующая форма, отражающая полную информацию о клиентах с возможностью пополнять записи, сохранять и удалить запись (рисунок 25).

Пациенты






Фамилия:	Иванов	    
Имя:	Иван	
Отчество:	Иванович	
Дело:	1456	
Договор:	98	
дата рождения:	02.04.2016	
дата прибытия:	02.04.2006	
впервые:	да	
на пост_обслуживание:	Да	
пол:	мужской	
№ комнаты:	№ 12	
паспортные данные:	1234 121212	
СНИЛС:	89678567457	
медицинский полис:	предоставлен	
Категория:	1-ая	

Рисунок 26 – Форма «Пациенты»

При нажатии кнопки «Мягкий инвентарь» открывается форма «Мягкий Инвентарь», отражающая полную информацию о медицинском инвентаре с возможностью пополнить, сохранить и удалить запись (рисунок 26).

Мягкий инвентарь





№ пациента:	1	Пол:	муж	   
Этаж:	1	№ Трусы:	1	
Прим:		№ Майка:	1	
Размер одежды:	42	№ Сорочка ночная:	1	
Размер обуви:	43	№ Носки:	1	
Рост:	178	№ Колготки:	1	
№ одежда зимняя:	1	№ Головной убор зимний:	1	
№ одежда демисезонная:	1	№ Головной убор летний:	1	
№ брики (х\б джинсы):	1	№ Обувь зимняя:	1	
№ Джемпер (свитер кофта):	1	№ Обувь:	1	
№ Спортивный костюм:	1	№ Обувь летняя:	1	
№ Футболка:	1	№ Тапочки:	1	
№ Платье:	1	№ Одеяло:	1	
№ Халат:	1	№ Подушка:	1	
№ Рейтузы шерстяные:	1			

Рисунок 27 – Форма «Мягкий инвентарь»

При нажатии кнопки «Индивидуальная программа реабилитации» открывается форма «Индивидуальная программа реабилитации», отражающая полную информацию о клиентах которые проходят лечение по индивидуальной программе реабилитации с возможностью пополнить, сохранить и удалить запись (рисунок 27).

индивидуальная программа реабилитации					
Фамилия:	Иванов	Имя:	Иван	Отчество:	Иванович
Этаж:	№ 12	Группа инвалидности:	1-ая		
Справка МСЭ №:	123				
Справка МСЭ дата выдачи:	27.08.2016				
Справка МСЭ срок действия:	20.05.2017				
ИПР №:	123				
ИПР дата выдачи:	13.04.2016				
ИПР срок действия:	22.07.2016				
Медицинская реабилитация:	да				
Профессиональная реабилитация:	да				
Социальная реабилитация:	да				
Наименование ТСП:	трость				
ТСП дата получения:	23.07.2016				
ТСП срок эксплуатации:	1 месяц				
ТСП дата след получения:	23.08.2016				
Дата постановки на учет в ФСС:	05.04.2016				

Рисунок 28 – Форма «Индивидуальная программа реабилитации»

При нажатии на кнопку «Отчеты» на главной кнопочной форме появляется форма со статистическими данными и отчетами, на которой располагаются кнопки: «ИПР и технические средства» (вызывает отчет «ИПР и технические средства»), «Мягкий инвентарь выданный клиентам» (вызывает отчет «Мягкий инвентарь выданный клиентам»), а также кнопки для печати отчетов (рисунок 28).

Отчеты

ИПР и технические средства

Мягкий инвентарь выданный пациентам

Рисунок 29 – Форма «Отчеты»

При нажатии на кнопку закрыть, на главной кнопочной форме – база данных закрывается.

2.5 Апробация базы данных

Апробация базы данных производилась на базе пансионата "Солнечный" г. Красноярска. На основной компьютер предприятия, расположенный в зоне приема клиентов, была установлена СУБД и загружена БД. Произведен краткий инструктаж с работниками по работе с системой. База данных эксплуатировалась в ознакомительном режиме в мае месяце.

В настоящий момент система переведена в рабочий режим и используется в пансионате.

ЗАКЛЮЧЕНИЕ

Целью дипломной работы являлось разработать базу данных «Клиенты КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»».

Мы решили следующие задачи:

- изучили организационную структуру пансионата "Солнечный";
- подобрали технологию СУБД, а именно MS Access, средствами которой мы создали таблицы с данными, запросы, отчеты, формы и макросы.
- разработали модель базы данных «Клиенты КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»»;
- создали Базу данных «Клиенты КГБУ СО Пансионат для граждан пожилого возраста и инвалидов «Солнечный»» в MS Access;
- описали алгоритм заполнения и работы Базы Данных.

Для систематизации, обработки и хранения информации о клиентах и проводимых работах была создана база данных в Microsoft Access. С ее помощью можно вести учет клиентов и медицинского инвентаря.

Пансионат "Солнечный" начал применять информационные технологии в своей работе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Симонович С.В. Информатика: учебник для вузов/ С.В. Симонович - Питер, 2014. 640 с.
- 2 Дейт, К. Введение в системы баз данных/ К Дейт Диалектика, 2008.
- 3 Хомоненко А.Д. Базы данных: учебник для вузов/ А.Д. Хороненко, М.Г. Мальцев, В.М. Циганков - СПб : «Корона» принт, 2009. 416 с.
- 4 Когаловский М.Р. Технология баз данных на персональных ЭВМ/ М.Р. Кагаловский - Москва : Финансы и статистика, 1992. 224 с.
- 5 Базы данных и системы управления базами данных. URL: <http://informatic.ugatu.ac.ru/lib/office/Access.htm>
- 6 Леонтьев В.П. Новейшая энциклопедия Интернет/ В.П. Леонтьев - Москва : ОЛМА-ПРЕСС, 2003. 781с.
- 7 Иллюстрированный самоучитель по Microsoft Access. URL: <http://www.taurion.ru/access>
- 8 Горев А. Эффективная работа с СУБД/ А. Горев, Р. Аханян, С. Макашариков СПб : «Питер», 1997. 700 с.
- 9 Программное обеспечение для работы с базами данных Access - Office.com. URL: <http://office.microsoft.com/ru-ru/access/>
- 10 Леонтьев В.П. Новейшая энциклопедия персонального компьютера/ В.П. Леонтьев - Москва : ОЛМА-ПРЕСС, 2009. 957с.
- 11 Глушаков С.А. Персональный компьютер: учебный курс/ С.А. Шлушаков, А.С. Сурядный - Харьков : Фолио, 2007. 519с.
- 12 Мэтью Р.Д. Web-сервер под UNIX / Р.Д. Мэтью - СПб : Символ-Плюс, 1998. 560с.
- 13 Симонович С.В. Информатика: Базовый курс / С. В. Симонович - СПб : Питер, 2011. 640 с.

- 14 Тимошок Т.В. Microsoft Access 2002. Самоучитель/ Т.В. Тимошок - Москва : Издательский дом «Вильямс», 2008. 352 с.
- 15 Робинсон С. Microsoft Access 2000: учебный курс/ С. Робинсон - СПб : Питер, 2010. 313 с.
- 16 Ковалевич И.А. Основы баз данных: учебное пособие/ И.А. Ковалевич - Красноярск : КГТУ, 2002. 133с.
- 17 Система управления базами данных. URL: https://ru.wikipedia.org/wiki/Система_управления_базами_данных
- 18 MS Access Кирпичики. URL: <http://msa.polarcom.ru/>
- 19 Microsoft Access. URL: http://ru.wikipedia.org/wiki/Microsoft_Access
- 20 Реляционная СУБД. URL: <http://ru.wikipedia.org/wiki/Реляционная СУБД>
- 21 Мэтью Мак-Дональд Access 2007. Недостающее руководство/ Мэтью Мак-Дональд - СПб.: «БХВ-Петербург», 2007. 784 с.