**Курсовая работа**

**Информационная система «Мебельная фабрика»**

**Введение**

Сегодня трудно себе представить сколько-нибудь значимую информационную систему, которая не имела бы в качестве основы или важной составляющей базу данных. Тем не менее, сейчас это вполне сложившаяся дисциплина, основанная на достаточно формализованных подходах и включающая широкий спектр приемов и методов создания баз данных. Назначение систем управления базами данных – обеспечение в течение длительного времени их сохранности, а также возможностей выборки и актуализации. Широкое использование баз данных различными категориями пользователей привело, с одной стороны, к созданию интерфейсов, требующих минимум времени на освоение средств управления системой, а с другой – к построению мощных, гибких СУБД, имеющих в том числе развитые средства защиты данных от случайного или преднамеренного разрушения.

Использование баз данных и информационных систем становится неотъемлемой составляющей деловой деятельности современного человека и функционирования преуспевающих организаций. От правильного выбора инструментальных средств создания информационных систем, определения подходящей модели данных, обоснования рациональной схемы построения базы данных, организации запросов к хранимым данным и ряда других моментов во многом зависит эффективность функционирования разрабатываемых систем.

Целью данной работы является автоматизация работы мебельной фабрики. Разработанная программа позволит существенно сократить время «бумажного учета», обеспечит легкость, эффективность и быстроту работы в вычислении стоимости заказов в бухгалтерии, выведении информации о необходимых заказах, сотрудниках и товарах за счет того, что вся работа будет осуществляться автоматически.

**1. Постановка задачи**

Задачей данной курсовой работы является построение базы данных и выполнение над ней заданных операций. База данных включает в себя таблицы, запросы, формы и отчеты. Данная работа предназначена для облегчения и координации работы менеджеров мебельной фабрики. При помощи программных средств пользователь может наглядно просмотреть интересующую его информацию. Сокращается трудоемкость; время поиска необходимого получателю товара, время поиска информации об интересующем заказе; производится автоматический учет выполненной работы, что облегчает и ускоряет процесс сбора и обработки информации. Для пользователя имеется возможность просмотра информации как о товарах и его заказчиках, так и о сотрудниках.

Пользователь, как секретарь фабрики, так и обычный студент или человек, интересующийся информацией о каком-то товаре, может вывести на экран список клиентов, сроки оплаты, адрес доставки и количество заказанного. Однако только сотрудник имеет возможность изменить информацию о товарах, внести информацию о новом поступившем товаре, предварительно введя пароль. Любой пользователь может просмотреть статистику, т.е. узнать количество товаров, имеющемся на складе. Значительную роль играет в программе функция поиска. Пользователь может найти любого сотрудника по его фамилии и имени или по его отделу. При этом программа выведет данные о сотруднике, который удовлетворяет введенным условиям.

Данная программа играет важную роль в ее практическом применении и освобождает пользователя от трудоемкой бумажной работы.

**2 Теоретическая часть**

**2.1 Общая характеристика СУБД Access 2002**

Программа Microsoft Access 2002 является реляционной СУБД, которая может функционировать под управлением операционных систем Windows 95/98, Windows NT 3.51 и выше, а также более новых версий Windows (в том числе Windows 2000 и Windows XP). Она является дальнейшим развитием предыдущей версии Microsoft Access 2000.

СУБД Access имеет стандартизированный интерфейс приложений Windows. Большинство действий по работе с различными элементами в среде Access можно выполнить с помощью следующих свойств: команд основного меню, кнопок панелей инструментов, команд контекстного меню и комбинаций клавиш. Отметим, что контекстное меню определенного элемента БД вызывается щелчком правой кнопкой мыши при размещении ее указателя над этимэлементом.

База данныхявляетсяосновным компонентом проекта приложения Ассеss и может включать в свой состав таблицы, формы, запросы, отчеты, макросы и модули.

Для работы с базами данных в Ассеss имеется стандартное окно, из которого можно вызвать любой ее объект просмотра, выполнения, разработки или) модификации. Пользователь для работы с базой данных может разработатьсвой интерфейс, основу которого обычно составляют формы.На формах размещаются различные элементы, такие как: поля таблиц, поля со списком кнопки, раскрывающиеся списки, выключатели, переключатели, флажки, рисунки, подчиненные формы и т.д.

За кнопками обычно закрепляют вызов функций. Функции обработки информации во время работы с базой данных задаются с помощью макросов или программ на Visual Basic for Application(VВА) – VВА-программ. Обычно в приложениях для работы с БД предусматривают автоматическое открытие главной кнопочной формы при открытии базы данных. В последующем работа пользователя происходит с помощью главной формы и при необходимости с помощью других форм и окон. Базу данных, имеющую интерфейс с пользователем, можно считать приложением, поскольку все описания базы данных (в том числе программные коды) интерпретируются системой Ассеss при работе пользователя.

Текущая открытая БД может взаимодействовать с внешними БД, которые используются как источник таблиц при импорте или присоединении, а так; как получатель при экспорте данных из текущей базы данных. С помощью запросов во внешней БД можно создавать таблицы. В качестве внешней БД может выступать любая база данных, поддерживающая протокол ОDВС, например, база данных SQL Server, расположенная на удаленном сервере, иля| одна из баз данных систем Paradox, dBASE, или Ассеss.

***Таблица*** представляет собой основную единицу хранения данных в ба Понятие таблицы в Ассеss полностью соответствует аналогичному понятию реляционной модели данных. В произвольной базе обычно имеете? совокупность связанных между собой таблиц. Между двумя таблицам можно устанавливать связи типа 1:1 и 1:М с помощью окна описания схемы данных. Основными операциями над таблицами являются: просмотр и обновление (ввод, модификация и удаление), сортировка, фильтрация я печать.

***Форма*** представляет собой объект базы данных Ассеss, в котором разработчик размещает элементы управления, принимающие действия пользователей или служащие для ввода, отображения и изменения данных в полях.

***Запрос*** представляет собой формализованное требование на отбор данных из таблиц или на выполнение определенных действий с данными. Запрос позволяет создать набор записей из данных, находящихся в разных таблицах, и использовать его как источник данных для формы или отчета.

В Ассеss можно создавать и выполнять следующие основные типы запросов: на выборку, обновление, удаление, или добавление данных. С помощью запросов можно также создавать новые таблицы, используя данные из одной или нескольких существующих таблиц.

Описание запроса можно выполнить с помощью бланка QBE или инструкции языка SQL.

***Макрос*** представляет последовательность макрокоманд встроенного языка Ассеss, задающих автоматическое выполнение некоторых операций, например: «Открыть Таблицу» (Ореn ТаЫе), «Закрыть» (С1оsе), «Найти Запись» (Find Record) и «Печать» (Print Out). В последующих версиях системы макросы используются для обеспечения совместимости с предыдущими версиями, и рекомендуется для процедур автоматизации использовать программный код VВА.1

**2.2 Создание основных элементов БД**

**2.2.1 Создание базы данных**

Access предоставляет 2 способа создания базы данных: создание пустой

БД (в последующем можно добавить нужные объекты) и создание непустой БД с помощью Мастера. Первый способ отличается большей гибкостью, так как требует отдельного определения каждого элемента БД. Второй способ ускоряет процесс создания БД и позволяет получить БД с образцами информации в таблицах. Он применим в случаях, когда пользователю подходит одна из предлагаемых типовых баз данных.

Независимо от способа создания базы данных можно в любое время легко ее изменить и расширить.

При создании БД возможны несколько вариантов диалога. К примеру, сразу после запуска Ассеss можно воспользоваться расположенной в правой части окна панелью задач, из которой создание новой БД инициируется щелчком мыши на ссылке «Новая база данных». В появившемся диалоговом окне Файл новой базы данных **(File New Database)** нужно выбрать папку размещения базы, ввести имя и нажать кнопку Создать (Create).

В результате появляется окно открытой БД, в котором выполняется работа по созданию требуемых элементов базы данных.

Если из начального окна пойти по ссылке «Создание с помощью шаблона» из раздела «Создание с помощью шаблона», то появится окно шаблонов с вкладками «Общие» и «Базы данных». Щелчком по соответствующей пиктограмме из вкладки «Общие» производится переход к созданию новой базе данных, для чего нужно будет задать папку и имя базы данных в окне Файл новой базы данных.

На вкладке Базы данных (Databases) можно выбрать как основу одну из многих готовых баз данных (здесь представлены различные темы, например, «Контакты», «Мероприятия», «Склад» и т.д.). Дальнейшее определение основных параметров базы данных выполняется с помощью Мастера баз данных.

При этом можно уточнить структуру одной или нескольких таблиц (в зависимости от типа выбранной базы), вид оформления экрана, свойства отчета для печати и т.д.

Созданная БД может быть автоматически открыта и снабжена справочным окном Access с изложением понятий по объектам базы данных. Открытые и созданные Мастером базы данных имеют главные кнопочные формы, позволяющие перейти к работе с данными (окно базы данных при этом свернуто).

К созданию БД можно также перейти путем нажатия кнопки Создать (Create) на панели инструментов или по команде меню Файл | Создать базу данных (File | New Database).

**2.2.2 Создание таблиц**

Перед созданием таблицы нужно открыть базу данных, в которой таблица будет находиться. Это можно сделать с помощью начального окна, а также нажатием кнопки Открыть базу данных (Open An Existing Database) панели инструментов или по команде меню Файл | Открыть (File | Open).

В открытой БД следует выбрать вкладку Таблицы (Tables) и нажать кнопку Создать (New). Начать создание таблицы можно также путем выбора в пункте Вставка (Insert) главного меню Access подпункта Таблица (Tables). В результате появляется окно.

Как следует из содержимого окна, возможны следующие пять вариантов создания таблиц:

* путем ввода данных в пустую таблицу, при сохранении данных в которой Access анализирует данные и автоматически присваивает соответствующий тип данных и формат каждому полю – *Режим таблицы (Datasheet View);*
* с помощью Конструктора – *Конструктор (Design View);*
* с помощью Мастера – *Мастер таблиц (Table Wizard);*
* из импортируемых таблиц – *Импорт таблиц (Import Tables);*
* путем создания таблиц, связанных с таблицами, находящимися во внешнем файле – *Связь с таблицами (Link Tables).*

Новые таблицы путем импортирования создаются пустыми или с данными. Это определяется параметрами (кнопка Параметры (Options)) в окне Импорт объектов (Import Objects), которое появляется при выборе исходной БД. При создании таблиц в режиме *Связь с таблицами* в окне БД появляются таблицы, которые находятся в другой базе данных. Фактически такие таблицы становятся разделяемым ресурсом, и их содержимое может изменяться из двух баз данных.

Независимо от способа создания изменение структуры таблицы можно выполнить в режиме *Конструктор,* предоставляющем наиболее гибкие и мощные возможности по определению параметров создаваемой таблицы.

Для перехода в режим *Конструктор* достаточно выделить вторую строку области выбора окна и щелкнуть на кнопке ОК. Появится окно, предназначенное для ввода характеристик создаваемых полей, в центре которого находится специальная форма (табличного вида) описания полей. Каждому полю в этой форме соответствует одна запись.

Имена полей вводятся в первой колонке, типы полей – во второй колонке, а необязательные комментарии – в третьей. Среди всего многообразия типов полей особый интерес представляет сравнительно недавно появившийся в современных СУБД новый тип полей – гиперссылки. При желании воспользоваться готовыми структурами таблиц можно прибегнуть к помощи Построителя таблиц, вызываемого при нажатии на кнопку панели инструментов. При этом появляется окно, позволяющее выбрать готовые описания полей.

**2.2.3 Связывание таблиц**

Для связывания таблиц БД нужно вызвать окно схемы данных.

После открытия основного окна БД окно схемы базы данных можно вызвать по команде Сервис | Схема данных (Tools | Relationships) основного меню или с помощью одноименной команды контекстного меню окна БД.

Построение схемы данных состоит в добавлении в нее таблиц и установлении связей между таблицами. Добавление таблиц в схему данных обычно производится с помощью соответствующей кнопки на панели инструментов.

В случае совпадения имен и типов полей у нескольких таблиц схемы данных и добавления в схему всех нужных таблиц, Access образует связи автоматически. Если образованные связи не устраивают пользователя, то их можно изменить.

Кнопка Объединение (Join Type) в правой части окна позволяет установить свойства объединения записей двух таблиц – как участвуют в связи остальные записи таблиц. В появившемся после нажатия этой кнопки окне для выбора предлагаются следующие варианты:

* объединение записей, в которых связанные поля таблиц совпадают;
* объединение всех записей из дополнительной таблицы и тех записей основной таблицы, в которых связанные поля совпадают;
* объединение всех записей из основной таблицы и тех записей дополнительной таблицы, в которых связанные ноля совпадают.

Если установление связей производится для непустых таблиц, то Access подвергает анализу всю имеющуюся информацию, и при наличии нарушений целостности сообщает об этом.

Для *изменения* или *удаления* имеющихся связей достаточно в схеме данных подвести указатель мыши к нужной связи, выделить ее щелчком и нажать правую кнопку. Появится контекстное меню, состоящее из двух пунктов, предлагающих соответственно изменить или удалить связь. При выборе первого пункта появляется окно. Удалить связь можно также, выделив ее мышью, а затем нажав клавишу <Del>.

Система Access при управлении связыванием таблиц отличается наглядностью отображения информации о связях. Надписи, стрелки и утолщения на стрелках связей характеризуют вид связи (1:1 или 1:М), вид объединения записей (стрелка указывается при объединении записей, в которых связанные поля совпадают), а также признак контроля целостности (утолщение окончаний связывающих линий).

**2.2.4 Создание запросов**

Перед *созданием* запроса нужно открыть базу данных, в которой он будет храниться. После этого следуем выбрать вкладку Запросы (Queries) и нажать кнопку Создать (New). Начать создание запроса можно также, выбрав в пункте Вставка (Insert) главного меню Access подпункт Запрос (Query). В результате появляется окно, в котором предлагается выбор варианта.

*Возможны следующие варианты создания запроса:*

* вручную с помощью конструктора – Конструктор (Design View);
* автоматическое создание простого запроса на выборку – Простой запрос (Simple Query Wizard);
* автоматическое создание перекрестного запроса для компактного представления данных в виде сводной (перекрестной) таблицы – Перекрестный запрос (Crosstab Query Wizard);
* автоматическое создание запроса на поиск записей с повторяющимися значениями полей – Повторяющиеся записи (Find Duplicates Query Wizard);
* автоматическое создание запроса на поиск записей в одной таблице, которые не имеют подчиненных записей в другой таблице – Записи без подчиненных (Find Unmatched Query Wizard).

Создаваемые запросы основаны на полях таблиц и / или запросов из базы данных. Все способы, кроме первого, реализуются с помощью Мастеров, упрощающих разработку запроса. Если созданный запрос не удовлетворяет требованиям, то можно воспользоваться Конструктором, либо создать заготовку запроса с помощью Мастера, которую затем подправить в режиме Конструктора.

Рассмотрим режим Конструктора как наиболее мощный и незаменимый при создании запросов, выходящих за рамки предлагаемых простейших вариантов.

Все эти действия выполняются в запросной форме, которую можно отнести к форме запроса на языке QBE. Запросная форма включает три основных элемента: заголовок (имя и тип запроса); область таблиц, их полей и связей между таблицами; бланк запроса по образцу.

Для *указания таблиц,* используемых в запросе, нужно поместить в запросную форму схемы этих таблиц или запросов и указать связи между ними. Включение объектов в запрос производится в окне Добавление таблицы (Show Table), которое вызывается автоматически (при создании запроса) или принудительно при работе с запросом путем нажатия кнопки панели инструментов.

При создании запроса Access по умолчанию предоставляет заготовку запроса на выборку. Изменить вид запроса можно с помощью пункта Запрос (Query) основного меню системы, где возможные виды запросов (выборка, обновление, добавление, удаление, создание таблицы, перекрестный) перечислены как подпункты меню.

Для создания запроса в виде инструкции языка SQL можно при наличии на экране запросной формы воспользоваться пунктом меню Запрос | Запрос SQL (Query | SQL), в котором выбрать нужный вид SQL-запроса (на объединение, к серверу или управляющий).

Условием отбора записей для вывода является выражение «>1000», означающее, что значения в поле «поле 15» должны быть больше 1000 (само поле не выводится). Выводимые записи сортируются по убыванию значений в поле «полеП».

Завершить создание запроса можно следующим образом:

* выбором пункта меню Файл | Сохранить (File | Save);
* выбором пункта меню Файл | Сохранить как / Экспорт (File | Save As/ Export);
* нажатием кнопки Сохранить (Save) на панели инструментов;
* путем закрытия окна создания запроса.1

**2.2.5 Создание форм**

Перед началом создания формы следует открыть базу данных, в которой она будет находиться. Собственно создание формы производится в окне открытой БД путем выбора вкладки Формы (Forms) и нажатия кнопки Создать (New). Начать создание формы можно также, выбрав в пункте Вставка (Insert) главного меню Access подпункт Форма (Form). При этом появляется окно Новая форма.

Запустив Мастер, пользователь может создать формы таких же видов, как и при выборе вариантов 3–5. Но здесь можно выбрать в качестве источников данных произвольное число запросов и / или таблиц, включив в форму нужные поля. Кроме того, можно изменить стиль фонового изображения. При необходимости произвести другие изменения макета формы, после завершения работы с Мастером, можно перейти в режим Конструктора.

Режим Конструктора является наиболее мощным, но и более трудоемким средством разработки форм. Чтобы сократить общее время разработки, целесообразно перед вызовом Конструктора воспользоваться одним из других способов создания формы, а Конструктор использовать для окончательного приведения формы к нужному виду. Рассмотрим основные возможности Конструктора.

*Вызов Конструктора форм* производится при создании формы или открытии существующей формы и переводе ее в режим Конструктора. В первом случае перейти в режим Конструктора форм можно в окне открытой БД из вкладки Формы (Forms) путем нажатия кнопки Создать (New) и нажатия ОК в появившемся окне (рис. 10.13), либо нажатием кнопки Конструктор (Design). Для перевода открытой формы в режим Конструктора достаточно щелкнуть мышью по кнопке на панели инструментов.

Окно разработки формы выглядит, как показано на рис. 10.14. Форма в режиме Конструктора в общем случае содержит следующие области: заголовок и примечание формы, верхний и нижний колонтитулы и область данных (в которой отображаются данные источников). Обязательной является область данных, остальные – необязательные. В области данных обычно размещаются поля таблиц.

Управлять отображением необязательных областей можно с помощью пункта меню Вид (View) при открытой форме в режиме Конструктора. Основные возможности Конструктора определяются составом инструментов Панели элементов (Toolbox), показанной на рис. 10.14. Панель элементов содержит следующие основные инструменты (перечислены слева направо и сверху вниз):

* Выбор объектов (Select Objects) – используется для выделения, изменения размера, перемещения и редактирования элементов управления.
* Мастера (Control Wizards) – кнопка включения / отключения автоматического вызова Мастеров создания элементов управления. Существуют Мастера по созданию группы, поля со списком, списка и командной кнопки.
* Надпись (Label) – предназначена для создания элемента управления, содержащего неизменяемый текст. По умолчанию большинство элементов управления содержит присоединенный текстовый элемент.
* Поле (Text Box) – используется для отображения, ввода или изменения данных, содержащихся в источнике записей, вывода результатов вычислений, а также приема данных, вводимых пользователем.
* Выключатель (Toggle Button), Переключатель (Option Button) и Флажок (Check Box) – различные по виду, но одинаковые по использованию элементы – предназначены для отображения логических значений. Выбор (включение) элемента приводит к вводу в соответствующее логическое поле значения «Да», «Истина» или «Вкл» (определяется значением свойства поля «Формат поля»). Повторный выбор элемента изменяет значение на противоположное: «Нет», «Ложь» или «Выкл». Эти элементы можно помещать в группу. Вид отображаемого в базовой таблице значения зависит от свойства Тип элемента управления (вкладка Подстановка свойств поля).
* Поле со списком (Combo Box) представляет составной элемент управления, объединяющий поле и раскрывающийся список. Для ввода значения в поле базовой таблицы, можно ввести значение в поле или выбрать значение из списка.
* Список (List Box) – предназначен для создания перечня (списка) возможных значений. Список можно создать, явно вводя данные, либо указав источник данных – таблицу или запрос. Список может содержать несколько столбцов, причем установка ширины при отображении любого из них делает этот столбец скрытым. Полю, с которым связан список, присваивается значение из множества значений одного любого столбца, в том числе скрытого.
* Кнопка (Command Button) – обычно используется для запуска закрепленного за ней макроса или программы на языке Visual Basic.
* Рисунок (Image) – предназначен для размещения в форме неизменяемого рисунка.
* Разрыв страницы (Page Break) – позволяет вставлять разрыв страницы в многостраничной форме.
* Набор вкладок (Tab Control) – позволяет создать в форме несколько вкладок, каждая из которых может содержать другие элементы управления.
* Подчиненная форма / отчет (Subform/Subreport) – предназначен для внедрения в форму некоторой другой (подчиненной) формы.
* Линия (Line) и Прямоугольник (Rectangle) – предназначены для создания соответствующих геометрических фигур. 1

**2.2.6 Создание отчетов**

Перед созданием отчета нужно открыть базу данных, в которой он будет находиться. Собственно создание отчета производится путем нажатия кнопки Создать (New) в окне открытой БД на вкладке Отчеты (Reports). Начать создание отчета можно также, задав в пункте Вставка (Insert) главного меню Access команду Отчет (Report).

Возможны следующие варианты создания отчета:

1. с помощью Конструктора – Конструктор (Design View);
2. с помощью Мастера – Мастер отчетов (Report Wizard);
3. автоматическое создание отчета стандартного вида, в котором каждая запись базового запроса или таблицы представлена в виде названия и значения поля – Автоотчет: в столбец (AutoreporfcColumnar);
4. автоматическое создание стандартного отчета, в котором данные записи базового запроса или таблицы выводятся в одной строке – Автоотчет: ленточный (AutoreportTabular);
5. создание отчета с диаграммой – Диаграмма (Chart Wizard);
6. создание отчета для печати почтовых наклеек – Почтовые наклейки (Label Wizard).

Проще всего создать отчет по вариантам 3 и 4, требующим наименьшее число параметров. В этих случаях получаются простейшие отчеты, включающие все поля источника данных (таблицы или запроса). После создания отчет доступен для просмотра или печати. В режиме Конструктора предоставляются более мощные средства, требующие больше знаний и времени для разработки отчета.

Техника работы с Конструктором отчетов мало чем отличается от работы с Конструктором форм. В частности, при этом используется такая же панель элементов. Остановимся на важнейших особенностях разработки отчетов в сравнении с формами.

При работе над отчетом используются те же области, что и при создании форм: области заголовка и примечания, области верхнего и нижнего колонтитулов, а также область данных. Кроме того, в отчеты можно включать области группировки записей. \*

Заголовок отчета и примечание выводятся один раз: в начале и конце отчета соответственно. Верхний/нижний колонтитулы помещаются в начало/ конец каждой страницы отчета. Содержимое области данных выводится один раз для каждой записи исходной таблицы или запроса. Если пользователь задал группировку записей отчета, то по каждому полю, по которому проводится группировка данных, Access формирует заголовок и примечание группы. Для создания в отчете области группировки нужно при открытом в режиме Конструктора отчете выбрать пункт меню Вид / Сортировка и группировка (View/Sorting And Grouping) или нажать кнопку на панели инструментов.

Существенное различие между отчетом и формой заключается в том, что отчеты предназначены исключительно для вывода данных на печать. Поэтому в них можно отказаться от использования (доступных для включения в отчет) управляющих элементов для ввода данных: списков, полей со списком, переключателей и т.п.

Отчеты могут находиться в двух режимах: Конструктора или *Просмотра.* Просмотреть готовый отчет можно после выделения нужного отчета при нажатии кнопки Просмотр (Preview) в окне БД, выборе пункта меню Файл/ Предварительный просмотр (File/Preview) или нажатии кнопки на панели инструментов.

**2.2.7 Парольная защита баз данных**

Парольная защита является простым и часто достаточным средством обеспечения защиты БД от открытия несанкционированными пользователями. Используемый при этом пароль называют паролем базы данных.

Зная пароль БД, любой пользователь сможет ее открыть и использовать, а также выполнить все необходимые операции с ней. Установка пароля может быть запрещена в случае, если для БД установлена защита на уровне пользователя и наложен запрет на парольную защиту.

*Процедура установки* парольной защиты БД включает следующие шаги.

1. Закрытие базы данных, если она открыта. Если база данных используется в сети, следует проверить, что все остальные пользователи тоже закрыли ее.

1. Выбор в меню команды Файл | Открыть (File | Open). Появится диалоговое окно Открытие файла базы данных (Open).
2. Установка с помощью соответствующего списка в правой нижней части окна открытия файла режима монопольного доступа (Монопольно (Exclusive)) и открытие базы данных.
3. Выбор команды Сервис | Защита | Задать пароль базы данных (Tools | Security | Set Database Password).
4. Ввод пароля в поле Пароль (Password) с учетом регистра клавиатуры.

**3. Проектная часть**

**3.1 Назначение предметной области**

В информационных системах, создаваемых на основе СУБД, способы организации данных и методы доступа к ним перестали играть решающую роль, поскольку оказались скрытыми внутри СУБД. Массовый, так называемый *конечный пользователь,* как правило, имеет дело только с внешним интерфейсом, поддерживаемым СУБД.

Эти преимущества, как уже понятно, не могут быть реализованы путем механического объединения данных в БД. Предполагается, что в системе обязательно существует специальное должностное лицо (группа лиц) – *администратор базы данных (ЛБД),* который несет ответственность за проектирование и общее управление базой данных. АБД определяет информационное содержание БД. С этой целью он идентифицирует объекты БД и моделирует базу, используя язык описания данных. Получаемая модель служит в дальнейшем справочным документом для администраторов приложений и пользователей. Администратор решает также все вопросы, связанные с размещением БД в памяти, выбором стратегии и ограничений доступа к данным. В функции АБД входят также организация загрузки, ведения и восстановления БД и многие другие действия, которые не могут быть полностью формализованы и автоматизированы.

Как очевидно, конечной целью проектирования является построение конкретной базы данных **мебельной фабрики**, иной степени воплощающей представление проектировщика о предметной области **и** задачах, решаемых пользователями с помощью созданной базы. Рассматривая базу данных как конкретную реализацию модели, мы по существу устанавливаем порядок процесса, отделяя этап определения принципов (то, какой база *должна* быть) от этапа воплощения этих принципов при реализации базы данных в конкретной среде СУБД, ОС и языках программирования. И как показывает практика, между реализациями баз данных и принципами их по строения всегда есть расхождения. Различия являются следствием разных причин, но чаще всего это явный или неявный отказ от некоторых принципиальных ограничений, налагаемых, например, моделью данных или базовыми (встроенными) алгоритмами обработки в пользу частного решения, которое, по мнению проектировщика, будет более эффективно, например, для понимания или обработки данных.

Проектирование базы данных – это упорядоченный формализованный процесс создания *системы взаимосвязанных описаний,* т.е. таких моделей предметной области, которые связывают (фиксируют) хранимые в базе данные с объектами предметной области, описываемыми этими данными. Прикладное назначение таких описаний состоит в том, чтобы пользователь, практически не имеющий представления об организации данных в БД (физическом размещении в памяти данных и механизмах их поиска), обращая запрос к базе, имел бы практическую возможность получить адекватную информацию о состоянии объекта предметной области.

Такая человекоориентированная модель практически полностью независима от физических параметров среды хранения данных, которой может быть как память человека, так и ЭВМ. Поэтому инфологическая модель не изменяется до тех пор, пока какие-то изменения в реальном мире (той его части, которая отнесена к предметной области) не потребуют изменения в модели соответствующего фрагмента описания, чтобы эта модель продолжала адекватно отражать предметную область.

**3.3 Создание таблиц БД**

Таблицы составляют основу базы данных – именно в них хранятся все данные. Таблицы должны быть тщательно спланированы. Прежде всего, должна быть спланирована структура каждой таблицы. Структура таблиц определяется содержанием тех выходных форм и отчетов, которые должны быть затем получены. При планировании таблиц необходимо избежать дублирования информации в разных таблицах.

В новой базы данных «Мебельная фабрика» необходимо создать следующие связанные таблицы со связанными полями:

**Таблица Выдачи:**

**–**Город – Текстовое поле, 50 символов;

– Телефон доставки – Числовое поле, длинное целое;

– Код расходов – Числовое поле, длинное целое;

– Стоимость доставки – Денежное поле, денежный формат;

– Вес упаковки – Числовое поле, длинное целое;

– индекс доставки – Числовое поле, длинное целое;

– Код фондов – Ключевое, Числовое поле, длинное целое.

**Таблица Договоры:**

– Начальная дата – поле Дата/время, Краткий формат даты, маска ввода: 99.99.00; 0;

– Конечная дата – поле Дата/время, Краткий формат даты, маска ввода: 99.99.00; 0;

– Имеется мест – Числовое поле, целое;

– Код сотрудника – Числовое поле, длинное целое;

– Код договора – Ключевое, Числовое поле, длинное целое;

– Код типа договора – Числовое поле, длинное целое.

**Таблица Записи работ:**

– Код фондов – Ключевое, Числовое поле, длинное целое;

– Затрачено часов – Числовое поле, одинарное с плавающей точкой;

– Оценочная цена – Денежное поле, денежный формат;

– Реальная цена – Денежное поле, денежный формат;

– Оплачено – Логическое поле, формат Да / Нет;

– ЧасовСПослОбсл – Числовое поле, целое.

**Таблица Клиенты:**

– Код клиента – Ключевое, Числовое поле, длинное целое;

– Код проекта – Числовое поле, длинное целое;

– Адрес – Текстовое поле, 50 символов;

– Город – Текстовое поле, 20 символов;

– Номер телефона – Числовое поле, длинное целое;

– Фамилия клиента – Текстовое поле, 20 символов;

– Последовательность – Числовое поле, длинное целое.

**Таблица Оплата:**

– Город – Текстовое поле, 50 символов;

– Фамилия клиента – Текстовое поле, 50 символов;

– Код заказа – Числовое поле, длинное целое;

– Код проекта – Ключевое, Числовое поле, длинное целое;

– Сумма оплаты – Денежное поле, денежный формат;

– Номер чека – Числовое поле, длинное целое.

**Таблица Расходы:**

– Код клиента – Числовое поле, длинное целое;

– Код расходов – Ключевое, Числовое поле, длинное целое;

– Сумма затрат – Денежное поле, денежный формат;

– Сумма задатка – Денежное поле, денежный формат;

– Дата покупки – поле Дата/время, краткий формат даты, маска ввода – 99.99.00; 0.

**Таблица Резервирование:**

– Код клиента – Числовое поле, длинное целое;

– Код договора – Числовое поле, длинное целое;

– Зарезервировано – Числовое поле, длинное целое;

– Остаток – Денежное поле, денежный формат;

– Выплаченная сумма – Ключевое, Денежное поле, денежный формат;

**Таблица Сведения о заказе:**

– Код товара **–** Числовое поле, длинное целое;

– Код заказа – Ключевое,Числовое поле, длинное целое;

– Количество **–** Числовое поле, двойное с плавающей точкой;

– Налог **–** Денежное поле, основной формат.

**Таблица Сотрудники:**

**–**Код сотрудника – Ключевое, Числовое поле, длинное целое;

– Код фондов – Числовое поле, основной формат;

– Фамилия – Текстовое поле, 50 символов;

– Домашний телефон – Числовое поле, длинное целое;

– Код отдела – Числовое поле, длинное целое;

– Номер паспорта – Числовое поле, длинное целое.

**Таблица Товары:**

– Код товара – Ключевое, Числовое поле, длинное целое;

– Наименование товара – Текстовое поле, 50 символов;

– На складе – Числовое поле, длинное целое;

– Заказано – Числовое поле, длинное целое;

– Минимальный заказ – Числовое поле, длинное целое;

– Время задержки – Числовое поле, длинное целое.

Создание таблиц на примере конструирования таблиц **Выдачи и Товары:**

В новой открытой базе данных в перечне объектов выбирается объект Таблицы и нажимается вкладка Создать. В появившемся окне выбирается Конструктор.

В режиме *Конструктора* в столбец *Имя поля* вводятся названия необходимых полей и определяются при этом типы данных в столбце *Тип данных*. Поле *Код фондов* делается ключевым, щелкнув правой кнопкой мыши и выбрав соответствующий пункт.

При активной вкладке *Общие* определяются необходимые параметры для каждого поля.

После выхода из режима *Конструктора*, присваивается созданной таблице имя **Выдачи**.

Возможен просмотр её в режиме *Таблицы* и при необходимости редактирование ширины столбцов.

Создание таблицы **Товары:**

Для этого выбирается объект *Таблицы* и нажимается вкладка *Создать*. В открывшемся окне ищется вариант создания с помощью *Мастера таблиц.*

При работе мастера выбирается в качестве образца таблица *Товары* и, дважды щелкая по образцам полей, включаются в создаваемую таблицу поля: *Код товара, Наименование товара, На складе, Заказано, Минимальный заказ и Срок*. Затем щелчок по кнопке *Далее.*

В окне *Создание таблиц* присваивается имя таблице **Товары** и происходит сохранение. Включенный флажок *Microsoft Access автоматически определяет ключ*. Щелчок по кнопке *Далее.*

Далеенажимается кнопка *Готово.* Переключение таблицы в режим *Конструктора* происходит при помощи щелчка по кнопке *Вид* инструментальной панели. Проводится необходимая корректировка *типов данных.*

В режиме *Таблицы* редактируется ширина столбцов.

Аналогично конструируются другие таблицы при помощи Конструктора и Мастера таблиц, редактируются и добавляются поля, изменяются типы данных.

**3.5 Создание схемы данных**

Для создания отношений необходимо указать поля в двух таблицах, которые содержат одни и те же данные. Обычно такое поле в одной из таблиц (главной) является ключевым. Имена связывающих полей могут отличаться, но *типы и свойства должны совпадать*.

Для этого нужно щелкнуть по кнопке инструментальной панели «*Схема данных*» или выполнить команду *Сервис / Схема данных.*

В окне *Добавление таблицы* последовательно выбираются все созданные таблицы, затем щелчок по кнопке *Добавить*.

Добавив последнюю таблицу, нужно щелкнуть по кнопке *Закрыть*.

В окне *Схема данных* необходимо установить связь между таблицей **Товары** и таблицей **Сведения о заказе**. Для этого:

Выделяется щелчком поле *Код товара* в таблице **Товары** и, удерживая

нажатой левую клавишу мыши, нужно переместить указатель на поле с тем же названием в таблице **Сведения о заказе**. В появившемся окне необходимо поставить галочку для обеспечения целостности данных и выполнить команду **Создать.** Появится связь между таблицами типа *один-ко-многим.*

Аналогичным образом связываются таблицы: Сведения о заказе – Оплата (по полю Код заказа), Оплата – Клиенты (по полю Код проекта), Клиенты – Расходы (по полю Код клиента), Расходы – Выдачи (по полю Код расходов, Выдачи – Записи работ (по полю Код фондов), Записи работ – Сотрудники (по полю Код фондов), Сотрудники – Договоры (по полю Код сотрудника), Договоры – резервирование (по полю Код договора). *Целостность данных* обеспечивается с помощью поставленного флажка. Таким образом, получается схема данных (Приложение 1, рис. 2).

**3.6 Разработка необходимых запросов, форм, отчетов.**

**3.6.1 Запросы**

Необходимо создать следующие запросы:

**1. Запрос по товарам –** простой запрос на выборку сведений из базы данных, включающий следующие поля:

**–**Код товара;

**–**Фамилия клиента;

**–**Код заказа;

**–**Сумма оплаты;

**–**Город;

**–**Количество.

Для этого нужно щелкнуть в окне *База данных* по объекту *Запросы*. В окне *Запросы* щелчок по кнопке *Создать*. В окне *Новый запрос* выбирается *Конструктор* и нажимается кнопка Ok. В появившемся окне *Добавить таблицу* добавляются таблицы **Сведения о заказе** и **Оплата**, затем окно закрывается.

Перетаскиваются названия полей из окон таблиц в бланк запроса, расположенный под схемой данных, соблюдается их заданный порядок.

Можно просмотреть выборку, выполнив команду *Вид / Режим таблицы* или щелкнув по кнопке инструментальной панели *Вид*. А в случае ошибок выборки, можно вернуться в режим*Конструктора запросов* и, если выборка содержит ошибки, и откорректировать запрос

При закрытии запроса в появившемся окне подтверждается необходимость его сохранения. Присваивается запросу имя **Запрос по товарам**.

**2. Выборка по оплате** включает следующие поля: Код фондов, Затрачено часов, Код сотрудника, Код договора, Оплачено из таблиц Выдачи, Записи работ, Сотрудники и Договоры. В поле *Оплачено* в строке *Условие отбора* указывается значение Да и убирается галочка в строке Вывод на экран. Запрос сохраняется (Приложение 2, рис. 2).

3. **Запрос на итоговую оплату с налогом** включает следующие поля: Наименование товара, Количество, Налог, Сумма оплаты, Фамилия клиента.

Далее нужно щелкнуть мышью в строке *Поле* первого свободного столбца бланка, а затем – по кнопке инструментальной панели *Построить*. Будет вызвано окно *Построителя выражений*

Вводится в окно построителя выражение: %**:**

**[Сумма оплаты]\*[Налог],**

где % **(**сумма налога) **–** заголовок столбца с вычисляемым полем,

**[Сумма оплаты]\*[Налог]** – поля таблиц **Оплата и Сведения о заказе.**

Используются для записи выражения инструменты *Построителя выражений*. После ввода выражения щелчок по кнопке Ok.

Щелчком мыши в ячейке поля следующего свободного столбца переходим в окно *Построителя выражений*.

Вводится в окне *Построитель выражений* формула:

**Итоговая сумма: [Сумма Оплаты]+[%],**

где **Итоговая сумма** – заголовок столбца с вычисляемым полем,

**[Сумма Оплаты] –** поле таблицы **Оплата.**

Для редактирования выражения в вычисляемом поле запроса можно использовать окно *Область ввода*, которое вызывается комбинацией клавиш **Shift+F2,** если ячейка с полем активизирована. То же самое можно сделать, выбрав пункт *Масштаб* в контекстном меню, вызываемом щелчком правой клавиши мыши по названию поля.

Включаются флажки вывода на экран вычисляемых полей. Запускается запрос. После отладки запрос сохраняется под именем **Запрос на итоговую оплату с налогом.**

**4. Запрос по городам** включает следующие поля: Код клиента, Код проекта, Адрес, Город, Номер телефона, Фамилия клиента, Последовательность из таблицы Клиенты. В поле *Город* в строке *Условие отбора* указывается [Введите город]. Запрос сохраняется. С помощью этого запроса просматривается информация заказов по городам в отдельности.

**6. Запрос по конечной дате** включает следующие поля: Начальная дата, Конечная дата, Имеется мест, Код сотрудника, Код договора, Код типа договора их таблицы Договоры. В строку Условие отбора поля Начальная дата необходимо установить курсор и щелкнуть по кнопке инструментальной панели Построить. В окне Построитель выражений ввести имя DMax. Для этого нужно раскрыть список Функции в левом подокне построителя и выбрать Встроенные функции. Далее в среднем подокне выбрать категорию функций По подмножеству, а в правом подокне – функцию DMax. Соответствующая функция появится в главном окне построителя с обозначением ее аргументов.

Удалите первый аргумент функции и вставьте вместо него имя поля *Начальная дата* либо непосредственным вводом с клавиатуры, либо путем раскрытия щелчком мышки списка *Запрос по конечной дате* в левом подокне, а затем выбрав *Начальная дата.* В последнем случае нужно удалить неиспользуемую часть строки *«Выражение».* Далее введите значения остальных аргументов функции, таким образом, чтобы функция приняла следующий окончательный вид:

**DMax(«[НачальнаяДата]»; «Договоры»)**

Для записи функции в ячейку таблицы описания запроса щелкните по кнопке Ok.

Закройте запрос сохранением макета и присвойте ему имя *Запрос по конечной дате*. Просмотрите результат выполнения запроса двойным щелчком мыши по его имени в окне базы данных (Приложение 3, рис 3).

**7. Подсчет зарезервированного клиентами** включает поля Код клиента и Зарезервировано всего из таблицы Резервирование. Щелкнув по кнопке Групповые операции в поле Зарезервировано всего выберем функцию count, которая обеспечивает подсчет числа записей в запросе (Приложение 4, рис. 1)

**8. Подсчет затрат клиентов по городам (перекр.).**

С помощью *Конструктора* создается новый запрос с использованием таблиц **Клиенты, Расходы и Выдачи.**

Вводится в 1-ый столбец бланка запроса поле *Фамилия клиента*, во 2-ой столбец – поле *Город*, в 3-ий столбец – поле *Сумма затрат*.

Щелчком по кнопке инструментальной панели *Тип запроса* выбирается из списка *Перекрестный*.

Выбираются значения в строке бланка *Перекрестная таблица* при развернутом списке в ячейках:

Для 1-го столбца *Заголовок строк*, для 2-го столбца – *Заголовок столбцов*, для 3-го столбца – *Значение*

Выбирается функция **Sum** для групповой операции в 3-м столбце.

Щелкнув по кнопке *Запуск*, можно просмотреть перекрестную выборку. Запрос сохраняется под именем **Подсчет затрат клиентов по городам (перекр.).**

Для создания итогового столбца нужно перейти в режим конструктора и вставить в бланк запроса еще одно поле *Сумма затрат*. Введите перед названием этого поля название столбца **Итого:** В строке *Групповая операция* выберите count, а в строке *Перекрестная таблица* – **Заголовки строк**.

**9. Подсчет оплаченного** включает следующие поля: Код клиента, Город, Количество оплаченного, Оплачено из таблиц Расходы, Выдачи и записи работ. В строке групповые операции для поля **Количество оплаченного** выбрать *count* (что обеспечит подсчет числа записей в запросе)*,* для поля Оплачено отменить вывод на экран и в строку Условие отбора ввести значение **Да**.

**10. Подсчет расходов по городам** включает следующие поля: Город и Код расходов таблицы Выдачи. В строке групповые операции для поля Код расходов выбрать функцию count и перед названием поля Код расходов ввести Всего, что позволит сосчитать количество записей в запросе.

**11.** Необходимо создать **запрос на обновление** базы данных, при выполнении которого произойдет увеличение оплаты товаров в 2 раза. Для этого создается с помощью конструктора новый запрос, где в качестве исходной таблицы принимают «Оплату».

Далее указывается тип запроса, выполняя команду ЗАПРОС/ОБНОВЛЕНИЕ.

Необходимо в поле «Сумма оплаты» в строку «Обновление» ввести с помощью «Построителя выражений» выражение: [Сумма оплаты]\*2. Запрос выполняется, нажав на панели инструментов кнопку «Запуск» и сохраняется под именем «**Запрос на обновление**»

**12**. Требуется составить **запрос на добавление** записей из одной таблицы в другую. На первом этапе создается таблица «Поступление новых товаров».

На втором этапе создается запрос с помощью конструктора. В окне *Добавление таблицы* выбирается ранее созданная таблица, нажимается кнопка Добавить и добавляются все поля таблицы. Указывается тип запроса, выполняя команду ЗАПРОС/ДОБАВЛЕНИЕ.

В окне *Добавление* введём имя таблицы «Товары», к которой должны добавляться строки исходной таблицы. Затем запрос сохраняется с именем **«Запрос на добавление».**

**3.6.2 Формы**

*Требуется создать формы:*

**1.** Необходимо создать простую форму по таблице Сотрудники.

Для этого нужно установите в окне базы данных вкладку «*Формы*» и щелкните по кнопке «*Создать*».

В окне «*Новая форма*» выбрать «**Мастер форм**» и указать в качестве источника данных таблицу *Сотрудники.*

В окне «*Создание форм*» выбрать все поля таблицы и нажмите кнопку «*Далее*».

Далее нужно выбрать внешний вид формы «*в один столбец*», нажать кнопку «*Далее*», выбрать стиль оформления «*Стандартный»* и нажать кнопку «*Далее*». Присвоив имя форме: «*Сотрудники – простая*», и нажать кнопку «*Готово*». Информация таблицы просматривается с помощью созданной формы.

Для **редактирования формы** нужно установить режим «*Конструктор*», с помощью вертикальной линейки создать достаточную область для ввода заголовка формы и, выбрав на **панели элементов** «*Надпись*», отметить границы надписи в области заголовка, где вводится текст в рамку надписи. Выделив рамку надписи, регулируется ее размер и местоположение.

Форма просматривается, если щелкнуть по кнопке инструментальной панели «*Вид*», установив режим «Форма».

Закрыв форму, необходимо сохранить изменения (Приложение 6, рис. 1).

**2.** Требуется создать **простую форму по связанным таблицам** для просмотра Списка клиентов, индексом доставки товаров с суммой затрат. Для создания формы необходимо использовать 2 таблицы: «*Клиенты*» и «*Расходы*».

Для этого в окне базы данных на вкладке «*Формы*» нужно щелкнуть по кнопке «*Создать*», в окне «**Новая форма**» выбрать «**Мастер форм**» и указать в качестве источника данных таблицу «*Группа*».

В окне «**Создание форм**» выбирается из таблицы «Клиенты» поле «Код клиента», из таблицы «*Расходы*» – поле «Сумма затрат» и «Индекс доставки» и нажимается кнопка «*Далее*».

На следующем шаге выбирается вид представления данных: *Клиенты* и нажимается кнопка «*Далее*». После выбирается внешний вид формы – «**ленточный»** и стиль – «**официальный**» и нажимается кнопка «*Готово*».

Установив режим «Конструктор», редактируется заголовок формы путем вызова «**Панель элементов**», щелкнув по соответствующей кнопке панели инструментов. Поместив панель элементов горизонтально над панелью задач Windows, нужно увеличить высоту области заголовка формы. Переместив контуры названий столбцов вниз, на освободившемся месте нужно вставить надпись для названия формы. Для перемещения надписи выделяется контур и перетаскивается мышкой расположенный в левом верхнем углу контура маркер. Вводится заголовок формы «*Список кодов клиента*».

Закрыв форму, присваивается имя «*Ленточная форма Кодов клиента*».

**3.** Необходимо создать форму со списком, которая позволяет просматривать данные каждого клиента. Форма должна быть снабжена списком клиентов, в котором для каждого клиента указывается его код, город, адрес и телефон. Для создания формы следует использовать таблицу «Клиенты» и специально подготовленный запрос «Расходы-Выдачи».

Запрос создается на основании таблиц Расходы и Выдачи, включающий поля Сумма затрат, Индекс доставки, Вес упаковки и Цена доставки.

На следующем шаге создается с помощью «Мастера форм» простая форма в один столбец по таблице «Клиенты», включив в нее поля: Фамилия клиента, Код клиента, Адрес, № телефона, Город.

В окне «Создание форм» форме присваивается имя «Форма Клиенты с полем для списка» и включается флажок «Изменить макет формы» для непосредственного перехода в режим конструктора. Для ознакомления с приемами конструирования форм нужно включить флажок «Вывести справку по работе с формой» и щелкнуть по кнопке «Готово».

В режиме конструктора, переместив вниз горизонтальную линию, разделяющую области данных и примечания, нужно щелкнуть на панели элементов по кнопке «Мастера» и выбрать на панели элементов конструирования формы элемент «Поле со списком». Указывается мышью местоположение окна для списка в области данных формы, где высота окна должна быть равна высоте кнопки, раскрывающей список. В окне Создание полей со списком необходимо оставить включенным переключатель объект «поле со списком», который будет использовать значения из запроса. В следующем окне включив переключатель Показать запросы, выбирается ранее созданный запрос Расходы-Выдачи и на следующем шаге включается в список все поля запроса. Далее нужно выбрать сортировку списка по возрастанию кодов клиента, а затем, перетаскивая границы заголовков столбцов, определить их ширину и щелкнуть по кнопке «Готово».

Необходимо добавить к списку заголовки строк. Для этого нужно перейти в режим конструктора и открыть окно свойств поля со списком, щелкнув правой клавишей мыши по маркеру в левом верхнем углу контура поля. В контекстном меню выбирается пункт Свойства и на вкладке Макет указывается для свойства Заглавия столбцов значение Да. Форма сохраняется с именем «Форма Клиенты с полем для списка».

**4.** Нужно создать составную форму, которая должна содержать главную форму и одну подчиненную форму табличного вида. Форма предназначена для вывода информации о городах.

Для этого создается простой запрос на выборку «Подсчет оплаченного», включив в него поля: Код клиента, Город и Количество оплаченного.

Также требуется создать табличную автоформу с именем «Подчиненная» со списком городов из таблицы «Клиенты». С помощью «Мастера форм» создается главная форма на основе запроса «Подсчет оплаченного», включив в нее все поля. Главная форма создается в один столбец стандартным стилем, используя тип представления данных по городу. Включив флажок «Изменить макет формы», задается имя формы: «Составная форма по городам».

В режиме конструктора нужно увеличить область для размещения данных и ввести элемент «Надпись» с текстом «Город». Далее размещается под надписью элемент «Подчиненная форма / отчет» и в появившемся окне мастера подчиненных форм выбирается из числа имеющихся форм – форму Подчиненная и щелкается по кнопке Далее. В окне определения полей связи сохранив включенным переключатель Выбор из списка, выбирается строка, предложенная мастером. В последнем окне мастера, не изменяя имени подчиненной формы, нужно щелкнуть по кнопке Готово. Форма сохраняется.

**5.** Требуется создать составную форму для просмотра товаров с пользовательскими кнопками для перехода по записям.

Для этого нужно создать на основании таблиц Оплата и Сведения о заказе запрос на выборку с итогами, содержащий поля: Код заказа, Количество, Сумма оплаты, Фамилия клиента, Город и Код товара. Новому запросу присваивается имя Запрос по товарам. Далее создается с помощью конструктора главная форма. Для этого в окне базы данных нужно щелкнуть по кнопке Создать и выбрать в качестве источника данных *Запрос по товарам*.

В окне конструктора форм, находясь в области данных, размещаются поля, перетаскивая их названия из окна со списком полей запроса и выравнивается местоположение и размеры полей и подписей к ним, перемещая маркеры их контуров. На следующем этапе нужно включить мастера конструирования форм, щелкнув по кнопке Мастер на инструментальной панели и, щелкнув по кнопке Подчиненная форма / отчет, разместить контур подчиненной формы в нижней части области построения. В окне мастера выбирается из числа имеющихся форм – Подчиненную, разработанную на предыдущем этапе.

Отключите в главной форме линейки (полосы) прокрутки, кнопки перехода по записям и область выделения. Для этого откройте окно свойств формы, щелкнув по кнопке Свойства на инструментальной панели конструктора форм. Установите вкладку Макет и измените на ней значения соответствующих свойств.

Для того чтобы создать в форме кнопку со стрелкой для перехода к следующему товару, нужно включить мастера панели элементов, щелкнув по соответствующей кнопке этой панели. Далее необходимо выбрать на панели элемент Кнопка и очертить контур будущей кнопки при нажатой левой клавише мыши. После отпускания клавиши мыши появится окно Создание кнопок .



В окне Создание кнопок выбирается категория Переходы по записям, а в нем – действие Следующая запись. В следующем окне включается флажок Показать все рисунки и выбирается рисунок. Не изменяя имя, предложенное Access, нужно щелкнуть по кнопке Готово. В режиме формы проверяется правильность работы кнопки. При ее нажатии должен происходить вывод сведений о следующем товаре.

Аналогичным образом создается кнопка для перехода к предыдущей записи. Проверив работу кнопок со стрелками в режиме формы, макет формы сохраняется.

**6.** Необходимо создать форму с вкладками для просмотра списка городов. На каждой вкладке должен располагаться список информации только одного города. На ярлычках вкладок должны отображаться названия городов.

Для этого с помощью мастера нужно создать четыре простых запроса на Воронеж, Киев, Киров и Москву для выборки из таблицы «Выдачи». В запрос необходимо включить все поля таблицы.

С помощью мастера создаются четыре табличные формы: форма на Воронеж, форма на Киев, форма на Киров, форма на Москву на основании вышеперечисленных запросов с использованием стандартного стиля.

С помощью конструктора создается новая форма (главная), не используя в качестве источника таблицу или запрос. В окне конструктора нужно щелкнуть по элементу «Вкладка» и очертить мышкой прямоугольный контур в области данных. Затем заменяются названия ярлычков вкладок. Для этого необходимо щелкнуть по ярлычку правой клавишей мыши и выбрать в контекстном меню пункт «Свойства», а затем ввести в окне свойств элемента новое имя на вкладке «Другие».



Выделив первую вкладку, нужно разместить на ней подчиненную форму «Форма на Воронеж» и удалить связанную с подчиненной формой подпись формы. Выделив вторую вкладку, нужно разместить на ней подчиненную форму «Форма на Киев» и удалить связанную с подчиненной формой подпись формы. Аналогично нужно сделать с другими формами. Форма сохраняется с именем «Форма с вкладками городов» (Приложение 8, рис. 2).

**3.6.3 Отчеты**

Для удобного просмотра сведений и предоставления каких-либо итогов используются *отчеты.*

Требуется создать **табличный отчет**, используя таблицу «Клиенты». Отчет должен включать следующие столбцы: Код клиента, Фамилия клиента, Город и Адрес.

Для этого в окне базы данных нужно выбрать объект «*Отчеты*» и щелкнуть по кнопке «*Создать*». В окне «*Новый отчет*» выбрать «**Мастер отчетов**» и в качестве источника данных – таблицу «Клиенты» и щелкнуть по кнопке «Ok».

В окне «*Создание отчетов*» необходимо выполнить следующие действия:

– выбрать поля запроса нужной последовательности;

– выбрать вид представления данных *по клиентам*;

– добавить уровень группировки *по адресу*;

– выбрать макет «*ступенчатый*»;

– выбрать стиль «*строгий*»;

– задать имя отчета «*Отчет по клиентам*»

и нажать кнопку «Готово».

Далее нужно перейти в режим конструктора, ввести текст в нижний колонтитул: «Ведомость составлена:» и сохранить отчет.

Необходимо создать **отчет с группировкой данных и подведением итогов**, используя запрос на выборку, с именем «**Запрос на итоговую оплату с налогом**». Необходимо в отчете подсчитать общую суммарную оплату с налогом по каждому товару и по каждому клиенту. Отчет должен содержать поля: Наименование товара, Фамилия клиента, Количество товара, Налог и Итоговая сумма.

Для этого нужно создать с помощью **мастера** новый отчет на основании указанного выше запроса, включив в запрос необходимые поля.

В окне создания отчета выполняются следующие шаги:

* выбирается тип представления данных – по полю «*Товар*»;
* добавляются уровни группировки *по Наименованию товара и Фамилии клиента;*
* настраивается режим сортировки *по количеству* и, щелкнув по кнопке «*Итоги*», нужно включить флажки «Sum» и «*Показать данные и итоги*»;
* выбирается макет отчета «*структура 1*»;
* выбирается *сжатый стиль* и нажимается кнопка «Готово».

Отчет сохраняется под именем «Запрос на итоговую оплату с налогом».

Необходимо заменить слово «**sum**» в итоговых строках на слово «**Всего по клиентам:**» для итогов по клиентам и «**Всего по товарам**:» для итогов по товарам. Отключаются все строки отчета с итогами по количеству записей и устанавливается одинаковый фиксированный формат числовых данных с двумя знаками после запятой для всех сумм оплаты.

**Заключение**

Созданная база данных «Мебельная фабрика» существенно упростит работу с поиском нужной информации. Во-первых, в базе содержится перечень товаров, имеющихся на складе, их минимальное количество, которое должно быть всегда и количество заказанного товара. Таким образом, сотрудник фирмы может следить за тем, чтобы количество товара на складе было не меньше, чем заказано. По коду товара можно узнать заказчика, исполнителя, сумму оплаты, срок изготовления и адрес доставки. При поступлении нового товара сотрудник без труда добавит его к уже имеющемуся.

Также имеется информация о клиентах фабрики: их адрес, № телефона, фамилия и город, т.е. имеется возможность быстро найти клиента, заказывающего определенный товар, и связаться с ним по мере необходимости. Также удобно следить за датой доставки, чтобы товары поставлялись без задержек.

Имеется возможность просмотреть информацию заказов по городам для удобства доставки.

Сотрудники фирмы следят за оплатой товара по срокам, вычисляют стоимость в зависимости от налога и повышении цен на товар.

Немаловажную роль имеет информация о сотрудниках фабрики, где представляется вся информация о том, с какими видами работ он связан на данный момент, его личные данные, отдел, адрес и записи работ.

Отчет на итоговую оплату с налогом позволяет следить за итоговой суммой по клиентам и товарам с учетом изменения налога и составителя ведомости.

Таким образом, данная база была создана для облегчения работы сотрудников фабрики с кругооборотом товаров.

**Список литературы**

1 Бекаревич Ю.Б., Пушкина Н.В. Самоучитель Microsoft Access 2002. – Санкт-Петербург.: БХВ - Петербург, 2002. – 720 с.

2 Голицина О.Л., Максимов Н.В., Попов И.И. Базы данных: учебное пособие. – М.: Форум: Инфа-М, – 2005. – 352 с.

3 Диго С.М. Проектирование и использование баз данных: Учебник, – М.: Финансы и статистика, 1995. – 348 с.

4 Карпова Т.С. Базы данных: модели, разработка, реализация, – Санкт-Петербург: Питер, 2001. – с. 304

5 Лисовский К.Ю., А.С. Марков Базы данных: Введение в теорию и методологию: Учебник. – М.: Финансы и статистика, 2004. – 512 с.

6 Мальцев М.Г., Хомоненко А.Д., Цыганков В.М. Базы данных: учебник, – 4-е изд., доп. и перераб. – Санкт-Петербург: Корона принт, 2004. – 736 с.

7 Михеева В., Харитонова И. Microsoft Access 2002. – Санкт-Петербург: БХВ-Петербург, 2002. – 1040 с.

8 Фаронов В.В., Шумаков П.В. Руководство разработчика баз данных. – М.: Нолидж, 1999. – 456 с.