

**ВВЕДЕНИЕ**

В последние годы на первый план выдвигается новая отрасль – информационная индустрия, связанная с производством технических средств, методов, технологий для производства новых знаний. Эта индустрия тесно связана с развитием компьютерных технологий.  
 В информационном обществе доминирует производство информационного продукта, а материальный продукт становится более информационно емким. Изменятся весь уклад жизни, система ценностей: возрастает значимость культурного досуга, возрастает спрос на знания, от человека требуется способность к интеллектуальному труду и творчеству. В результате появились противоречия между ограниченными возможностями человека по восприятию и переработке информации и существующими массивами хранящейся и передаваемой информации.  
 Возникло большое число избыточной информации, в которой иногда трудно сориентироваться и выбрать нужные сведения.  
 Для решения подобных проблем применяются автоматизированные базы данных. Они стали неотъемлемой частью практически всех компьютерных систем - от отрасли до отдельного предприятия. За последние несколько лет вырос уровень потребительских качеств систем управления базами данных (СУБД): разнообразие поддерживаемых функций, удобный для пользователя интерфейс, сопряжение с программными продуктами, в частности с другими СУБД, возможности для работы в сети и т.д. СУБД позволяет сводить воедино информацию из самых разных источников (электронные таблицы, другие базы данных) и помогает быстро найти необходимую информацию, донести ее до окружающих с помощью отчетов, графиков или таблиц.  
 К настоящему времени накоплен значительный опыт проектирования БД, предназначенных для управления производством, это позволяет сделать процесс создания БД более эффективным.  
 Исходными данными для написания работы являются теоретические источники информации, усредненные статистические данные деятельности организации предметной области.  
 Объектом исследования в ходе курсовой работы являются базы данных предприятия.  
 Предметом исследования является процесс разработки реляционной базы данных.  
 Целью данной работы является разработка базы данных « Документооборот предприятия» для предприятия с использованием продукта Microsoft Access, входящего в пакет Microsoft Office.  
 Задачи для достижения поставленной цели:

– описать предметную область;

– составить инфологическую модель;

– разработать таблицы и схемы данных;

– создать запросы;

– создать удобный интерфейс пользователя;

– создать основную кнопочную форму.  
 Объектом работы является базы данных для автоматизации документооборота.

Предметом данной работы – процесс разработки БД.

Теоретическая база исследования – основные теоретические подходы и концепции, раскрывающие сущность базы данных и ее структурные элементы.   
 Работа состоит из введения, трех глав, заключения и списка использованной литературы.

**1 Теоретические основы базы данных и СУБД**  
  
  **1.1 Понятие базы данных и СУБД**

База данных – это организованная структура для хранения информации.  
С понятием базы данных тесно связано понятие системы управления базой данных. Это набор программных средств, предназначенных для создания структуры новой базы данных, наполнения ее контентом, редактирования контента и визуализации информации. Визуализация информации базы означает выбор, упорядочение, форматирование и выдачу данных, отображаемых в соответствии с заданным критерием, после устройств вывода или передачи по каналам связи [3].  
 Существует множество различных систем управления базами данных, так называемые «СУБД», такие как: IMS, Cetop, Oracle, Clipper, FoxPro, Access. Но, прежде чем продолжить изучение базы данных, рассмотрим некоторые основные понятия.  
 Данные – систематизированная и структурированная каким-либо образом информация.  
 Поле – основной и минимальный элемент данных.  
 Запись – несколько полей.  
 Связи – логические взаимосвязи между записями или полями.  
 База данных – это набор данных, связанных с определенной темой или целью.  
 База данных может быть сконфигурирована как массив обработки и хранения данных, который может использоваться одним или несколькими пользователями.  
 В зависимости от разных моделей данных база данных может быть иерархической, относительной сети. Кроме того, будут рассматриваться только реляционные базы данных.  
 Таблица представляет собой серию записей (строк), состоящих из отдельных полей (столбцов), в которых хранится информация и компонент, фактически содержимое базы данных.  
 Таблицы являются основной формой отображения информации, содержащейся в базе данных. Без таблицы форма не может быть разработана; Запросы и отчеты составляются на основании таблиц.  
 Таблицы базы данных, как правило, позволяют работать с гораздо большим количеством разных типов данных. Например, базы данных Microsoft Access работают со следующими типами данных:  
 1) текстовый – это тип данных, используемый для хранения простого текста, не отформатированный в ограниченном размере (до 255 символов);  
 2) числовой – тип данных для хранения действительных чисел;  
 3) поле Мемо – это специальный тип данных для хранения большого количества текста (до 65 535 символов). Физически текст не сохраняется в поле. Он хранится в другом месте в базе данных, и указатель на него хранится в поле, но для пользователя такое разделение не всегда чувствуется;  
 4) дата / время – тип данных для хранения дат и текущего времени;  
 5) денежный – тип данных для хранения денежных сумм. Теоретически можно будет использовать поля числового типа для их записи, но для денежных сумм есть некоторые особенности (например, правила общего округления), которые делают более удобным использование специального типа данных, а не настройки числового типа;  
 6) счетчик – специальный тип данных для уникальных в натуральных числах с автоматическим наращиванием. Используется для нумерации записей.  
 7) логический – тип для хранения логических данных (они могут принимать только два значения, например, да или нет).  
 8) гиперссылка – это специальное поле для хранения URL-адресов интернет-объектов. Щелчок по ссылке автоматически отправляет в браузер и воспроизводит объект в его окне.  
 9) мастер подстановок. Не является специальным типом данных. Это объект, с помощью которого вы можете автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать их из раскрывающегося списка [5].  
 СУБД (система управления базами данных) – это программа, которая управляет данными, хранит, извлекает, ищет и редактирует информацию, хранящуюся в базе данных. СУБД также делятся на иерархические, сеточные, реляционные, в зависимости от данных, которые они обрабатывают.  
Система управления базами данных (СУБД) – это система, которая обеспечивает поиск, хранение, восстановление данных и формирование ответов на запросы. Система обеспечивает сохранность данных, их конфиденциальность, трафик и связь с другим программным обеспечением. Основные функции, которые пользователь может выполнять с помощью СУБД:

– создать структуру базы данных;

– заполнение базы данных информацией;

– изменять (редактировать) структуру и содержание базы данных;

– поиск информации в базе данных;

– сортировка данных;

– защита базы данных;

– проверить целостность базы данных.

Современные СУБД позволяют включать в них не только текстовую и графическую информацию, но и голосовые фрагменты и даже видеоролики.  
 Удобство использования СУБД позволяет создавать новые базы данных, не прибегая к программированию, а только используя встроенные функции. СУБД обеспечивают точность, целостность и согласованность данных, а также легкий доступ к ним [9].  
 Популярная СУБД – FoxPro, Access for Windows, Paradox.  
 Поэтому необходимо различать фактические данные (БД) – аккуратные наборы данных и системы управления базами данных (СУБД) – программы для управления хранением и обработкой данных. Например, приложение Access, включенное в пакет программ Microsoft Office, является СУБД, которая позволяет пользователю создавать и компилировать табличные базы данных.  
 **1.2** **Классификация баз данных**  
  
 Существует следующая классификация баз данных:

1. По характеру хранимой информации:

– фактографические – содержит краткую информацию об описанных объектах, представленную только в определенном формате   
 – документальные – содержит различные документы (информацию): текст, графику, аудио, мультимедиа.  
 2. По способу хранения данных:  
 – централизованный (хранится на одном компьютере),  
 – распределенный (используется в локальных и глобальных компьютерных сетях).  
 3. По данным организационной структуры:  
 – реляционные (табличные),  
 – нереляционные.  
 Термин «реляционный» указывает, что эта модель хранения данных построена на взаимосвязи между ее составными частями. Реляционная база данных, по сути, представляет собой двумерную таблицу. Каждая строка такой таблицы называется записью. Столбцы таблицы называются полями: каждое поле характеризуется своим именем и типом данных. Поле БД – это столбец таблицы, который содержит значения для определенного свойства.  
 Характеристики относительной модели данных:  
 – каждый элемент в таблице является отдельным элементом данных;  
 – все поля таблицы однородны, то есть существует один тип;  
 – нет идентичных записей в таблице;  
 – порядок записей в таблице может быть произвольным и может характеризоваться несколькими полями, типом данных.  
 Иерархическая база данных называется базой данных, в которой информация упорядочена следующим образом: один элемент считается основным компонентом, остальные подчиняются. В иерархической базе данных записи располагаются в определенной последовательности, например, по шкале, и вы можете искать данные, непрерывно «отбрасывая» их от шага к шагу. Эта модель характеризуется такими параметрами, как уровни, узлы, соединения. Принцип модели таков, что несколько узлов более низкого уровня связаны посредством связи с одним узлом более высокого уровня.  
 Узел – это информационная модель элемента, расположенного на заданном уровне иерархии.  
 Характеристики иерархической модели данных:  
 – некоторые узлы самого низкого уровня связаны только с одним узлом самого высокого уровня;  
 – иерархическое дерево имеет только одну вершину (корень), оно не подчинено какой-либо другой вершине;  
 – каждый узел имеет свое имя (ID);  
 – существует только один способ из корневой записи записать больше личных данных.  
 Сетевой называется БД, в которой к вертикальным иерархическим связям добавляются горизонтальные связи. Каждый объект может быть главным и подчиненным.  
 Программное обеспечение, предназначенное для работы с базами данных, называется системой управления базами данных (СУБД). СУБД используются для упорядоченного хранения и обработки больших объемов информации [13].

**1.3 Функции СУБД**  
  
 Современная система управления базами данных должна выполнять следующие функции:  
 – определения данных. СУБД должна допускать определения данных (внешние схемы, концептуальные схемы, внутренние схемы и все связанные сопоставления) в исходной форме для преобразования этих настроек в форму соответствующих объектов, т.е. СУБД должна включать компонент языкового процессора для определения различных языков данных.  
 – обработка данных. СУБД должна уметь обрабатывать запросы пользователя на выборку, изменение или удаление соответствующих данных в базе данных или на добавление новых данных в базу данных, т. е. СУБД должна включать в себя компонент процессора языка обработки данных (ЯОД).  
 – «запланированные» и «незапланированные» запросы. Запланированный запрос – это запрос, необходимость которого предусмотрена заранее. Системный администратор должен настроить физический проект базы данных таким образом, чтобы обеспечить удовлетворительную производительность для таких запросов. Это типично для «запуска приложений». Незапланированный запрос – это особый запрос, необходимость которого не была предусмотрена заранее. Это типично для приложений «поддержки решений» [16].  
 – безопасность и целостность данных. СУБД должна отслеживать запросы пользователей, чтобы прекратить попытки нарушить правила безопасности и целостности, определенные администратором базы данных.  
Для пользователей информационной системы недостаточно, чтобы база данных просто отражала объекты в реальном мире. Важно, чтобы это отражение было однозначным и последовательным.  
 – восстановление данных и репликация. Восстановление и репликация данных выполняется СУБД или администратором.  
 – введение словаря данных. Словарь данных – это полный набор таблиц или файлов, который представляет собой каталог всех описаний данных. Он также может содержать информацию о пользователе, разрешения и т. Д., Которые доступны только для базы данных администратора. Словарь данных является основным источником информации для СУБД, АБД всех пользователей.  
 – производительность. СУБД должна выполнять все указанные функции с максимально возможной эффективностью.  
 Как правило, целью СУБД является обеспечение пользовательского интерфейса с базой данных.

**2 Проектирование базы данных документооборота предприятия  
   
 2.1 Описание предметной области документооборота предприятия**

Документы – это основные информационные ресурсы предприятия, работа с которыми требует правильной организации. Документы обеспечивают информационную поддержку принятия управленческих решений на всех уровнях и сопровождают ведение всех бизнес-процессов. Документооборот – это непрерывный процесс движения документов, объективно отражающий деятельность предприятия и позволяющий оперативно управлять им. Документооборот – это движение документов с момента их создания или получения до завершения исполнения, отправки или сдачи в дело [8] .  
 Организация документооборота – это правила, в соответствии с которыми происходит движение документов в учреждении.  
 Документооборот является важным звеном делопроизводства, так как определяет не только инстанции движения документов, но и скорость этого движения.  
 Основными характеристиками документооборота являются маршрут движения, который включает все инстанции на пути движения документа от создания черновика (или от получения) до подшивки в дело, и время, затрачиваемое на прохождение документов по этому маршруту. Отсюда главное правило организации документооборота – оперативное прохождение документа по наиболее короткому и прямому маршруту с наименьшими затратами времени. Кроме того, при организации движения документов необходимо выполнение следующих правил (принципов):  
 – централизация операций по приему, первичной обработке и отправке документов (это означает, что все документы, поступающие в организацию всеми способами доставки (по почте, факсу, электронной почте и др.) и отправляемые из организации, обрабатываются в службе ДОУ);  
 – максимальное сокращение числа инстанций прохождения и исключение возвратных движений документа, не обусловленных деловой необходимостью (возвратным считается такое движение документа, при котором он поступает в ту инстанцию, в которой уже проходил обработку);  
 – маршрутизация отдельных видов документов (разработка схем прохождения документов);  
 – однократность регистрации документов (документы, поступающие в организацию и создаваемые в организации, регистрируются один раз: входящие – при поступлении, внутренние и исходящие – при создании; регистрация документов может осуществляться в службе ДОУ учреждения или в структурном подразделении);  
 – организация предварительного рассмотрения документов (распределение документов, поступивших в организацию, на документы, направляемые на резолюцию руководителю, его заместителям и передаваемые на исполнение непосредственно в подразделения; предварительное рассмотрение проводит специалист службы ДОУ (секретарь-референт или помощник руководителя);  
 – вынесение конкретных резолюций, по возможности точно определяющих исполнителя, поручение, срок исполнения;  
 – исключение необоснованных согласований, организация параллельного согласования документов одновременно несколькими работниками (копирование проекта документа или передача его по электронной почте одновременно всем заинтересованным лицам);  
 – снижение уровня подписания документов.  
 Соблюдение этих правил позволяет реализовать основной организационный принцип службы делопроизводства: специализацию, возможность централизованного выполнения однородных средств оргтехники и персонала [2].  
 Кроме того, соблюдение этих правил позволяет улучшить документационное обеспечение аппарата управления за счет:  
 – прямого направления документопотоков, исключения дублирующих и повторных операций по подготовке и обработке документов;  
 – ритмичности движения документов, которая обеспечивает их равномерное поступление и способствует пропорциональной загрузке как специалистов аппарата управления, так и службы делопроизводства.  
  
 **2.2 Постановка задачи**  
  
 Актуальность разработки базы данных «Документооборот предприятия» заключается в необходимости хранить и учитывать сведения о документах. Использование базы данных позволяет более эффективно отслеживать и корректировать работу предприятия.

Создается база данных для учета документов и их «маршрута» сотрудниками предприятия.

Для автоматизации процесса нужно выполнить:

1. запись информации о предприятии,
2. запись информаци о истории «пути» документов,
3. запись информации о сотрудниках предприятия,
4. поиск записей по определенным запросам,
5. составление отчетов о документах,
6. предоставление пользователю найденной информации в электронном виде,   
    7) предоставление пользователю найденной информации в печатном виде.  
     
    **2.3 Построение информационно-логической модели базы данных**  
     
    Нормализация баз данных заключается в приведении структуры хранения данных к нормальным формам (NF). Всего таких форм существует 8, но часто достаточным является соблюдение первых трех.

На начальном этапе проектирования была составлена одна ненорма-лизованная таблица, которая содержит 15 полей (номер документа, содержание, дата, статус, правки и изменения, тип документа, количество, дата и время получения, дата и время отправления, этап «маршрута», ФИО сотрудника, заработная плата, номер телефона сотрудника, должность, подразделения.). Подобная база данных сложна для восприятия и заполнения информации. Следовательно, перед дальнейшей разработкой БД необходимо провести нормализацию до третьей нормальной формы.

После нормализации мы получили 9 таблиц. Перед разработкой ин-формационно-логической модели реляционной БД рассмотрим, из каких информационных объектов должна состоять эта БД.

Можно выделить девять объектов, которые не будут обладать избыточностью: «Документ», «Должность», «История», «Маршрут», «Сотрудник», «Тип документа», «Подразделение», «Действие», «Пункт маршрута».

Представим состав реквизитов этих объектов в виде «название объекта» (перечень реквизитов):

– «Тип документа» (Тип документа),

– «Документ» (Код документа; дата; содержание; статус; правки и исправления; тип документа, количество),

– «История пути договора» (Код истории; маршрут; код сотрудника; код документа, дата и время отправления/получения документа),

– «Сотрудники» (Код сотрудника; ФИО сотрудника; телефон; должность; подразделение),

– «Подразделение» (Код подразделения; название подразделения),

– «Должность» (Код должности; название должности),

– «Действие» (Код действия; название; количество дней на выполнение)  
 – «Маршрут» (Код маршрута; название маршрута),  
 –«Пункт маршрута» (Код маршрута; название маршрута; код документа; тип документа; действие).

Рассмотрим связи между объектами «Пункт маршрута» и «Маршрут». Между данными таблицами установлены две связи. Первая связь по коду маршрута, что соответствует связи один-к-одному, так как код является уникальным. А вторая связь между названием маршрута, она обычная.  
 Рассмотрим связь между объектами «Тип документа» и «Пункт маршрута». Связь один-ко-многим, так как один и тот же тип документа может иметь несколько «маршрутов»  
 Рассмотрим связь между объектами «Пункт маршрута» и «Действие». Связь один-ко-многим, так как одно и то же действие может быть применено на разных этапах «маршрута» документа.  
 Рассмотрим связь между «История пути документа» и «Сотрудники». Связь один-ко-многим, так как один сотрудник может заниматься несколькими документами.

Рассмотрим связь между объектами «Документ» и «История пути документа». Здесь связь один-ко-многим, так как один документ может отправляться от одного сотрудника к другому.

Рассмотрим связь между объектами «Сотрудники» и «Подразделения». Здесь связь один-ко-многим, так как в одном подразделении может быть более двух сотрудников.

Рассмотрим связи между объектами «Сотрудники» и «Должность». Здесь связь один-ко-многим, так как несколько соьрудников могут занимать одинаковые должности.  
 Рассмотрим последнюю связь между «Документ» и «Тип документа». Связь один-ко-многим, так как несколько документов могут быть одного и того же типа.

**3 Разработка базы данных   
  
 3.1 Разработка таблиц и схемы базы данных**  
  
 Для разработки данной базы данных понадобятся следующие таблицы: Действие, Документ, История пути маршрута, Маршрут, Сотрудники, Должность, Подразделение, Тип документа, Пункт маршрута. Которые уже связаны для обеспечения непротиворечивости данных и проиллюстрированы на рисунке 2.

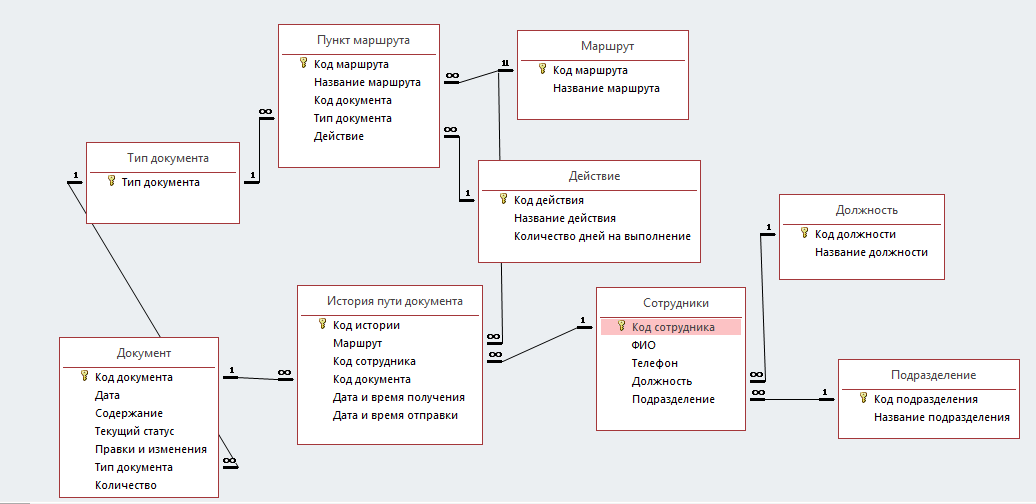


Рисунок 2 – Схема данных БД «Документооборот предприятия»

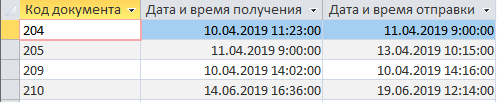
Следующим действием мы начали заполнять таблицы с помощью форм. Для этого нужно создать формы каждой таблицы.  
 Для заполнения таблицы «Пункт маршрута» сначала нужно заполнить таблицы «Тип документа» и «Действие», для заполнения таблицы «История пути документа» – таблицы «Документ» и «Сотрудники», для заполнения которых, в свою очередь, нужно заполнить таблицы «Тип документа», «Должность» и «Подразделение». А для заполнения таблицы «Маршрут» нужно изначально заполнить таблицу «Пункт маршрута».   
 Начнем заполнения базы данных с таблицы «Действие». Представленная таблица состоит из 3 полей: код действия, название действия и количество дней на выполнение. Тип данных у всех столбцов – это короткий текст. Ключевым полем является Код действия.  
 Продолжим заполнение с таблицы «Тип документов». Данная таблица состоит из 2 полей: код типа документа и тип документа. Тип данных у поля – это короткий текст. Ключевым полем является Код типа документа.  
 Следующей мы заполним таблицу «Пункт маршрута». У данной таблицы имеется 5 полей: код маршрута, название маршрута, код документа, код типа документа, код действия. Тип данных у поля Номер клиента – короткий текст. Ключевое поле данной таблицы – Код маршрута.  
 Далее мы заполним таблицу «Маршрут». Данная таблица состоит из 3 полей: код маршрута, названия маршрута и этапа. Тип данных у поля – короткий текст. Ключевое поле – Код маршрута.  
 Следующей заполняется таблица «Документы». Данная таблица состоит из 7 полей: код документа, дата, содержание, текущий статус, правки и изменения, код типа документа и количество. Тип данных у поля дата– это дата и время, у поля правки и изменения – логический. У всех остальных – короткий текст. Ключевым является поле Код документа.  
 Продолжим заполнение с таблицы «Должность». Данная таблица состоит из 2 полей: код должности и название. Тип данных у обеих полей – короткий текст. Ключевым полем является Код должности.  
 Далее заполняем таблицу «Подразделение». Она состоит из 2 полей: код подразделения и название подразделения. Тип данных – короткий текст, а ключевое поле – Код подразделения.  
 Продолжим заполнение с таблицы «Сотрудники». Данная таблица состоит из 6 полей: код сотрудника, ФИО, телефон, код должности, код подразделения и З/П. Тип данных у поля З/П – денежный, у всех остальных – короткий текст. Ключевым полем является Код сотрудника.  
 Закончим заполнение с таблицы «История пути документа». Данная таблица состоит из 6 полей: код истории, код маршрута, код сотрудника, код документа, дата и время получения, дата и время отправки. Тип данных у полей дата и время отправки и получения – дата и время, у остальных – короткий текст. Ключевым полем является Код истории.  
 Построение структуры БД происходит в СУБД реляционного типа MS Access 2016 в соответствии с разработанной логической моделью БД БД «Документооборот предприятия».  
  
 **3.2 Запросы системы**  
  
 Запросы позволяют отбирать необходимую информацию из таблиц и представлять ее в табличном виде. Запросы создаются с помощью команды SELECT. Для данной системы базы данных было создано 15 запросов.  
 Запрос 1 создается по данным таблицы «История пути документа» и является запросом с вычисляемым полем. Этот запрос показывает определенные даты получения и отправки документов. Код SQL для данного запроса выглядит следующим образом:   
 SELECT [История пути документа].[Код документа], [История пути документа].[Дата и время получения], [История пути документа].[Дата и время отправки]  
 FROM [История пути документа]  
 WHERE ((([История пути документа].[Дата и время получения]) Between #4/10/2019# And #6/19/2019#));  
 Фрагмент результата выполнения запроса представлен на рисунке 3.  
 

Рисунок 3 – Фрагмент результата запроса 1

Запрос 2 создается по данным таблиц «Документ» и является дополнительным запросом для главной кнопочной формы и показывает все документы без внесенных изменений. Код SQL для данного запроса выглядит следующим образом:  
 SELECT Документ.[Код документа], Документ.Содержание, Документ.[Текущий статус], Документ.[Правки и изменения]  
 FROM Документ  
 WHERE (((Документ.[Правки и изменения])=No));  
 Фрагмент результата запроса показан на рисунке 4.

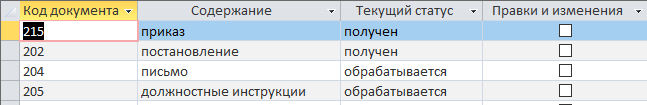


Рисунок 4 – Фрагмент результата запроса 2

Запрос 3 создается по данным таблиц «Документ» и является дополнительным запросом для главной кнопочной формы и показывает все документы с внесенными изменениями. Код SQL для данного запроса выглядит следующим образом:

SELECT Документ.[Код документа], Документ.Содержание, Документ.[Текущий статус], Документ.[Правки и изменения]  
 FROM Документ  
 WHERE (((Документ.[Правки и изменения])=Yes));

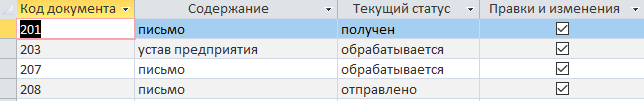
Фрагмент результата запроса показан на рисунке 5.  


Рисунок 5 – Фрагмент результата запроса 3

Запрос 4 создается по данным таблиц «Тип документа», «Пункт маршрута», «Маршрут» и является запросом с использованием внешнего соединения таблиц (правое и левое). Данный запрос показывает на каком этапе какой документ находится. Код SQL для данного запроса выглядит следующим образом:  
 SELECT Пункт маршрута.[Код маршрута], Тип документа.[Тип документа], Маршрут.[Этап]  
 FROM Маршрут RIGHT JOIN ([Тип документа] RIGHT JOIN [Пункт маршрута] ON [Тип документа].[Код типа документа] = [Пункт маршрута].[Код типа документа]) ON Маршрут.[Код маршрута] = [Пункт маршрута].[Код маршрута];  
 Фрагмент выполненного запроса показан на рисунке 6.

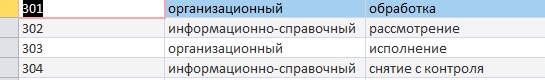


Рисунок 6 – Фрагмент результата запроса 4

Запрос 5 создается по данным таблиц «Пункт маршрута», «Маршрут» и является запросом с использованием внутреннего соединения по одному полю. С помощью этого запроса мы можем наблюдать маршрут каждого документа. Код SQL для данного запроса выглядит следующим образом:

SELECT Пункт маршрута.[Код маршрута], Пункт маршрута.[Код документа], Маршрут.[Название маршрута]

FROM Маршрут INNER JOIN Пункт маршрута ON Маршрут.[Код маршрута] = Пункт маршрута.[Код маршрута];

Фрагмент выполненного запроса показан на рисунке 7.

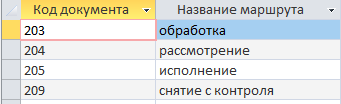


Рисунок 7 – Фрагмент результата запроса 5

Запрос 6 создается по данным таблицы «Документ» и является запросом с вычисляемым полем. Этот запрос показывает количество обрабатываемых документов. Код SQL для данного запроса выглядит следующим образом:

SELECT Документ.\*, Документ.[Текущий статус]

FROM Документ

WHERE (((Документ.[Текущий статус])="обрабатывается"));

Фрагмент выполненного запроса показан на рисунке 8.



Рисунок 8 – Фрагмент результата запроса 6

Запрос 7 создается по данным таблиц «Маршрут», «Пункт маршрута», «История пути документа» и является запросом с использованием косвенно связанных таблиц. С помощью этого запроса мы можем отследить время получения и отправки документа. Код SQL для данного запроса выглядит следующим образом:

SELECT Маршрут.[Код маршрута], [История пути документа].[Дата и время получения], [История пути документа].[Дата и время отправки], [Пункт маршрута].[Код документа]

FROM (Маршрут INNER JOIN [Пункт маршрута] ON (Маршрут.[Название маршрута] = [Пункт маршрута].[Название маршрута]) AND (Маршрут.[Код маршрута] = [Пункт маршрута].[Код маршрута])) INNER JOIN [История пути документа] ON [Пункт маршрута].[Код маршрута] = [История пути документа].[Код маршрута];

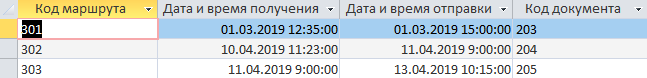
Фрагмент выполненного запроса показан на рисунке 9.

Рисунок 9 – Фрагмент результата запроса 7

Запрос 8 создается по данным таблиц «Должность», «Сотрудники» и является запросом с использованием соединений по отношению. Данный запрос выводит должность каждого сотрудника. Код SQL для данного запроса выглядит следующим образом:

SELECT Сотрудники.[Код сотрудника], Сотрудники.[ФИО], Сотрудники.[Код должности], Должность.[Название должности]  
 FROM Должность INNER JOIN Сотрудники ON Должность.[Код должности] = Сотрудники.[Код должности];

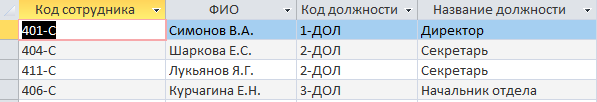
Фрагмент данного запроса представлен на рисунке 10.

Рисунок 10 – Фрагмент результата запроса 8

Запрос 9 создается по данным таблиц «Пункт маршрута», «Маршрут» и является запросом с использованием таблиц, связанных более чем по одному полю. С помощью данного запроса выводятся все документы, находящиеся на этапе обработки или отчета. Код SQL для данного запроса выглядит следующим образом:

SELECT [Пункт маршрута].[Код маршрута], [Пункт маршрута].[Название маршрута], [Пункт маршрута].[Код документа], [Пункт маршрута].[Код действия], Маршрут.Этап  
 FROM Маршрут INNER JOIN [Пункт маршрута] ON (Маршрут.[Название маршрута] = [Пункт маршрута].[Название маршрута]) AND (Маршрут.[Код маршрута] = [Пункт маршрута].[Код маршрута])  
 WHERE (((Маршрут.Этап)="отчет" Or (Маршрут.Этап)="обработка"));

На рисунке 11 представлен выполненный запрос.

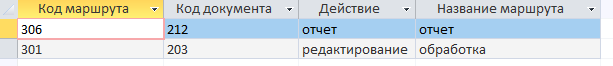


Рисунок 11 – Фрагмент результата запроса 9

Запрос 10 создается по данным таблицы «Документ» и является запросом с использованием функции агрегирования. С помощью этого запроса мы рассчитываем сумму количества документов с правками и изменениями, с содержанием «письмо» и статусом «получен». Код SQL для данного запроса выглядит следующим образом:

SELECT Sum([Количество]) AS Выражение1  
 FROM Документ  
 WHERE (((Документ.[Правки и изменения])=Yes) AND ((Документ.Содержание)="Письмо") AND ((Документ.[Текущий статус])="получен"));

Результат выполненного запроса представлен на рисунке 12.

  
Рисунок 12 – Фрагмент результата запроса 10

Запрос 11 создается по данным таблицы «Документ» и является запросом с использованием функции агрегирования. С помощью этого запроса мы рассчитываем сумму количества всех документов на предприятии. Код SQL для данного запроса выглядит следующим образом:

SELECT Sum([Количество]) AS Выражение1  
 FROM Документ;

На рисунке 13 представлен результат выполненного запроса.



Рисунок 13 – Фрагмент результата запроса 11

Запрос 12 создается по данным таблицы «Документ» и является запросом с использованием функции агрегирования. С помощью этого запроса мы рассчитываем сумму всех документов с правками и изменениями. Код SQL для данного запроса выглядит следующим образом:

SELECT Sum([Количество]) AS Выражение1  
 FROM Документ  
 WHERE (((Документ.[Правки и изменения])=Yes));

На рисунке 14 представлен результат выполненного запроса.



Рисунок 14 – Фрагмент результата запроса 12

Запрос 13 создается по данным таблицы «Документ» и является запросом с использованием функции агрегирования. С помощью этого запроса мы рассчитываем сумму всех документов без правок и изменений. Код SQL для данного запроса выглядит следующим образом:

SELECT Sum([Количество]) AS Выражение1

FROM Документ

WHERE (((Документ.[Правки и изменения])=No));

На рисунке 14 представлен результат выполненного запроса.



Рисунок 14 – Фрагмент результата запроса 13

Запрос 14 создается по данным таблиц «Документ», «Сотрудники» и является запросом с объединением. С помощью данного запроса мы объединили в одну таблицу документы и сотрудников. Код SQL для данного запроса выглядит следующим образом:

SELECT [Код документа]  
 FROM [Документ]  
 UNION SELECT [ФИО]  
 FROM [Сотрудники];  
 На рисунке 15 представлен результат выполненного запроса.

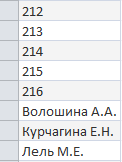


Рисунок 15 – Фрагмент результата запроса 14

Запрос 15 создается по данным таблицы «Действие» и является перекрестным запросом. С помощью данного запроса мы сделали перекрестную таблицу типов действий и сроком их выполнения. Код SQL для данного запроса выглядит следующим образом:

TRANSFORM Sum(Действие.[Количество дней на выполнение]) AS [Sum-Количество дней на выполнение]

SELECT Действие.[Код действия]

FROM Действие

GROUP BY Действие.[Код действия]

PIVOT Действие.[Название действия];

На рисунке 16 представлен результат выполненного запроса.

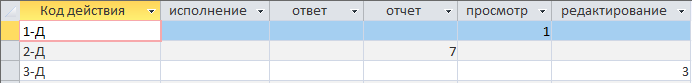


Рисунок 16 – Фрагмент результата запроса 15

**3.3 Интерфейс пользователя**  
  
 В представленной базе данных создавалась форма для ввода данных, предназначенная для каждой таблицы по отдельности. В каждой форме были созданы макросы для перелистывания страниц, а также был создан макрос для поиска записи и закрытия формы.  
 Форма «Документ» была создана из одноименной таблицы и представлена на рисунке 17.

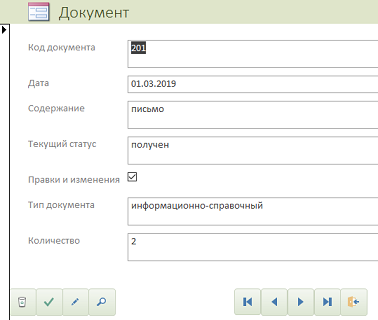


Рисунок 17 – Форма «Документ»

Форма «Маршрут» была создана из одноименной таблицы и представлена на рисунке 18.

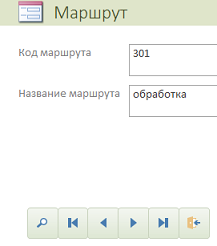


Рисунок 18 – Форма «Маршрут»

Форма «Сотрудники» была создана из одноименной таблицы и представлена на рисунке 19.

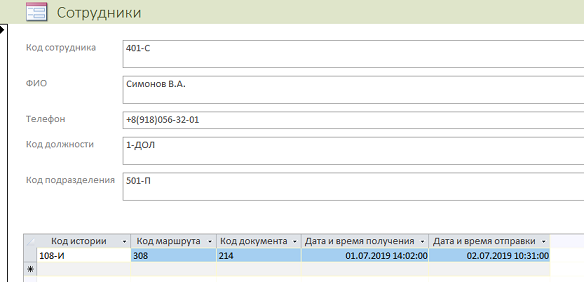


Рисунок 19 – Форма «Документ»

Форма «Тип документа» была создана из одноименной таблицы и представлена на рисунке 20.

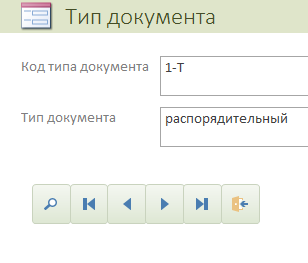


Рисунок 20 – Форма «Тип документа»

Форма «Подразделение» была создана из одноименной таблицы и представлена на рисунке 21.

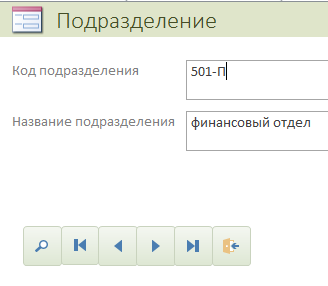


Рисунок 21 – Форма «Подразделение»

Форма «Должность» была создана из одноименной таблицы и представлена на рисунке 22.

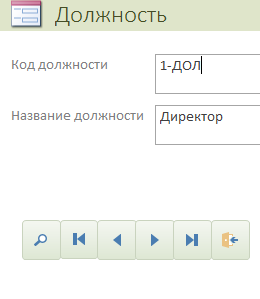


Рисунок 22 – Форма «Должность»

Форма «Действие» была создана из одноименной таблицы и представлена на рисунке 23.

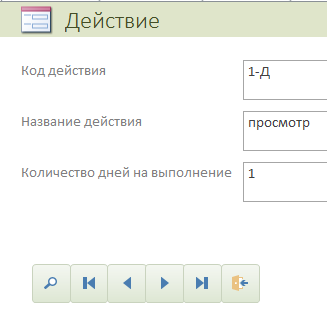


Рисунок 23 – Форма «Действие»

Форма «Пункт маршрута» была создана из одноименной таблицы и представлена на рисунке 24.

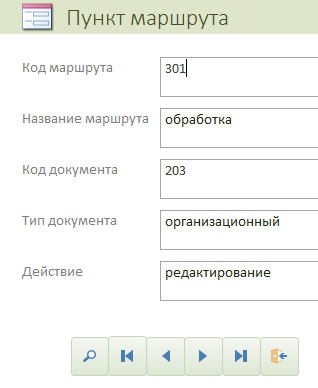


Рисунок 24 – Форма «Пункт маршрута»

Форма «История пути документа» была создана из одноименной таблицы и представлена на рисунке 25.

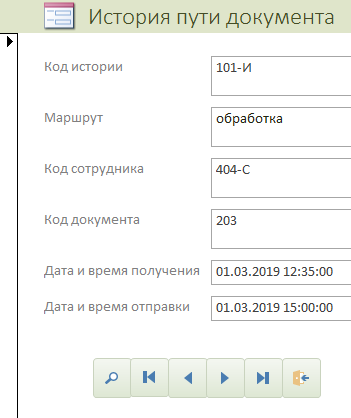


Рисунок 25 – Форма «История пути документа»

Разработку базы данных мы продолжили созданием отчетов. Мы создали 6 отчетов.  
 Отчет «Документ» создан на основе таблицы «Документ». Фрагмент этого отчета представлен на рисунке 26.

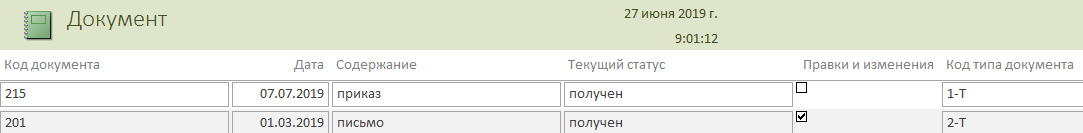


Рисунок 26 – Фрагмент отчета «Документ»

Отчет «Документы без правок» предоставляет информацию о не редактированных документах. Фрагмент этого отчета представлен на рисунке 27.

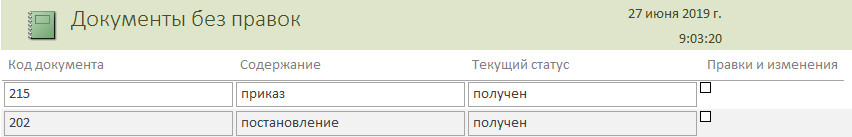


Рисунок 27 ­– Фрагмент отчета «Документы без правок»

Отчет «Документы с правками» предоставляет информацию о редактированных документах. Фрагмент этого отчета представлен на рисунке 28.

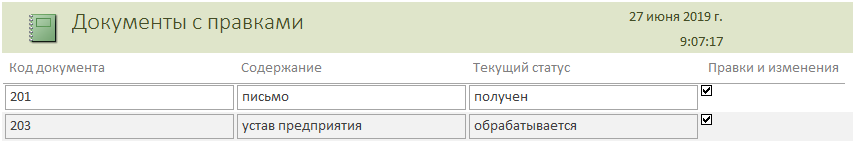


Рисунок 28 – Фрагмент отчета «Документы с правками»

Отчет «Должности» создан на основе таблицы «Должности». Фрагмент этого отчета представлен на рисунке 29.

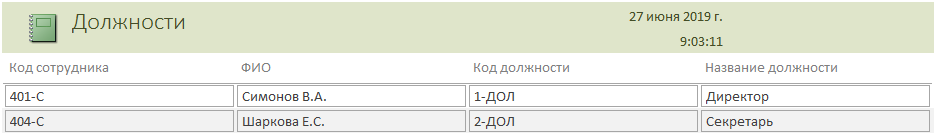


Рисунок 29 – Фрагмент отчета «Должности»

Отчет «История пути документа» предоставляет информацию о дате и времени отправки и получения документов. Фрагмент этого отчета представлен на рисунке 30.

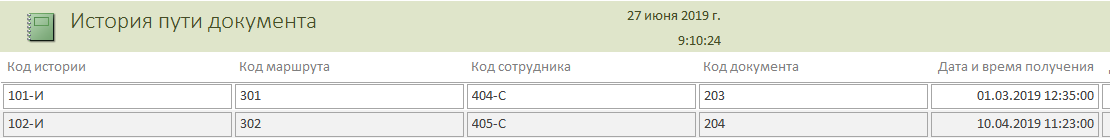


Рисунок 30 – Фрагмент отчета «История пути документа»

Отчет «Сотрудники» основан на таблице «Сотрудники». Фрагмент этого отчета представлен на рисунке 31.

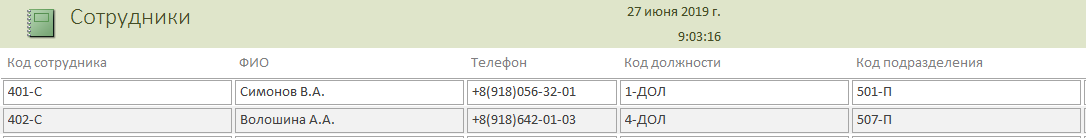


Рисунок 31 – Фрагмент отчета «Сотрудники»

Также была создана главная кнопочная форма, которая открывается автоматически при запуске программы. Она состоит из кнопок, с помощью которых можно открыть нужную форму, отчет, подсчеты или выйти из базы данных. На рисунке 32 представлена эта форма.

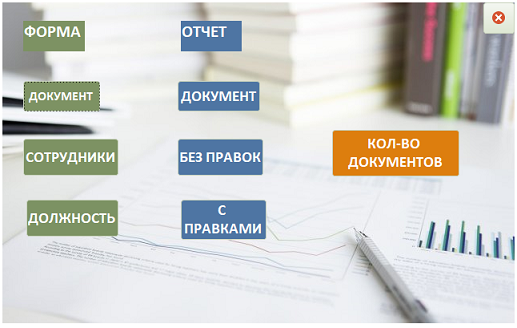


Рисунок 32 – Главная кнопочная форма

Всего было создано 14 макросов.  
 Три макроса было создано для открытия форм: Документ, История пути документа, Сотрудники.  
 Пять макросов было сделано для открытия отчетов: Документы, Документы без правок, Документы с правками, История пути и Сотрудники.  
 Три макроса открывают запросы, которые отвечают за количество всех документов, отредактированных и не отредактированных.  
 Также был создан макрос, который закрывает базу данных и который автоматически открывает главную кнопочную форму при открытии про-граммы.   
 Используя данные интерфейсы можно решить следующие задачи:  
 – упрощение внесения данных,  
 – автоматизация создания отчетов,  
 – поиск нужных записей,  
 – уменьшение риска допустить ошибку,  
 – упрощение хранения информации,  
 – возможность вывода необходимых данных с помощью написания запросов в режиме SQL или создания их в конструкторе запросов.

**ЗАКЛЮЧЕНИЕ**

На основе базы данных создан новый уровень информационных технологий, которые эффективно используются во многих областях человеческой деятельности. Организациям также нужны специально разработанные базы данных. Они способствуют наиболее эффективной работе организации.   
 Microsoft Access основан на модели реляционных баз данных, предназначенной для создания быстрых и эффективных баз данных, которые используются в повседневной жизни и в бизнесе. Кроме того, он может подключаться к другим базам данных, создавая широкий фронт для работы с данными, где бы они ни находились.   
 Оценивая преимущества и недостатки СУБД Microsoft Access и ее функциональности, можно утверждать, что в этой системе есть все инструменты, необходимые для создания, редактирования, хранения и ежедневного использования баз данных. Интерфейс программы прост и удобен, работа не требует большого количества дополнительных знаний. В ходе курса была решена проблема создания базы данных для предприятия.   
 Цель, изложенная во введении, была достигнута. Разработана база данных документооборота предприятия, с помощью которой мы можем создать:  
 – таблицы для хранения данных,  
 – запросы для поиска и извлечения только необходимых данных,  
 – формы для просмотра, добавления и изменения данных в таблицах,  
 – отчеты для анализа и печати данных в определенном формате.  
 Эта база нормализуется до третьей нормальной формы. Связи между объектами осуществляются с помощью заполнения самой базы данных. Разработанные запросы, макросы, отчеты и формы. Все это помогает упростить работу сотрудников.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1 Агрегатные функции SQL [Электронный ресурс]. – Режим доступа: https://codetown.ru/sql/agregatnye-funkcii/  
 2 Бобылева, М.П. «Управленческий документооборот: от бумажного к электронному. Вопросы теории и практики» 2 издание / М.П. Бобылева. – М.: Термика, 2019.  
 3 Бегг, К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / К. Бегг, Т. Конноли. – М.: Юрайт, 2017.  
 4 Вычисляемые поля в запросах Access. [Электронный ресурс]. – Режим доступа: https://accesshelp.ru/vychisljaemye-polja-v-zaprosah-access/  
 5 Гордеев, С.И. Организация баз данных в 2 частях. Часть 1 / С.И. Гордеев, В.Н. Волошина. – М.: Юрайт, 2017.  
 6 Дэйт, К. Дж. Введение в системы баз данных / К. Дж. Дэйт. – М.: Вильямс, 2018.  
 7 Илюшечкин, В.М. Основы использования и проектирования баз данных / В.М. Илюшечкин. – М.: Юрайт, 2016.  
 8 Кабашов, С.Ю. Делопроизводство и архивное дело в терминах и определениях / С.Ю. Кабашов. – М.: Флинта, 2018.  
 9 Кумскова, И.А. Базы данных / И.А. Кумскова. – М.: КноРус, 2016.  
 10 Латыпова, Р.Р. Базы данных. Курс лекций / Р.Р. Латыпова. – М.: Проспект, 2017.  
 11 Новиков, Б.А. Основы технологий баз данных. Учебное пособие / Б.А. Новиков, Е.А. Горшкова. – М.: ДМК-Пресс, 2019  
 12 Осипов, Д.Л. Технологии проектирования баз данных / Д.Л. Осипов. – М.: ДМК-Пресс, 2019.  
 13 Петкович, Д. SQL Server 2012. Руководство для начинающих: Пер.с англ. – СПб.: БХВ-Петербург, 2013.  
 14 Советов, Б.Я. Базы данных / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. – 2016.  
 15 Стружкин, Н.П. Базы данных: проектирование. Практикум. Учебное пособие для академического бакалавриата / Н.П. Стружкин, В.В. Годин. – 2016.  
 16 Тарасов, С.В. СУБД для программиста. Базы данных изнутри / С.В. Тарасов. – М.: Солон-пресс, 2015.  
 17 Уроки по Access для Office 365 [Электронный ресурс]. – Режим доступа: https://support.office.com/ru-ru/access  
 18 Фуфаев, Э.В. Разработка и эксплуатация удаленных баз данных / Э.В. Фуфаев, Д.Э. Фуфаев. – М.: Academia, 2014.  
 19 Цехановский, В.В. Управление данными / В.В. Цехановский, В.Д. Чертовской. – М.: Лань, 2015.  
 20 Шустова Л. И. Базы данных: учебник / Л.И. Шустова, О.В. Тараканов. – М.: НИЦ ИНФРА-М, 2016.