

Московский Государственный Технический Университет имени Н.Э.Баумана

Факультет "Информатика и системы управления"

Кафедра "Компьютерные системы и сети"

А.В. Брешенков, А.М. Губарь

Методическое пособие по лабораторным работам

Проектирование баз данных в Access 2002

Москва 2013

ОГЛАВЛЕНИЕ

Интегрированная среда Access 2000.....	3
Проектирование таблиц в Access 2000.....	14
Проектирование запросов в Access 2002 с помощью конструкторов.....	30
Проектирование запросов в Access 2002 с помощью мастеров.....	53
Проектирование форм в Access 2002 с помощью конструктора.....	66
Проектирование форм в Access 2002 с помощью мастеров.....	93
Проектирование отчетов в Access 2002 с помощью конструктора.....	115
Проектирование отчетов в Access 2002 с помощью мастеров.....	129
Использование макросов в Access 2002.....	143

Интегрированная среда Access 2002.

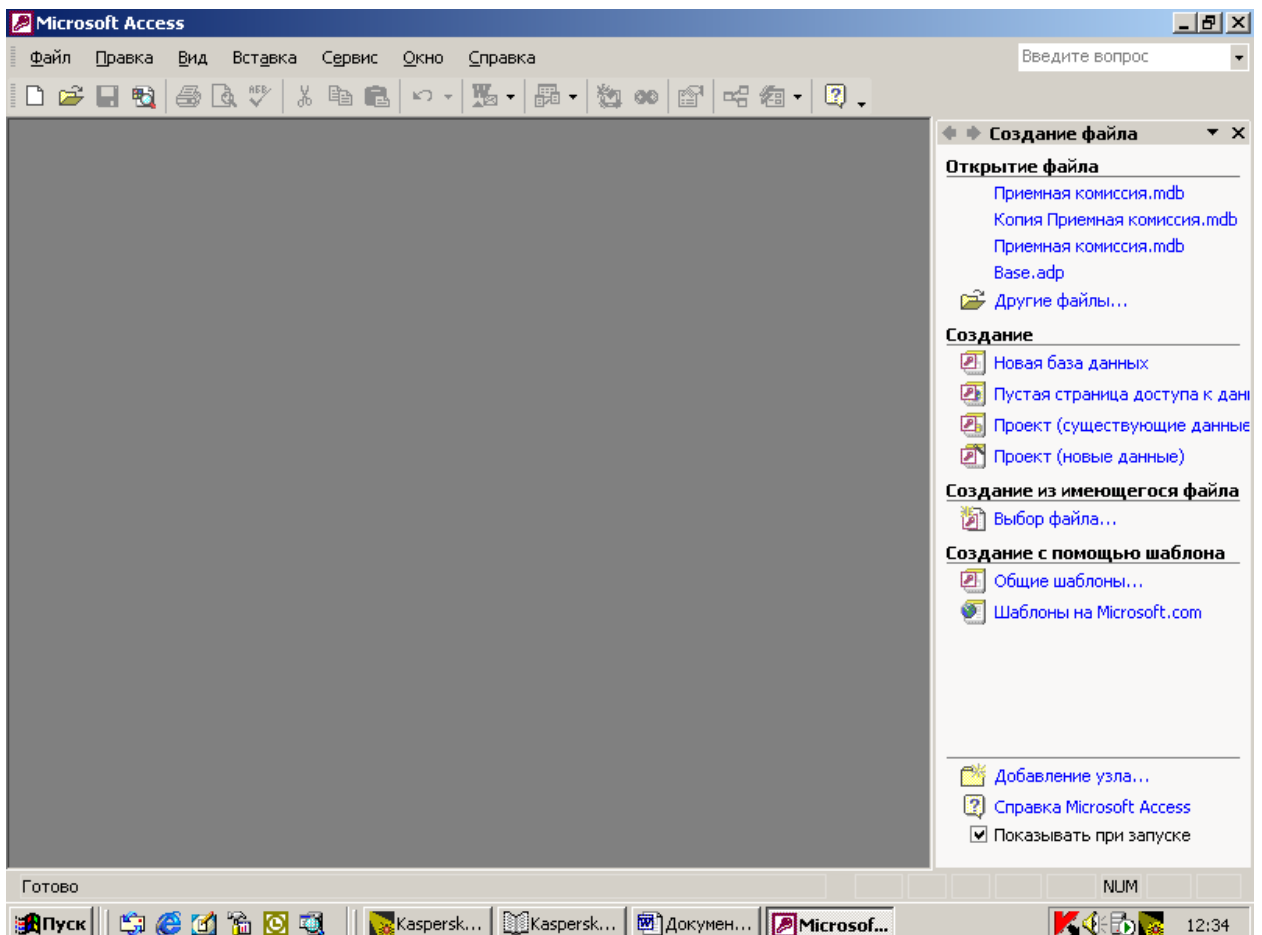
Введение

Access 2002 является одним из самых популярных приложений в семействе настольных систем управления баз данных (СУБД). Кроме того, данная система позволяет проектировать клиент-серверные базы данных (БД) средней сложности. Access 2002 и другие его версии имеют очень удобный графический интерфейс, позволяющий оперативно проектировать БД и средства их управления. Особенностью Access является то, что он хранит всю информацию в одном файле. Достоинством Access является то, что он входит в полный пакет Microsoft Office, что определяет его малую стоимость, доступность и существенно сокращает трудоемкость его освоения – общесистемные средства во всех приложениях Microsoft Office реализованы одинаково.

1. Загрузка Access 2002

Загрузка Access 2000 осуществляется аналогично всем приложениям Windows.

После его загрузки формируется следующее окно.



В верхней части окна стандартное меню приложений Microsoft Office.

В правой части окна предложены режимы загрузки и формирования БД. Ниже они кратко прокомментированы.

Открытие файла.

Данный режим позволяет загрузить одну из БД, с которой пользователь работал в предыдущие сеансы. В данном случае предлагаются 4-е БД. В настройках Access можно установить другое число

Создание.

Данный режим позволяет:

- создать новую БД. Загрузится пустое базовое окно Access, в котором проектируются все объекты БД;
- загрузить пустую страницу доступа к данным, на данной странице проектируется форма для Интернет приложения, которая связывается с БД;
- загрузить существующий проект - клиентское приложение Access, связанное с БД на сервере;
- создать новый проект - связать клиентское приложение Access с БД на сервере.

Создание из имеющегося файла.

Данный режим аналогичен подрежиму **Другие файлы** режима **Открытие файла** и позволяет загрузить БД для модификации или использования.

Создание с помощью шаблона.

Этот режим обеспечивает возможность использования разработанных фирмой Microsoft БД для эксплуатации или дальнейшего усовершенствования. При этом БД можно загружать и по Интернету.

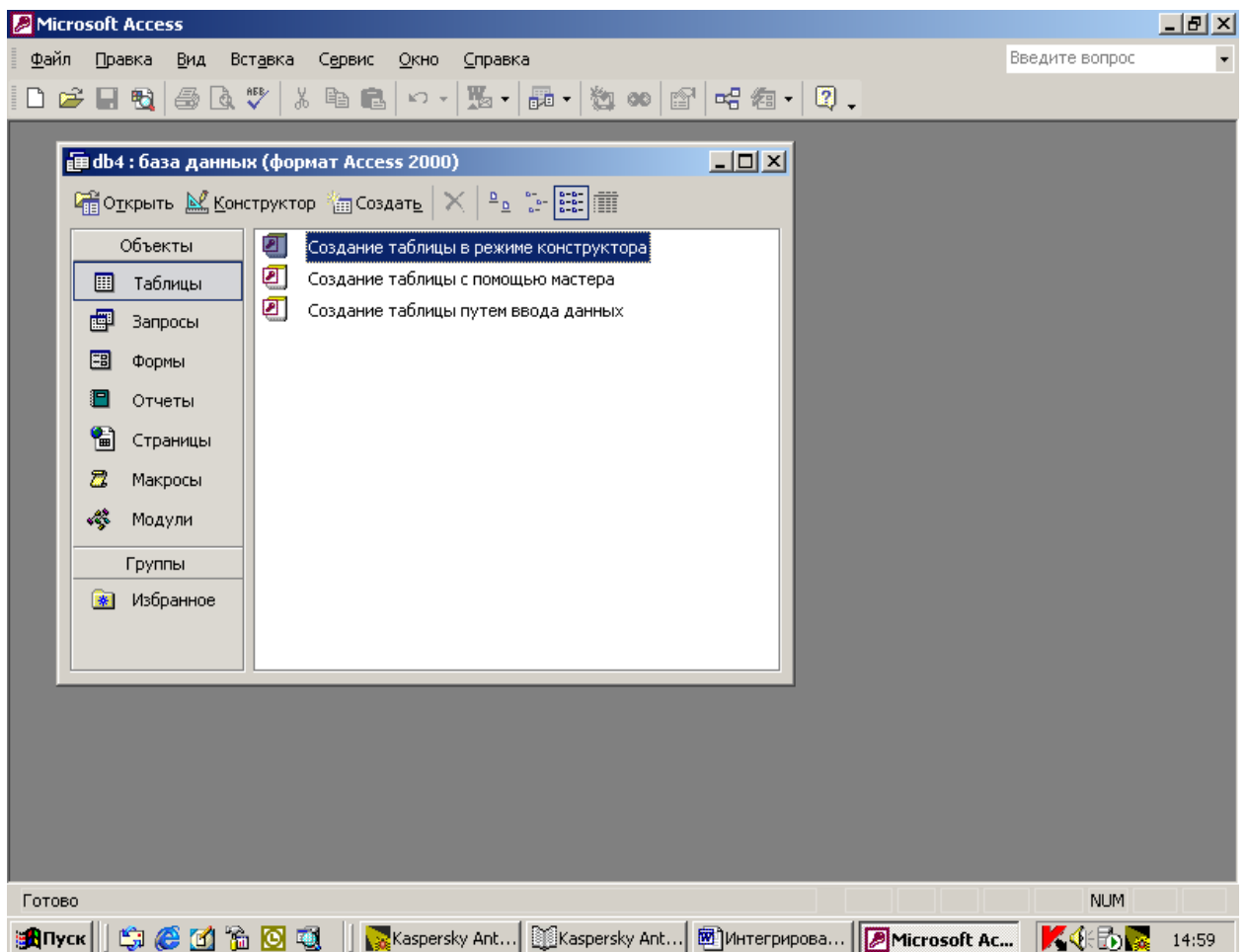
Добавление узла.

Позволяет запустить мастер добавления узла, позволяющего создавать и использовать общие папки на WEB узлах.

Посредством флажка **Показывать при запуске** вывод списка режимов в правой части окна при загрузке Access можно отменить. В этом случае действия аналогичные перечисленным выполняются с помощью меню.

2. Интегрированная среда Access.

Кратчайший путь загрузки интегрированной среды Access - это выбор подрежима **создать новую БД** после загрузки Access. После выбора данного подрежима запрашивается имя создаваемой БД и место ее размещения на диске. Затем сформируется следующее окно (базовое окно Access 2000):



В левой части окна задействованы кнопки, соответствующие объектам Access – Таблицы, Запросы, Формы, Отчеты, Страницы, Макросы, Модули. Нажатие каждой из 7-и кнопок обеспечивает переход в режим создания и редактирования соответствующих объектов.

В центральной части окна для объектов Таблицы, Запросы, Формы, Отчеты, Страницы расположены 2-е или 3-и позиции, используемые для создания этих объектов в соответствующих режимах. Например, для таблиц - **Создание таблицы с помощью конструктора**, **Создание таблицы с помощью мастера**, **Создание таблицы путем ввода данных**. Там же, в центральной части окна, отображаются объекты, спроектированные разработчиком БД.

В верхней части базового окна расположена кнопка **Открыть**, с помощью которой можно открыть любой предварительно выделенный объект. С помощью кнопки **Конструктор** можно открыть любой предварительно выделенный объект в режиме конструктора и осуществить его модификацию. Например, в конкретной таблице можно добавить, изменить, удалить поля, описать их свойства. С помощью кнопки “Создать” дублируются 2-е или 3-и позиции, используемые для создания объектов в соответствующих режимах. Кроме того, нажатие данной кнопки позволяет использовать оригинальные режимы. Для таблиц - **Импорт таблиц** и **Связь с таблицами**. Для запросов – “Перекрестный запрос”, “Повторяющиеся записи” и “Записи без подчиненных”. Для

форм – Автоформа в столбец, Автоформа ленточная, Автоформа табличная, Автоформа сводная таблица, Автоформа сводная диаграмма, Диаграмма, Сводная таблица. Для отчетов – Автоотчет в столбец, Автоотчет ленточный, Мастер диаграмм, Почтовые наклейки.

Режимы меню базового окна повторяет большую часть режимов меню приложений Word и Excel. Имеются и специфические для данного приложения команды.

В меню Файл посредством режима 'Внешние данные' можно осуществить импорт данных из других БД и Excel. Причем из БД, реализованных в Access можно импортировать не только таблицы, но и все остальные объекты БД. Посредством этого же режима можно осуществить связь с таблицами другой БД, которая в общем случае может быть размещена на другом компьютере, в частности на сервере. При этом со связанными таблицами можно осуществлять практически все манипуляции, которые возможны с 'родными' таблицами БД.

Меню Вставка дублирует кнопку **Создать** базового окна.

Меню Сервис существенно отличается от других приложений. Ниже кратко рассматриваются наиболее важные его особенности.

Подчиненное меню Схема данных позволяет создать или загрузить схему, на которой отображаются связи между таблицами. На схеме формируются и отображаются типы и способы связывания таблиц. Впоследствии связи между таблицами используются для проектирования запросов, форм, отчетов.

Подчиненное меню Анализ позволяет, в частности, выполнить анализ заполненной таблицы, дать рекомендации по ее нормализации. Кроме того, эта команда позволяет с помощью режима **Архивариус** собрать статистическую информацию обо всех объектах загруженной БД с различной степенью детализации.

Подчиненное меню Служебные программы позволяет преобразовать БД в другие версии Access. Следует иметь в виду, что преобразование осуществляется далеко не всегда успешно. В частности, из-за различия нотаций Visual Basic. С помощью этой команды можно сжать и восстановить БД. Особенностью Access является то, что при непродолжительной работе с БД соответствующий файл имеет тенденцию неоправданного увеличения объема (не только за счет данных). В связи с этим администратор БД должен предусмотреть ее регулярное сжатие. Восстановление БД иногда помогает восстановить нарушенные связи и вернуть работоспособность БД. **Диспетчер связанных таблиц** помогает восстановить связи со связанными таблицами, если они утрачены, с его же помощью можно узнать, где находятся связанные таблицы. Команда 'Разделение баз данных' позволяет преобразовать БД в две БД. В одной будут находиться все таблицы, в другой - все остальные объекты и связи с таблицами первой БД. Таким образом, реализуется псевдо клиент-серверная технология, которая обеспечивает доступ к данным

из нескольких приложений, но не обеспечивает полноценной клиент-серверной технологии. **Диспетчер кнопочных форм** позволяет запустить Мастер, который позволит создать формы с кнопками, связать их между собой и закрепить за каждой кнопкой вызов подчиненной кнопочной формы или выполнение какого-либо действия, например, открытие формы запуск макроса или открытие отчета. **Мастер преобразования в SQL сервер** позволяет на основе существующей БД построить полноценное клиент-серверное приложение на основе Access (клиент) и Microsoft SQL Server (сервер). Однако, ограничения на объем данных остаются такие же, как и в Access. **Создать MDE файл** позволяет создать из БД более компактный откомпилированный файл с форматом MDE, (обычный формат MDB), в котором будет невозможно редактировать объекты Visual Basic.

Подчиненное меню Защита позволяет сформировать пароль для БД, описать группы ее пользователей, назначить им права.

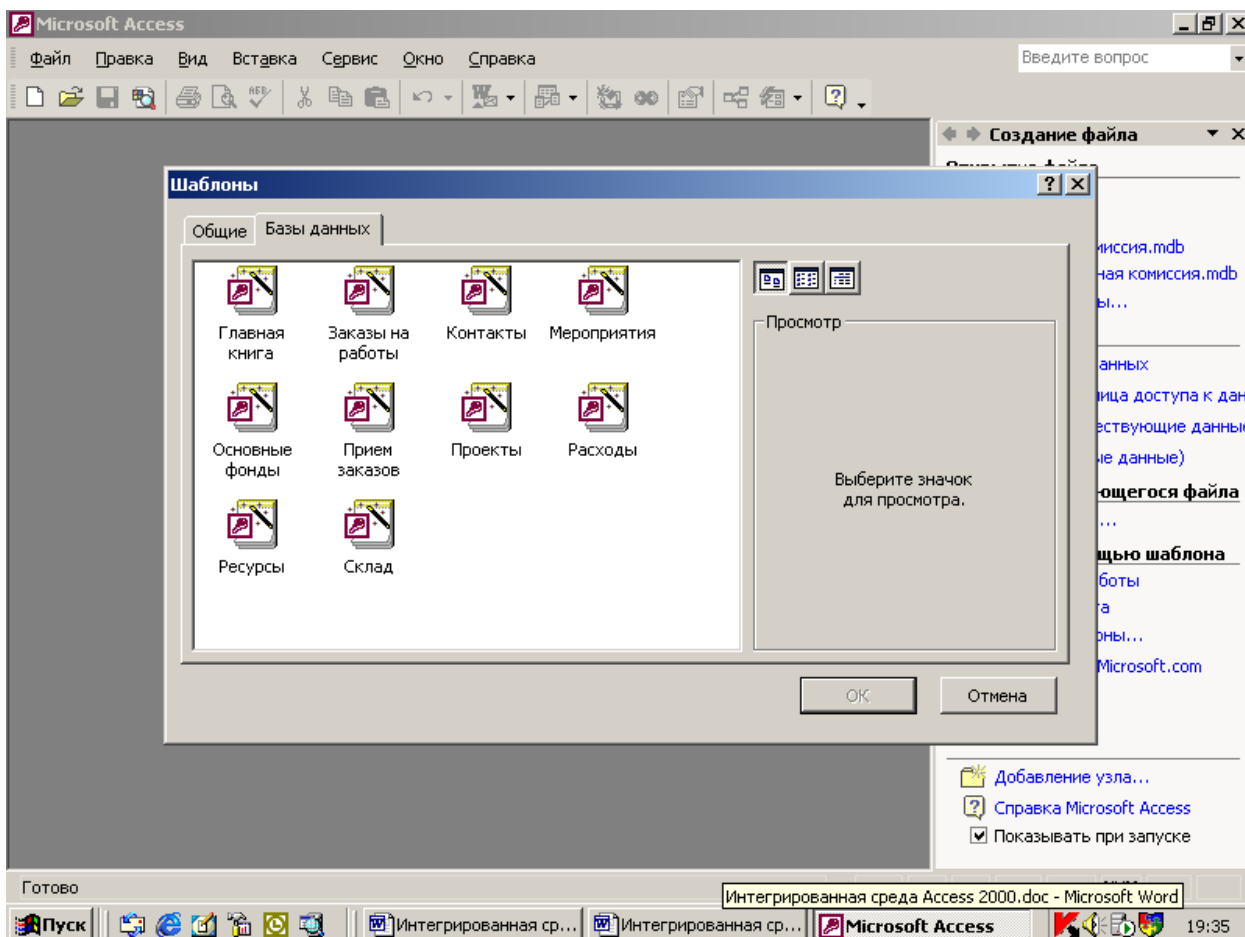
Подчиненное меню Репликация позволяет создать реплики БД – ее синхронизируемые копии, которые, в частности, могут находиться на различных компьютерах в сети.

Подчиненное меню Параметры запуска позволяет задать параметры, которые будут задействованы после запуска данной БД. В частности, можно определить форму, которая будет автоматически загружаться после запуска БД. Автоматически загружать форму можно также и с помощью макроса с именем Autoexec.

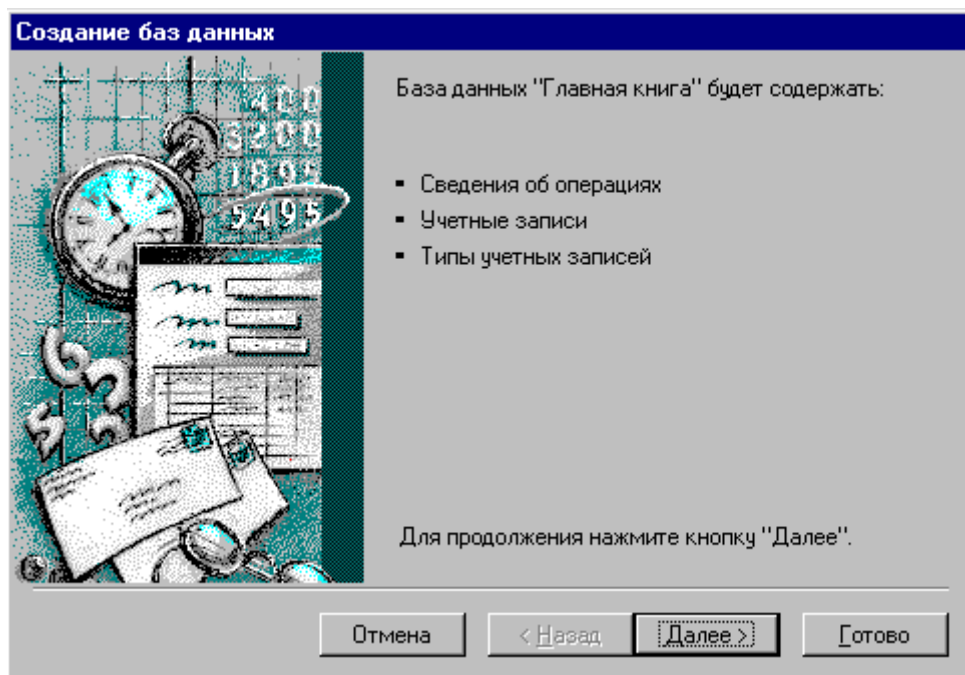
Подчиненное меню Элементы ActiveX позволяет определить **Элементы ActiveX**, которые будут доступны в данном приложении, в частности **Календарь**.

3. Создание баз данных с помощью шаблонов.

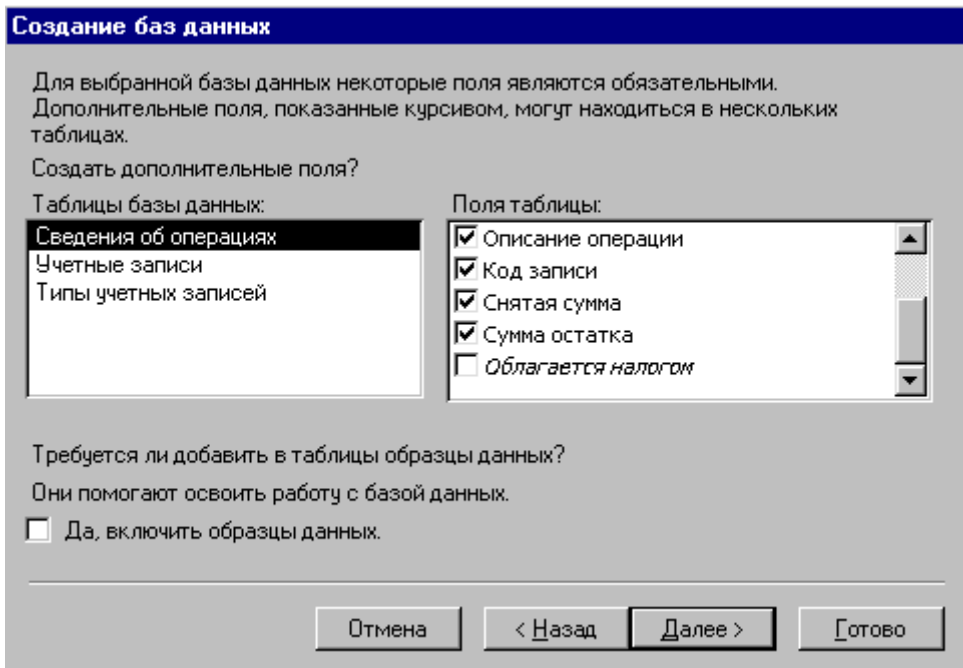
Для формирования БД с помощью шаблонов необходимо после загрузки Access выбрать режим **Общие шаблоны**. Если после загрузки Access режимы не отображается, необходимо в меню **Файл** выполнить команду **Создать**. После выбора режима **Общие шаблоны** загрузится следующее окно:



Затем выбирается нужный шаблон – наиболее близкий по сути к проектируемой БД. Например, если выбрать шаблон **Главная книга**, загрузится следующее окно:

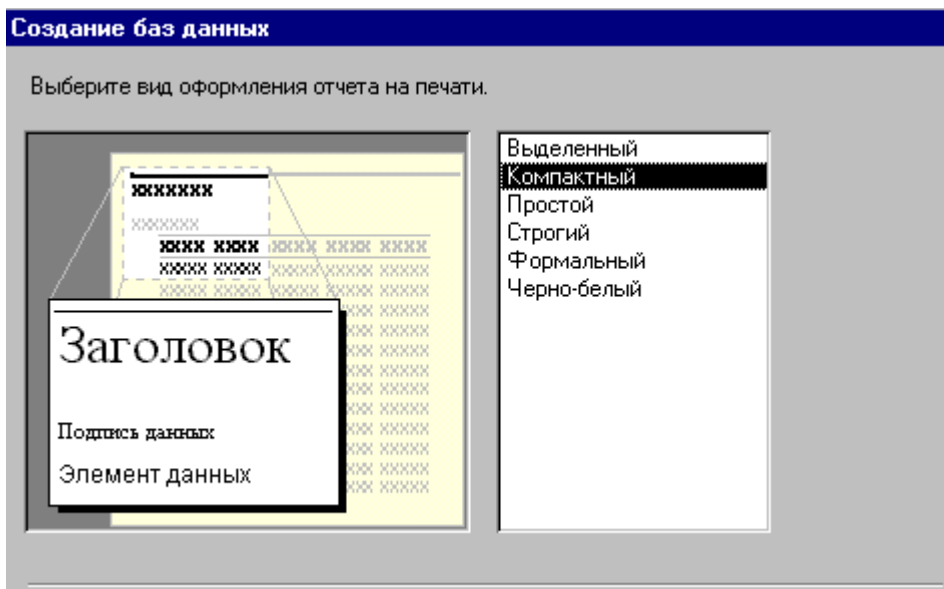
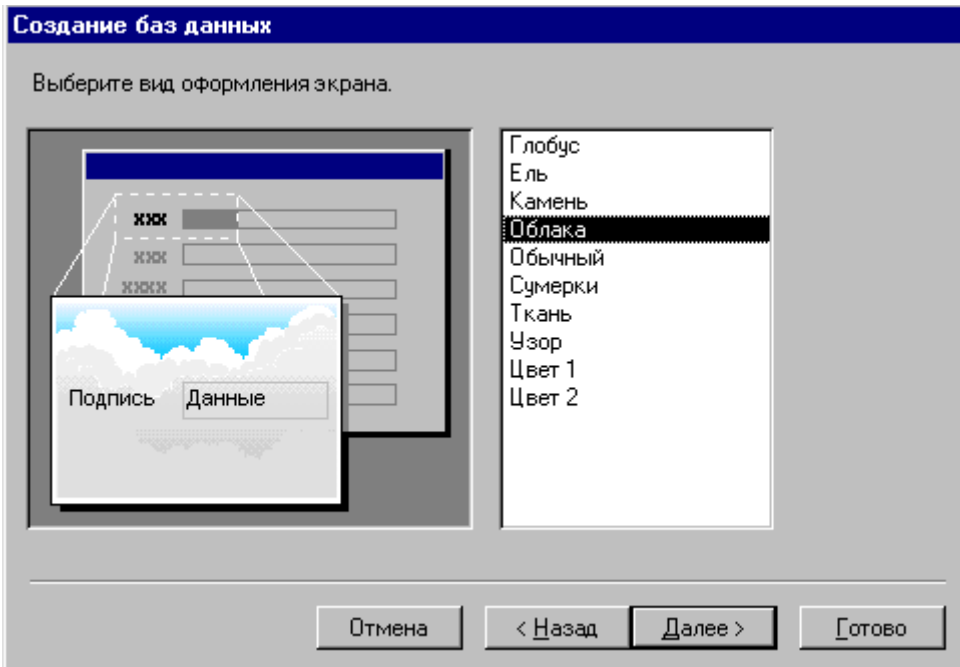


Это окно Мастера построения БД. Работа с Мастером осуществляется как обычно – по шагам. На следующем шаге сформируется окно:

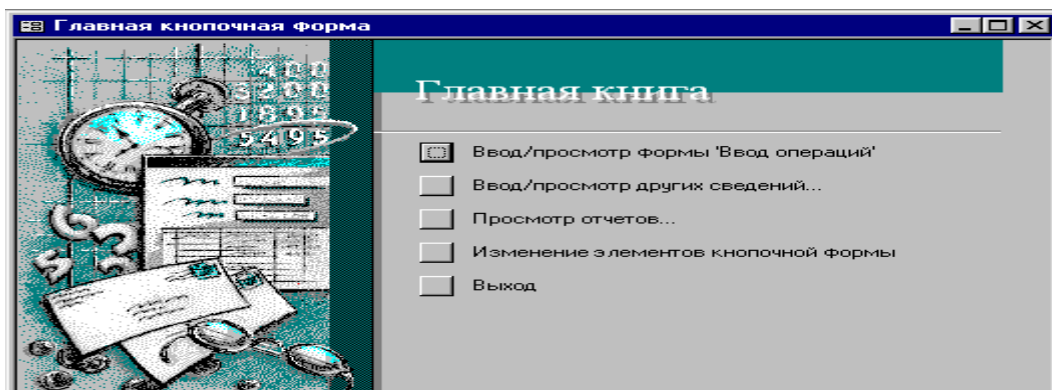


В этом окне указываются таблицы, которые задействованы в создаваемой БД и их поля. Необязательные поля выделены курсивом. Невыделенные поля отменить невозможно. С помощью флажка можно включить в таблицы образцы данных.

Выполняя следующие шаги Мастера, можно выбрать вид оформления форм и отчетов. Эти средства представлены на следующих рисунках:



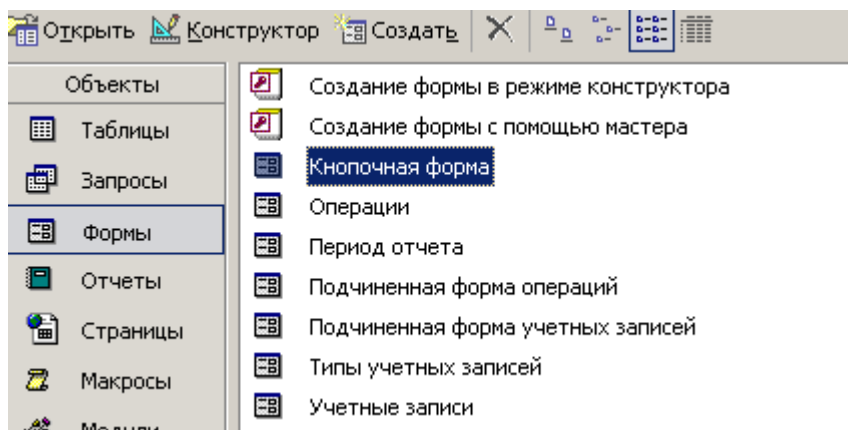
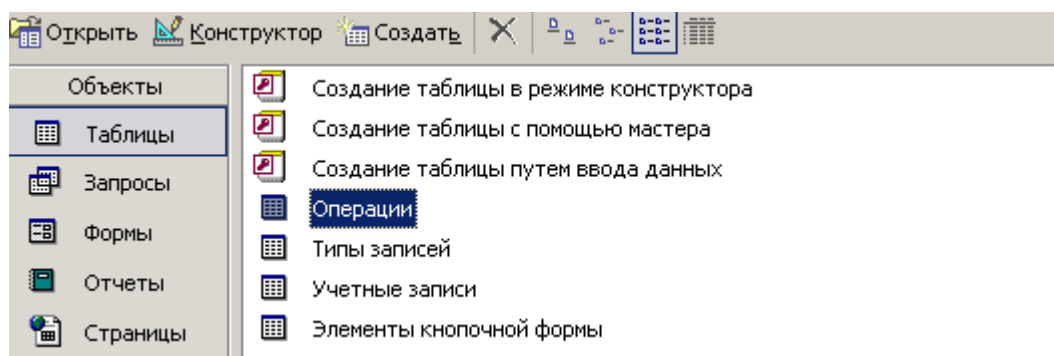
На следующем шаге запрашивается имя БД. После указания имени БД и нажатия кнопки **Готово** будет загружена БД **Главная книга**, внешний вид которой представлен на рисунке:

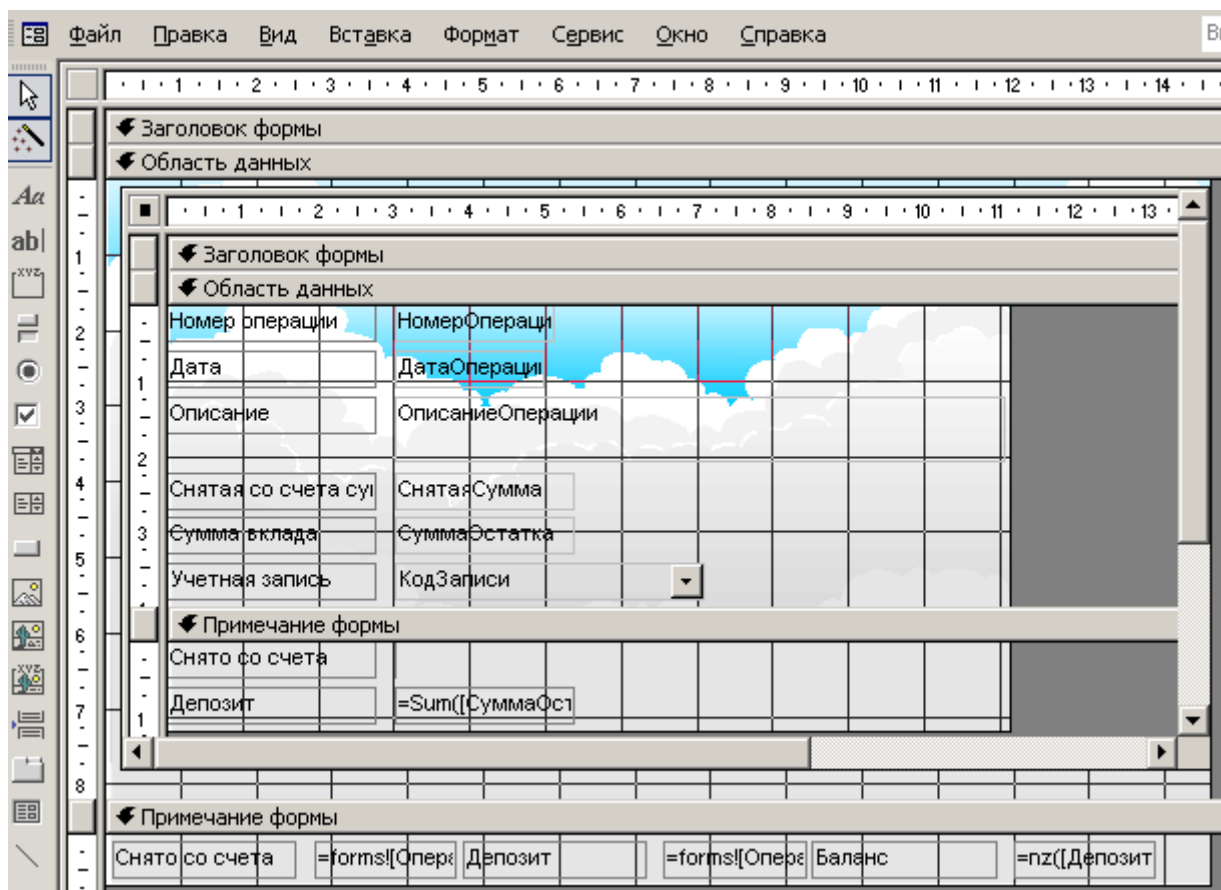


В окне представлена кнопочная форма. С помощью кнопок можно загружать подчиненные кнопочные формы и выполнять конкретные действия. Например, ввод/просмотр сведений, просмотр отчетов.

Для оперативного ознакомления с основами построения объектов Access оправданно выполнить все режимы работы БД, а затем с помощью конструктора просмотреть объекты данной БД, посредством которых эти режимы реализуются. Доступ к объектам БД обеспечивается с помощью меню **Окно** и выбора позиции с именем БД, в данном случае **Главная**. После этого загрузится базовое окно Access. В этом окне можно выделять нужные объекты и открывать их в режиме конструктора. Для возврата в БД можно в меню **Окно** выбрать позицию **Главная кнопочная форма**.

Данная БД представлена только таблицами, формами и отчетами. Список этих объектов представлен на рисунках:





В таблице виден список полей и их типы. Перемещаясь по полям, можно просмотреть и изменить их свойства.

В форме видно то, что в основную форму включена подчиненная. Видны поля таблицы, которые выводятся в форме, видны вычисляемые поля и области форм. В левой части окна расположена панель элементов, которая напоминает панель элементов многих приложений. Поэтому можно с элементами поэкспериментировать.

Если в режиме конструктора были внесены какие-либо изменения, то при закрытии объекта эти изменения можно сохранить или нет.

Задание на лабораторную работу.

1. Загрузить Access.
2. Просмотреть режимы загрузки и охарактеризовать их.
3. Создать новую БД.
4. Просмотреть формы для проектирования объектов Access и охарактеризовать их.
5. Просмотреть меню Access и охарактеризовать все его команды.
6. Создать БД на основе какого-либо шаблона с анализом каждого шага построения БД.
7. Выполнить все режимы сформированной БД (поработать в качестве пользователя).

8. Открыть базовое окно Access.
9. Просмотреть все объекты сформированной БД.
10. Установить соответствие между объектами БД и их местом в среде БД.
11. Открыть все объекты БД в режиме конструктора.
12. Проанализировать элементы объектов в режиме конструктора.
13. Внести изменения в объекты и просмотреть результаты.
14. Создать БД на основе какого-либо другого шаблона с анализом каждого шага построения БД.
15. Выполнить п.п. 7-14.

Проектирование таблиц в Access 2002.

Введение

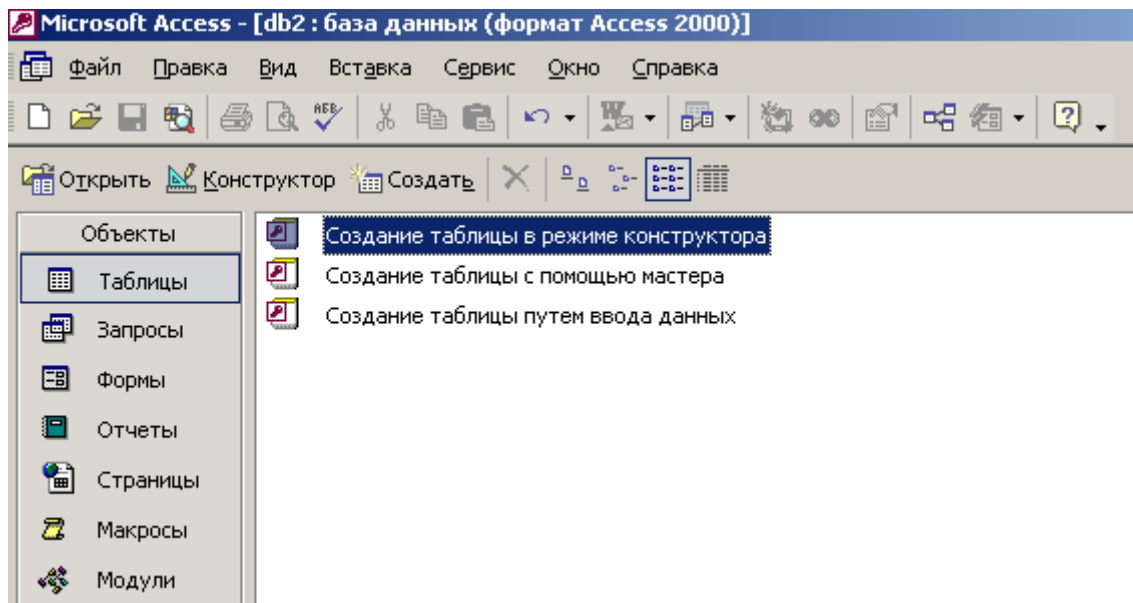
Таблицы являются основным объектом всех БД, в том числе и Access 2002. Они служат для хранения и доступа к данным. Таблицы представляют собой совокупность записей. Каждая запись включает в себя одно или более полей. Поле определяется типом и набором свойств. То, насколько продуманы таблицы перед их проектированием и реализацией в конечном итоге определяет количественные и качественные характеристики БД. В Access 2002, в отличие от многих СУБД, можно перепроектировать даже заполненные таблицы. Но это удастся далеко не всегда. Поэтому состав и структуру таблиц необходимо тщательно проанализировать до создания других объектов Access 2002, а тем более до эксплуатации БД. При проектировании БД в Access 2002 и, в частности ее таблиц, следует помнить, что максимальный объем БД в Access 2002 2 Гб, максимальная длина имени поля 64 символа, максимальное количество полей – 255, максимальная длина записи 2 Кбайта. Теоретически в таблице может содержаться 2 000 000 000 записей.

В Access возможны 3- способа создания таблиц: путем ввода данных, с помощью мастера и с помощью конструктора.

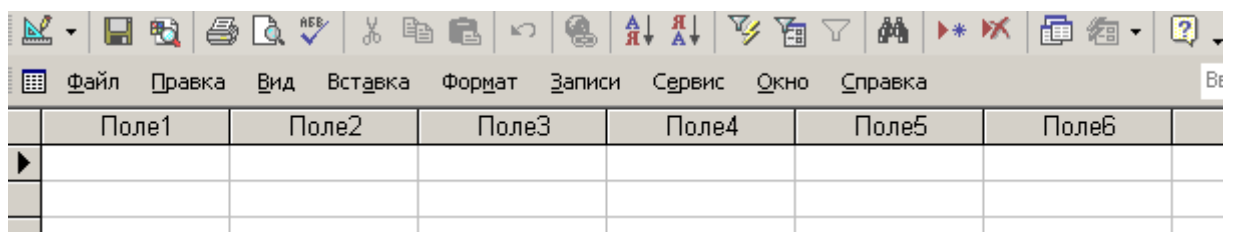
4. Создание таблиц путем ввода данных.

Самым простым, но не самым лучшим способом создания таблиц, является создание таблиц Access 2002 путем ввода данных.

Окно, в котором представлены ярлыки для выбора способа создания таблиц, приведено ниже:



Для формирования таблиц путем ввода данных необходимо выбрать соответствующий ярлык, после чего загрузится окно:




В поля данной таблицы вводятся данные. При этом надо следить за тем, чтобы типы данных вводимых в одноименном столбце были одинаковыми.

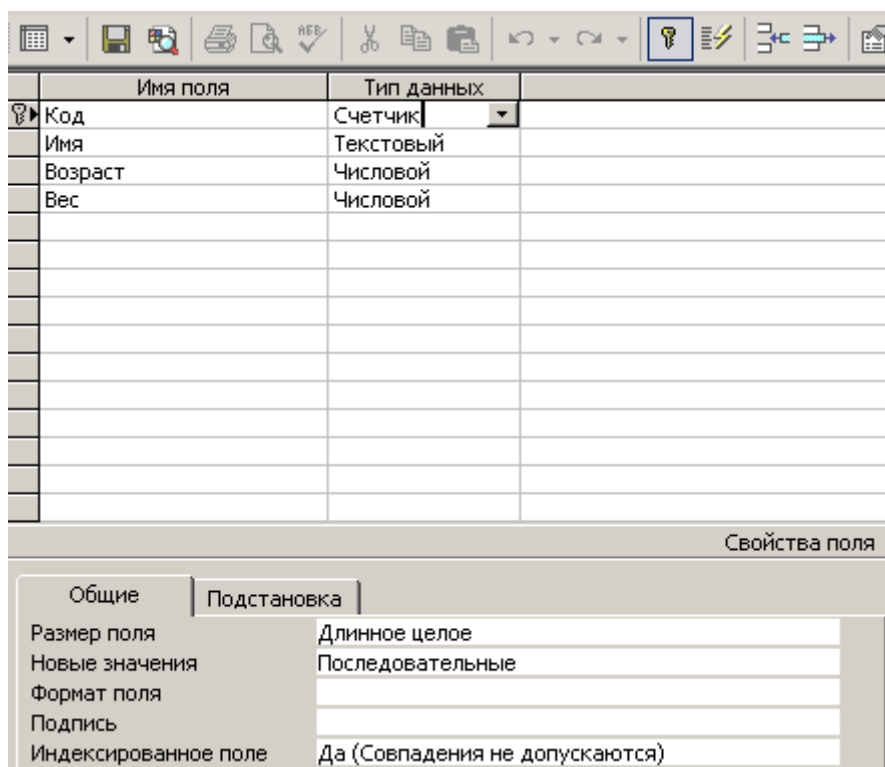
Заголовки полей можно поменять. Для этого нужно щелкнуть правой кнопкой по заголовку, выбрать из контекстного меню **Переименовать столбец** и ввести новое имя столбца.

Посредством контекстного меню можно вставить, удалить столбец, а также выполнить другие полезные команды, характерные для Word и/или Excel (Многие из них отображены на панели инструментов). Режим контекстного меню **Столбец подстановок** позволяет запустить Мастер подстановок, который предложит использовать для списка существующую таблицу или создать новый список. После выполнения шагов мастера подстановок активное поле преобразуется в поле со списком.

По окончании ввода данных необходимо закрыть и сохранить таблицу. При этом система предложит создать ключевой столбец. Если согласится с предложением, сформируется дополнительное поле с типом **Счетчик**. На рисунке приведен пример заполненной таблицы:

	Имя	Возраст	Вес	Поле4	Поле5
	Вася	12	55,5		
	Петя	14	60,3		
	Боря	13	50,8		

Эта же таблица, открытая в режиме конструктора, выглядит следующим образом:

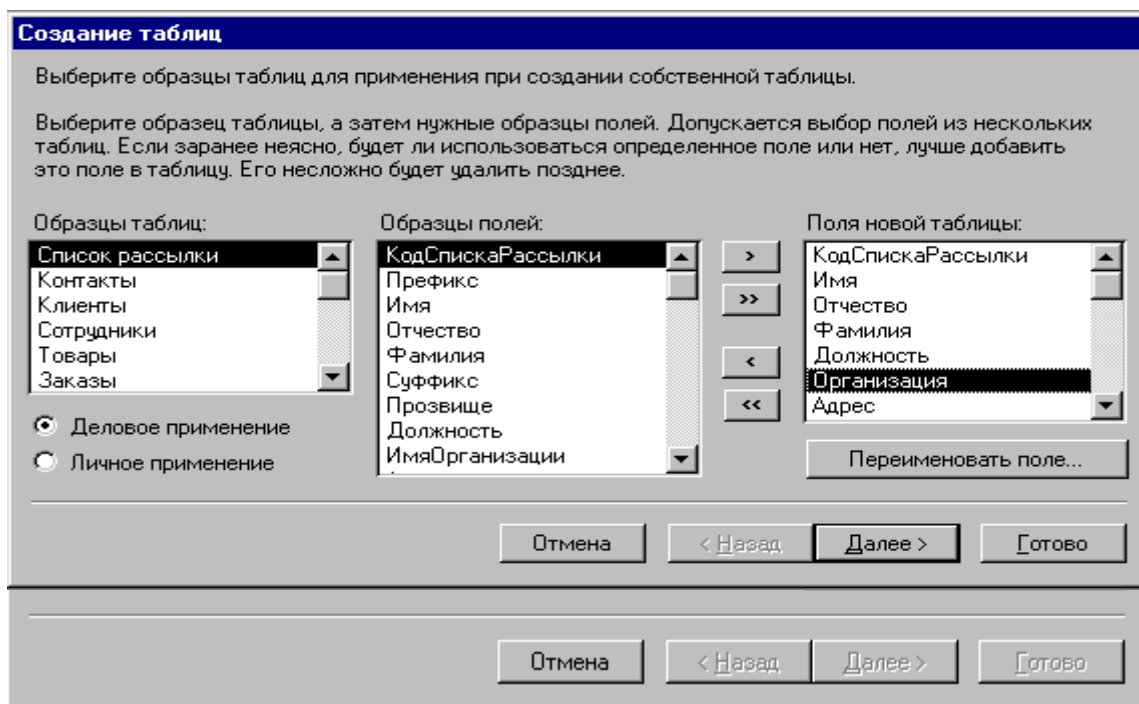


Как видно из рисунка, сформировано дополнительное поле с типом данных **Счетчик**. Для других полей автоматически определены их типы и свойства. Они сформированы исходя из контекста введенных данных. Несмотря на то, что поля 'Возраст' и 'Вес' числовые, свойства у них отличаются. Их размер соответственно 'Длинное целое' и **Двойное с плавающей точкой**. Это связано с представлением данных в таблице при ее заполнении (см. предыдущий рисунок). Для просмотра и изменения свойств поля необходимо щелкнуть в строке этого поля.

В режиме конструктора можно переименовать, удалить, добавить поля изменить их свойства. Однако, при изменении свойств полей возможны ошибки преобразования. Конечно, можно удалить данные из таблицы, а затем переопределить ее структуру.

5. Создание таблиц с помощью мастера.

Для формирования таблиц с помощью мастера необходимо выбрать соответствующий ярлык, после чего загрузится окно:



Как видно из рисунка, в левой части окна представлен список спроектированных таблиц, в центральной части – их поля, в правой части - используемые поля.

Всего представлено несколько десятков таблиц. Причем примерно половина из них для делового применения и половина для личного применения. Каждая таблица включает в себя группу полей. Так, например, таблица **Список рассылки** включает в себя 37 полей различного типа. С помощью кнопки '>' можно перенести отдельные поля в новую (создаваемую) таблицу, с помощью кнопки '>>' можно перенести все поля в новую таблицу. Посредством кнопок '<', '<<' поля удаляются из новой таблицы. Новую таблицу можно формировать из полей нескольких таблиц. Таким образом, можно не только проанализировать состав атрибутов описываемого объекта, но и компоновать новый состав атрибутов из нескольких объектов.

В качестве примера выбраны следующие поля:

Как видно из рисунка, одно поле переименовано. Это реализуется посредством выделения нужного поля, нажатия кнопки **Переименовать поле** и ввода нового имени.

На следующем шаге мастера вводится имя таблицы и выполняется один из приведенных ниже режимов работы с таблицей:

Указаны все сведения, необходимые для создания таблицы с помощью мастера.

Дальнейшие действия после создания таблицы:

изменение структуры таблицы

непосредственный ввод данных в таблицу

ввод данных в таблицу с помощью формы, создаваемой мастером

Вывести справку по работе с таблицей.

Отмена < Назад Далее > Готово

Если выбрать режим **Изменение структуры таблицы**, то спроектированная таблица загрузится в режиме конструктора:

	Имя поля	Тип данных
☺	КодСпискаРассылки	Счетчик
	Имя	Текстовый
	Отчество	Текстовый
	Фамилия	Текстовый
	Должность	Текстовый
	Организация	Текстовый
	Адрес	Текстовый
	ДомашнийТелефон	Текстовый
	РабочийТелефон	Текстовый
	АдресЭлектроннойПочты	Текстовый
	ДатаРождения	Дата/время
	ДатаВступления	Дата/время
	СуммаВзноса	Денежный
	ДатаВыплатыВзноса	Дата/время
	СуммаВыплаты	Денежный
	ДатаВыплаты	Дата/время
	Фотография	Поле объекта OLE
	Заметки	Поле MEMO

Свойства поля

Общие	Подстановка
Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	Код списка рассылки
Индексированное поле	Да (Совпадения не допускаются)

В таблице задействовано 6 типов данных. В данном режиме можно изменить состав и свойства полей.

Мастер построения таблиц полезен в том случае, если разработчик БД не четко представляет состав и свойства атрибутов описываемого с помощью таблицы объекта. В противном случае для описания таблиц лучше всего пользоваться конструктором.

6. Создание таблиц с помощью конструктора.

После выбора соответствующего режима загрузится окно аналогичное предыдущему, только пустое.

В этом окне описываются нужные поля. В области **Имя поля** вводятся имена полей. В области **Типы данных** описываются их типы. В области **Описание** вводятся необязательные примечания к полям. В области **Свойства** назначаются свойства полей.

В подавляющем большинстве случаев таблицы следует проектировать так, чтобы они удовлетворяли требованиям нормализации. Эти требования связаны с совокупным составом полей таблиц. В большинстве таблиц для обеспечения уникальности записей и связывания таблиц необходимо назначить ключевые поля.

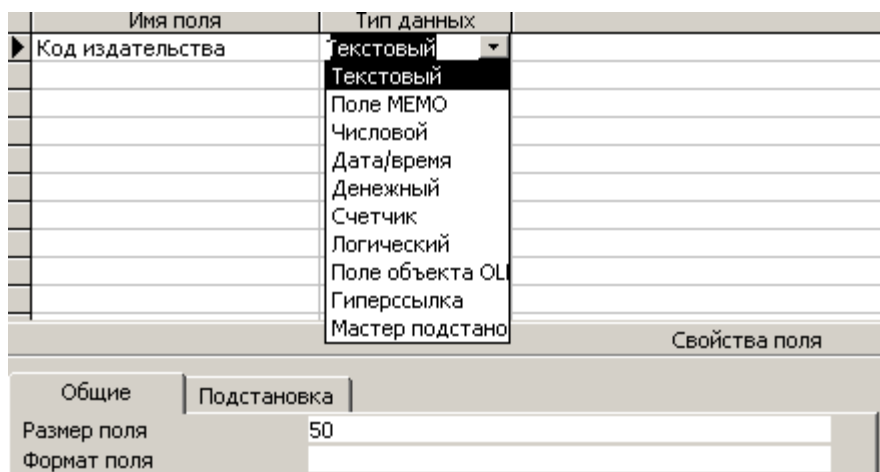
Собственно проектирование таблицы заключается в определении состава полей, описания их имен, типов и свойств. Далее кратко рассмотрены эти действия.

4.1. Описание имен полей.

Имена полей должны удовлетворять следующим требованиям: имена должны быть уникальны, длина имен не должна превышать 64 символа, имена не должны начинаться с символа пробела, в именах не должны содержать специальные символы Access (';', '''', '!', '[', ']'). В именах полей допустимы пробелы. Однако, в этом случае при обращении к таким именам их необходимо заключать в квадратные скобки. Эти же правила действительны для имен всех объектов Access.

4.2. Назначение типов полей.

Типы полей выбираются из предлагаемого списка, приведенного на рисунке:



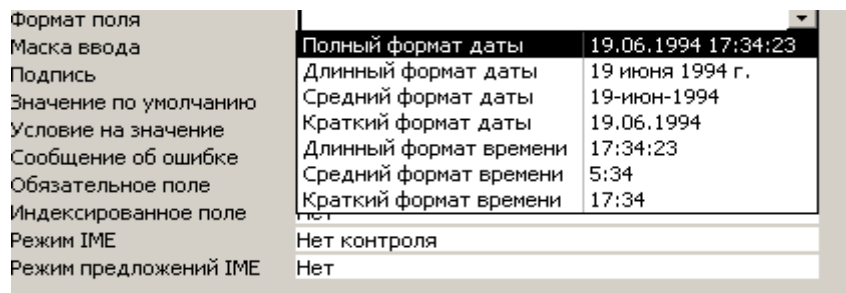
Ниже кратко охарактеризованы типы полей Access.

Текстовый – символьные данные до 255 символов. Независимо от назначенной длины в поле данного типа хранится только введенное количество символов.

Поле MEMO может содержать до 65 535 символов. Поле данного типа не может быть ключевым или индексным.

Числовой тип используется для хранения числовых данных. Имеет подтипы байт, целое, длинное целое, одинарное с плавающей точкой, двойное с плавающей точкой, код репликации, действительное. Подтип выбирается в области свойств в строке 'Размер поля'.

Дата/время используется для представления даты и времени. В Access возможно использование дат от 100 г. до 9999 г. Формат этого поля выбирается посредством свойства 'Формат'. Возможные форматы приведены на рисунке:



Денежный тип данных предназначен для хранения данных, точность представления которых до 4 знаков после запятой. Целая часть содержит до 15 знаков.

Счетчик представляет собой поле, в котором Access автоматически формирует уникальное значение. Поле данного типа, как правило, используется в качестве первичного ключа.

Логический тип данных используется для хранения одного из 2-х значений Истина/Ложь.

Поле объекта OLE содержит ссылку на OLE объект. Объем объекта ограничен только имеющимся в наличии дисковым пространством. Ссылка на конкретный объект реализуется в процессе заполнения поля. Поле данного типа не может быть ключевым или индексным.

Гиперссылка позволяет ввести в поле гиперссылку.

Мастер подстановок позволяет запустить мастер подстановок, который позволит определить поле как поле со списком и указать список для выбора.

4.3. Назначение свойств полей.

Многие свойства полей для различных типов полей совпадают, но есть и оригинальные свойства.

Ниже кратко охарактеризованы свойства полей Access.

Подпись может быть введена почти для всех типов полей. То, что введено в это поля автоматически отображается в качестве подписей полей при формировании форм и отчетов. При отсутствии подписи будут использоваться имена полей. Подпись может содержать до 2048 символов.

Обязательное поле определяет обязательный или необязательный ввод данных в данное поле.

Формат поля указывает формат вывода данных в режиме Таблицы. В режиме Формы формат поля можно изменить.

Маска ввода позволяет указать маску для ввода данных, например для ввода номера телефона. Для описания маски используются специальные символы.

Размер поля позволяет указать длину поля.

Значение по умолчанию позволяет указать значение данного поля, которое задается автоматически. Его, конечно, при необходимости можно изменить.

Условие на значение позволяет ввести выражение, в соответствии с которым будут проверяться введенные в поле данные. Например, для поля 'Возраст' может быть введено условие ≥ 0 and < 200 .

Сообщение об ошибке используется для ввода сообщения, которое будет выводиться, если после ввода данных в поле не выполняется условие на значение.

Пустые строки запрещает или разрешает использовать в качестве данных пустые строки. Следует обратить внимание на то, что пустое поле это не пустая строка и не 0. Пустое поле имеет значение Null. Разработчик БД, в случае необходимости, должен предусмотреть обработку полей с такими значениями в каждом конкретном случае, в зависимости от назначения поля.

Индексированное поле определяет индекс, задаваемый по данному полю. Индекс ускоряет выполнение запросов, в которых используются индексированные поля и операции сортировки и группировки. Например, если часто выполняется поиск по полю **Фамилия** в таблице **Сотрудники**, следует создать индекс для этого поля. При назначении индекса можно указать, допускаются ли совпадения значений данного поля.

Свойства полей назначаются на вкладке **Общие**.

Вкладка **Подстановка** становится доступной, когда для поля использован мастер подстановок. На этой вкладке отображаются и редактируются свойства поля со списком.

Для того чтобы какое либо поле назначить ключевым необходимо выделить это поле и щелкнуть по инструменту



Для того, чтобы какое либо группу полей назначить ключом необходимо, удерживая клавишу Ctrl, выделить нужные поля и щелкнуть по тому же инструменту.


7. Связывание таблиц на схеме данных.


а. Введение.

Важнейшими свойствами реальных БД является возможность получения информации из нескольких таблиц одновременно, целостность и непротиворечивость данных. Таблицы отображают реальные объекты, а также возникают в результате мероприятий по нормализации исходных таблиц. В результате нормализации таблица может быть представлена двумя или более таблицами. Связи между таблицами позволяют вывести совокупную информацию, автоматически отслеживать целостность данных.

Для связывания таблиц используются ключевые поля – первичные и внешние. Первичные ключи уникальны, внешние могут повторяться, их значения равны значениям первичных ключей в других таблицах.

В качестве примера для связывания таблиц спроектированы 2-е таблицы – Издательства и Книги:

Издательства : таблица	
Имя поля	Тип данных
 Код издательства	Счетчик
Наименование	Текстовый
Адрес	Поле объекта OLE
Логотип	Текстовый
Сайт	Гиперссылка

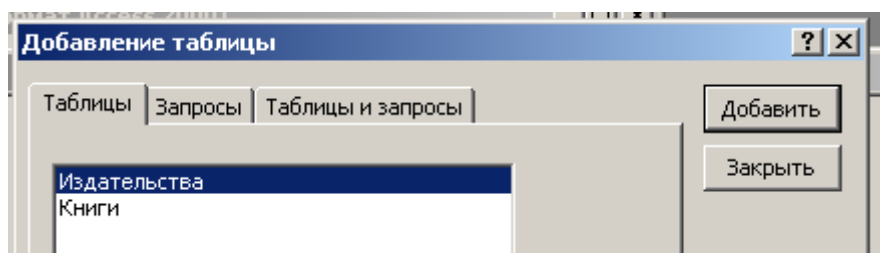
Книги : таблица	
Имя поля	Тип данных
 Код книги	Счетчик
Автор	Текстовый
Название	Текстовый
Цена	Денежный
Код издательства	Числовой
Дата издания	Дата/время
Твердый переплет	Логический
Аннотация	Поле MEMO

При проектировании таблиц задействованы все типы полей, тип поля следует из его назначения. В качестве ключевых полей назначены счетчики, т.к. счетчики автоматически изменяют свое значение. Назначение счетчиков в качестве ключевых полей необязательно. Кроме того, можно использовать несколько полей в качестве ключа. Использование счетчика автоматически обеспечивает уникальность записи и удобство формирования записи. **Код издательства** и **Код книги** являются первичными ключами для соответствующих таблиц. Поле **Код издательства** присутствует и в таблице **Книги** в качестве обычного поля. Таким образом, оно может быть использовано в качестве внешнего ключа. Тип этого поля должен совпадать с типом первичного ключа. Счетчик имеет числовой тип (длинное целое). Имена первичного и внешнего ключей не обязательно должны совпадать, но назначение совпадающих имен упрощает дальнейшие действия как Access, так и разработчика БД.

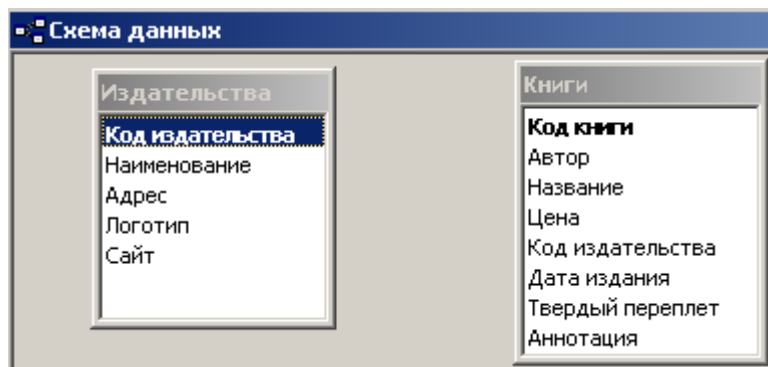
Для формирования, отображения и редактирования схемы данных используется инструмент



После выбора данного инструмента загрузится окно:



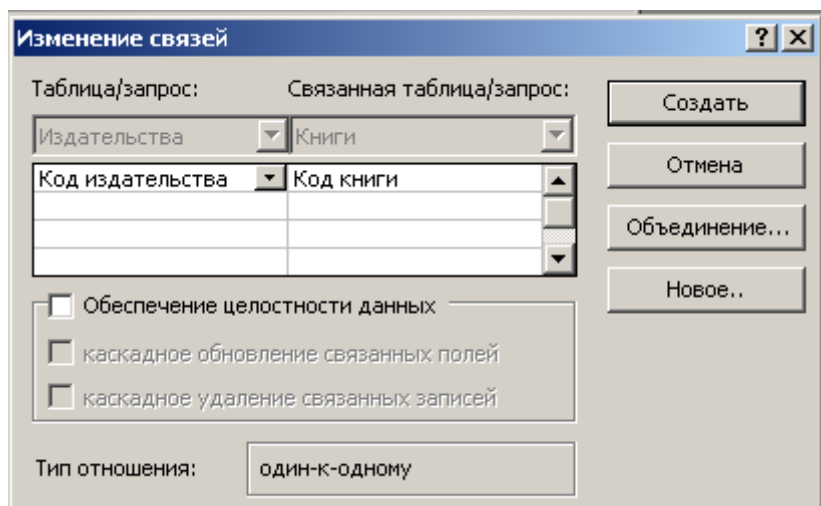
В этом окне выбираются нужные таблицы и (запросы) и добавляются в схему данных. На рисунке представлена схема данных для 2-х таблиц:



Существует 4-е типа связей между таблицами: один к одному, один ко многим, многие к одному и многие ко многим. 2-й и 3-й типы связей различаются только тем, как по отношению друг к другу размещены таблицы.

в. Связь один к одному.

Для рассматриваемого случая связь один к одному можно создать связь, связав первичные ключи таблиц - **Код издательства** и **Код книги**. Это будет означать, что каждому издательству ставится в соответствие только одна книга. Это маловероятная ситуация. Чтобы связать таблицы, необходимо щелкнуть на ключевом поле одной таблицы и пробуксировать мышью до ключевого поля другой таблицы. В результате сформируется окно:



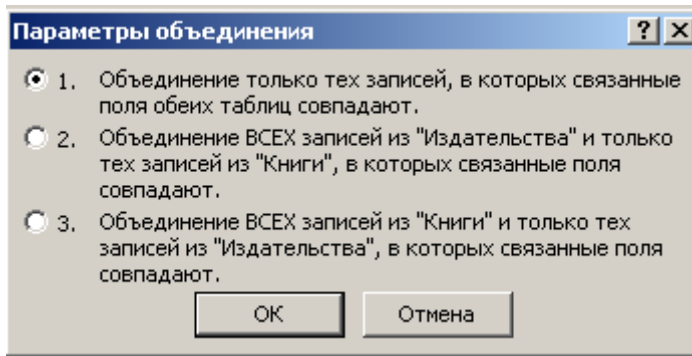
В этом окне описывается связь.

Если задействовать флажок **Обеспечение целостности**, то при формировании ключевого поля в какой-либо таблице выполняется проверка - совпадает ли его значение со значением ключевого поля в связанной таблице. В случае несовпадения, выдается соответствующее сообщение и значение не вводится.

Если задействовать флажок **Каскадное обновление связанных полей**, то при изменении значения ключевого поля в какой-либо таблице автоматически выполняется изменение значения соответствующего ключевого поля в связанной таблице. (Значение поля типа Счетчик изменить невозможно).

Если задействовать флажок **Каскадное удаление связанных полей**, то при удалении какой-либо записи в таблице выполняется автоматическое удаление связанной записи в другой таблице.

При формировании связей необходимо правильно установить параметры объединения. От этого зависит то, что впоследствии будет отображено посредством запроса к данным таблицам. Для выбора параметров объединения необходимо щелкнуть по кнопке **Параметры**. В результате сформируется окно:



Эти объединения называются соответственно внутреннее, левое и правое. Существует, кроме названных параметров объединения, внешнее объединение. Но в Access для формирования такого объединения необходимо выполнить специальные мероприятия.

Внешний вид связи зависит от значения флажка **Обеспечение целостности** и параметров объединения.

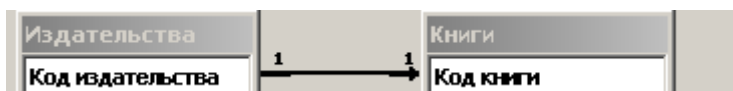
Флажок не установлен, объединение внутреннее:



Флажок установлен, объединение внутреннее:



Флажок установлен, объединение левое:



Флажок установлен, объединение правое:



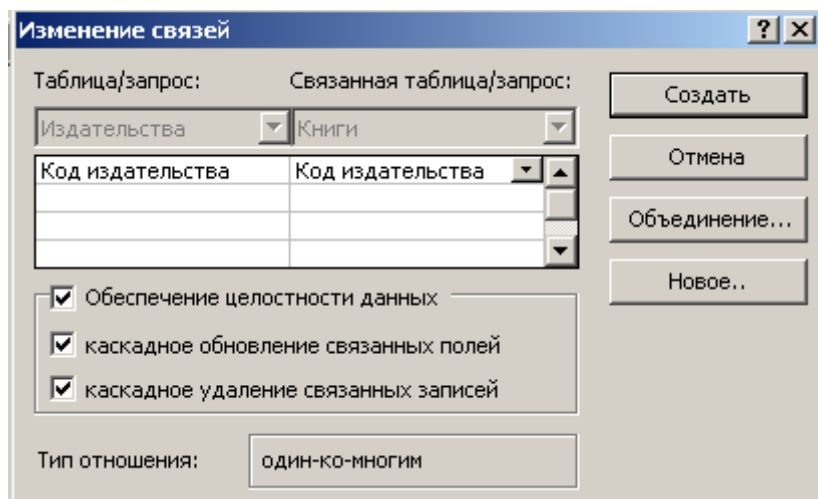
Свойства связи можно редактировать. Для этого необходимо выполнить по связи двойной щелчок. Кроме того, связь можно удалить традиционным способом.

с. Связь один ко многим.

Следует отметить, что связь один к одному в рассмотренном случае нелогична. Связь такого типа оправдана, например, для связывания таблиц с личными данными студента и его успеваемостью.

В данном случае более логично сформировать связь один ко многим. Действительно, каждое издательство может издавать множество книг.

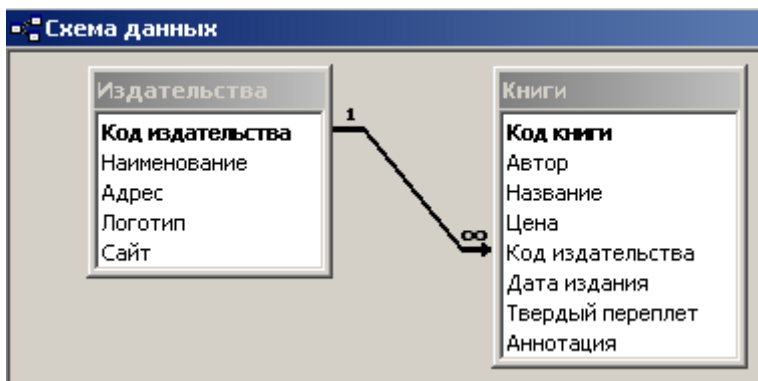
Формирование и описание связи реализуется по аналогии с предыдущим случаем. Только в данном случае связывается первичный ключ **Код издательства** таблицы **Издательства** и внешний ключ **Код издательства** таблицы **Книги**.



В этом случае автоматически формируется связь один ко многим, что видно из комментария в нижней части окна.

Флажки имеют тот же смысл, что и в предыдущем случае. Введенные флажки означают то, что при формировании записи о книге допустимо ссылаться только на существующее (ранее описанное) издательство; при изменении **Кода издательства** в таблице **Издательства** автоматически изменяются ссылки на издательства в таблице **Книги**; при удалении записи из таблицы **Издательства**, удаляются записи о всех книгах, которые изданы в данном издательстве из таблицы **Книги**.

Внешний вид связи, как и в предыдущем случае, зависит от значения флажка **Обеспечение целостности** и параметров объединения. Единственное отличие на конце связи со стороны 'многие' изображается знак бесконечности. Например:



Связь такого рода обычно предусматривается – в таблицах описываются соответствующие поля.

d. Связь многие ко многим.

Для БД **Издательства-Книги** более логична связь многие ко многим. Действительно, различные издательства могут издавать одни и те же книги. И наоборот одна и та же книга может издаваться в различных издательствах.

Для формирования связей такого типа вводится третья таблица, в которой задействованы первичные ключи первых двух таблиц. Если есть необходимость, в эту таблицы вводятся и другие поля. Ниже представлена данная таблица в режиме конструктора.

Связи : таблица

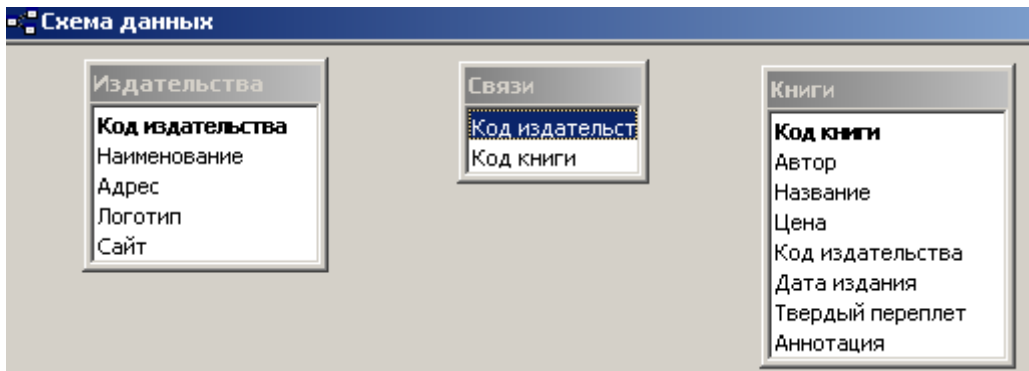
Имя поля	Тип данных
<input checked="" type="checkbox"/> Код издательства	Числовой
<input type="checkbox"/> Код книги	Числовой
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	

Свойства поля

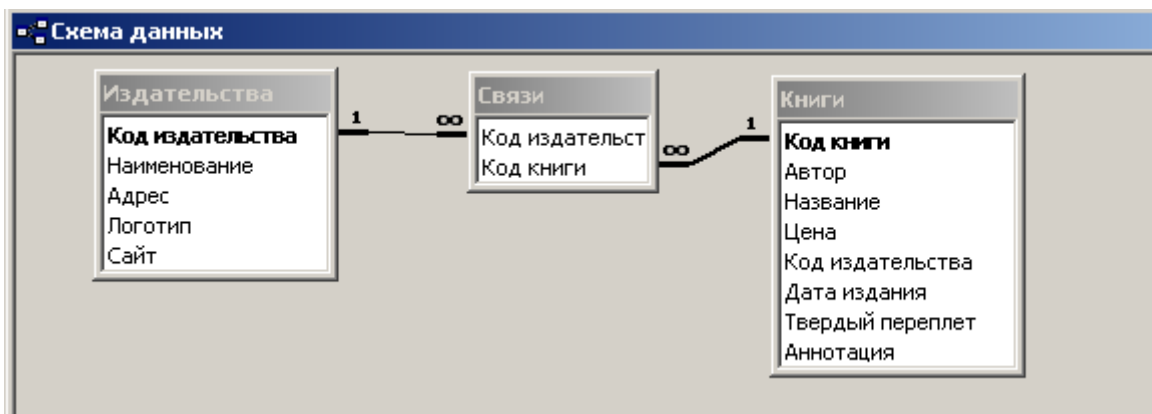
Общие	Подстановка
Размер поля	Длинное целое
Формат поля	
Число десятичных знаков	Авто
Маска ввода	
Подпись	
Значение по умолчанию	0
Условие на значение	
Сообщение об ошибке	
Обязательное поле	Нет
Индексированное поле	Да (Допускаются совпадения)

Поля лучше назвать так же, как и в связываемых таблицах. Типы полей в данном случае нужно выбрать числовое, длинное целое (как у счетчика). В свойствах полей 'Индексированное целое' необходимо выбрать 'Допускается совпадения'. Это связано с тем, что в таблице **Связи** значения данных полей могут повторяться – связь многие ко многим.

После описания таблицы **Связи** ее следует включить в схему данных. Для этого надо загрузить схему данных, выбрать из контекстного меню 'Добавить таблицу' и добавить в схему данных таблицу **Связи**. В результате сформируется окно:



После этого необходимо сформировать связь один ко многим между таблицами **Издательства** и **Связи** и сформировать связь один ко многим между таблицами **Книги** **Связи**. В результате сформируется следующая схема данных:



Как видно из рисунка, между таблицами **Издательства** и **Книги** сформирована связь многие ко многим.

При формировании запроса на основе этой схемы данных можно просмотреть все издательства и книги, которые в них изданы.

Задание на лабораторную работу.

16. Сформировать таблицу путем ввода данных, задействовав в ней поля различных типов.
17. Переименовать эти поля.
18. Просмотреть таблицу в режиме конструктора, изменить свойства полей.
19. Создать таблицы с помощью мастера.
20. Просмотреть таблицу в режиме конструктора, изменить свойства полей.
21. Создать 2-е таблицы в режиме конструктора. При этом задействовать все типы полей и все свойства полей. В одной из таблиц назначить первичный ключ, в другой – первичный и внешний ключи.
22. Загрузить схему данных и добавить в схему связываемые таблицы.
23. Сформировать связь один к одному.
24. Задействовать флажки и параметры связи во всех сочетаниях.
25. Для каждого сочетания выполнить ввод данных в таблицы и проанализировать результаты ввода.
26. Сформировать связь один ко многим.
27. Задействовать флажки и параметры связи во всех сочетаниях.
28. Для каждого сочетания выполнить ввод данных в таблицы и проанализировать результаты ввода.
29. Создать таблицу с помощью мастера для организации связи многие ко многим.
30. Загрузить схему данных и добавить в схему эту таблицу.
31. Сформировать связь многие ко многим.
32. Задействовать флажки и параметры связей во всех сочетаниях.
33. Для каждого сочетания выполнить ввод данных в таблицы и проанализировать результаты ввода.

Проектирование запросов в Access 2002 с помощью Конструктора.

Введение

Запросы, наряду с таблицами, являются основными объектами БД. По сути БД и разрабатывается для того, чтобы пользователь имел возможность запрашивать необходимую информацию в нужном виде.

Запросы могут строиться на базе нескольких таблиц. При этом выборка может осуществляться по столбцам, по строкам или в соответствии с комплексным критерием выборки.

Кроме отображения необходимых данных запросы могут быть задействованы для обновления данных, удаления данных, создания новых таблиц, добавления данных одной таблицы к записям другой.

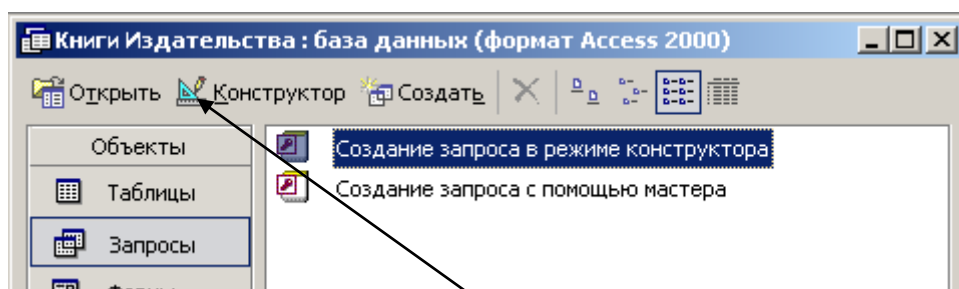
В запросах можно выполнять вычисления с использованием значений полей, сортировать выводимые данные по значениям полей.

На базе запросов, также как и на базе таблиц, могут быть спроектированы формы и отчеты. Некоторые запросы могут быть использованы для ввода данных.

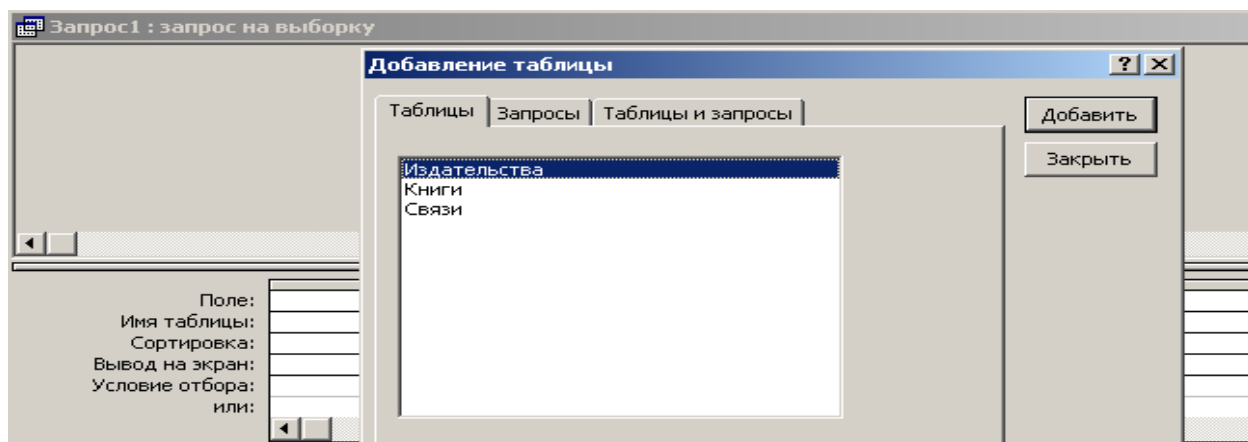
В Access, как и во многих СУБД, запросы можно проектировать с помощью бланка по образцу (QBE), а также посредством языка SQL.

2. Проектирование запросов на выборку

Базовое окно Access 2002 после выбора объектов **Запросы** выглядит следующим образом:

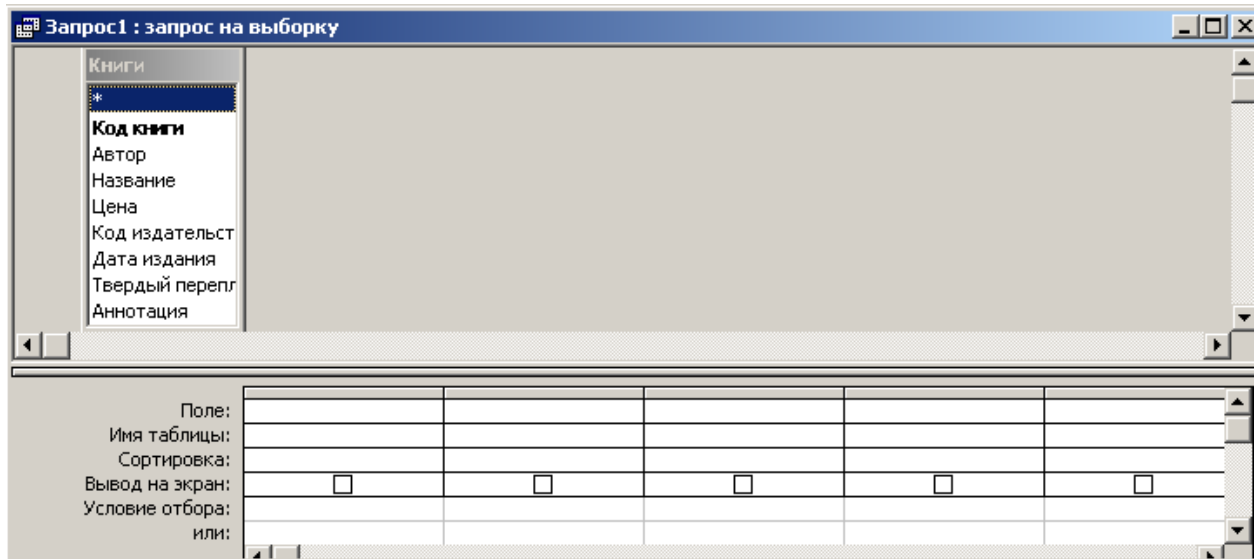


Для того чтобы инициировать проектирование запроса в режиме **Конструктора** необходимо щелкнуть по строке **Создание запроса в режиме конструктора**. Первый запрос можно создать посредством инструмента **Конструктор**. После инициации формирования запроса загрузится окно:

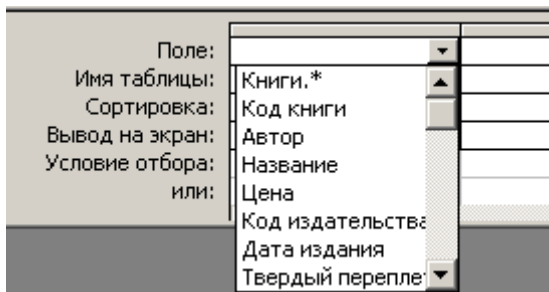


Запрос можно формировать на базе таблицы или на базе другого запроса. Для выбора типа объекта, на базе которого строится запрос, используются 3-и вкладки. На вкладке **Таблица** представлены все таблицы, которые имеются в БД. Чтобы добавить в бланк

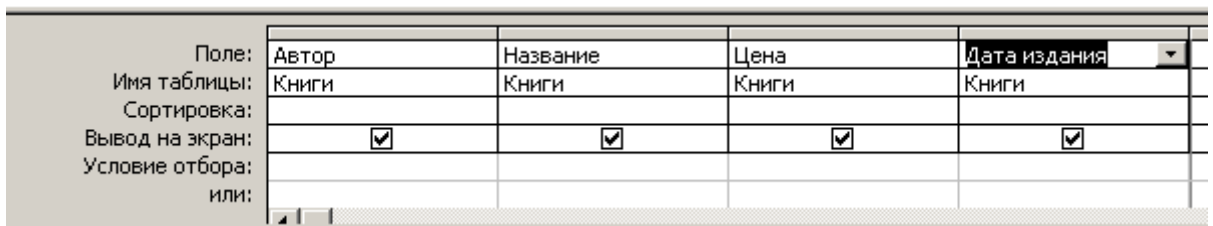
запроса таблицу, нужно ее выделить и щелкнуть по кнопке **Добавить**. После добавления всех таблиц, на базе которых предполагается строить запрос, нужно щелкнуть по кнопке **Закреть**. Результат добавления таблицы представлен на рисунке:




Для включения поля в запрос можно дважды щелкнуть по нужному полю из списка полей таблицы **Книги**. Кроме того, для выбора нужного поля можно использовать список, который формируется после щелчка в поле бланка **Поле:**



Если выбрать “*”, то запрос будет формироваться для всех полей. В приведенном примере задействованы 4-е поля:



В строке бланка запроса **Вывод на экран** можно отменить отображение данных из любого поля и использовать его, например, для вычисления значения.

Результат выполнения запроса можно просмотреть, не покидая **Конструктора**. Для этого надо щелкнуть по инструменту  (**запуск**). Результат запуска данного запроса следующий:

Запрос1 : запрос на выборку				
	Автор	Название	Цена	Дата издания
▶	Шиллер	Разбойники	120,00р.	01.01.2002
	Шекспир	Отелло	55,00р.	01.01.2003
	Мильтон	Потерянный рай	77,00р.	02.02.2002
	Данте	Божественная комедия	66,00р.	02.02.2002
	Гомер	Эллиада	55,00р.	02.02.2002
	Гомер	Одиссея	55,00р.	02.02.2003
	Грин	Алые паруса	77,00р.	01.01.2004
	Жюль Верн	Машина времени	90,00р.	01.01.2004
*			0,00р.	

Как видно из рисунка, записи не отсортированы. Для выполнения сортировки в бланке запроса используется строка **Сортировка**:

Поле:	Автор	Название	Цена	Дата издания
Имя таблицы:	Книги	Книги	Книги	Книги
Сортировка:	по возрастанию			
Вывод на экран:	<input checked="" type="checkbox"/>	по возрастанию	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		по убыванию		
или:		(отсутствует)		

Результат запуска данного запроса следующий:

	Автор	Название	Цена	Дата издания
▶	Гомер	Одиссея	55,00р.	02.02.2003
	Гомер	Эллиада	55,00р.	02.02.2002
	Грин	Алые паруса	77,00р.	01.01.2004
	Данте	Божественная комедия	66,00р.	02.02.2002
	Жюль Верн	Машина времени	90,00р.	01.01.2004
	Мильтон	Потерянный рай	77,00р.	02.02.2002
	Шекспир	Отелло	55,00р.	01.01.2003
	Шиллер	Разбойники	120,00р.	01.01.2002

Для фильтрации записей таблицы в бланке запроса используется строка **Условие отбора**.

Условия отбора может выглядеть следующим образом:

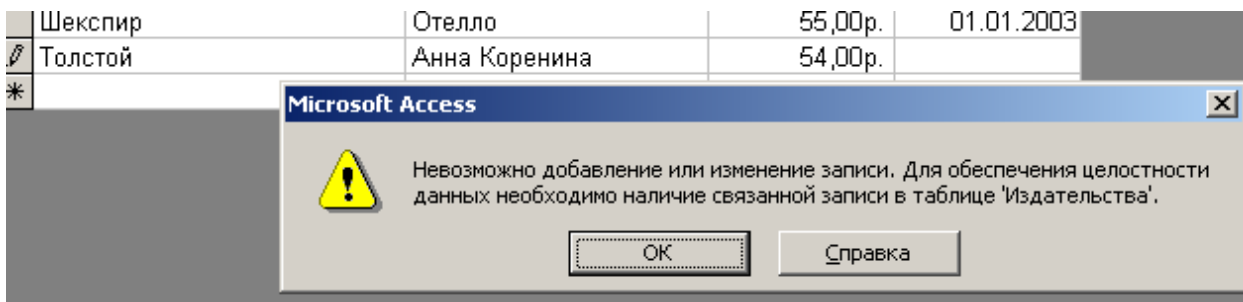
Поле:	Автор	Название	Цена	Дата издания
Имя таблицы:	Книги	Книги	Книги	Книги
Сортировка:	по возрастанию			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			<90	>#01.09.2002#
или:				

Условия отбора можно использовать, практически, для каждого поля. Тогда эти условия объединяются союзом **И (AND)**. Есть возможность объединять условия и союзом **ИЛИ (OR)**. Для этого в бланке запроса может быть задействована строка **или**. Причем таких строк можно задействовать практически неограниченное количество.

Для того, чтобы отличить поле типа дата от других типов полей его необходимо окружать символами **"#"**. Результат запуска данного запроса следующий:

	Автор	Название	Цена	Дата издания
▶	Гомер	Одиссея	55,00р.	02.02.2003
	Грин	Алые паруса	77,00р.	01.01.2004
	Шекспир	Отелло	55,00р.	01.01.2003
*			0,00р.	

Запросы на выборку нередко можно использовать для ввода данных, об этом свидетельствует последняя запись, которая заполнена только значением по умолчанию для поля **Цена**. Однако, в некоторых случаях ввод данных в запросе невозможен. В частности, если в таблице имеется поле, которое ссылается на соответствующее ключевое поле другой таблицы, то это поле, как правило, является обязательным для заполнения. В рассматриваемом случае это **Код издательства**. Оно в запросе не предусмотрено и поэтому этот запрос нельзя использовать для ввода данных. При попытке добавления записи возникнет сообщение об ошибке:



Этот и другие запросы можно просмотреть в формате SQL, заодно и изучить основные конструкции языка запросов. Для этого необходимо выбрать меню **Вид/Режим SQL**. В формате SQL данный запрос выглядит следующим образом:

```

Книги выборка : запрос на выборку
SELECT Книги.Автор, Книги.Название, Книги.Цена, Книги.[Дата издания]
FROM Книги
WHERE (((Книги.Цена)<90) AND ((Книги.[Дата издания])>#9/1/2002#))
ORDER BY Книги.Автор;

```

Команда **SELECT** служит для указания на то, что этот запрос является запросом на выборку данных. После **SELECT** перечисляются выводимые поля. Так как поля могут выводиться из нескольких таблиц, то перед полем указывается имя таблицы. Конструкция **FROM** указывает таблицу, из которой выводятся данные. Конструкция **WHERE** используется для указания условий отбора. Конструкция **ORDER BY** используется для указания поля, по которому осуществляется сортировка. В квадратные скобки берутся имена полей, в которых присутствуют пробелы. В этом режиме запрос можно редактировать, например, указать еще одно поле для вывода, например, **Код издательства**:

```

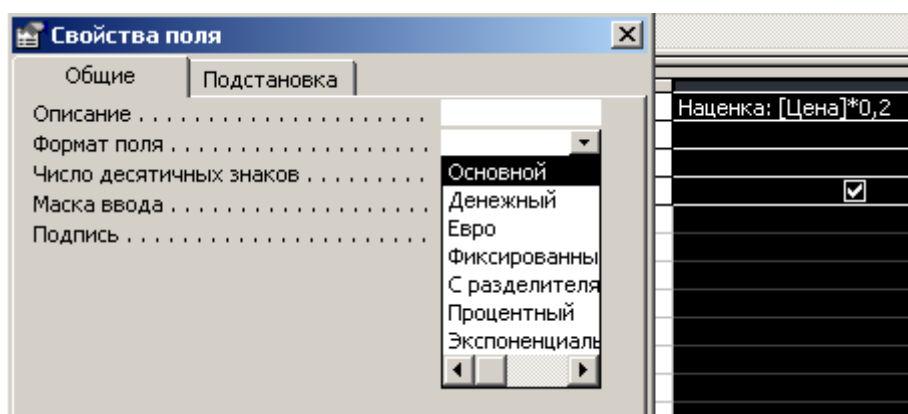
Книги выборка : запрос на выборку
SELECT Книги.Автор, Книги.Название, Книги.Цена, Книги.[Дата издания], Книги.[Код издательства]
FROM Книги
WHERE (((Книги.Цена)<90) AND ((Книги.[Дата издания])>#9/1/2002#))
ORDER BY Книги.Автор;

```

В запрос можно включить вычисляемое поле, например, поле **Наценка**:

Поле:	Автор	Название	Цена	Дата издания	Наценка: [Цена]*0,2
Имя таблицы:	Книги	Книги	Книги	Книги	
Сортировка:	по возрастанию				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			<90	>#01.09.2002#	
или:					

Чтобы формат вывода наценки был денежный необходимо обратиться к свойству данного поля и выбрать **Формат поля Денежный**. Для этого надо выделить столбец **Наценка**, щелкнуть правой клавишей мыши и выбрать позицию контекстного меню **Свойства**:

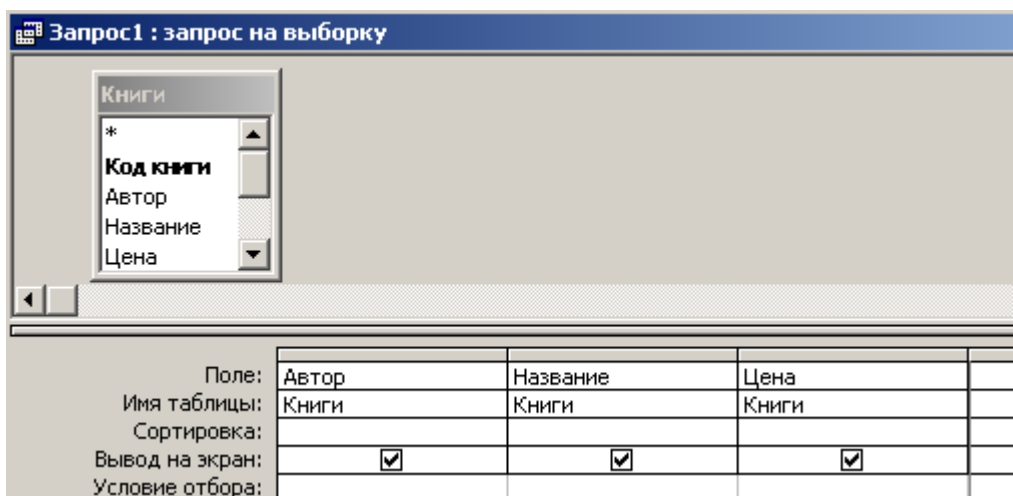


Результат запуска данного запроса следующий:

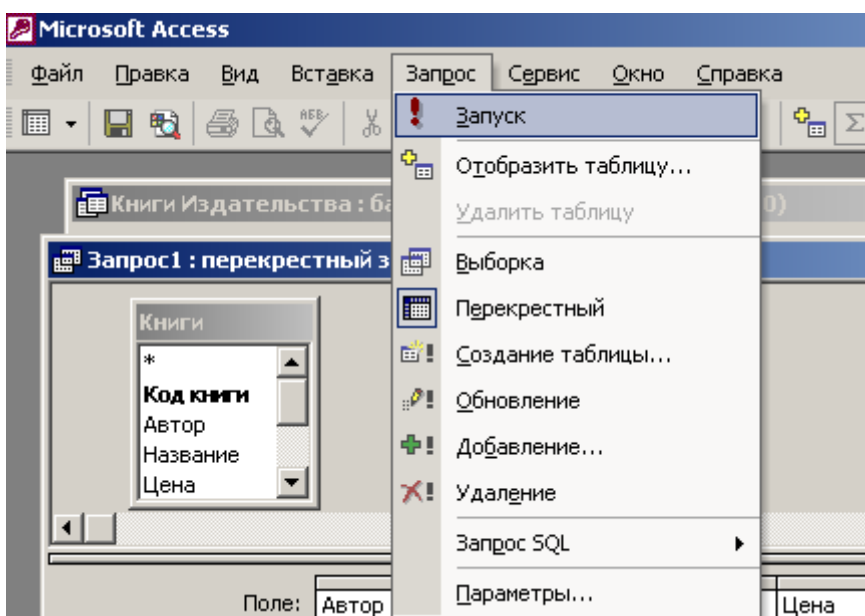
	Автор	Название	Цена	Дата издания	Наценка
▶	Гомер	Одиссея	55,00р.	02.02.2003	11,00р.
	Грин	Алые паруса	77,00р.	01.01.2004	15,40р.
	Шекспир	Отелло	55,00р.	01.01.2003	11,00р.

3. Проектирование перекрестных запросов

Перекрестные запросы подобны таблицам Excel, в которых присутствуют заголовки строк и заголовки столбцов. В ячейках таблицы располагаются числовые данные. Примером такой таблицы может быть ведомость зарплаты. Заголовки строк - фамилии, заголовки столбцов – даты, в ячейках располагаются суммы. В Access обычную таблицу можно преобразовать в перекрестную таблицу (запрос). В качестве примера для преобразования используется следующий запрос на выборку:



Для преобразования запроса в перекрестный необходимо выбрать меню **Запрос/Перекрестный**:



После этого запрос преобразуется в следующий вид:

Поле:	Автор	Название	Цена
Имя таблицы:	Книги	Книги	Книги
Групповая операция:	Группировка	Группировка	Группировка
Перекрестная таблица:			
Сортировка:			
Условие отбора:			

Как видно из рисунка, в бланке запроса возникли 2-е новые строки: **Групповая операция** и **Перекрестная таблица**. По умолчанию в строке **Групповая операция** установлено значение **Группировка**. **Группировка** означает то, что записи группируются по одинаковым значениям. Например, если в таблице **Книги** есть несколько одноименных книг одного и того же автора, то при группировке по **Автору** и по **Названию** в перекрестном запросе эта одноименная книга будет представлена один раз. По полю **Цена** группировать данные не имеет смысла, в этом поле имеет смысл выбрать из предлагаемого списка сумму (Sum):

Поле:	Автор	Название	Цена
Имя таблицы:	Книги	Книги	Книги
Групповая операция:	Группировка	Группировка	Sum
Перекрестная таблица:			
Сортировка:	Заголовки строк		
Условие отбора:	Заголовки столбцов		
или:	Значение (не отображается)		

В строке бланка запроса **Перекрестная таблица** для всех полей необходимо выбрать из списка их назначение. Результат выбора:

Поле:	Автор	Название	Цена
Имя таблицы:	Книги	Книги	Книги
Групповая операция:	Группировка	Группировка	Sum
Перекрестная таблица:	Заголовки строк	Заголовки столбцов	Значение

Для следующих данных представленных в таблице Книги

Книги : таблица					
	Код книги	Автор	Название	Цена	Код издательс
▶	11	Гомер	Одиссея	65,00р.	1
	6	Гомер	Одиссея	55,00р.	3
	5	Гомер	Эллиада	55,00р.	3
	12	Грин	Алые паруса	83,00р.	3
	7	Грин	Алые паруса	77,00р.	1
	4	Данте	Божественная комедия	66,00р.	3
	8	Жюль Верн	Машина времени	90,00р.	1
	3	Мильтон	Потерянный рай	77,00р.	3
	2	Шекспир	Отелло	55,00р.	2
	1	Шиллер	Разбойники	120,00р.	2
*	(Счетчик)			0,00р.	0

спроектированный перекрестный запрос в режиме просмотра будет выглядеть следующим образом:

	Автор	Алые паруса	Божественная	Машина време	Одиссея	Отелло
▶	Гомер				120,00р.	
	Грин	160,00р.				
	Данте		66,00р.			
	Жюль Верн			90,00р.		
	Мильтон					
	Шекспир					55,00р.
	Шиллер					

Как видно из результатов запроса, суммарная стоимость книг Гомера “Одиссея” 120р., а суммарная стоимость книг Грина “Алые паруса” 160 р. Этот пример, возможно, не очень удачен, но не трудно придумать множество примеров, когда перекрестные запросы могут быть очень полезны.

В режиме SQL перекрестный запрос выглядит следующим образом:

```

TRANSFORM Sum(Книги.Цена) AS [Sum-Цена]
SELECT Книги.Автор
FROM Книги
GROUP BY Книги.Автор
PIVOT Книги.Название;

```

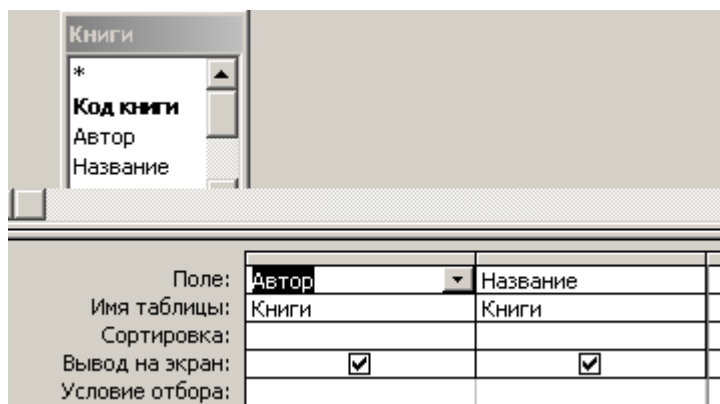
Ключевое слово **TRANSFORM** обозначает перекрестный запрос. Конструкция **Sum(Книги.Цена)** указывает на то, что необходимо выполнять сложение значений полей **Цена** для одноименных строк и столбцов запроса. Конструкция **AS [Sum-Цена]** используется для назначения суммарной цене имени **Sum-Цена**. Конструкция **SELECT Книги.Автор FROM Книги** указывает на то, что выбираются данные об авторах из таблицы **Книги**. Конструкция **GROUP BY Книги.Автор** указывает на то, что данные группируются по значению поля **Автор** и в качестве заголовков строк перекрестной таблицы используется значения поля **Автор** таблицы **Книги**. Конструкция **PIVOT Книги.Название** указывает на то, что в качестве привязки к столбцам запроса используется поле **Название** таблицы **Книги**.

4. Проектирование запроса на создание таблицы

Существует немало причин, приводящих к необходимости создания таблиц на основе одной или нескольких таблиц. В частности, бывает необходимо сформировать и переслать вертикальный или горизонтальный срез какой-либо информации, или бывает полезно

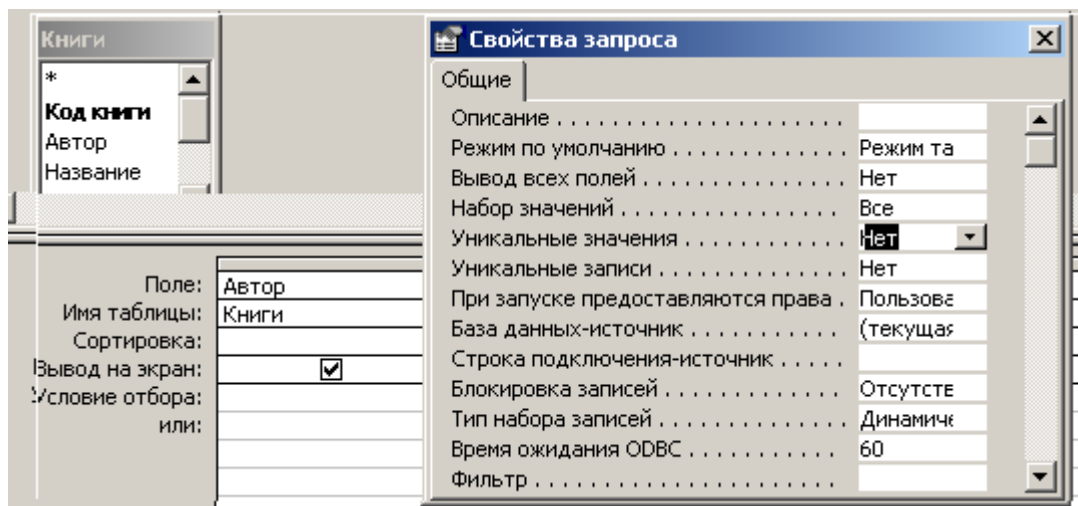
сформировать таблицу, которую можно использовать в качестве временного хранилища данных.

Запрос на создание таблицы строится на основе запроса на выборку. В качестве запроса на выборку в примере используется следующий запрос:

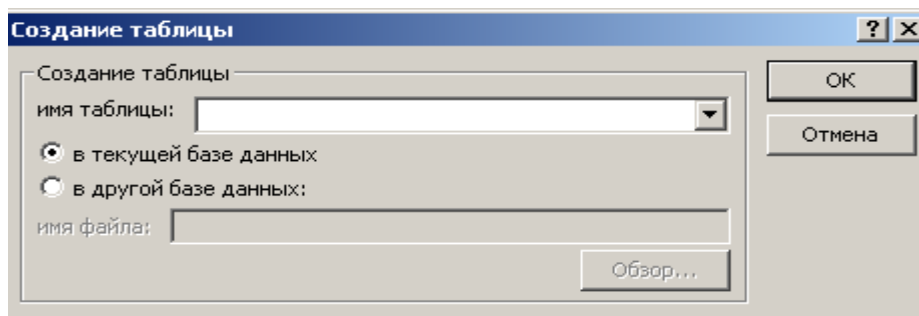


Здесь, как и во многих других типах запросов, можно задействовать строки бланка **Сортировка** и **Условие отбора**.

Предполагается, что в новой таблице будут храниться книги и их авторы. Однако, в исходной таблице встречаются повторения названных сочетаний. В свойствах запроса можно исключить дублирование. Для доступа к свойствам запроса необходимо дважды щелкнуть по серой области бланка и в строке **Уникальные значения** выбрать **Да**:



Для преобразования этого запроса в запрос на создание таблицы необходимо обратиться к меню **Запрос/Создание таблицы**. После обращения к данному меню будет запрошено имя создаваемой таблицы:



Как видно из рисунка, таблицу можно создать и в другой базе данных. После ввода имени новой таблицы и выполнения запроса сформируется новая таблица, содержимое которой имеет вид:

	Автор	Название
▶	Гомер	Одиссея
	Гомер	Эллиада
	Грин	Алые паруса
	Данте	Божественная комедия
	Жюль Верн	Машина времени
	Мильтон	Потерянный рай
	Шекспир	Отелло
	Шиллер	Разбойники
*		

Для сохранения запроса, сформированного в режиме конструктора, также как и для сохранения любого объекта можно использовать меню **Файл/Сохранить**. Для открытия запроса, как и любого другого объекта, можно дважды щелкнуть по значку запроса в базовом окне Access.

Спроектированный в бланке запрос в формате SQL выглядит следующим образом:

```
SELECT DISTINCT Книги.Автор, Книги.Название INTO [Книги без дублирования]
FROM Книги;
```

Конструкции **SELECT DISTINCT Книги.Автор, Книги.Название** позволяют выбрать поля **Книги** и **Автор** из таблицы **Книги**. Ключевое слово **DISTINCT** позволяет исключить дублирование записей. Конструкции **INTO [Книги без дублирования] FROM Книги** обеспечивает создание новой таблицы **Книги без дублирования** из таблицы **Книги**.

5. Проектирование запроса на добавление

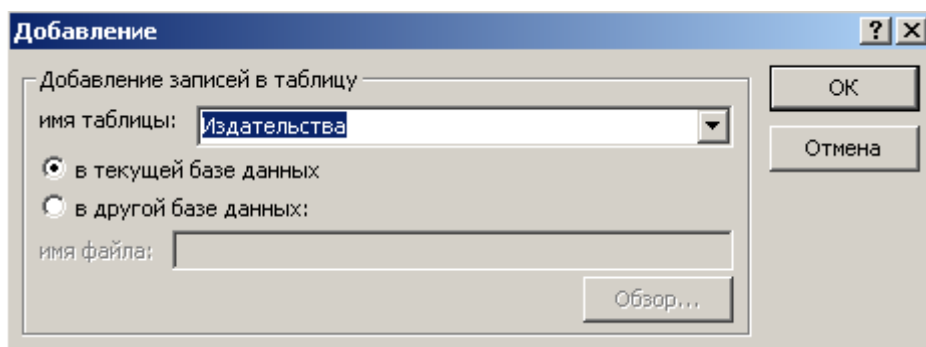
Запрос на добавление позволяет добавить записи одной таблицы к записям другой таблицы. Запрос такого рода нередко может оказаться полезным. Например, по электронной почте поступают данные о вновь открывшихся издательствах в виде таблиц. Их нужно добавлять в обобщенную таблицу **Издательства**.

Запрос на добавление, как и многие другие запросы, строится на основе запроса на выборку. Спроектирован запрос на выборку на базе таблицы **Издательства1**:

Поле:	Код издательства	Наименование	Адрес	Логотип	Сайт
Имя таблицы:	Издательства1	Издательства1	Издательства1	Издательства1	Издательства1
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:					

В данном запросе можно использовать строку **Условие отбора**.

Для формирования запроса на добавление необходимо выбрать меню **Запрос/Добавление**. После этого откроется уже знакомое окно, в котором нужно указать имя таблицы приемника:



После щелчка по кнопке **ОК** запрос примет вид:

Поле:	Код издательства	Наименование	Адрес	Логотип	Сайт
Имя таблицы:	Издательства1	Издательства1	Издательства1	Издательства1	Издательства1
Сортировка:					
Добавление:	Код издательства	Наименование	Адрес	Логотип	Сайт
Условие отбора:					

В бланке запроса появилось новая строка **Добавление**. В этой строке представлены поля таблицы приемника. Так как в рассматриваемом случае структуры обеих таблиц совпадают, то в строке **Добавление** исправлять ничего не надо, В противном случае исправления могут потребоваться.

Содержимое таблицы **Издательства** до выполнения запроса было следующее:

	Код издательств	Наименование	Адрес	Логотип	Сайт
▶	+	1 Мир	Бауманская ,2	soft PowerPoint	6666@aaa.ru
	+	2 Прометей	Подбельского,	Visio.Drawing.6	
	+	3 Эксмо	11-я парковая,;		111@222

Содержимое таблицы **Издательства1** до выполнения запроса было следующее:

	Код издательств	Наименование	Адрес	Логотип	Сайт
▶		4 Машиностроен	Королева,15		
		5 Энергия	Уральская,77		

После выполнения запроса содержимое таблицы **Издательства** примет вид:

	Код издательств	Наименование	Адрес	Логотип	Сайт
▶	+	1 Мир	Бауманская ,2	soft PowerPoint	6666@aaa.ru
	+	2 Прометей	Подбельского,	Visio.Drawing.6	
	+	3 Эксмо	11-я парковая,;		111@222
	+	4 Машиностроен	Королева,15		
	+	5 Энергия	Уральская,77		

В формате SQL спроектированный запрос имеет вид:

```
INSERT INTO Издательства ( [Код издательства], Наименование, Адрес, Логотип,
Сайт )
SELECT Издательства1.[Код издательства], Издательства1.Наименование,
Издательства1.Адрес, Издательства1.Логотип, Издательства1.Сайт
FROM Издательства1;
```

Конструкции **SELECT Издательства1.[Код издательства], Издательства1.Наименование, Издательства1.Адрес, Издательства1.Логотип, Издательства1.Сайт FROM Издательства1** позволяют выбрать из таблицы **Издательства1** 5 полей.

Конструкции **INSERT INTO Издательства ([Код издательства], Наименование, Адрес, Логотип, Сайт)** позволяют добавить значения выбранных полей в таблицу **Издательства** в соответствии с перечисленными полями этой таблицы.

6. Проектирование запроса на обновление

Нередко требуется изменить значения каких-либо полей. Например, нужно увеличить цену всех книг на какое-либо значение или уменьшить стоимость дорогих книг.

Для реализации операций такого рода над данными используются запросы на добавление.

Запросы на добавление формируются на основе запроса на выборку. В качестве примера используется следующий запрос на выборку:

The screenshot shows a query editor window. On the left, a list of fields is visible: * (selected), Код книги, Автор, Название, and Цена. Below this, a table defines the query parameters:

Поле:	Цена	
Имя таблицы:	Книги	
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		

Для преобразования запроса на выборку в запрос на обновление необходимо выбрать меню **Запрос/Обновление**. После этого бланк запроса примет вид:

The screenshot shows the same query editor window, but the 'Обновление' (Update) option is now selected in the table below the field list:

Поле:	Цена	
Имя таблицы:	Книги	
Обновление:		
Условие отбора:		
или:		

Как видно из рисунка, в бланке запроса появилась строка **Обновление**. В этой строке можно ввести выражение для обновления. Например:

The screenshot shows the query editor with the 'Обновление' field populated with the expression '[Цена]+10':

Поле:	Цена	
Имя таблицы:	Книги	
Обновление:	[Цена]+10	
Условие отбора:		

Для того, чтобы поле отличить от текста в выражении его необходимо окружить квадратными скобками

После выполнения этого запроса стоимость всех книг увеличится на 10 рублей.

В результате содержимое таблицы **Книги** примет вид:

	Код книги	Автор	Название	Цена	Код издательс	Дата и
▶	11	Гомер	Одиссея	75,00р.	1	
	6	Гомер	Одиссея	65,00р.	3	02.0
	5	Гомер	Эллиада	65,00р.	3	02.0
	12	Грин	Алые паруса	93,00р.	3	
	7	Грин	Алые паруса	87,00р.	1	01.0
	4	Данте	Божественная комедия	76,00р.	3	02.0
	8	Жюль Верн	Машина времени	100,00р.	1	01.0
	3	Мильтон	Потерянный рай	87,00р.	3	02.0
	2	Шекспир	Отелло	65,00р.	2	01.0
	1	Шиллер	Разбойники	130,00р.	2	01.0

В запросе на обновление нередко используют условие отбора. Например, требуется уменьшить на 5 рублей книги, которые стоят больше 70-и рублей. Для этого необходим запрос вида:

Поле:	Цена
Имя таблицы:	Книги
Обновление:	[Цена]-5
Условие отбора:	>70

В результате выполнения этого запроса содержимое таблицы **Книги** примет вид:

	Код книги	Автор	Название	Цена	Код издательс
▶	11	Гомер	Одиссея	70,00р.	1
	6	Гомер	Одиссея	65,00р.	3
	5	Гомер	Эллиада	65,00р.	3
	12	Грин	Алые паруса	88,00р.	3
	7	Грин	Алые паруса	82,00р.	1
	4	Данте	Божественная комедия	71,00р.	3
	8	Жюль Верн	Машина времени	95,00р.	1
	3	Мильтон	Потерянный рай	82,00р.	3
	2	Шекспир	Отелло	65,00р.	2
	1	Шиллер	Разбойники	125,00р.	2

В формате SQL данный запрос имеет вид:

```
UPDATE Книги SET Книги.Цена = [Цена]-5
WHERE (((Книги.Цена)>70));
```

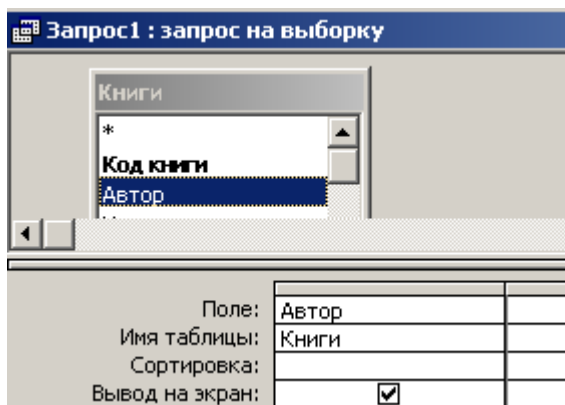
Конструкция **UPDATE Книги** свидетельствует о том, что обновляется таблица **Книги**. Конструкция **SET Книги.Цена = [Цена]-5** обозначает уменьшение прежней цены на 5. Конструкция **WHERE (((Книги.Цена)>70))** указывает на то, какие поля подлежат обновлению. В данном случае все скобки после ключевого слова **WHERE** можно удалить. Они сформированы автоматически на случай использования сложного выражения.

7. Проектирование запроса на удаление

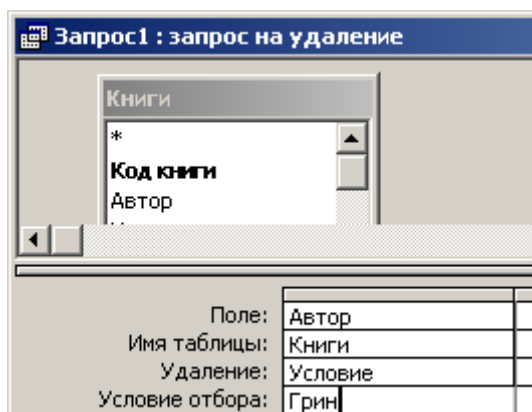
В процессе работы с БД иногда возникает необходимость удаления некоторых записей таблицы. Например, нужно удалить книги какого-либо автора или удалить книги какого-либо издательства.

Для реализации операций такого рода над данными используются запросы на удаление.

Запросы на удаление формируются на основе запроса на выборку. В качестве примера используется следующий запрос на выборку:

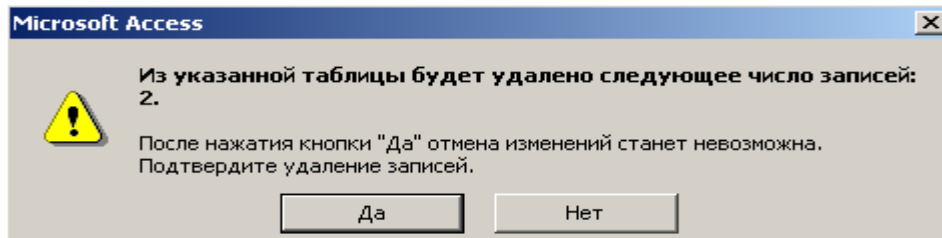


Для преобразования запроса на выборку в запрос на удаление необходимо обратиться к меню **Запрос/Удаление**. В результате запрос преобразуется к следующему виду:



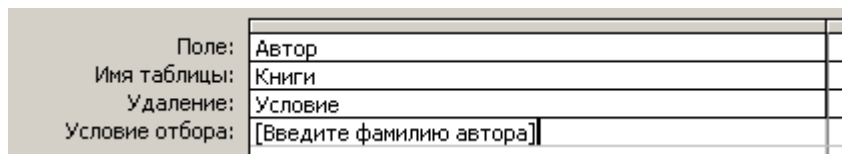
Как видно из рисунка, бланк запроса изменился. Если в строке Условие отбора ничего не вводить, то при выполнении запроса очистится вся таблица. Бывают такие случаи, когда это нужно. Например, нельзя добавить записи к несуществующей таблице, а к пустой таблице, из которой удалены все записи, можно. Но, как правило, удаляют не все записи

таблицы, а записи соответствующие введенным критериям. В данном случае в качестве критерия используется автор **Грин**. Следует помнить, что удаление записей операция необратимая, поэтому проектировать и использовать запросы такого рода нужно аккуратно. Правда, стандартные средства Access, если они не отменены разработчиком БД, предупреждают пользователя о результатах, к которым приведут запросы на создание таблицы, добавление, обновление, удаление. Например, при попытке выполнить данный запрос отобразится окно:

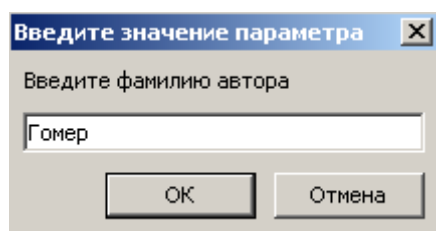


Пользователь БД может подтвердить или отменить удаление записей.

При выполнении запроса на удаление и многих других типов запросов зачастую удобно использовать не фиксированное условие отбора, а использовать вводимое пользователем условие отбора. Для этого применяются параметрические запросы. Для обозначения параметра условие отбора берется в квадратные скобки. Например:



В качестве параметра используется предложение **Введите параметры автора**. Такое сложное имя параметра выбрано в связи с тем, что это имя выводится при запросе значения параметра. При обращении к данному запросу сформируется форма, в которой пользователь должен указать фамилию автора:



В результате выполнения запроса содержимое таблицы **Книги** примет вид:

Книги : таблица					
	Код книги	Автор	Название	Цена	Код издательс
▶	12	Грин	Алые паруса	88,00р.	3
	7	Грин	Алые паруса	82,00р.	1
	4	Данте	Божественная комедия	71,00р.	3
	8	Жюль Верн	Машина времени	95,00р.	1
	3	Мильтон	Потерянный рай	82,00р.	3
	2	Шекспир	Отелло	65,00р.	2
	1	Шиллер	Разбойники	125,00р.	2

Как видно из рисунка, удалены записи с книгами Гомера. Условий отбора и, соответственно, параметров может быть несколько. Если задействовано несколько параметров, то при выполнении запроса значение каждого параметра запрашивается.

В формате SQL данный запрос выглядит следующим образом:

```
DELETE Книги.Автор
FROM Книги
WHERE (((Книги.Автор)=[Введите фамилию автора]));
```

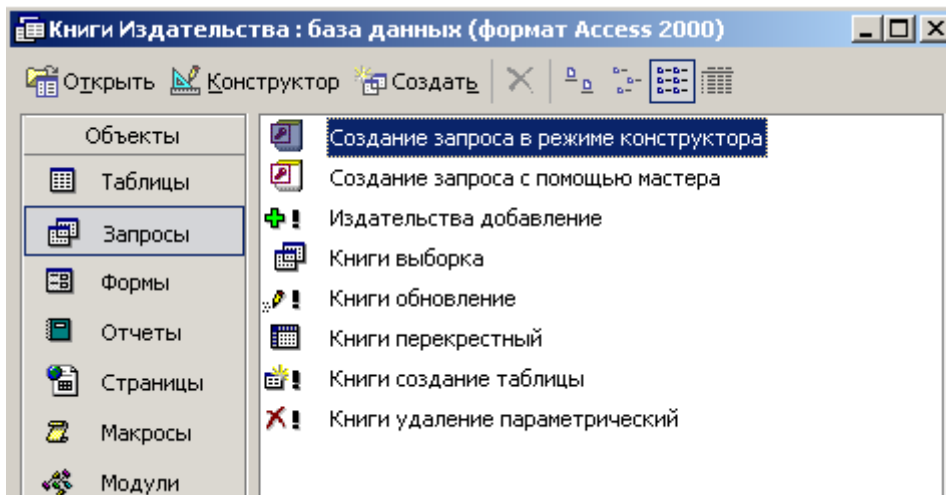
Конструкция `DELETE Книги.Автор FROM Книги` свидетельствует о том, что удаляются записи из таблицы Книги, а в качестве поля для условия отбора используется Автор. Конструкция `WHERE (((Книги.Автор)=[Введите фамилию автора]))` указывает условие отбора.

При проектировании запросов Access сам назначает им имена по умолчанию. Запросу приписывается имя Запрос1. Если такое имя уже есть, то запросу приписывается имя Запрос2 и так далее. Это характерно и для других объектов. При сохранении запроса в предлагаемой форме можно ввести другое имя:



Имеет смысл присваивать запросам информативные поля, для того, чтобы при проектировании объектов на базе спроектированных запросов было легче разобраться в назначении запросов.

Спроектированные запросы в базовом окне Access выглядят следующим образом:

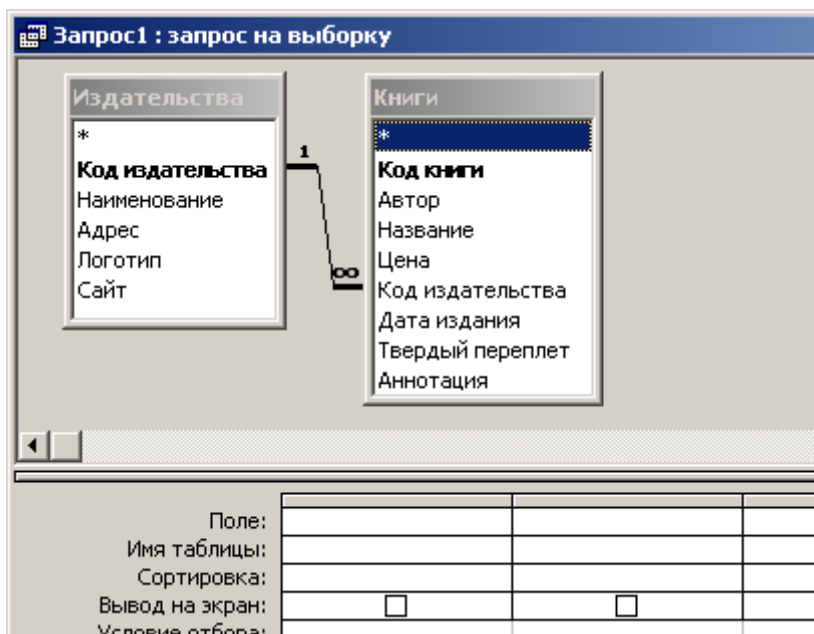


Как видно из рисунка, каждому типу запроса ставится в соответствии свой значок.

Запросы, как и другие объекты, легко переименовать. Это можно сделать с помощью меню **Правка/Переименовать**. Или посредством выделения нужного запроса, щелчка в области его имени и ввода нового имени.

8. Проектирование запросов на основе нескольких таблиц

В БД нередко требуется выводить записи из нескольких таблиц. Для проектирования запроса на основе нескольких таблиц необходимо добавить в бланк запроса эти таблицы. В качестве примера в бланке запроса добавлены таблицы **Издательства** и **Книги**:



Между таблицами автоматически сформировалась связь. Это произошло потому, что эта связь была сформирована при построении схемы данных. Эту связь можно удалить. А

если в схеме данных связь не установлена или схема данных отсутствует, то ее можно сформировать в бланке запроса.

Для удаления связи необходимо щелкнуть по ней и нажать клавишу **Delete**. На рисунке приведен бланк запроса с удаленной связью и двумя выводимыми полями из разных таблиц:

Издательства	Книги
* Код издательства Наименование Адрес Логотип Сайт	* Код книги Автор Название Цена Код издательства Дата издания Твердый переплет Аннотация
Поле:	Наименование Автор
Имя таблицы:	Издательства Книги

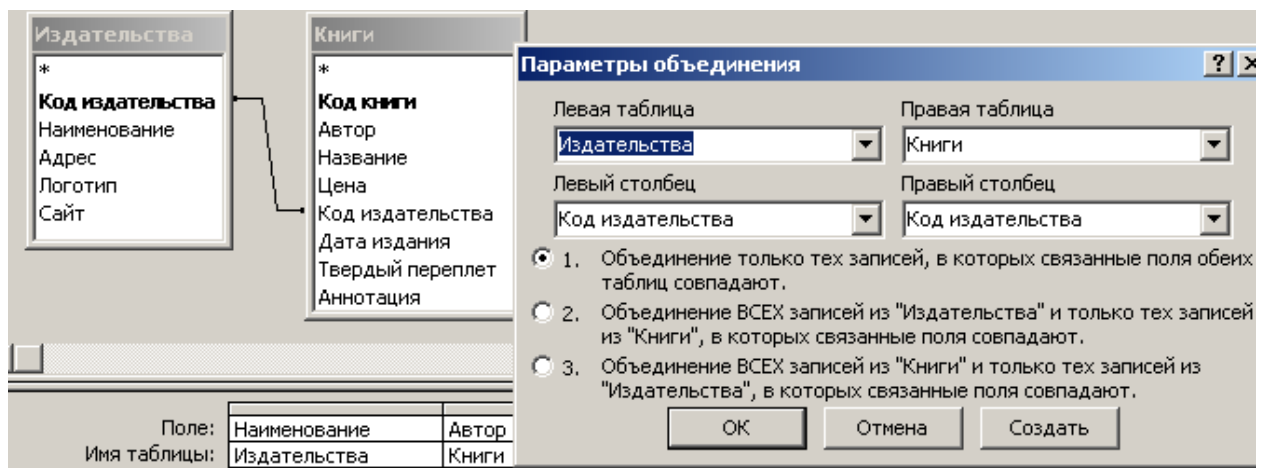
Результатом выполнения этого запроса будет декартово произведение 2-х полей – будет выведено каждое издательство со всеми авторами. Часть записей отображаемых при выполнении данного запроса приведена на рисунке:

	Наименование	Автор
▶	Мир	Шиллер
	Мир	Шекспир
	Мир	Мильтон
	Мир	Данте
	Мир	Грин
	Мир	Жюль Верн
	Мир	Грин
	Прометей	Шиллер
	Прометей	Шекспир
	Прометей	Мильтон
	Прометей	Данте
	Прометей	Грин
	Прометей	Жюль Верн
	Прометей	Грин
	Эксмо	Шиллер
	Эксмо	Шекспир

Всего сформировано 35 записей. Количество сформированных записей равно произведению количества книг в таблице **Книги** (7) на количество издательств в таблице **Издательства** (5).

Проектировать такой запрос для данных таблиц не имеет смысла. Но иногда запрос такого типа может оказаться полезным. Например, при использовании таблиц **Студенты** и **Дисциплины** аналогичным образом можно сформировать индивидуальные ведомости.

В нашем случае оправдано использование связи между таблицами. Для восстановления связи необходимо щелкнуть по внешнему ключу **Код издательства** таблицы **Издательства** и пробуксировать до внешнего ключа **Издательства** таблицы **Книги**. В Access можно задействовать один из 3-х типов связей. Для назначения типа связи необходимо дважды щелкнуть по связи, в результате чего отобразится форма параметры объединения:



Если выбрать 1-й тип объединения, то результат выполнения запроса будет иметь вид:

	Наименование	Автор
▶	Мир	Грин
	Мир	Жюль Верн
	Прометей	Шиллер
	Прометей	Шекспир
	Эксмо	Мильтон
	Эксмо	Данте
	Эксмо	Грин

В формате SQL этот запрос имеет вид:

```
SELECT Изда.Наименование, Книги.Автор
FROM Изда INNER JOIN Книги ON Изда.[Код издательства] =
Книги.[Код издательства];
```

Конструкция **SELECT** **Издательства.Наименование**, **Книги.Автор** позволяет выбрать поля **Наименования** из таблицы **Издательства** и **Автор** из таблицы **Книги**. Конструкции **FROM** **Издательства** **INNER JOIN** **Книги** **ON** **Издательства.[Код издательства] = Книги.[Код издательства]** указывает на то что между таблицами **Издательства** и **Книги** используется внутреннее соединение (**INNER JOIN**). При этом условием соединения является равенство ключевых полей таблиц (**Издательства.[Код издательства] = Книги.[Код издательства]**). Внутреннее соединение это такое соединение, при котором связанные поля обеих таблиц совпадают.

Если выбрать 2-й тип объединения, то результат выполнения запроса будет иметь вид:

	Наименование	Автор
▶	Мир	Грин
	Мир	Жюль Верн
	Прометей	Шиллер
	Прометей	Шекспир
	Эксмо	Мильтон
	Эксмо	Данте
	Эксмо	Грин
	Машиностроен	
	Энергия	

В формате SQL этот запрос имеет вид:

```
SELECT Издательства.Наименование, Книги.Автор
FROM Издательства LEFT JOIN Книги ON Издательства.[Код издательства] =
Книги.[Код издательства];
```

В данном запросе используется левое соединение (**LEFT JOIN**). Левое соединение - это такое соединение, при котором выбираются все записи левой таблицы, и только те записи правой таблицы, в которых все связанные поля совпадают.

Если выбрать 3-й тип объединения, то результат выполнения запроса будет иметь вид:

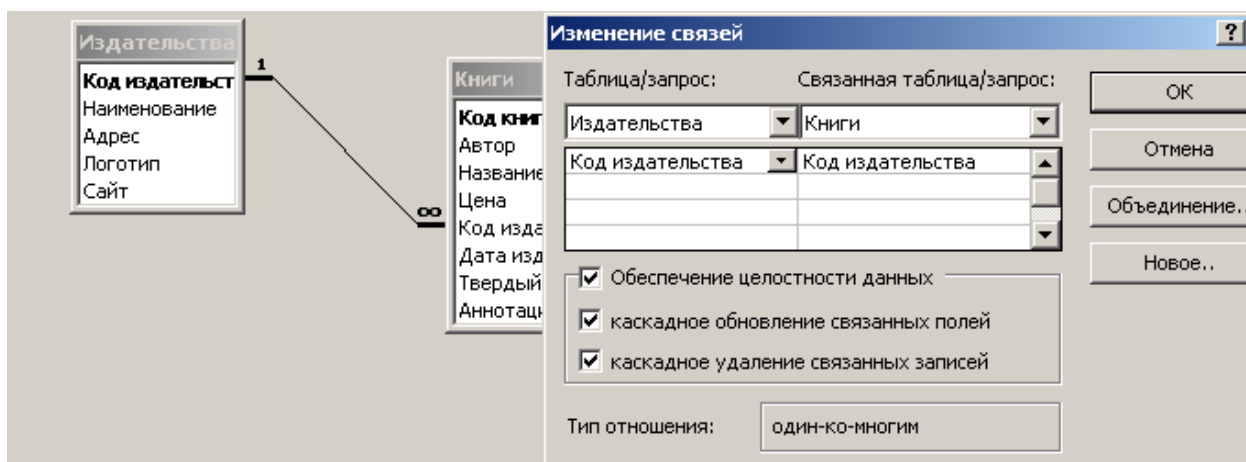
	Наименование	Автор
▶	Мир	Грин
	Мир	Жюль Верн
	Прометей	Шиллер
	Прометей	Шекспир
	Эксмо	Мильтон
	Эксмо	Данте
	Эксмо	Грин

В формате SQL этот запрос имеет вид:

```
SELECT Издательства.Наименование, Книги.Автор
FROM Издательства RIGHT JOIN Книги ON Издательства.[Код издательства] =
Книги.[Код издательства];
```

В данном запросе используется правое соединение (**RIGHT JOIN**). Правое соединение - это такое соединение, при котором выбираются все записи правой таблицы, и только те записи левой таблицы, в которых все связанные поля совпадают.

Из рисунков видно, что результаты выполнения запросов в 1-м и 3-м случаях аналогичны. Это произошло в связи с тем, что при вводе книг в таблицы **Книги** всегда вводился **Код издательства**. И поэтому в таблице **Книги** нет книг, в которых нет связанных записей в таблице **Издательства**. В рассматриваемом случае просто невозможно сформировать запись в таблице **Книги**, не указав **Код издательства**. Это обеспечено посредством схемы данных:



Флажок **Обеспечение целостности данных** обязывает пользователя вводить **Код издательства** при формировании записи в таблице **Книги**.

Задание на лабораторную работу.

1. Спроектировать запрос на выборку для одной таблицы и выполнить этот запрос.
2. В запросе на выборку выполнить сортировку по различным полям.
3. В запросе реализовать условия отбора для нескольких полей и просмотреть результаты выполнения запроса.
4. Просмотреть и проанализировать запрос в формате SQL.
5. Спроектировать перекрестный запрос. Выполнить этот запрос.
6. Просмотреть и проанализировать запрос в формате SQL.
7. Спроектировать запрос на создание таблицы. Выполнить этот запрос.
8. Просмотреть и проанализировать запрос в формате SQL.
9. Спроектировать запрос на добавление. Выполнить этот запрос.
10. Просмотреть и проанализировать запрос в формате SQL.
11. Спроектировать запрос на обновление. Выполнить этот запрос.
12. Просмотреть и проанализировать запрос в формате SQL.
13. Спроектировать запрос на удаление. Выполнить этот запрос.
14. Просмотреть и проанализировать запрос в формате SQL.
15. Спроектировать параметрический запрос. Выполнить этот запрос.
16. Просмотреть и проанализировать запрос в формате SQL.
17. Спроектировать запрос для нескольких таблиц с 1-м параметром объединения.
Выполнить этот запрос.
18. Просмотреть и проанализировать запрос в формате SQL.
19. Спроектировать запрос для нескольких таблиц со 2-м параметром объединения.
Выполнить этот запрос.
20. Просмотреть и проанализировать запрос в формате SQL.
21. Спроектировать запрос для нескольких таблиц с 3-м параметром объединения.
Выполнить этот запрос.
22. Просмотреть и проанализировать запрос в формате SQL.
23. Проанализировать свойства полей запроса.
24. Проанализировать свойства запроса.

Проектирование запросов в Access 2002 с помощью Мастеров.

Введение

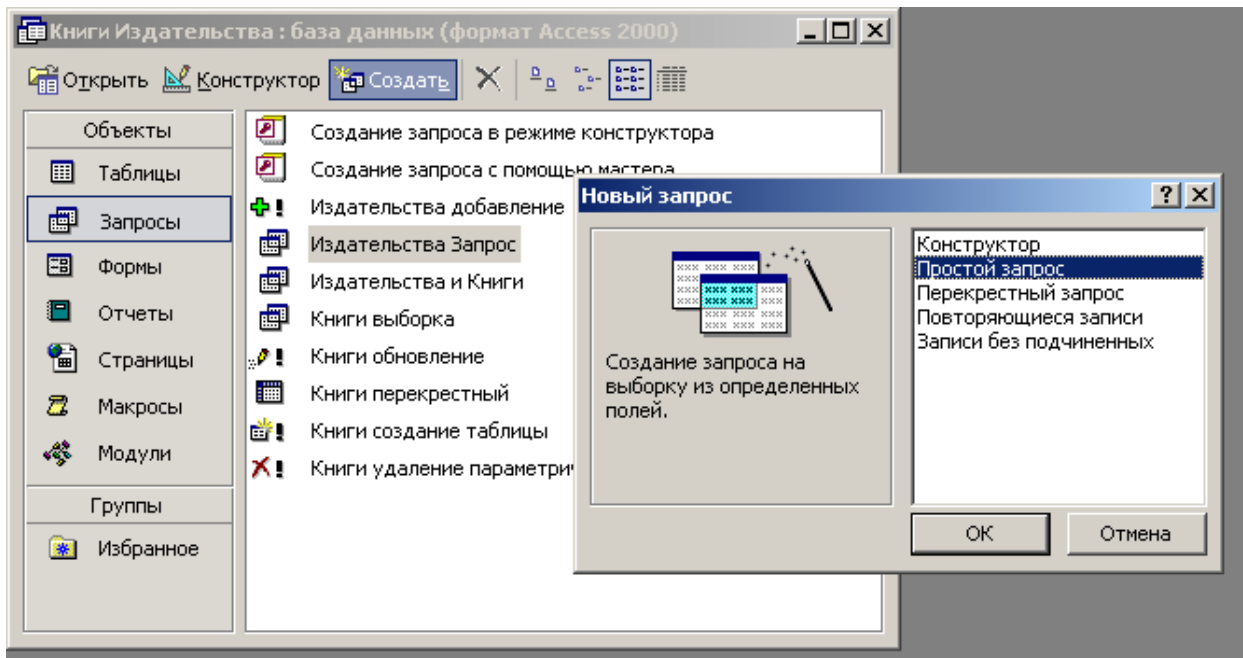
В некоторых случаях запросы удобнее строить с помощью мастеров. Использование мастера, как правило, существенно ускоряет процесс проектирования запроса и сводит к минимуму ошибки проектирования.

В Access 2002 можно использовать следующие мастера для создания запросов: Простой запрос, Перекрестный запрос, Повторяющиеся записи, Записи без подчиненных. Все эти типы запросов можно построить в режиме конструктора. Однако не всегда очевидно, как это делать.

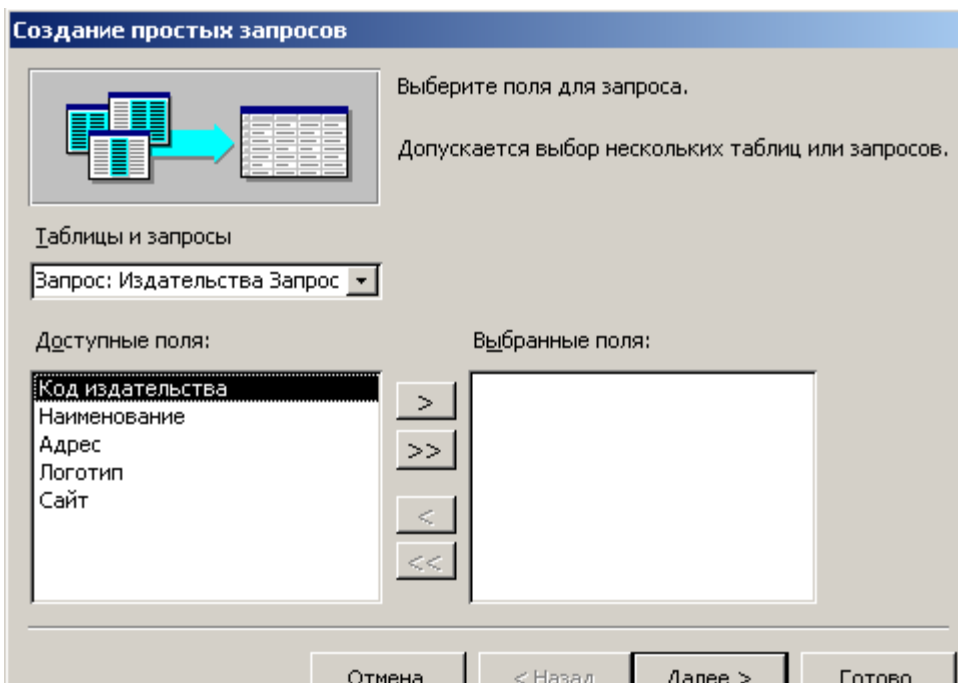
Разумным подходом к проектированию многих запросов является следующий подход – запросы строятся с помощью мастера, а затем модифицируются при необходимости посредством конструктора. Суть модификации для каждого конкретного случая своя и зависит от требований к запросу. Это может быть формирование критериев сортировки, формирование условий отбора, описание вычислений и другие модификации.







2. Формирование простого запроса.

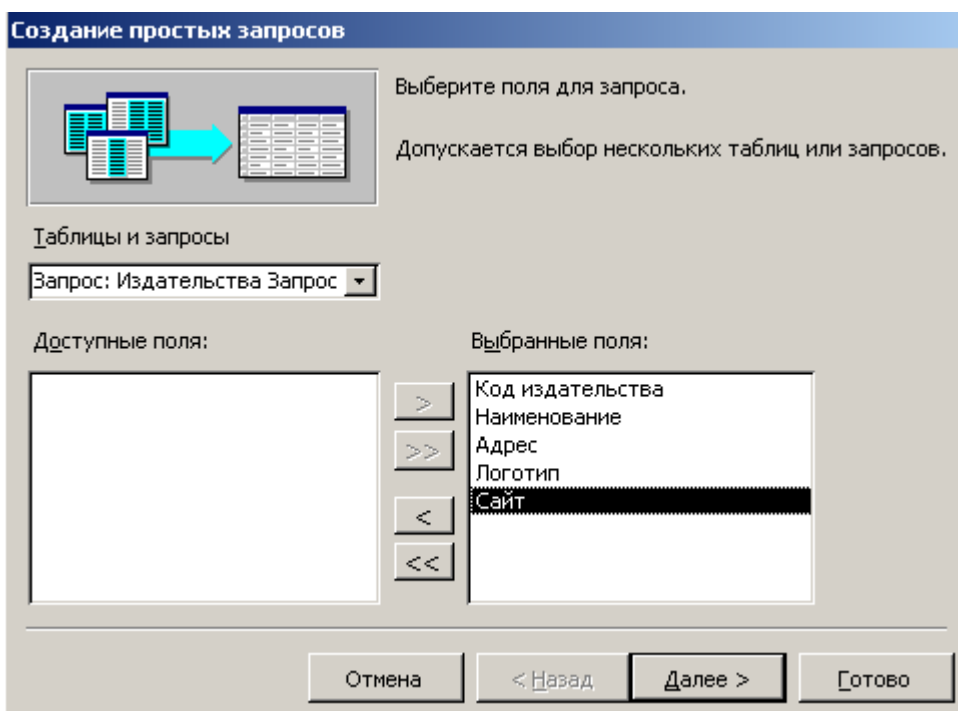
Для создания запросов с помощью какого-либо мастера необходимо в базовом окне Access выбрать вкладку **Запросы** и щелкнуть по инструменту **Создать** или выбрать меню **Вставка/Запрос**. После этого сформируется окно:



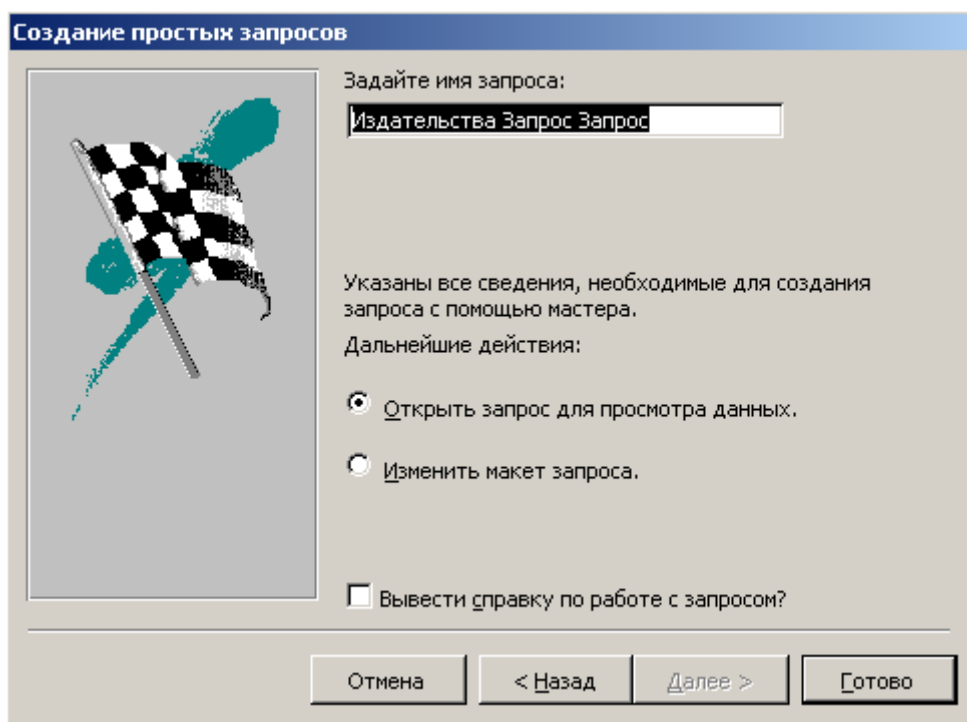
В форме **Новый запрос** нужно выбрать запрос, который предполагается строить и щелкнуть по кнопке **ОК**. При выборе позиции **Простой запрос** сформируется окно:



В поле **Таблицы и запросы** данного окна необходимо выбрать таблицу или запрос из предлагаемого списка, на базе которого предполагается строить запрос. После чего в левой области окна сформируется список полей выбранного объекта. Чтобы включить поля в проектируемый запрос используются кнопки  и . Кнопка  позволяет выбрать для создаваемого запроса предварительно выделенные поля, а кнопка  позволяет выбрать все поля. Кнопки  и  позволяют исключить поля из списка выбранных полей. В рассматриваемом случае выбраны все поля запроса **Издательства запрос**:



После нажатия кнопки **Далее** сформируется форма следующего вида:



Как видно из рисунка, построенному запросу по умолчанию присвоено имя. Оно формируется из имени объекта, на котором строится запрос и слова “запрос”. Это имя можно изменить или оставить предложенное. При необходимости его можно будет изменить позже в базовом окне Access.

Посредством радиокнопки **Изменить макет запроса** сформированный запрос можно отобразить в режиме конструктора с целью анализа и модификации. Посредством радиокнопки **Открыть запрос для просмотра данных** можно просмотреть результаты выполнения построенного запроса. После нажатия кнопки **Готово** результаты выполнения запроса отобразятся в следующем виде:

Издательства Запрос Запрос : запрос на выборку					
	Код издательст	Наименование	Адрес	Логотип	Сайт
▶	1	Мир	Бауманская ,2	verPoint.Slide.8	6666@aaa.ru
	2	Прометей	Подбельского,7	ft Visio Drawing	
	3	Эксмо	11-я парковая,		111@222
	4	Машиностроени	Королева,15		
	5	Энергия	Уральская,77		
*	(Счетчик)				

Как видно из рисунка, некоторые заголовки или значения полей не помещаются в отведенных областях. Перед сохранением запроса эти области можно расширить и при последующих просмотрах запроса результаты будут выводиться в отредактированном виде.

После закрытия окна с результатами запроса с помощью кнопки оконного меню необходимо подтверждение на сохранение запроса.

3. Формирование перекрестного запроса.

Перекрестный запрос позволяет вывести данные в таком виде, который предусматривает вывод заголовков столбцов, заголовков строк и значений в областях пересечения заголовков строк и столбцов. При проектировании перекрестного запроса следует знать, что в качестве заголовков столбцов можно задействовать значения одного поля, в качестве заголовков строк можно задействовать значения от одного до трех полей, в качестве значений можно задействовать только значения одного поля (как правило, это поле числового типа). В связи с этим следует строить перекрестный запрос на базе той таблицы или запроса, для которых перекрестный запрос имеет смысл. Например, для таблицы с зарплатами сотрудников или для таблицы с ценами товаров использование перекрестного запроса может оказаться полезным.

Для создания перекрестного запроса надо в форме **Новый запрос** выбрать позицию **Перекрестный запрос**. После этого сформируется следующее окно:

Создание перекрестных таблиц

Выберите поля, значения которых будут использованы в качестве заголовков строк.

Допускается выбор не более трех полей.

Выберите поля по порядку сортировки данных. Например, можно сначала выполнить сортировку значений по странам, а затем по городам.

Доступные поля:

- Код книги
- Название
- Цена
- Код издательства
- Дата издания
- Твердый переплет

Выбранные поля:

- Автор

Образец:

Автор	Заголовок1	Заголовок2	Заголовок3
Автор1	ИТОГИ		
Автор2			
Автор3			
Автор4			

Отмена < Назад Далее > Готово

В данном окне необходимо выбрать от одного до трех полей, которые в перекрестном запросе будут использоваться в качестве заголовков строк. Как правило, в качестве таких полей выбираются текстовые поля или поля с типом дата. В данном случае выбрано поле **Автор**.

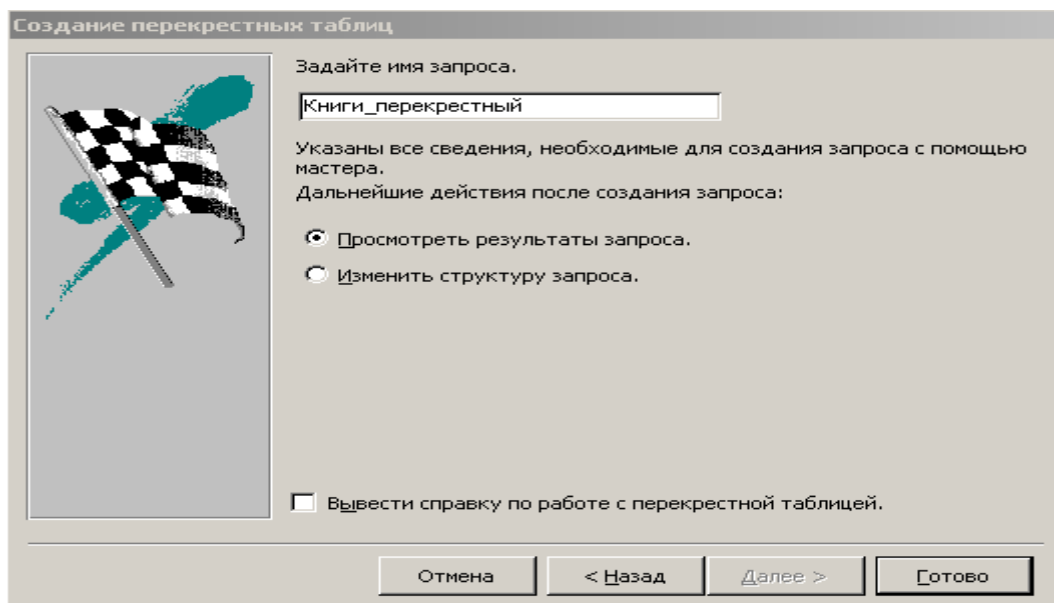
В нижней части окна представлен образец перекрестного запроса, который облегчает разработчику решение задачи построения запроса.

После нажатия кнопки **Далее** мастер перейдет к следующему шагу. На этом шаге отобразится окно, в котором по аналогии с предыдущим окном, необходимо указать поле, значения которого будут использоваться в качестве заголовков столбцов. В данном случае в качестве заголовков столбцов задействованы значения поля **Код книги**.

На следующем шаге работы мастера сформируется окно вида:

В данном окне нужно выбрать поле, по которому будут формироваться итоги в ячейках запроса. В данном случае имеет смысл в качестве поля для итогов выбрать поле **Цена**, а в качестве функции, которая будет выполняться над значениями данного поля – **Sum** (Сумма). Посредством флажка **Вычислить итоговые значения для каждой строки?** можно обеспечить формирование дополнительного поля, в котором для данного случая будут суммироваться стоимости книг каждого автора.

После выбора итогового поля и перехода к следующему шагу мастера сформируется окно:



Это последнее окно, которое отображает мастер запросов. Назначение этого окна аналогично назначению последнего окна мастера **Простого запроса**.

Результаты выполнения спроектированного запроса будут представлены в следующем виде:

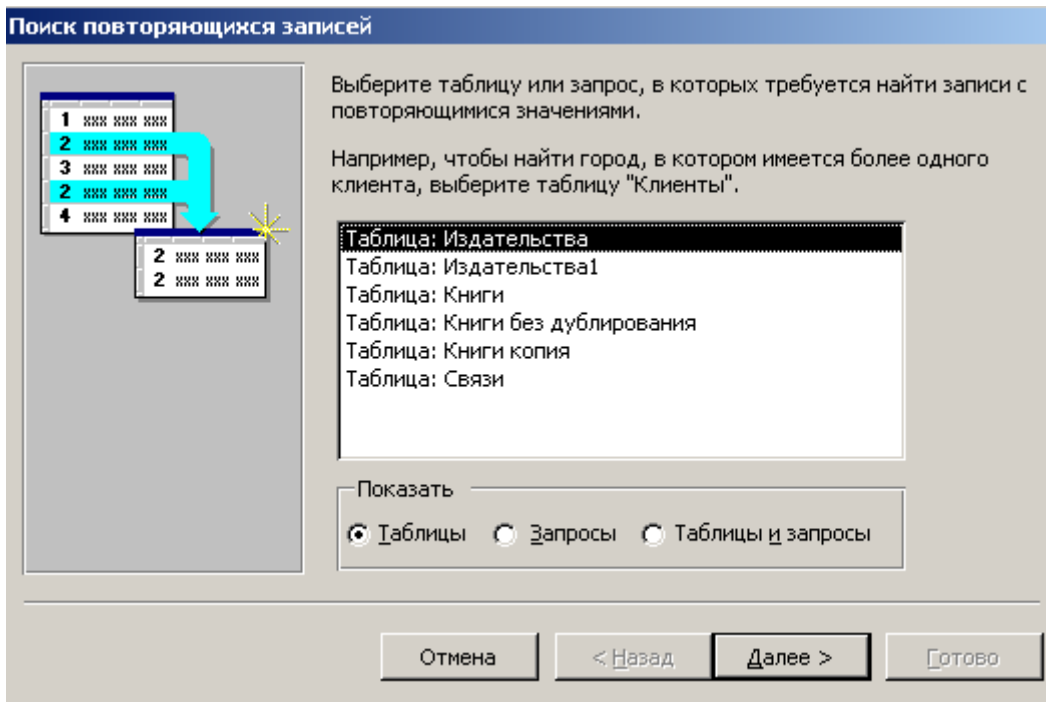
Книги_перекрестный : перекрестный запрос									
Автор	Итоговое значе	1	2	3	4	7	8	12	
Трин	170,00р.					82,00р.		88,00р	
Данте	71,00р.				71,00р.				
Жюль Верн	95,00р.						95,00р.		
Мильтон	82,00р.			82,00р.					
Шекспир	65,00р.		65,00р.						
Шиллер	125,00р.	125,00р.							

Как видно из рисунка, сформировано поле **Итоговое значение**, в котором расположены суммарные цены для каждой строки.

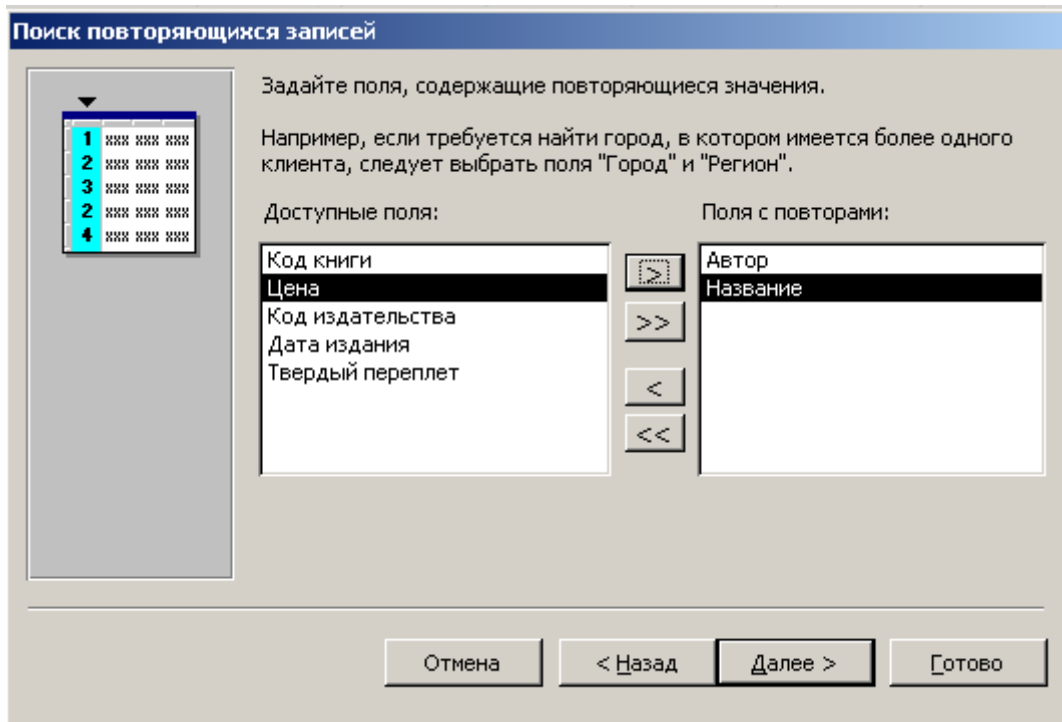
4. Выборка повторяющихся записей.

Нередко бывает полезно просмотреть поля или записи таблицы, которые встречаются неоднократно. В таблицах, в которых задействованы ключевые поля с уникальными значениями, повторяющихся записей в принципе быть не может. Поэтому и строить запрос на выборку повторяющихся записей для таких таблиц не имеет смысла. Для таблиц такого рода можно построить запрос на выборку повторяющихся значений отдельных полей. В некоторых таблицах могут быть не задействованы ключевые поля. Для таких таблиц может представлять интерес выборка повторяющихся записей.

Для создания запроса на выборку повторяющихся записей надо в форме **Новый запрос** выбрать позицию **Повторяющиеся записи**. После этого сформируется следующее окно:

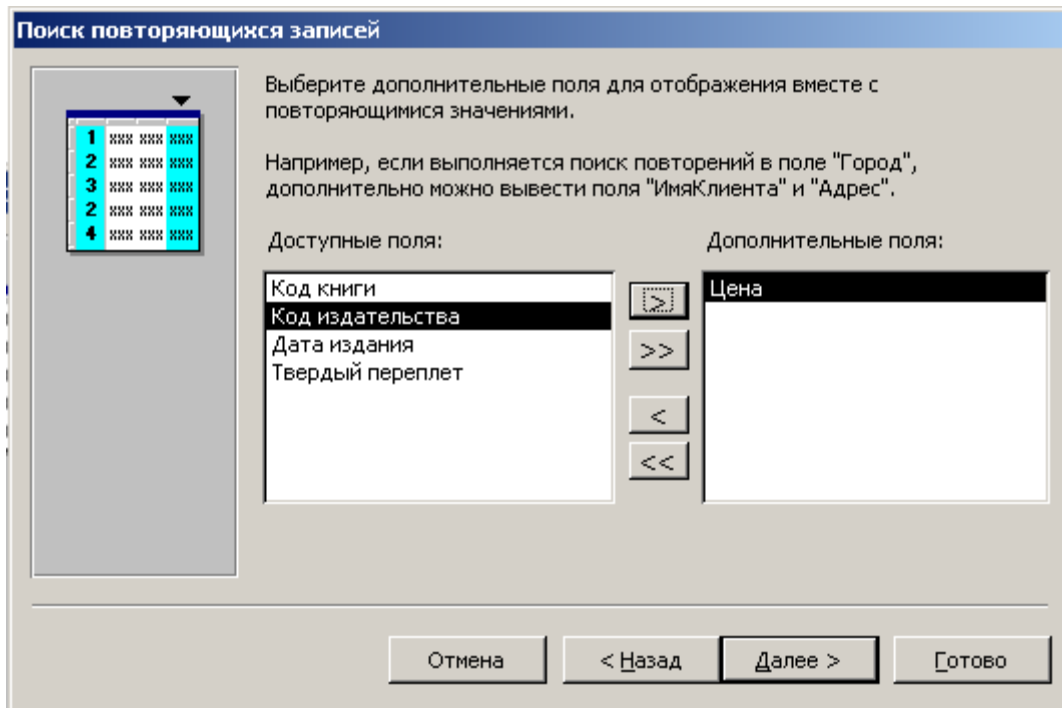


В данном окне необходимо выбрать таблицу или запрос, из которых требуется вывести повторяющиеся записи. После выбора объекта и перехода к следующему шагу мастера сформируется окно:



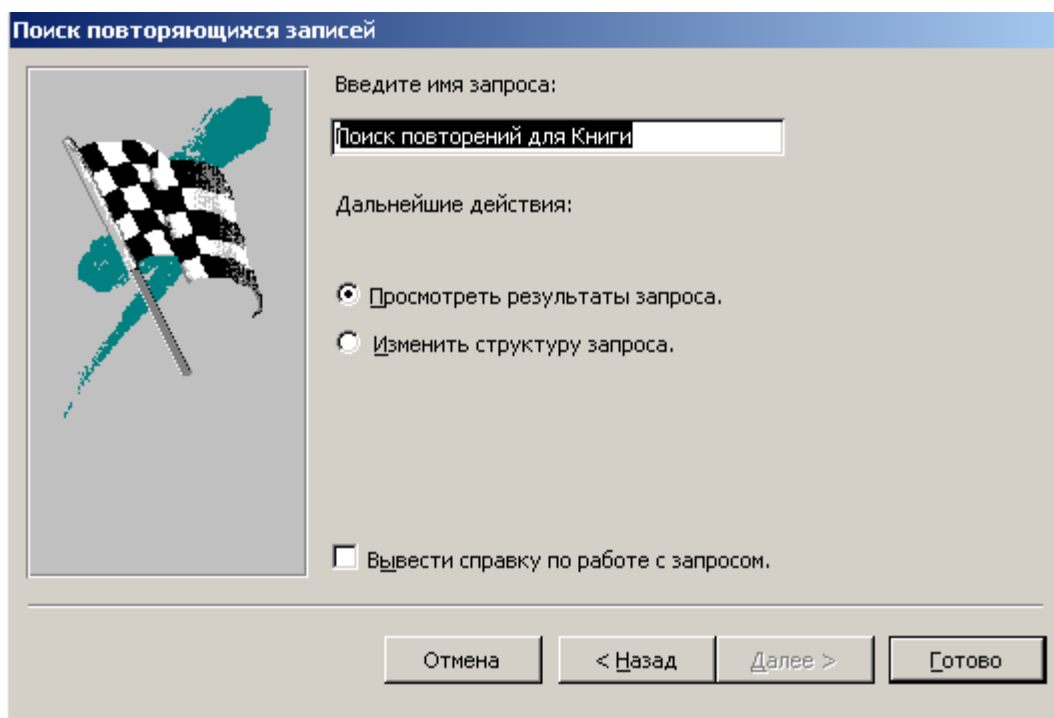
В этом окне необходимо выбрать поля, для которых требуются найти повторения. Если выбрать все поля, то повторяющихся записей не будет, так как среди этих полей есть поле **Код книги**, имеющее уникальные значения. В данном случае для анализа выбраны поля **Автор** и **Название**.

На следующем шаге мастера отобразится окно:



В данном окне необходимо выбрать поля, которые будут отображаться при выполнении запроса вместе с повторяющимися значениями. В рассматриваемом случае в качестве такого поля выбрано поле **Цена**.

На следующем шаге мастера отобразится его последнее, стандартное окно:



В этом окне можно изменить предлагаемое имя запроса. Запрос можно загрузить в режим конструктора или просмотреть результаты его выполнения. В режиме просмотра сформированный запрос выглядит следующим образом:

Поиск повторений для Книги : запрос на выборку			
	Автор	Название	Цена
▶	Грин	Алые паруса	88,00р.
	Грин	Алые паруса	82,00р.
*			0,00р.

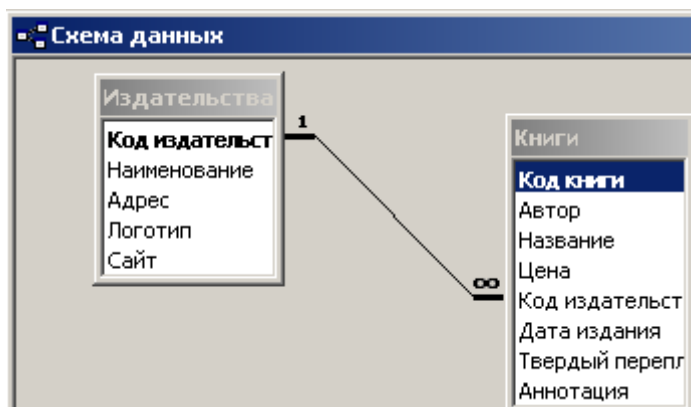
Как видно из рисунка, повторяются две пары полей **Автор-Название**. В качестве дополнительного поля отображено поле **Цена**, которое имеет разные значения.

5. Поиск записей без подчиненных.

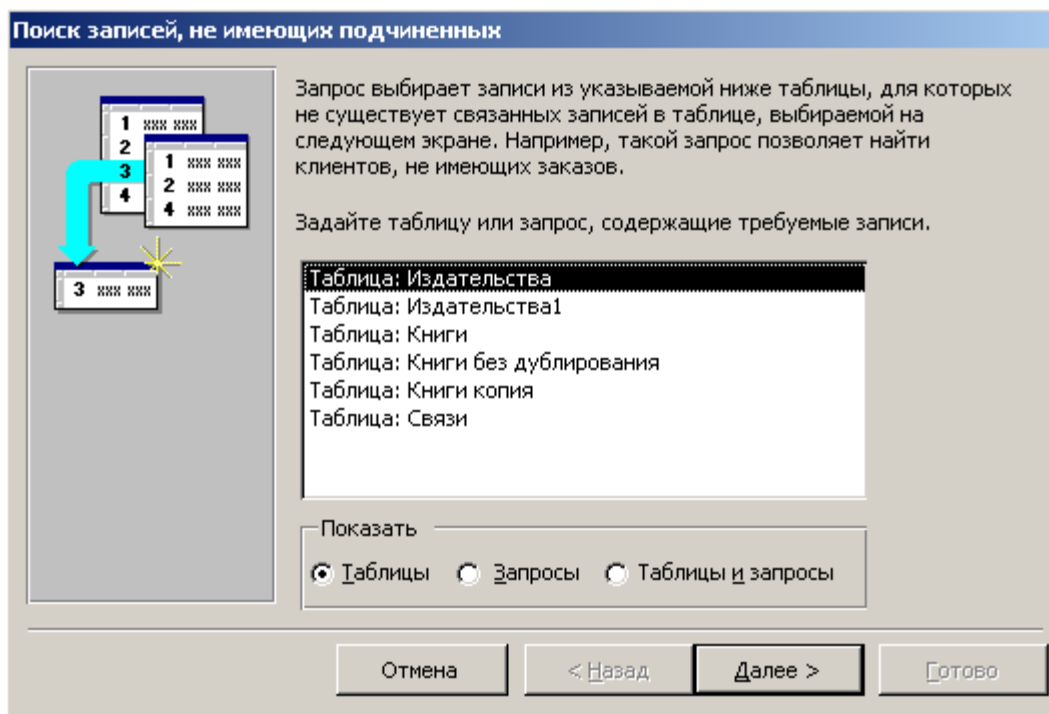
В схеме данных таблицы могут быть связаны между собой посредством ключевых полей. При этом одним из типов возможных связей является связь Один ко многим. Таким образом, назначается основная и подчиненная таблицы. Записи подчиненной таблицы посредством внешнего ключа могут ссылаться к записям основной таблицы. В рассматриваемом примере в качестве основной таблицы выступает таблица **Издательства**. В качестве подчиненной таблицы выступает таблица **Книги**. Они связаны между собой посредством ключевых полей, которые в обеих таблицах называются **Код издательства**.

Нередко пользователю БД интересно узнать, какие записи основной таблицы не имеют подчиненных записей в подчиненной таблице. Для рассматриваемого случая это означает вывод списка издательств, которые в настоящее время не издали ни одной книги.

Схема данных, для которой строится запрос рассматриваемого типа, представлена на рисунке:

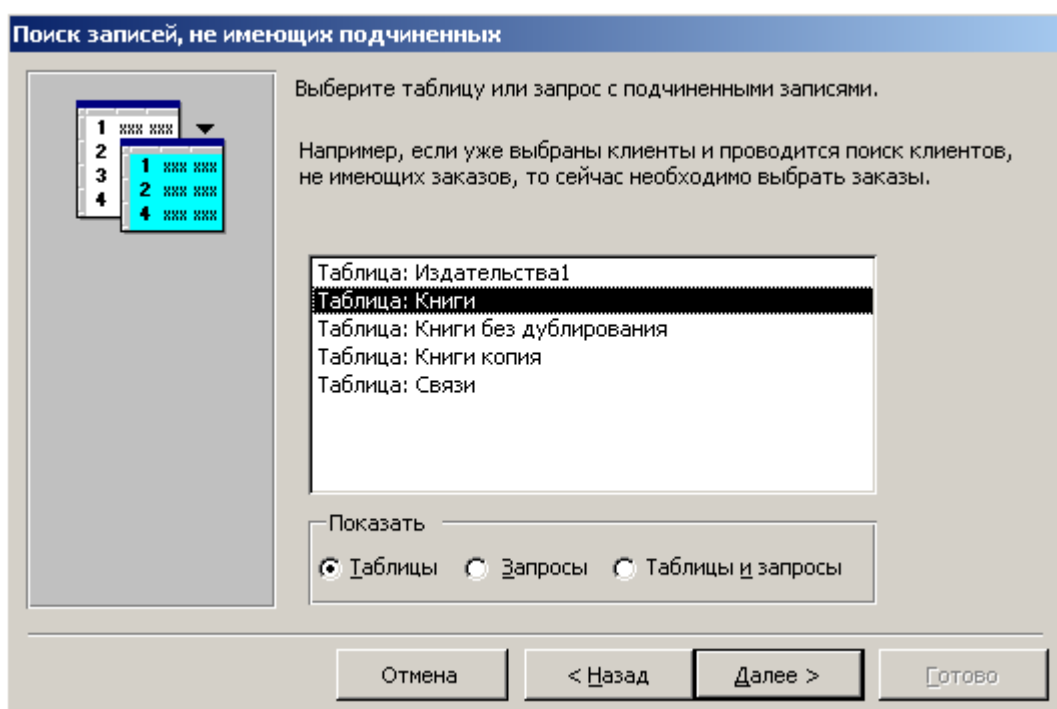


Для создания запроса на выборку **Записи без подчиненных** надо в форме **Новый запрос** выбрать позицию **Записи без подчиненных**. После этого сформируется следующее окно:

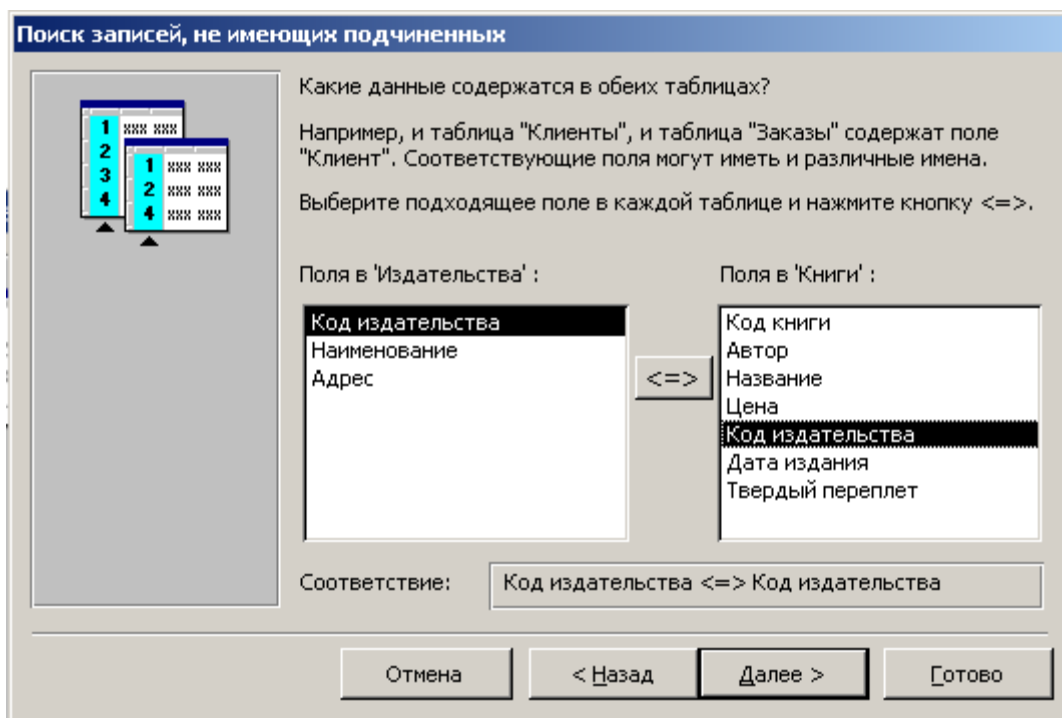



В этом окне необходимо выбрать таблицу или запрос, у которой есть подчиненная таблица или запрос. В качестве такой таблицы в рассматриваемом случае выбрана таблица **Издательства**.

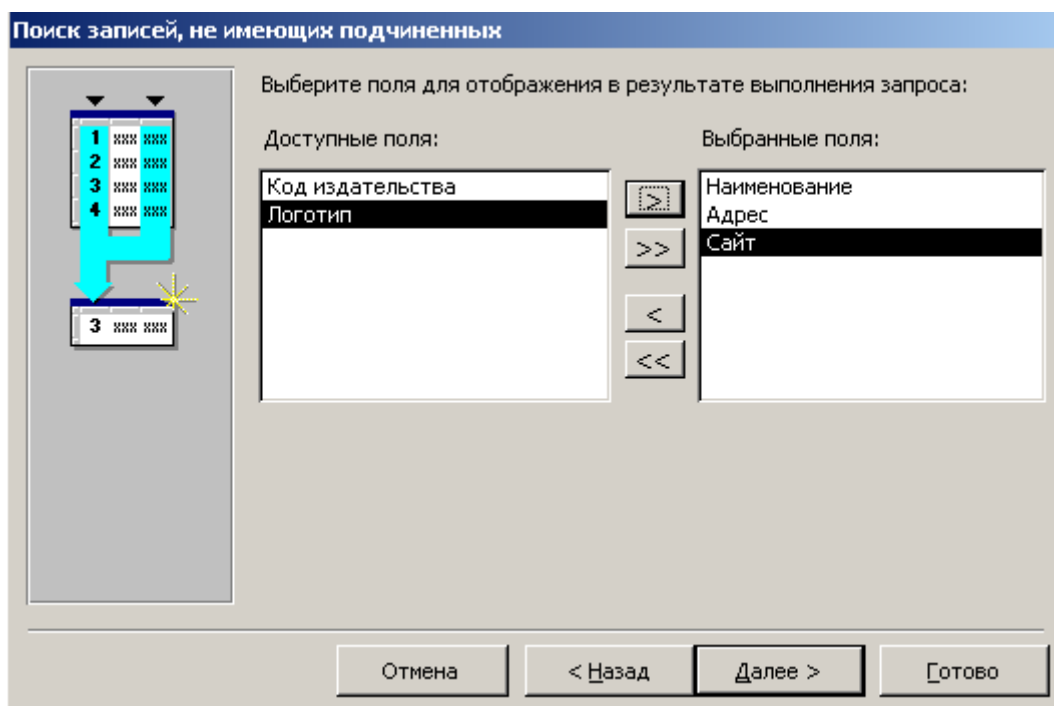
Выполнение следующего шага мастера приведет к загрузке окна:



В этом окне необходимо выбрать подчиненную таблицу. Этот выбор необходим, так как у каждой таблицы могут быть несколько подчиненных таблиц. В данном случае выбрана таблица **Книги**. На следующем шаге мастера сформируется окно:



В этом окне необходимо указать поля, по которым связаны главная и подчиненная таблицы. Для указания связи необходимо выделить поле в области полей таблицы **Издательства**, выделить соответствующее поле в области полей таблицы **Книги** и щелкнуть по кнопке . В рассматриваемом случае в качестве поля для связи в обеих таблицах выбрано поле **Код издательства**. На следующем шаге мастера сформируется окно:



В этом окне надо выбрать поля основной таблицы, которые будут отображены в создаваемом запросе. В рассматриваемом случае выбраны поля **Наименование**, **Адрес** и **Сайт**.

На следующем шаге мастера загрузится стандартное последнее окно мастера:

Поиск записей, не имеющих подчиненных

Задайте имя запроса:

Указаны все сведения, необходимые для создания запроса с помощью мастера.

Дальнейшие действия после создания запроса:

Просмотреть результаты запроса.

Изменить структуру запроса.

Вывести справку по работе с запросом.

Отмена < Назад Далее > Готово

В этом окне можно ввести имя запроса и открыть его в нужном режиме.

В режиме просмотра результатов запроса результат будет выглядеть следующим образом:

	Наименование	Адрес	Сайт
▶	Машиностроени	Королева,15	
	Энергия	Уральская,77	
*			

Как видно из результатов выполнения запроса, издательства **Машиностроение** и **Энергия** не издали ни одной книги.

Задание на лабораторную работу.

25. Сформировать простой запрос.
26. Выполнить запрос.
27. Просмотреть и проанализировать запрос в режиме конструктора.
28. Просмотреть и проанализировать запрос в формате SQL.
29. Сформировать перекрестный запрос.
30. Выполнить запрос.

31. Просмотреть и проанализировать запрос в режиме конструктора.
32. Просмотреть и проанализировать запрос в формате SQL.
33. Выполнить выборку повторяющихся записей.
34. Просмотреть и проанализировать запрос в режиме конструктора.
35. Просмотреть и проанализировать запрос в формате SQL.
36. Выполнить поиск записей без подчиненных.
37. Просмотреть и проанализировать запрос в режиме конструктора.
38. Просмотреть и проанализировать запрос в формате SQL.
39. Спроектировать перекрестный запрос. Выполнить этот запрос.
40. Просмотреть и проанализировать запрос в формате SQL.

Проектирование форм в Access 2002 с помощью конструктора.

2004

Введение

Средства проектирования форм с помощью **Конструктора** предоставляют разработчику интерфейса базы данных большие возможности. В форме можно разместить любое количество полей таблицы или запроса, расположить их произвольным образом, установить нужные размеры полей, отформатировать поля и области формы.

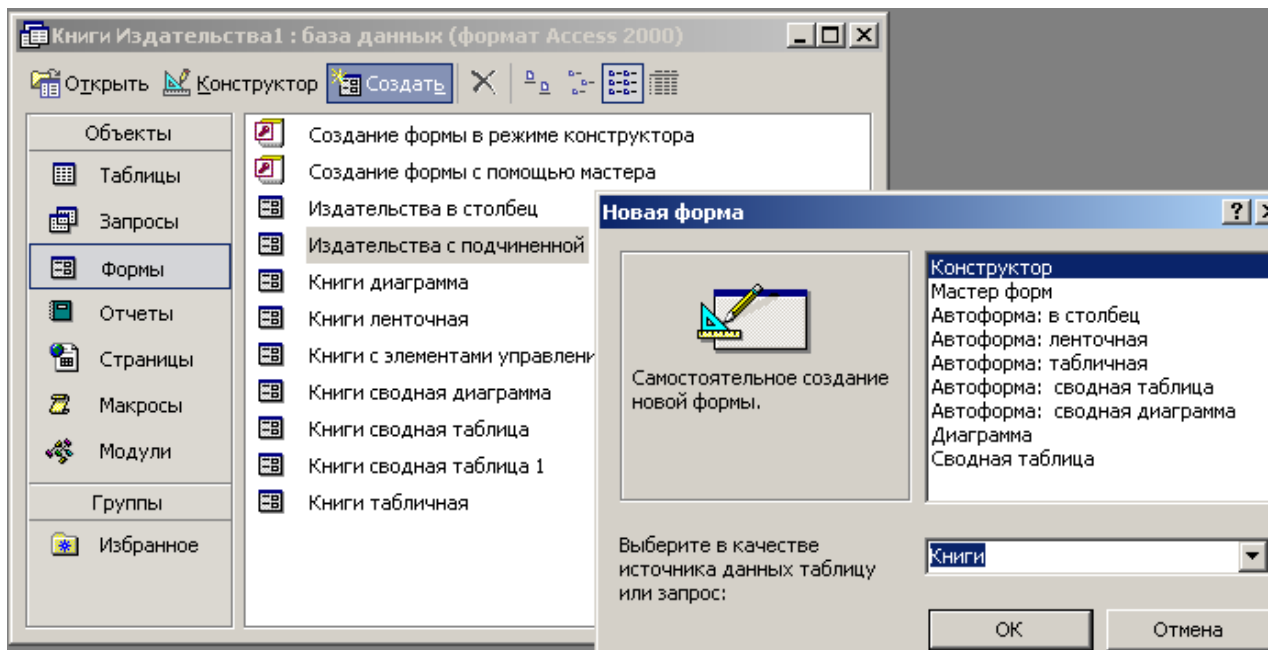
В режиме **Конструктора** в форме можно: сформировать надписи; спроектировать вычисляемые поля; сформировать группы элементов различного вида, для выбора альтернативы в процессе работы; сформировать поля со списком и списки; спроектировать кнопки и закрепить за ними какие-либо действия; внедрить рисунок в форму; внедрить объект OLE; внедрить подчиненную форму; расположить элементы формы на нескольких вкладках.

С помощью **Конструктора** форм можно отредактировать формы, построенные посредством **Мастеров**, и выполнить с ними все перечисленные выше действия.

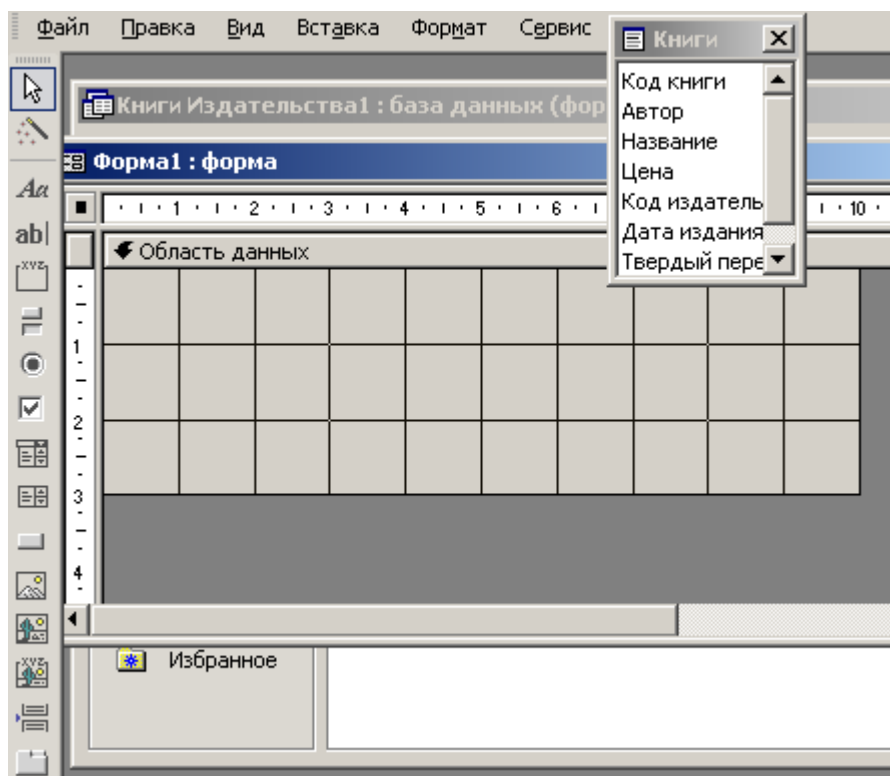
Посредством свойств формы и ее элементов, которые доступны в режиме **Конструктора**, можно: выполнить ряд действий по формированию и изменению макета формы или ее элементов; осуществить комплекс назначений свойств формы или ее элементов относительно данных; закрепить за любым событием, возникающим в ходе работы с формой или ее элементом, действие, которое обрабатывает это событие.

8. Создание формы посредством Конструктора

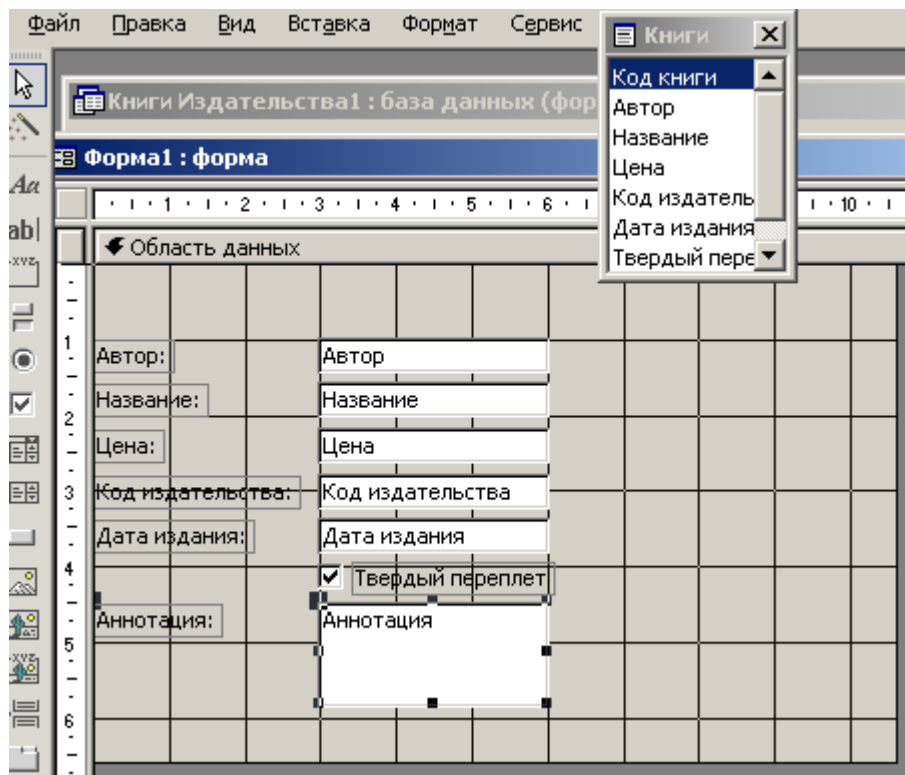
Для доступа к средствам проектирования формы в режиме **Конструктора** необходимо в базовом окне Access щелкнуть по инструменту **Создать**. В результате сформируется окно **Новая форма** со списком, в 1-й позиции которого расположен нужный режим:



Для инициации создания формы посредством **Конструктора** необходимо выделить соответствующую позицию в списке новых форм, затем в нижней части окна выбрать таблицу или запрос, на базе которых предполагается проектировать форму и щелкнуть по кнопке **ОК**. После этого сформируется окно:



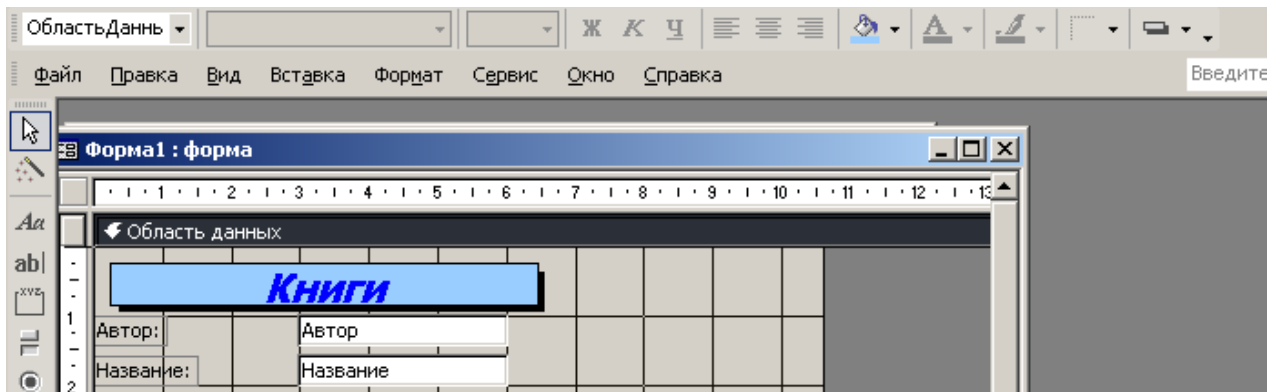
В этом окне приведен список полей таблицы **Книги**. Поля, которые необходимо отображать в форме, следует перетащить из этого списка в область данных. Так как поля не помещаются в области данных, то ее необходимо расширить посредством перетаскивания ее рамки. В результате этих действий форма в режиме **Конструктора** будет выглядеть следующим образом:



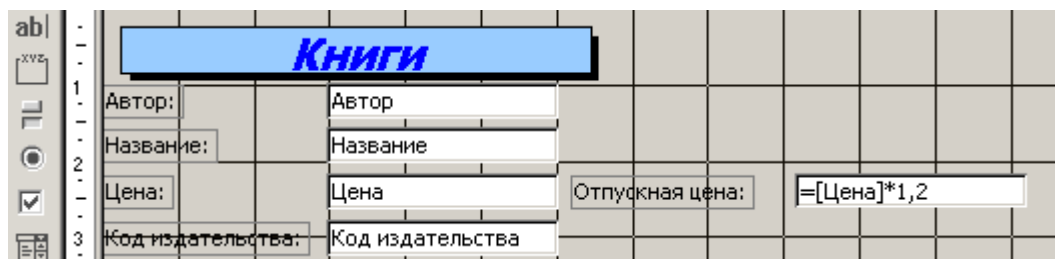
Размеры и положение объектов формы (полей, подписей) можно изменять. Для этого их надо выделить, щелкнув внутри объекта. На рисунке выделено поле **Аннотация**. Для перемещения поля необходимо взяться за левый верхний угол выделенного поля и перетащить в нужное место. Для изменения размеров поля служат квадратики, расположенные по периметру поля.

В левой части окна расположена панель элементов, которая может быть использована для оформления формы и изменения ее возможностей. Ниже рассматривается назначение этих элементов.

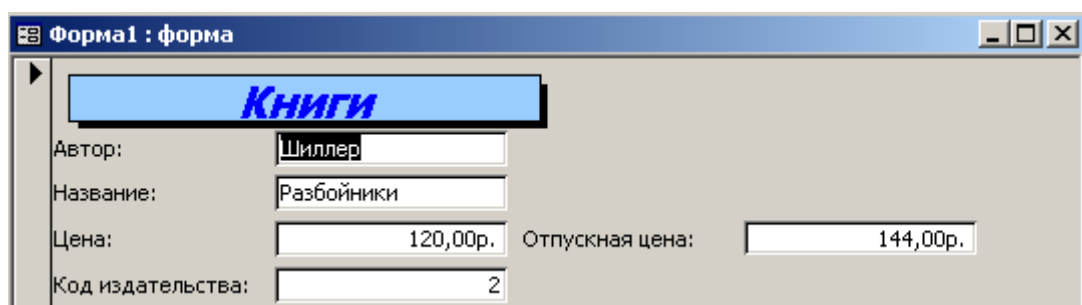
Элемент **Аннотация** (надпись) используется для ввода текстовой строки внутри формы. На рисунке представлен результат применения этого поля, а также панели инструментов **Форматирование**, которая выводится посредством команды меню **Вид/Панели инструментов/Формат**.








Элемент **ab** (поле) используется для формирования полей в области формы. В этих полях, в частности, можно ввести формулу. Для использования этого элемента необходимо щелкнуть по нему, нарисовать рамку внутри формы и ввести имя поля или формулу. На рисунке представлена формула для расчета отпускной цены.



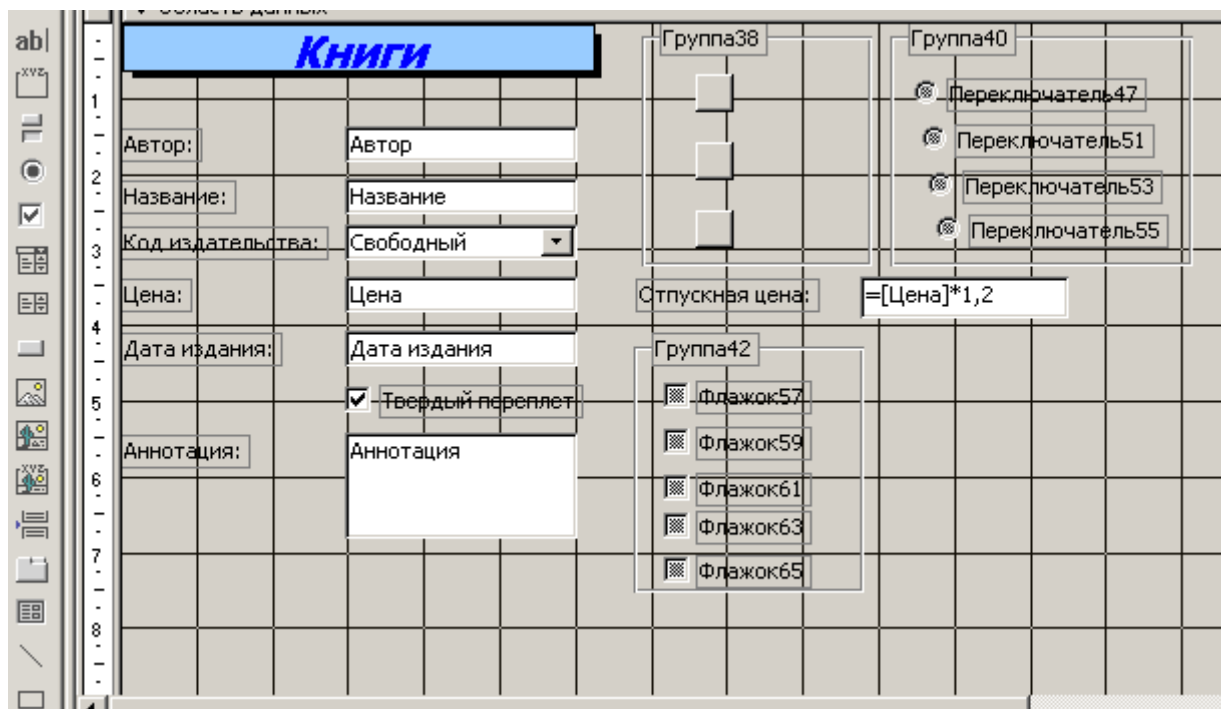
В режиме просмотра этот фрагмент выглядит следующим образом:



Элемент  (группа переключателей) используется для размещения в форме набора **выключателей** , **переключателей** , **флажков** , представляющих набор альтернативных (взаимоисключающих) значений.

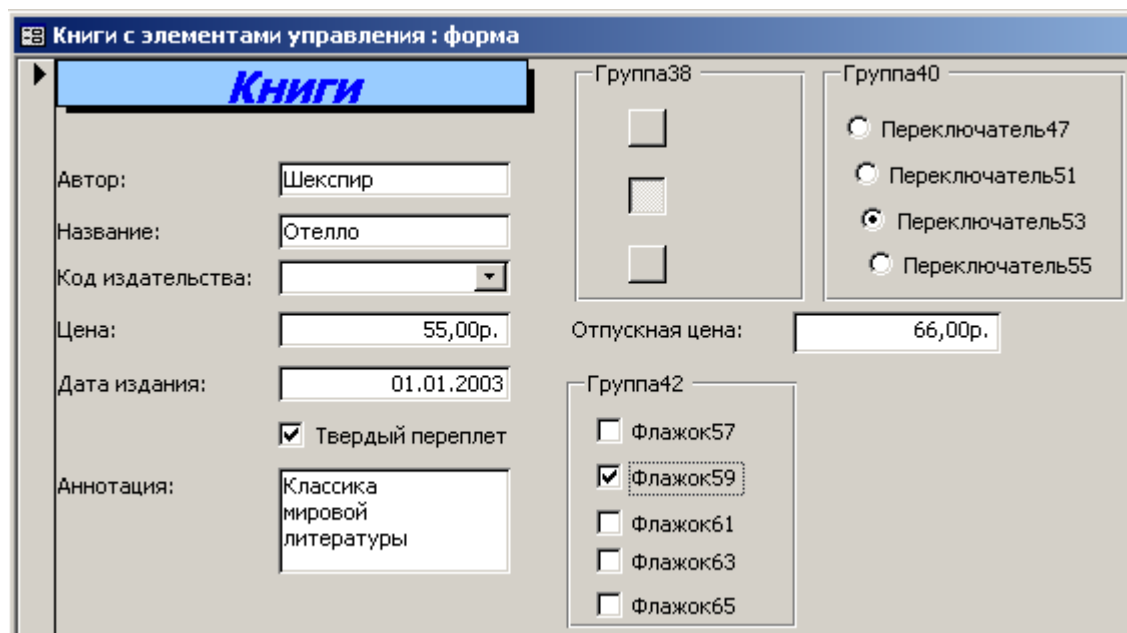
Для создания групп переключателей необходимо щелкнуть по элементу , а затем щелкнуть в нужной области формы, после чего появится рамка для группы переключателей. После этого в полученной рамке необходимо расположить нужное количество **выключателей, переключателей или флажков**.

На рисунке представлены примеры всех групп переключателей:








Подписи у переключателей сформировались автоматически. При желании их можно поменять на более информативные подписи.

На рисунке представлена форма с переключателями в режиме просмотра.

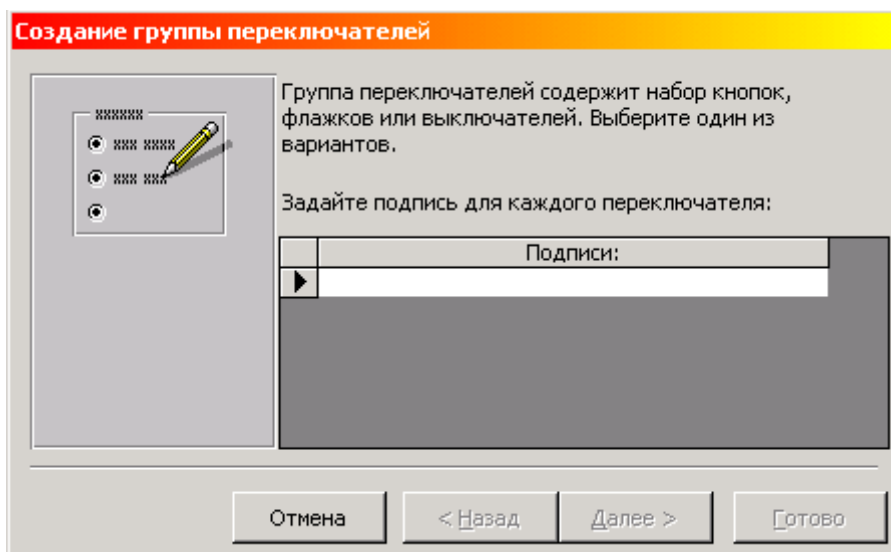


В группе переключателей одновременно может быть нажат только один переключатель. Это позволяет пользователю выбрать один из предлагаемых режимов. Разработчик формы должен с помощью макроса или процедуры реализовать анализ состояний переключателей и описать соответствующую реакцию на них. Например, 1-й выключатель открывает запрос на обновление таблицы, 2-й выключатель открывает запрос


на просмотр книг в твердом переплете, переключатель51 открывает форму Издательства, флажок61 открывает итоговый отчет.

Как правило, группу **выключателей** , **переключателей**  или **флажков**  удобнее проектировать с помощью **Мастера** . Для этого нужно щелкнуть по **Мастеру**, а затем щелкнуть по элементу  (группа переключателей). **Мастер** в режиме **Конструктора** может быть уже включен. Далее нужно следовать указаниям **Мастера**.

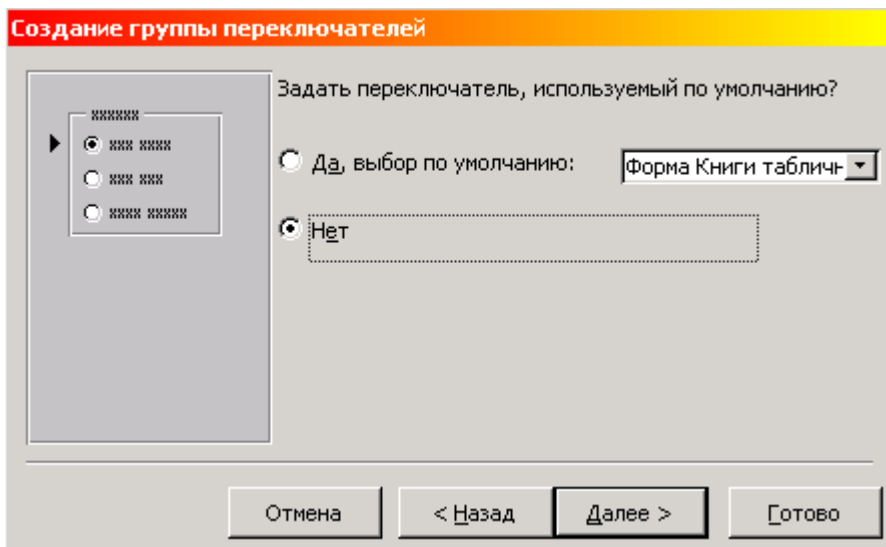
После выбора элемента **группа переключателей** при включенном **Мастере** и размещении его в области формы сформируется окно, в котором необходимо ввести подписи для каждого размещаемого переключателя:



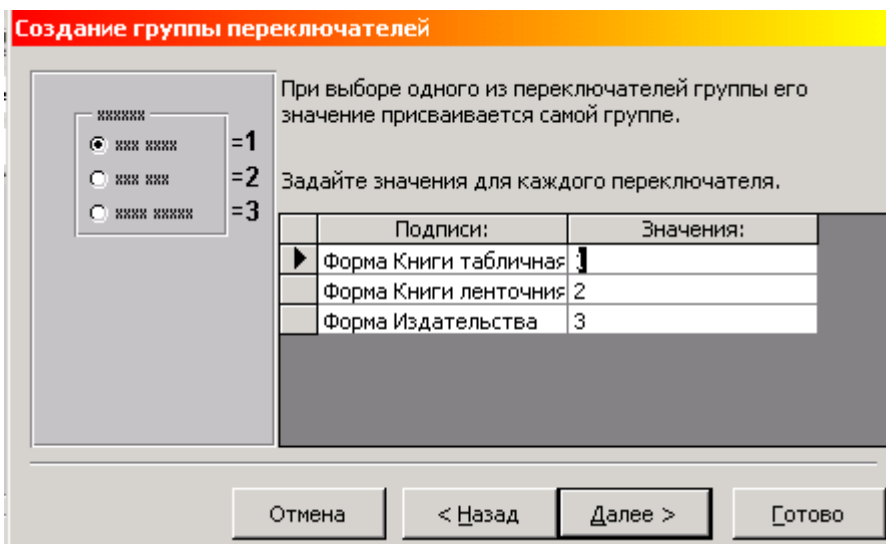
Например, можно ввести следующие подписи:

	Подписи:
	Форма Книги табличная
	Форма Книги ленточная
	Форма Издательства
*	

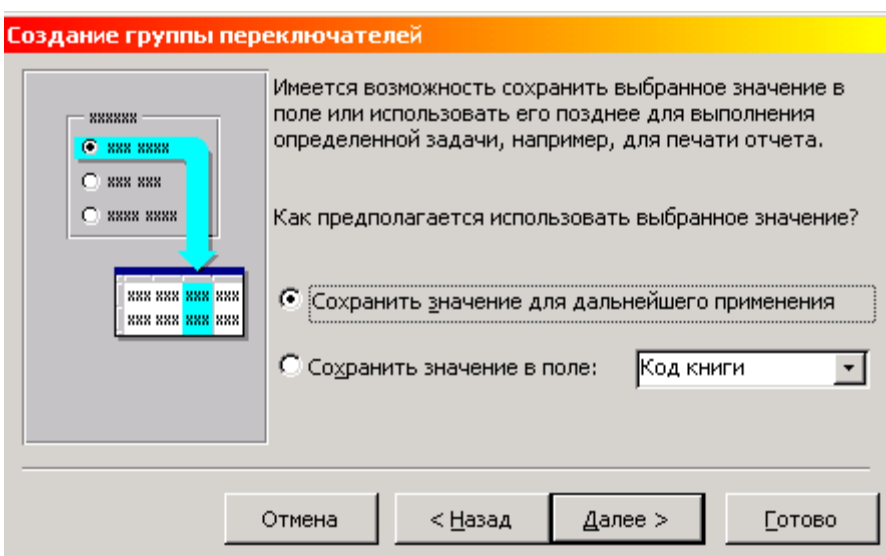
После ввода подписей и нажатия кнопки далее сформируется окно:



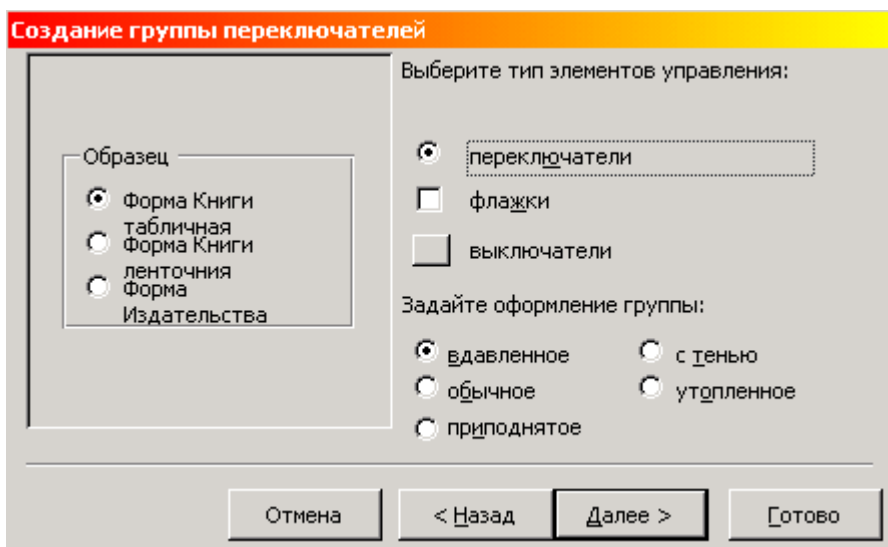
Здесь разработчик должен выбрать способ задания переключателей. После выбора способа и нажатия кнопки **Далее** формируется окно:



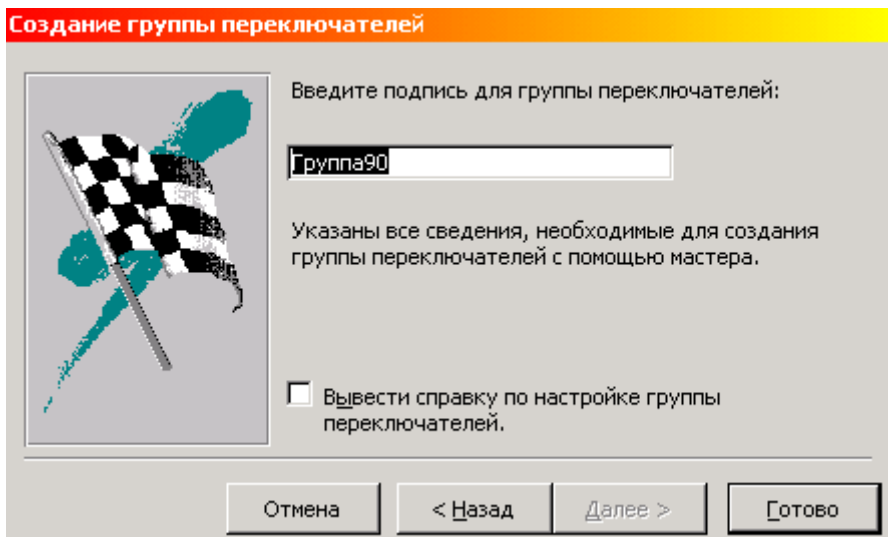
В этом окне предложенные значения переключателей можно заменить на другие (числовые) значения. Нажатие кнопки **Далее** инициирует открытие окна:



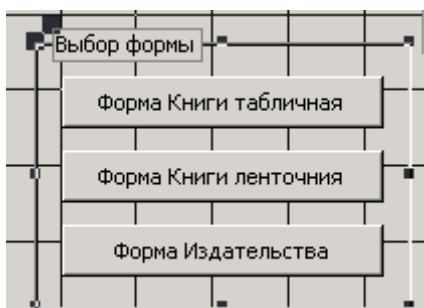
В окне необходимо указать, в каком поле формы будет сохраняться значение переключателя. Значение переключателя можно не сохранять в поле, а приписать ему впоследствии какие-либо действия. Нажатие кнопки **Далее** инициирует открытие окна:



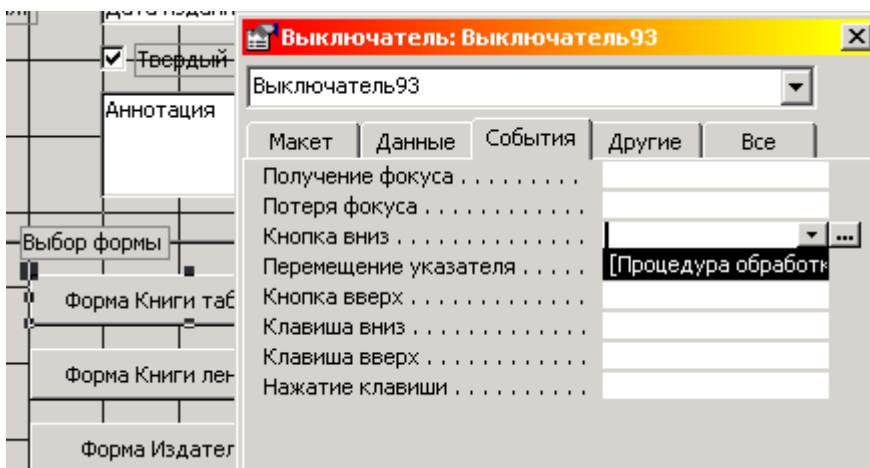
В этом окне можно выбрать тип элементов управления и их оформление. Нажатие кнопки **Далее** инициирует открытие окна:



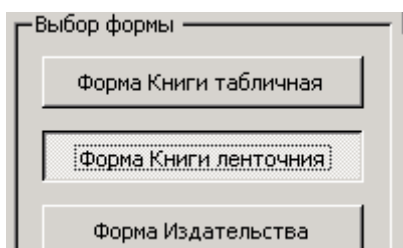
В данном окне можно ввести подпись для группы переключателей. После нажатия кнопки **Готово** сформируется окно:






Это внешний вид созданной группы переключателей в режиме **Конструктора**. Так как на предыдущих шагах не предусмотрено сохранение значения переключателя в каком-либо поле, то необходимо с каждым переключателем связать какое-либо действие. Для этого нужно дважды щелкнуть по переключателю и свойству **Кнопка вниз** вниз, поставить в соответствие процедуру или макрос.

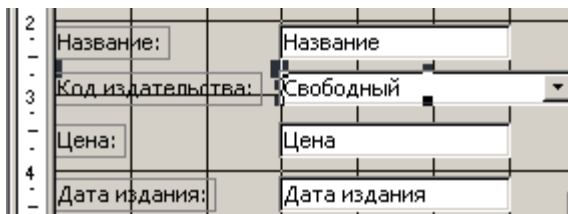


Созданная группа переключателей в режиме просмотра выглядит следующим образом:



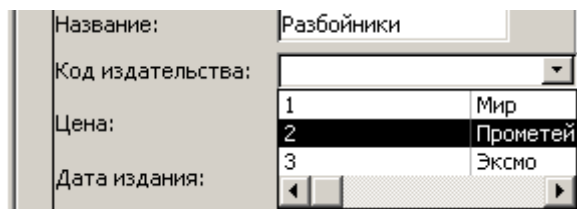
Элемент  (**поле со списком**) позволяет объединить поле и раскрывающийся список. Из этого списка пользователь может выбрать нужные значения. В рассматриваемом случае представляется рациональным поле **Код издательства** заменить на **поле со списком**, из которого данные будут вводиться в поле **Код издательства** рассматриваемой формы. Действительно, пользователю нелегко запомнить все издательства и их коды.

Для формирования такого поля необходимо включить мастер , если он не включен. Затем щелкнуть по элементу  и в области формы, где необходимо расположить **поле со списком**. После этого, следуя шагам мастера, выбрать в качестве источника данных поле **Код издательства** таблицы **Издательства** и поставить элементы списка в соответствии **Коду издательства** текущей формы. В режиме конструктора сформированное поле будет выглядеть следующим образом:




2	Название:	Название
3	Код издательства:	Свободный
4	Цена:	Цена
5	Дата издания:	Дата издания


Тот же фрагмент в режиме просмотра с раскрытым списком выглядит так:



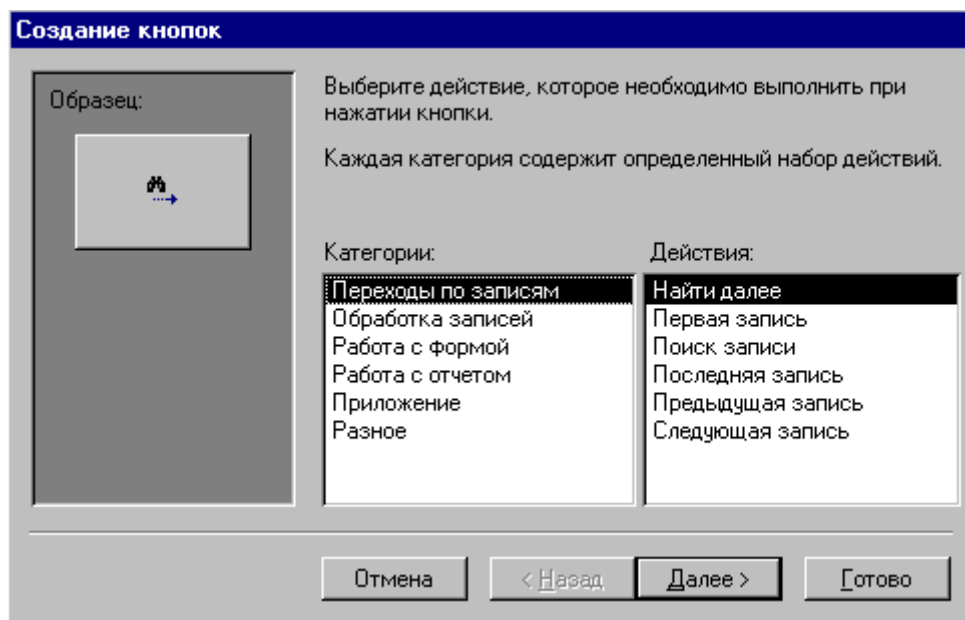
Название:	Разбойники
Код издательства:	Свободный
Цена:	1 Мир
	2 Прометей
	3 Эксмо
Дата издания:	

Элемент  (**список**) по сути представляет собой то же самое, что и предыдущее поле, но список развернут постоянно, а не разворачивается как в предыдущем случае.

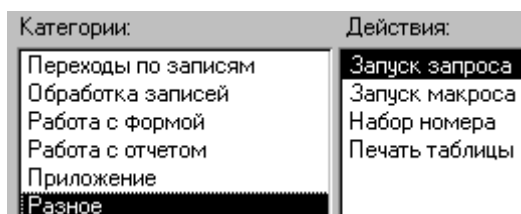
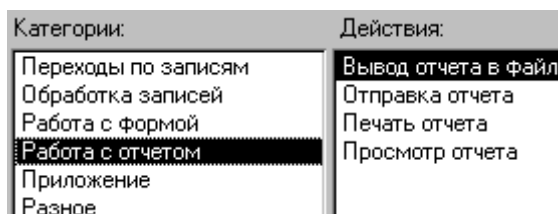
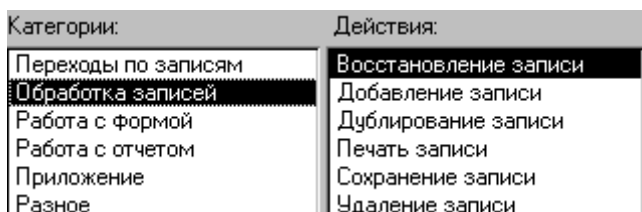
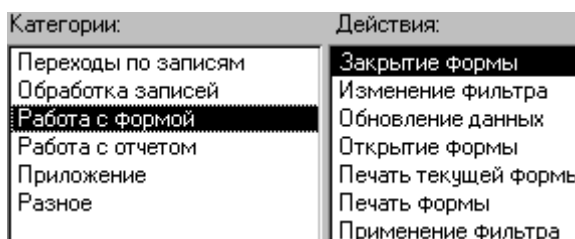
Формируется **Список** аналогично **Полю со списком**.

Элемент  (**кнопка**) используется для расположения кнопки в форме. Нажатие кнопки ставится в соответствие какое-либо действие – открытие формы, открытие отчета, запуск программы и др. Как правило, это действия стандартные. В связи с этим мастер предусматривает автоматизированное связывание с кнопкой ряда стандартных действий. Поэтому кнопка, как правило, формируется при включенном мастере.

Для формирования кнопки необходимо щелкнуть по соответствующему элементу и щелкнуть в области формы в том месте, где предполагается поместить кнопку. Если мастер включен, то сформируется следующее окно:

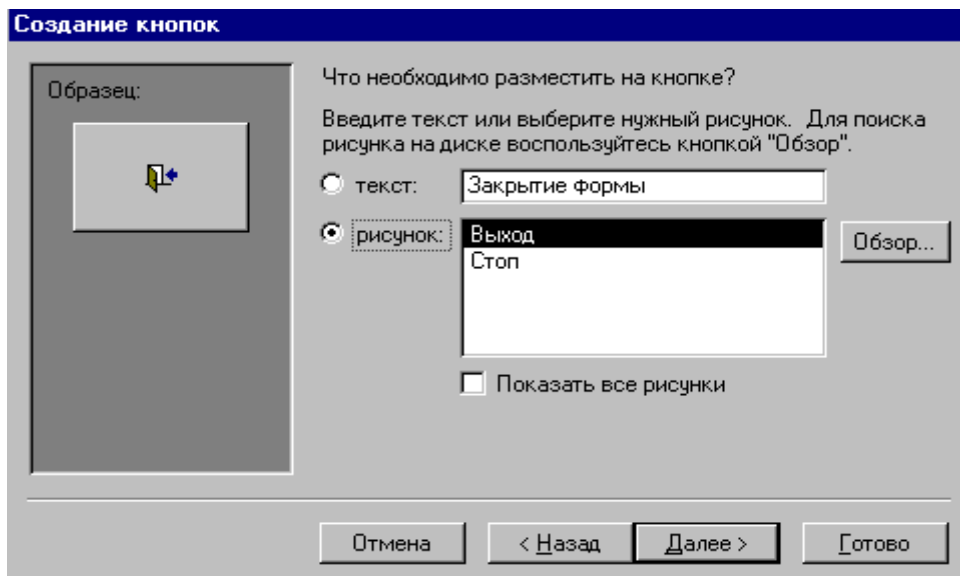


В списке **Категории** приведены категории средств, которые могут быть связаны с кнопкой. В списке **Действия** приведены все действия, которые можно связать с кнопкой в рамках выбранной категории. На рисунке представлены действия для категории **Переходы по записям**. На следующих рисунках приведены остальные категории и соответствующие им действия.



Приведенные категории и соответствующие им действия в большинстве случаев говорят сами за себя и поэтому не комментируются. Ниже, в качестве примера,

рассматривается действие **Закрытие формы** категории **Работа с формой**. После выбора этого действия и нажатия кнопки **Далее** сформируется окно:



В данном окне можно выбрать рисунок, который будет расположен на кнопке или ввести текст, который будет расположен на кнопке.


После нажатия кнопки **Далее** будет запрошено имя кнопки. Имя кнопки формируется мастером автоматически, например Кнопка1, но его можно изменить для того, чтобы оно было более информативно и более удобно для возможного использования в процедурах.


После нажатия кнопки **Далее** сформируется кнопка вида:

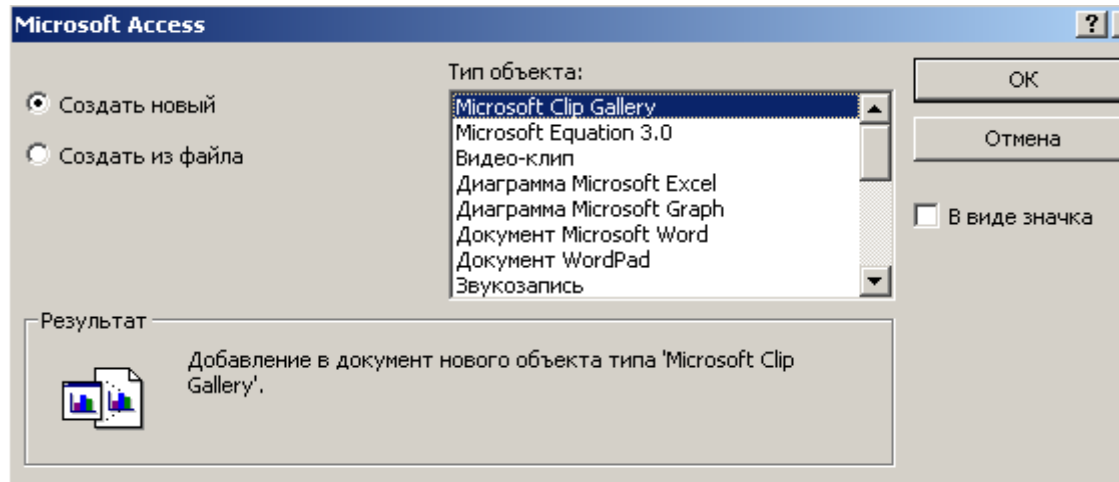


В режиме просмотра формы, при нажатии на данную кнопку осуществляется закрытие формы.

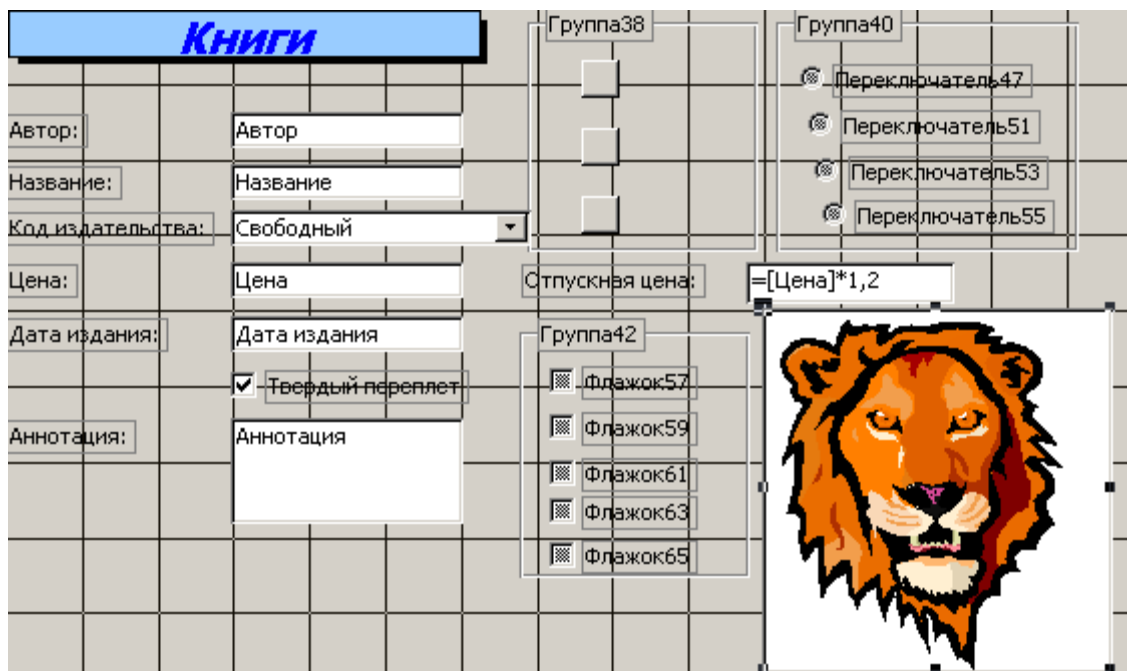
Посредством данного элемента можно сформировать набор кнопок в форме, за которым будет закреплен набор средств пользователя. Более того, можно создать форму, в которой будут размещены только кнопки. Кнопкам можно поставить в соответствие различные действия, в том числе и открытие других кнопочных форм, и таким образом разработать удобный интерфейс пользователя, который будет ничем не хуже традиционного меню.


Элемент  (**рисунок**) позволяет внедрить в форму рисунок. После щелчка по этому элементу и щелчка в области формы сформируется стандартное окно для навигации по файловой системе Windows. Посредством этого окна можно найти и вставить нужный рисунок.


Элемент  (**свободная рамка объекта**) позволяет вставить какой-либо объект в форму. Использование этого элементов приводит к формированию окна, в котором можно выбрать нужный объект:




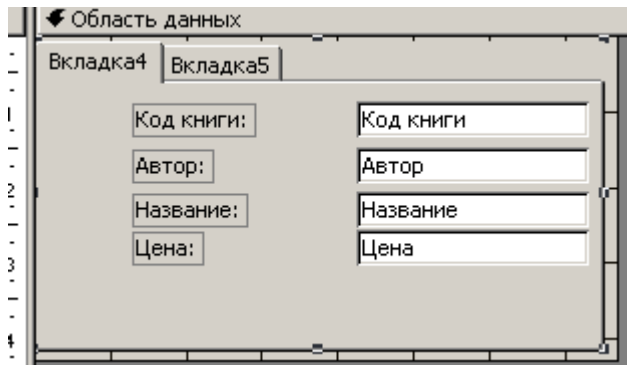
Например, использование объекта Microsoft Clip Gallery позволяет внедрить рисунок:



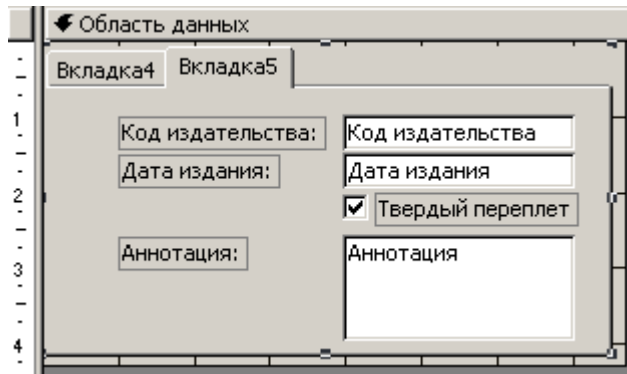
Элемент  (**присоединенная рамка объекта**) позволяет связать какой-либо объект с полем формы. Тип поля должен быть **OLE**.

Элемент  (**разрыв страницы**) позволяет указать начало нового экрана в форме. В режиме просмотра формы переход с экрана на экран осуществляется посредством клавиш **PageUp** и **PageDown**. Разрыв страницы имеет смысл использовать в том случае, если в форме очень много полей и они не помещаются на экране.

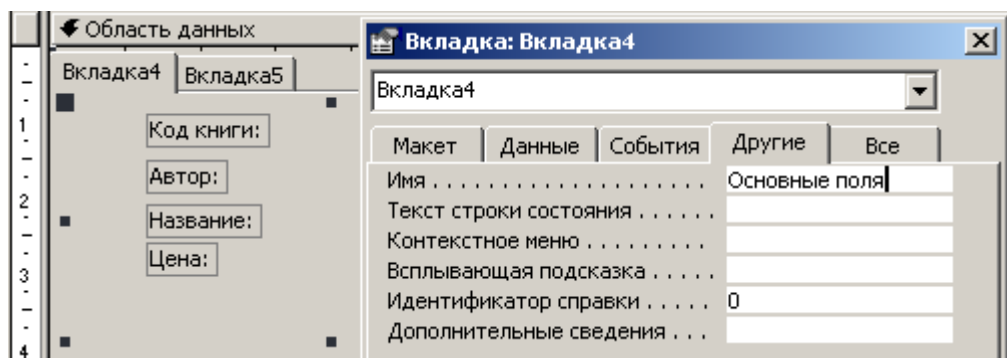
Элемент  (**вкладка**) позволяет создать форму с несколькими вкладками. Для использования вкладок надо создать пустую форму с помощью режима конструктора, затем расположить вкладки в форме. После этого из списка полей таблицы, на основе которой создается форма, на каждую из вкладок перетаскиваются поля. Например, на первой вкладке можно расположить следующие поля:



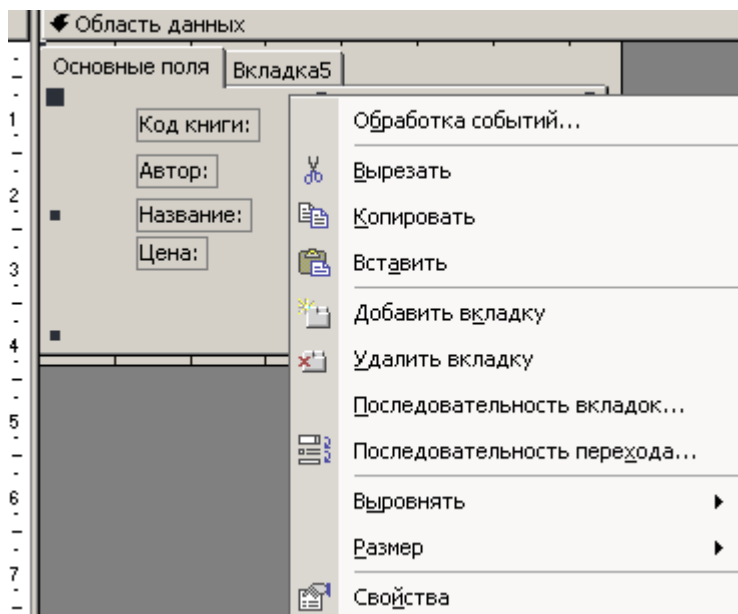
На второй вкладке расположены остальные поля таблицы Книги:






Имя вкладки можно поменять и сделать более информативным для этого достаточно дважды щелкнуть по ярлычку вкладки и в окне свойств в поле **Имя** набрать новое имя:



Число вкладок можно увеличить. Для этого надо щелкнуть правой клавишей мыши внутри вкладки. Появится контекстное меню, которое позволяет добавить вкладку и выполнить другие действия:

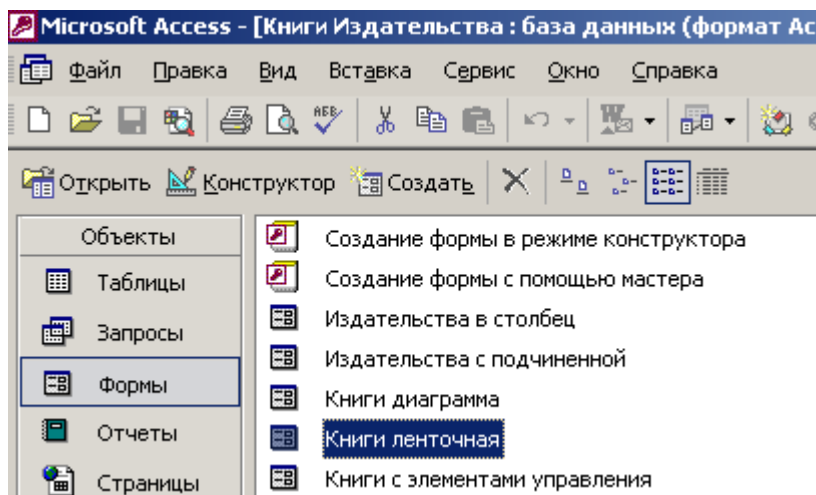


 (**подчиненная форма**) позволяет внедрить в форму подчиненную форму. В рассматриваемом случае имеет смысл внедрить подчиненную форму для формы **Издательства**. При этом в качестве подчиненной можно использовать форму **Книги**. При включенном мастере это нетрудно сделать. Однако создание подчиненной формы реализуется значительно проще способом, который рассмотрен в пункте 9.

Элемент  (**линия**) позволяет нарисовать в форме линию. Элемент  (**прямоугольник**) позволяет нарисовать в форме прямоугольник.

9. Редактирование формы посредством Конструктора

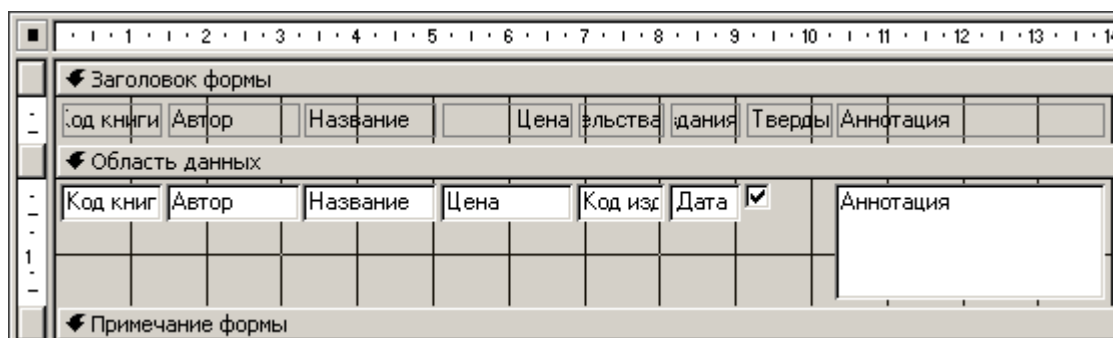
Форму, спроектированную одним из перечисленных способов, можно редактировать в режиме конструктора. Для этого в базовом окне Access необходимо выделить редактируемую форму, а затем щелкнуть по инструменту **Конструктор**.



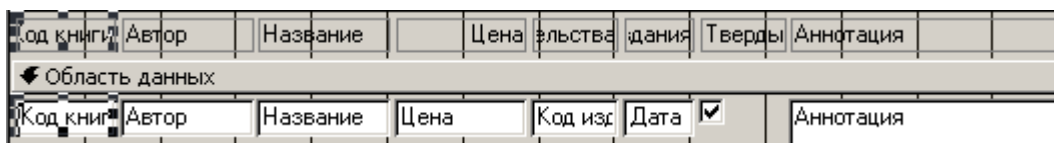
Форма **Книги ленточная** в режиме просмотра выглядит следующим образом:

Код книги	Автор	Название	Цена	Код издания	Дата издания	Тверды	Аннотация
1	Шиллер	Разбойники	120,00р.	2	1.2002	<input checked="" type="checkbox"/>	Классика мировой литературы
2	Шекспир	Отелло	55,00р.	2	1.2003	<input checked="" type="checkbox"/>	Классика мировой литературы
3	Мильтон	Потерянный	77,00р.	3	2.2002	<input checked="" type="checkbox"/>	
4	Данте	Божественн	66,00р.	3	2.2002	<input type="checkbox"/>	
5	Гомер	Эллиада	55,00р.	3	2.2002	<input type="checkbox"/>	

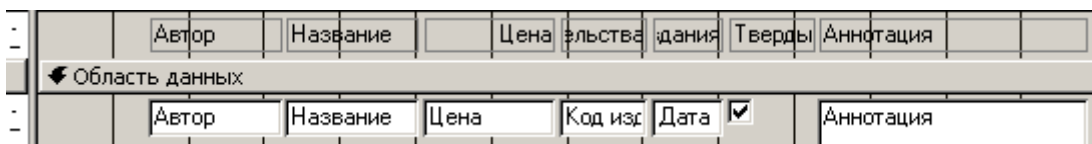
Как видно из рисунка, данная форма нуждается в редактировании. В частности: **Код книги** не является информативным и его можно удалить (в соответствующей таблице он все равно будет формироваться автоматически); заголовки некоторых полей видны не полностью; содержимое поля **Дата издания** не помещается в отведенной области; высота поля **Аннотация** не позволяет отобразить много записей в окне и при этом есть место в окне справа. Далее рассматривается, как можно выполнить устранение этих и других недостатков. В режиме конструктора форма выглядит следующим образом:



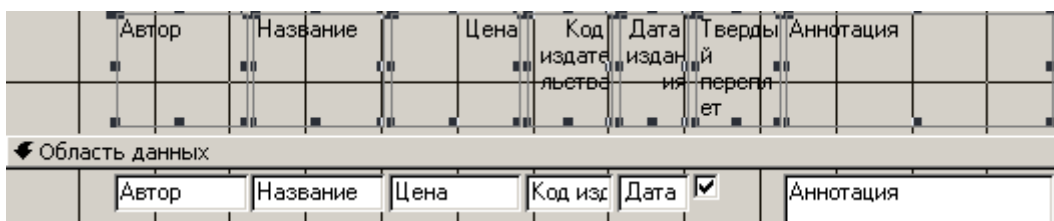
Для удаления поля **Код книги** необходимо выделить это поле и его заголовок и нажать на клавиатуре клавишу **Delete**. Для того чтобы выделить в форме несколько объектов необходимо при выделении удерживать клавишу **Shift**. Результат выделения:



Результат удаления:



Для изменения размера объектов, в частности заголовков полей, их необходимо выделить, а затем потянуть за один из маркеров выделения и изменить размер по вертикали и горизонтали. В данном случае имеет смысл увеличить размеры всех заголовков по вертикали. Для этого необходимо выделить все заголовки и потянуть за нижний маркер любого заголовка. Для выделения смежных объектов можно нажать левую клавишу мыши вне объекта и пробуксировать по выделяемым объектам. Результат выделения группы заголовков и расширения их по вертикали:



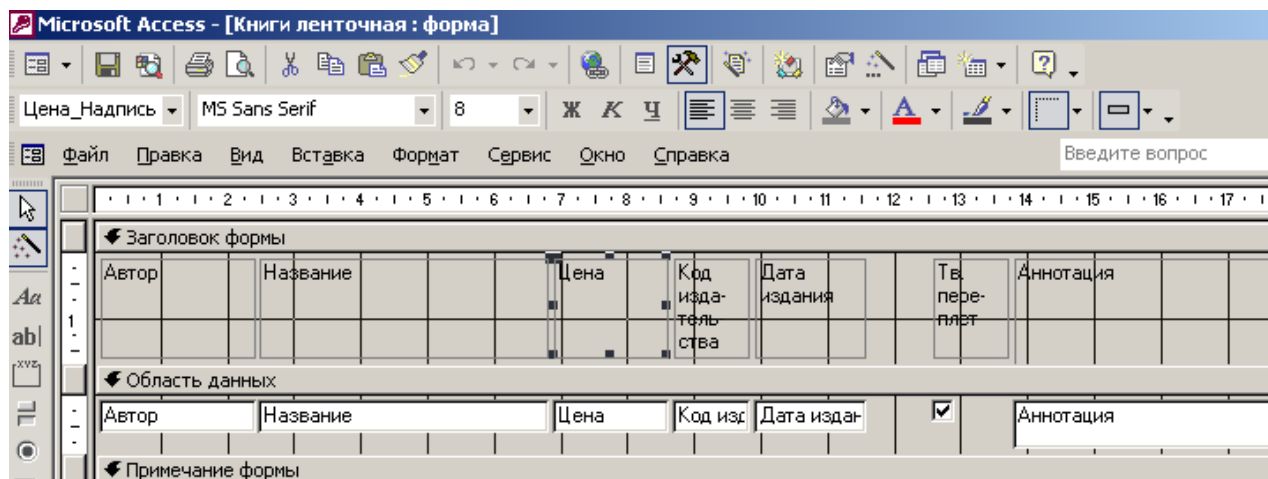
Дальнейшие манипуляции с формой могут привести к следующему виду:



В режиме просмотра форма выглядит теперь следующим образом:

Автор	Название	Цена	Код изда- тель- ства	Дата издания	Тв. пере- плет	Аннотация
Шиллер	Разбойники	120,00р.	2	01.01.2002	<input checked="" type="checkbox"/>	Классика мировой литературы
Шекспир	Отелло	55,00р.	2	01.01.2003	<input checked="" type="checkbox"/>	Классика мировой литературы
Мильтон	Потерянный рай	77,00р.	3	02.02.2002	<input checked="" type="checkbox"/>	
Данте	Божественная комедия	66,00р.	3	02.02.2002	<input type="checkbox"/>	
Гомер	Эллиада	55,00р.	3	02.02.2002	<input type="checkbox"/>	
Гомер	Одиссея	55,00р.	3	02.02.2003	<input type="checkbox"/>	
Грин	Алые паруса	77,00р.	1	01.01.2004	<input checked="" type="checkbox"/>	
Жюль Верн	Машина времени	90,00р.	1	01.01.2004	<input checked="" type="checkbox"/>	

Как видно из рисунка, форма приняла более наглядный вид. Однако для наглядности формы исчерпаны не все ресурсы. В частности, для оформления объектов формы можно использовать панель форматирования, которая имеет стандартный набор инструментов.

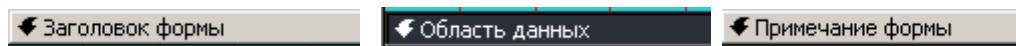


Результат использования панели форматирования форма в режиме просмотра может принять следующий вид:

Автор	Название	Цена	Код изда- тель- ства	Дата издания	Тв. пере- плет	Аннотация
Шиллер	Разбойники	120,00р.	2	01.01.2002	<input checked="" type="checkbox"/>	Классика мировой литературы
Шекспир	Отелло	55,00р.	2	01.01.2003	<input checked="" type="checkbox"/>	Классика мировой литературы

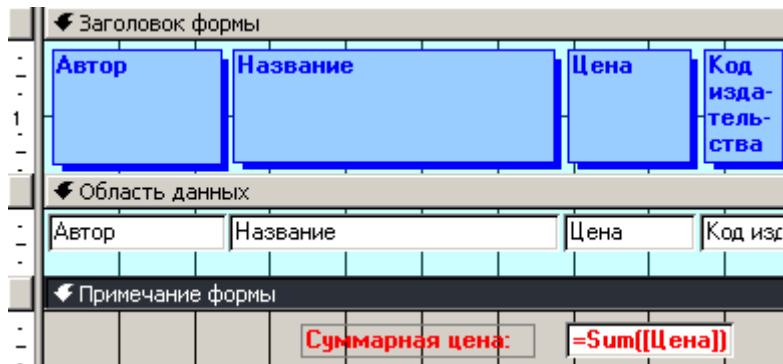
Для форматирования любого объекта формы необходимо выделить формируемый объект, а затем щелкнуть по нужному инструменту на панели инструментов Форматирование. Это касается и области **Заголовка формы** и **Области данных**. Для

выделения области необходимо щелкнуть по заголовку области. Ниже приведен вид 3-х заголовков областей:

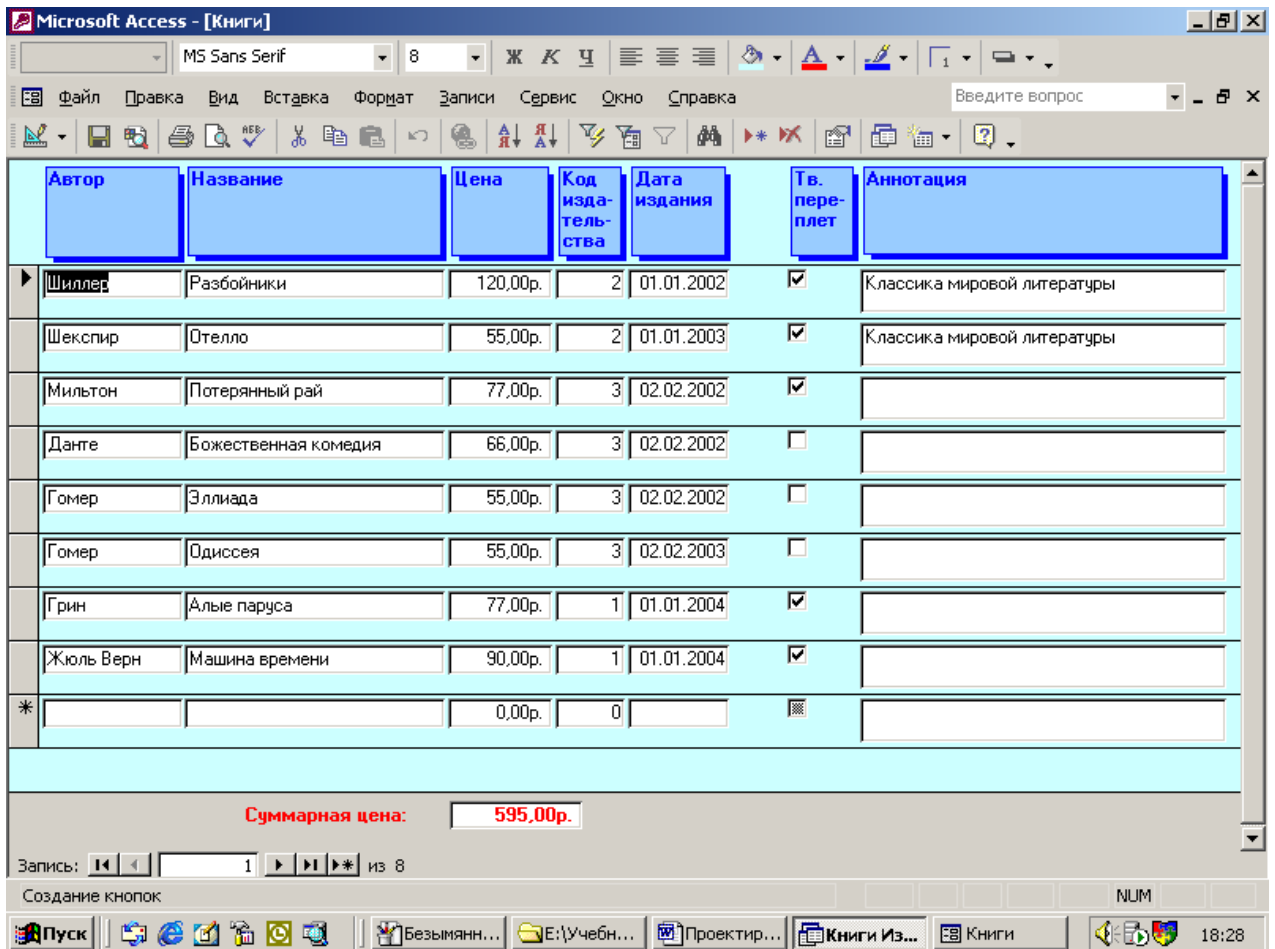


В приведенном примере выделена **Область данных**.

В области **Заголовок формы**, как правило, располагаются заголовки полей. В области **Область данных** располагаются поля с данными, вычисляемые поля. В области **Примечание формы** удобно располагать итоговые поля. Во всех областях можно расположить и другие элементы, которые формируются посредством панели элементов, рассмотренной выше. В качестве примера, в области **Примечание** сформировано итоговое поле, в котором подсчитывается суммарная стоимость книг:

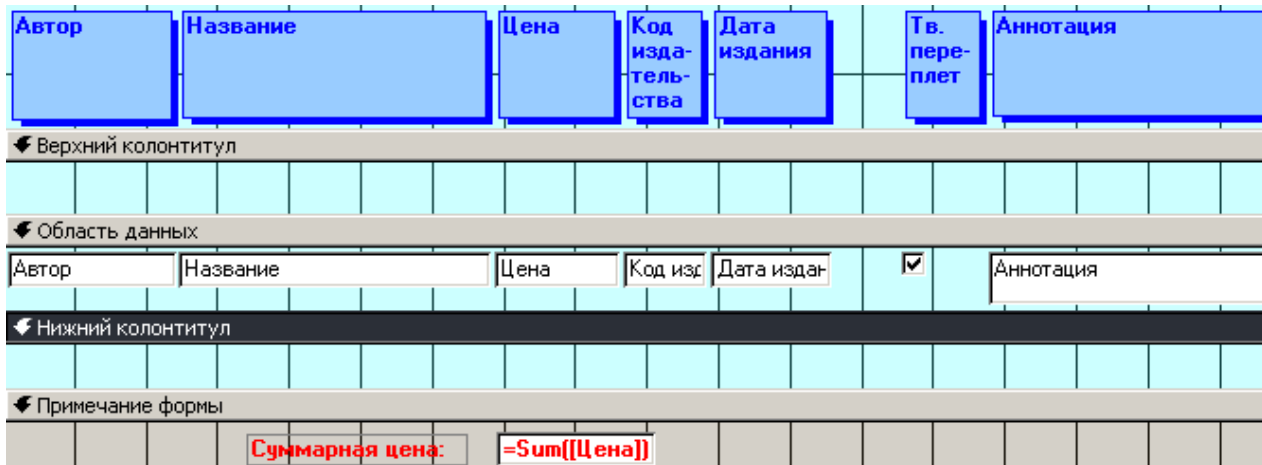


Поле, над которым выполняется функция суммирования, взято в квадратные скобки. Это необязательно, если имя поля не имеет внутри пробелов. Форма в режиме просмотра теперь будет выглядеть следующим образом:



В левом нижнем углу формы расположены кнопки для навигации по записям формы (**Кнопки перехода**). Эти кнопки могут быть удалены посредством установки соответствующего свойства формы, которое будет рассмотрено позже. Вместо этих кнопок для большей наглядности формы можно сформировать свои кнопки навигации. Действия по созданию этих кнопок аналогичны действиям по формированию кнопки закрытия формы, которые описаны ранее.

Кроме названных областей в форме могут быть задействованы области **Колонтитулы**. Эти области формируются при обращении к меню **Вид/Колонтитулы** в режиме **Конструктора**. В результате применения этого меню форма в режиме **Конструктора** примет следующий вид:

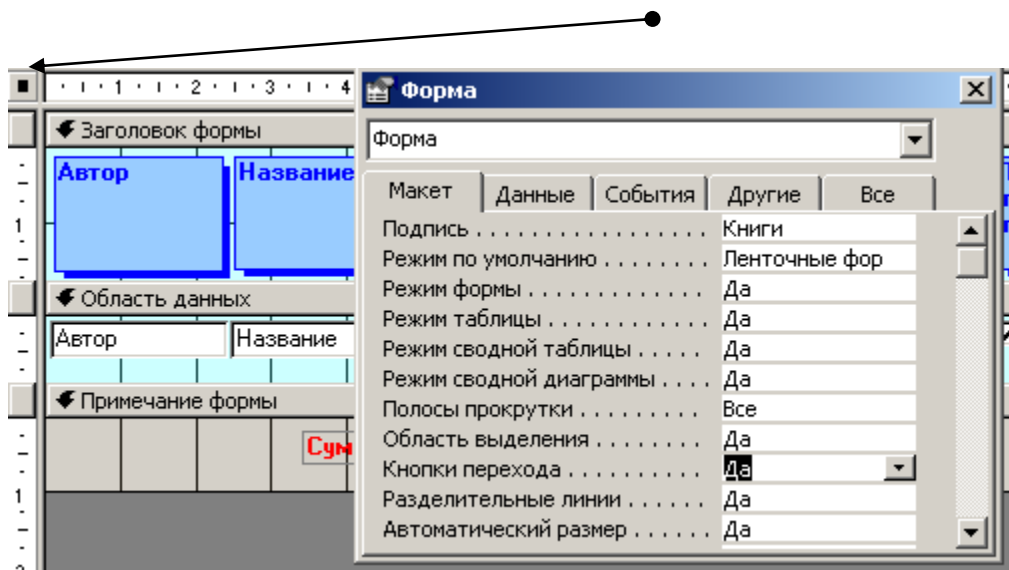


В областях **Верхний колонтитул** и **Нижний колонтитул** можно расположить текст, вычисляемые поля. Но они не будут отображаться в режиме просмотра формы. Выводиться они будут в начале и конце каждой страницы при выводе формы на печать.

10. Свойства формы и ее элементов.

4.1 Введение

За формой и всеми ее элементами закреплен набор свойств, которые определяют их внешний вид и многие другие характеристики. При формировании формы и ее элементов свойства назначаются по умолчанию. Нередко возникает потребность в их изменении. Чтобы обратиться к свойствам формы, необходимо в режиме **Конструктора** дважды щелкнуть по квадратику в верхнем левом углу формы.



После этого на экране сформируется набор свойств формы, расположенный на 5-й вкладке. Всего форма имеет более 100 свойств. Они сгруппированы по смыслу в 4-е

группы, на 5-й вкладке расположены все свойства. Ниже комментируются основные свойства формы.

4.2 Вкладка Макет.

На вкладке **Макет** сгруппированы свойства формы, которые определяют ее внешний вид.

Подпись позволяет задать текст, который выводится над формой в режиме ее просмотра.

Режим по умолчанию позволяет выбрать режим по умолчанию, в котором открывается форма. С помощью средств программирования можно открыть форму в другом режиме: в режиме таблица, сводная таблица и др.

Свойства **Режим формы, Режим таблицы, Режим сводной таблицы, Режим диаграммы** позволяют открыть форму в одном из названных режимов. Для этого необходимо всем названным свойствам, кроме одного, назначить значение Нет.

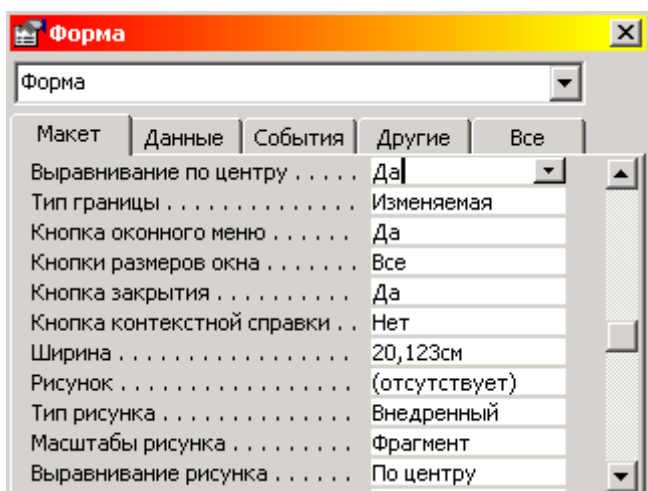
Свойство **Полосы прокрутки** позволяет отображать или нет в форме полосы прокрутки. Для кнопочных форм их отображать обычно не нужно.

Свойство **Кнопки перехода** позволяет отображать или нет стандартные навигационные кнопки в нижней части и формы.

Свойство **Разделительные линии** позволяет отображать линии, которые разделяют записи в ленточной форме.

Свойство **Автоматический размер задает** значение, показывающее будет ли окно формы при открытии автоматически изменять свой размер так, чтобы отобразились все записи.

На рисунке представлен следующий набор свойств вкладки **Макет**:



Свойство **Выравнивание по центру** обеспечивает центрирование формы на экране при ее открытии.

Свойство **Изменяемая граница** позволяет разрешить изменять границы формы в режиме ее просмотра.

Кнопка оконного меню, кнопка размеров окна, кнопка закрытия позволяют отображать или нет соответствующие элементы формы в режиме ее просмотра.

Свойство **Кнопка контекстной справки** определяет, будет ли в области заголовка формы отображаться кнопка контекстной справки **Что это такое?**

Свойство **Ширина** позволяет вручную задать ширину формы.

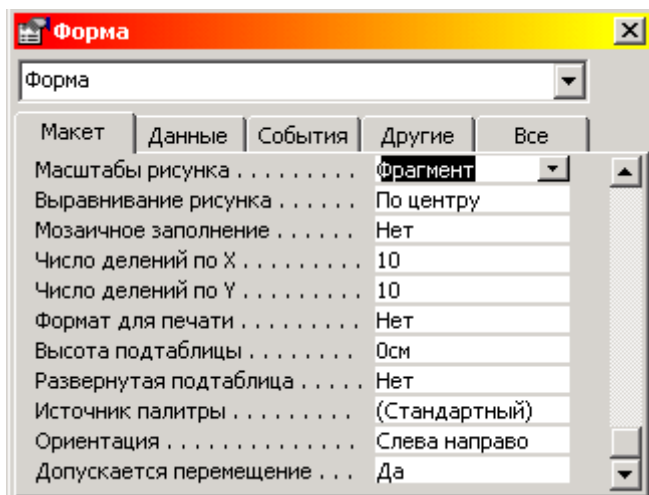
Свойство **Рисунок** позволяет сослаться на рисунок, на фоне которого располагаются элементы формы.

Тип рисунка может быть внедренный или ссылка. Рисунок типа ссылка позволяет значительно уменьшить объем базы данных, однако, при переносе БД на другой компьютер необходимо переносить и все связанные рисунки. Причем рисунки на другом компьютере необходимо располагать в одноименных папках.

С помощью свойства **Масштабы рисунка** рисунок можно разместить как фрагмент или вписать в рамку.

Выравнивание рисунка позволяет разместить рисунок в различных частях формы.

На рисунке представлен оставшийся набор свойств вкладки **Макет**:



Мозаичное заполнение позволяет разместить рисунок в форме в виде мозаики или как один рисунок.

Число делений по X и Y определяет минимальный шаг, на который можно перемещать элементы формы в режиме **Конструктора**.

Формат для печати позволяет установить используется или нет форматирование при выводе формы на печать. От этого зависит выбор шрифта: используются экранные шрифты или шрифты принтера.

Высота подтаблицы обычно используется в свойствах таблицы для назначения высоты связанной таблицы.

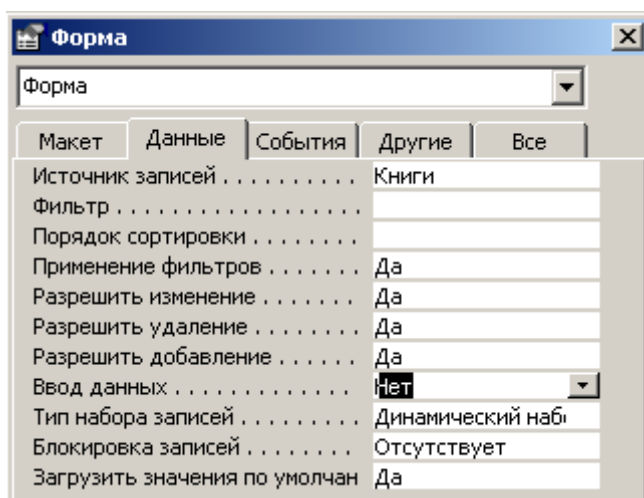
Развернутая подтаблица обычно используется в свойствах таблицы для запоминания состояния связанных таблиц.

Источник палитры позволяет использовать нестандартный набор цветов.

Ориентация определяет ориентации данных при выводе на печать

4.3 Вкладка Данные

На вкладке Данные сгруппированы следующие свойства:



Источник записей позволяет назначить таблицу или запрос, на базе которых формируется форма

Свойство **Фильтр** позволяет ввести выражение для фильтрации записей, выводимых в форме.

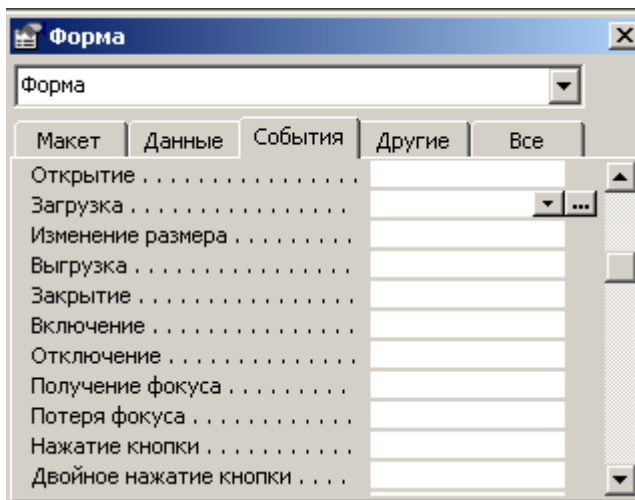
Разрешить изменение, Разрешить удаление, Разрешить добавление разрешают соответствующие режимы редактирования в форме.



Ввод данных позволяет использовать форму только для ввода данных. В форме будет представлена пустая запись для заполнения данными.

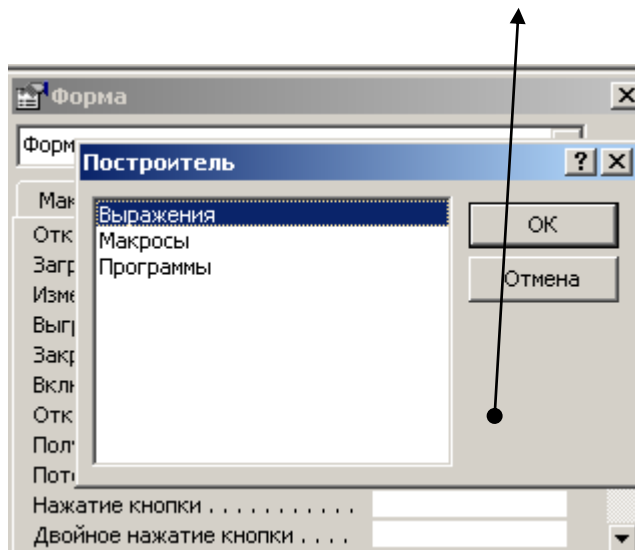
Блокировка записей позволяет заблокировать запись пользователю, если с ней работает другой пользователь.

4.4 Вкладка События

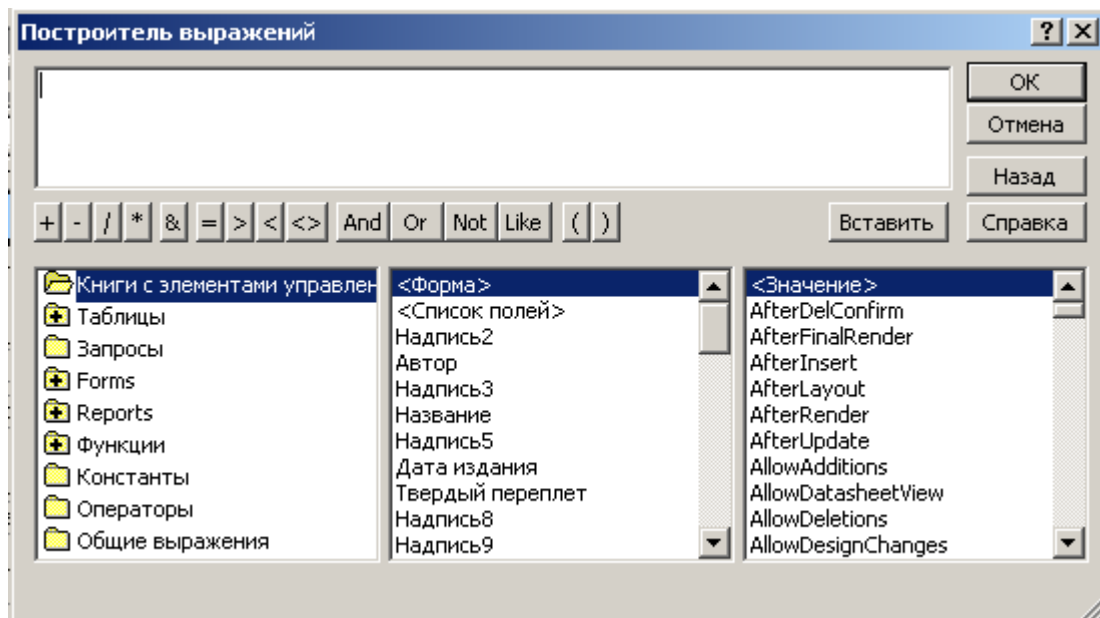
На вкладке **События** сгруппировано множество событий, которые могут возникать при работе с формой:



Каждому событию можно поставить в соответствие математическое выражение, запуск макроса или запуск программы. Для выбора макроса из списка существующих макросов необходимо щелкнуть по маркеру списка , а для построения нового выражения, макроса или процедуры необходимо щелкнуть по построителю выражения . После щелчка по построителю сформируется небольшое меню:



Если выбрать **Выражения**, откроется построитель выражений, с помощью средств которого нетрудно построить нужное выражение:



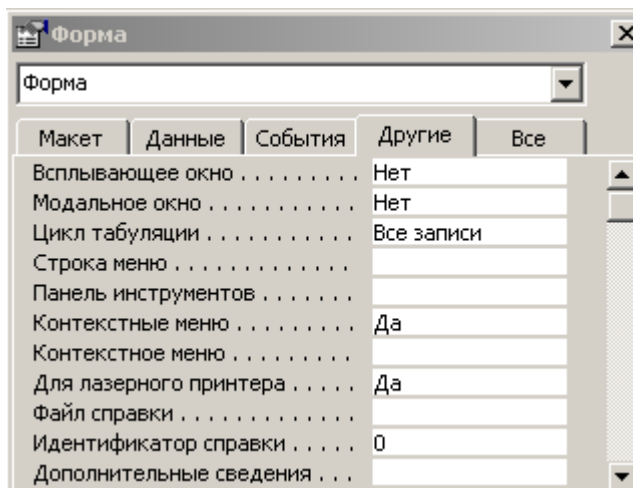
Для этого последовательно используются 3-и области и знаки операций.

Если выбрать **Макросы**, загрузится окно для описания макросов.

Если выбрать **Программы**, то загрузится редактор **Visual Basic**, который можно использовать для формирования процедуры. Эта процедура связывается с соответствующим событием и запускается при его наступлении.

4.5 Вкладка Другие

На вкладке **Другие** представлены события, которые не группируются по смыслу:



Свойство *Модальное окно* позволяет назначить окно как модальное. Если модальное окно будет открыто, то открыть другие окна будет нельзя, пока не закроется модальное окно.

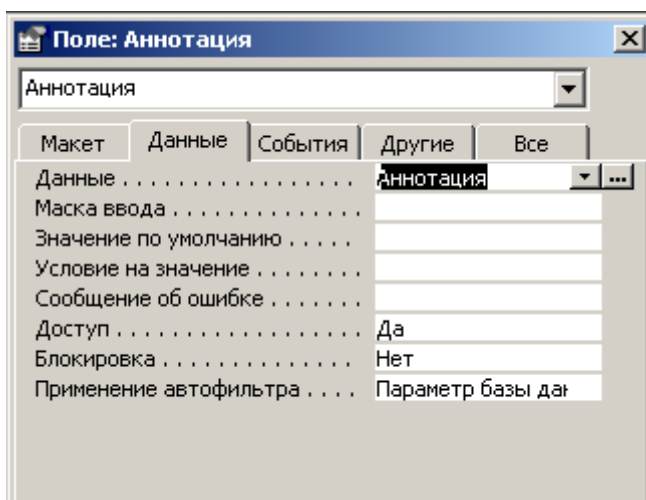
Цикл таблицы позволяет установить то, каким образом будет перемещаться указатель в форме при нажатии клавиши **Tab**.

С формой может быть связано *меню, контекстное меню, панель инструментов, справка.*

На вкладке **Все** расположены все свойства.

У всех областей формы и их элементов также имеются свойства. При этом большая часть этих свойств совпадает с рассмотренными свойствами. Однако некоторые свойства обусловлены спецификой объекта.

Например, группа свойств **Данные** для поля значительно отличается от одноименной группы свойств формы:



Как видно из рисунка, в качестве свойств поля задействованы *Маска ввода, Значение по умолчанию, Условие на значение, Сообщение об ошибке.* Эти свойства аналогичны соответствующим свойствам полей в таблицах. В связи с этим в форме можно назначить те свойства для полей, которые не назначены в таблицах.

Задание на лабораторную работу.

41. Создать форму в режиме Конструктора.
42. Разместить поля таблицы в этой форме.
43. Сформировать элемент Надпись и отформатировать его.
44. Создать элемент поле. Ввести в поле формулу.
45. Создать группы выключателей, переключателей, флажков.
46. Сформировать поле со списком и список
47. Спроектировать несколько кнопок для выполнения действий всех категорий.
48. Вставить рисунок в форму.

49. Вставить несколько OLE объектов в форму.
50. Создать пустую форму, расположить в ней элемент Вкладка, расположить на вкладках поля таблицы.
51. Внедрить в форму подчиненную форму.
52. Нарисовать и отформатировать линии и прямоугольники
53. Открыть форму, созданную мастером, в режиме конструктора.
54. Выполнить мероприятия по редактированию и форматированию формы и ее элементов.
55. Просмотреть все свойства формы, областей формы и ее элементов.
56. Выполнить эксперименты с основными свойствами формы, областей формы и ее элементов.

Проектирование форм в Access 2002 с помощью мастеров.

Введение

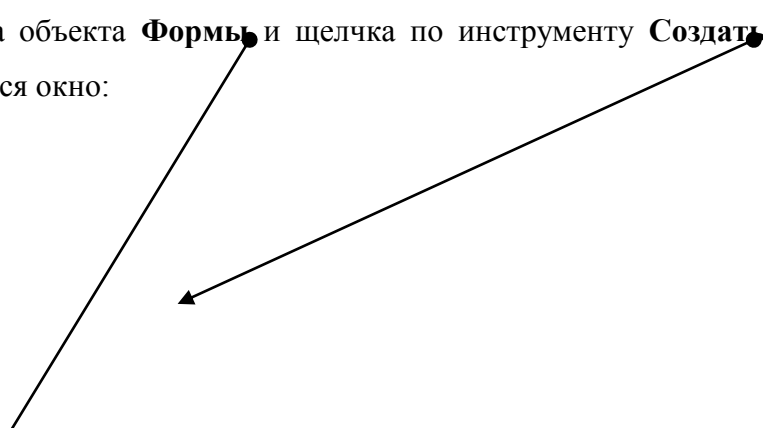
Формы являются средством организации интерфейса между пользователем и БД во всех СУБД, в том числе и в Access 2002. В основном формы используются для доступа к данным, причем в одной форме могут быть представлены данные из нескольких таблиц и/или запросов. Формы могут быть задействованы для открытия других форм, открытия таблиц, запросов, отчетов, запуска макросов и процедур. В Access 2002 предусмотрен ряд средств для оперативного проектирования форм и их качественного оформления.

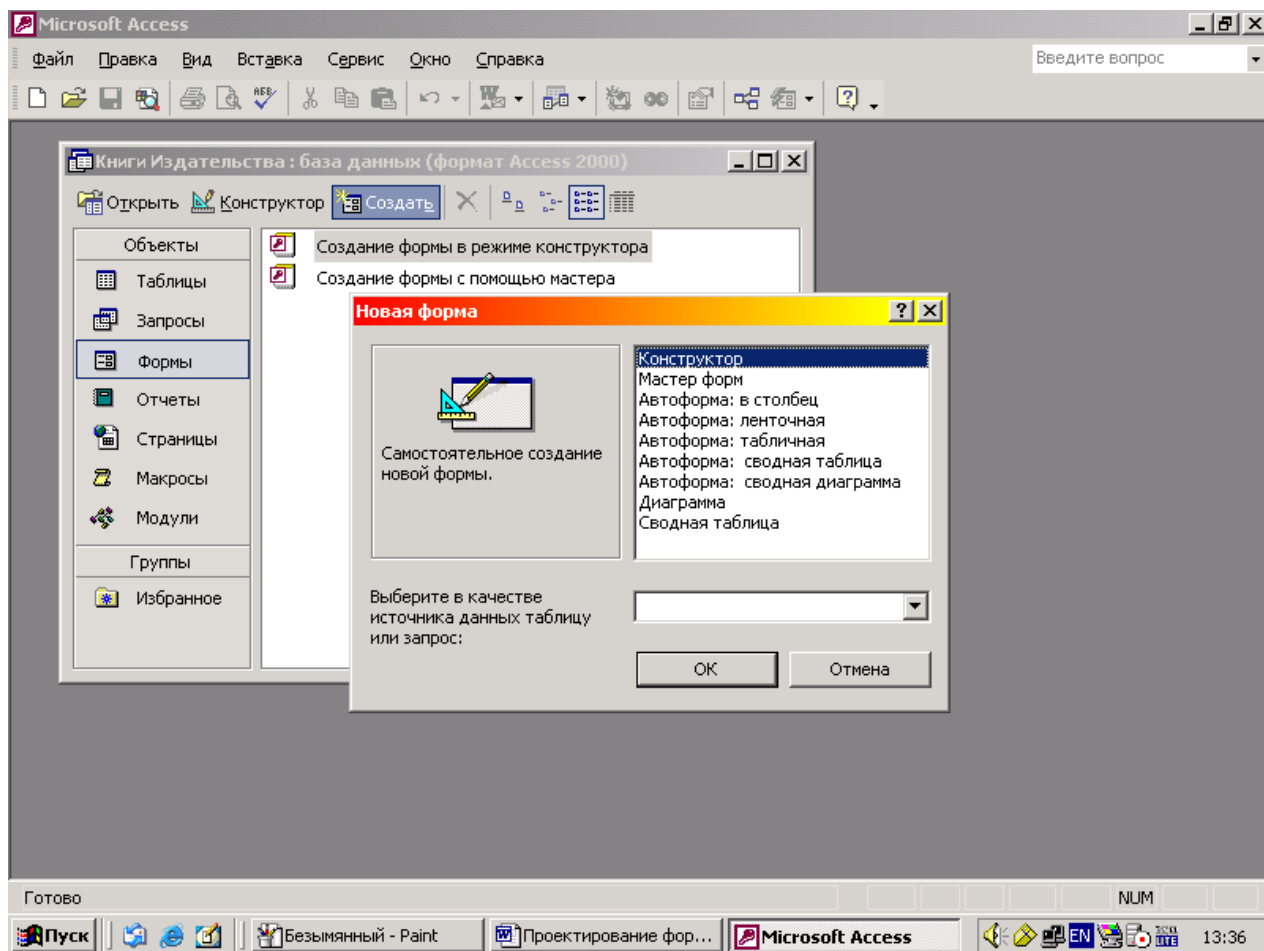
От того насколько хорошо разработчиком БД продуманы формы и их взаимосвязь существенно зависит коммерческая ценность разработанного программного продукта.

В Access возможны 9 способов создания форм. Для этого используются следующие средства: конструктор, мастер форм, автоформа в столбец, автоформа ленточная, автоформа табличная, автоформа сводная таблица, автоформа сводная диаграмма, диаграмма, сводная диаграмма.

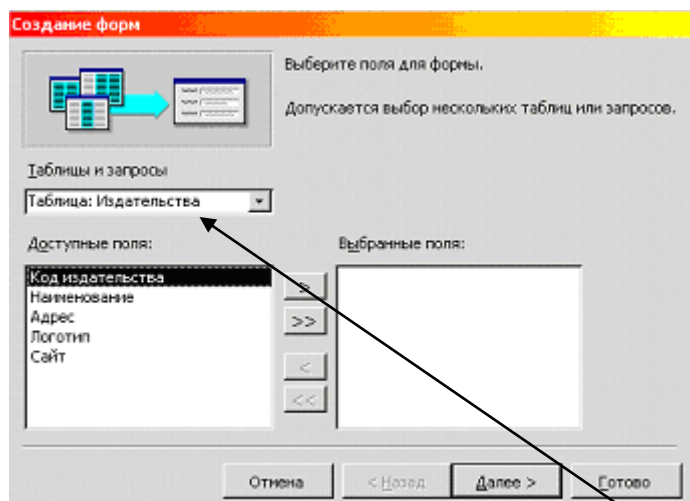
11. Проектирование форм с помощью Мастера форм.

После выбора объекта **Формы** и щелчка по инструменту **Создать** в базовом окне Access сформируется окно:



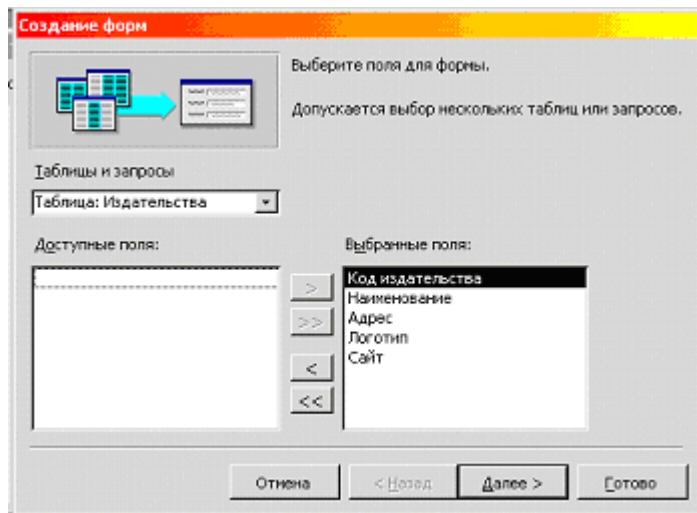


Для проектирования формы с помощью **Мастера** форм необходимо выбрать из списка позицию **Мастер форм**, затем в поле со списком (в нижней части окна) выбрать таблицу или запрос, для которых формируется форма и щелкнуть по кнопке **ОК**. После чего сформируется окно:

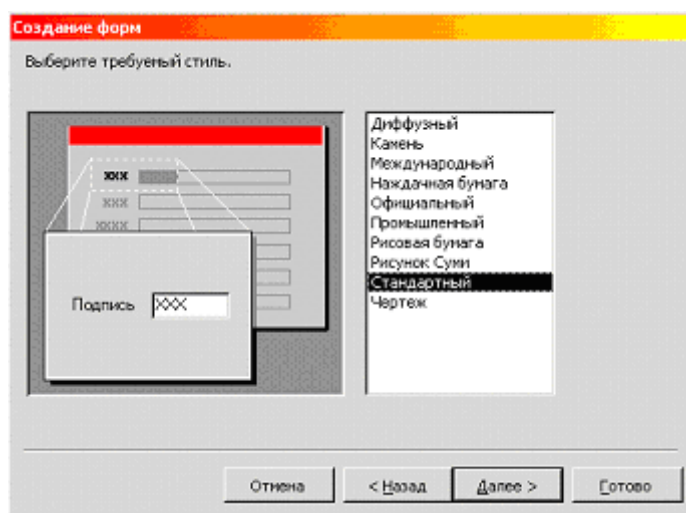


В данном случае создается форма для таблицы **Издательства**. Поэтому название этой таблицы и отображается в левой верхней части окна. При необходимости в этом поле можно выбрать другую таблицу или другой запрос. В нижней левой части окна необходимо выбрать поля, которые разработчик формы планирует в нее включить. Для

этого используются кнопки “>” и “>>”. В форме, приведенной ниже, предполагается задействовать в форме все поля таблицы, хотя это необязательно.

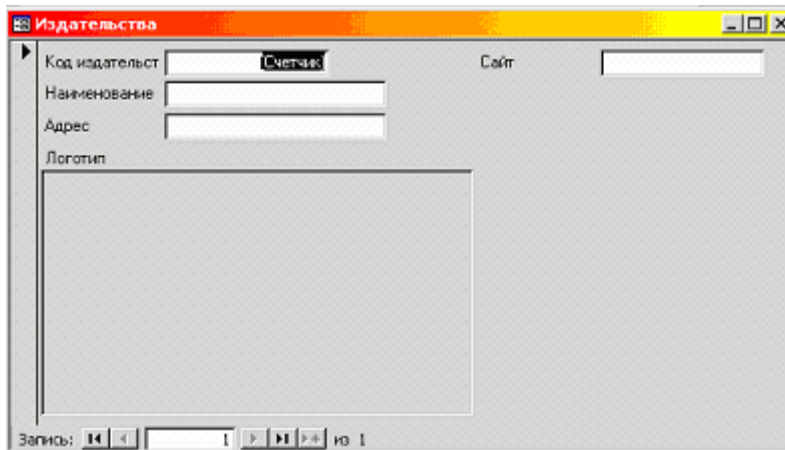


После нажатия кнопки **Далее** сформируется окно:



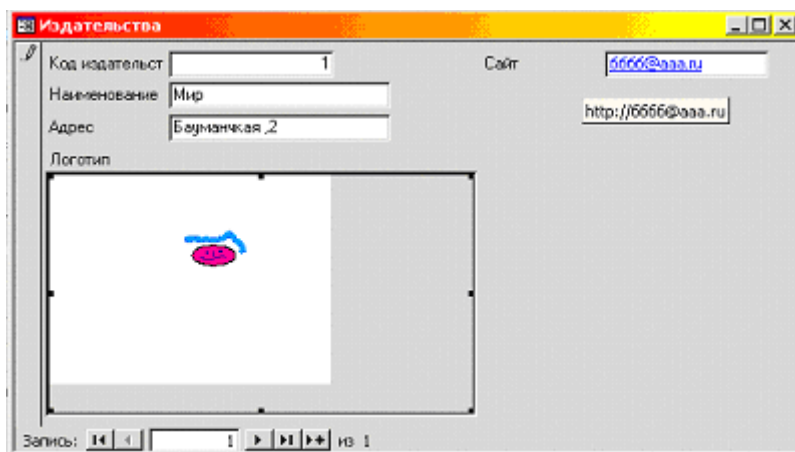
В нем можно выбрать один из стилей проектируемой формы. Стиль включает в себя совокупность элементов оформления формы – цвет, шрифт, форматы полей и т.п.

Если выбрать **Стандартный** стиль и нажать кнопку **Готово**, сформируется следующая форма:




В ней, как нетрудно заметить, задействованы все 5 полей таблицы **Издательства**. Размеры и расположение полей выбраны разработчиками Access исходя из различных соображений, в частности, в зависимости от числа полей, их типов длины и др. Например, для поля типа **OLE** отведено значительно больше места, чем для других полей – в этом поле может быть размещена объемная информация (рисунки, фотографии). В режиме Конструктора можно изменить внешний вид и содержание формы.

Ниже представлена форма с заполненными полями:

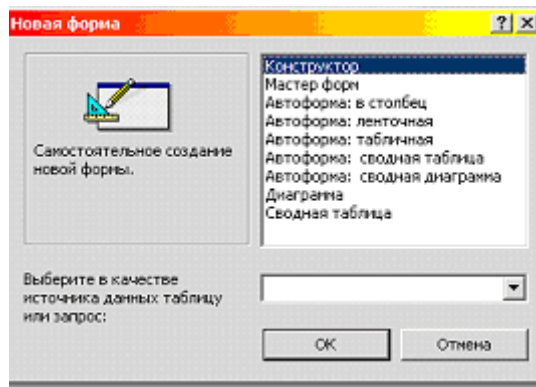


Поле **Код издательства** типа **счетчик** заполняется автоматически. Поле **Логотип** типа **OLE** заполняется с помощью меню **Вставка/объект**. Остальные поля заполняются прямым вводом данных.

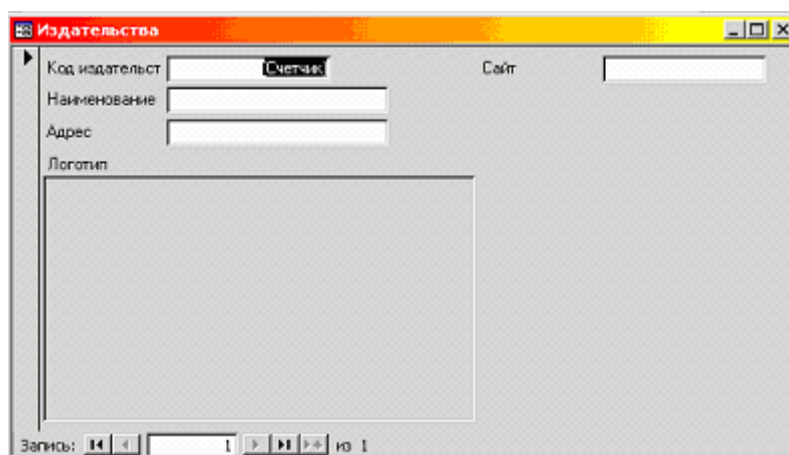
В нижней части окна расположены **кнопки перехода** . Эти кнопки позволяют перейти к следующей, предыдущей, первой, последней записи. Крайняя правая кнопка позволяет сформировать новую запись.

3. Автоформа в столбец.

Для создания **Автоформы в столбец** необходимо выбрать соответствующую позицию из списка типов новых форм, выбрать в поле со списком имя таблицы или запроса, для которых формируется форма, и щелкнуть по кнопке **ОК**.



После этого сформируется форма вида:



Нетрудно заметить, что эта форма имеет точно такой же вид, что и форма, созданная с помощью мастера форм. Поэтому этот режим формирования формы предпочтительней, если нужно оперативно сформировать форму в столбец. Однако, если нет необходимости использования всех полей формы, если имеется необходимость выбора стиля, оправдано использование мастера.

4. Автоформа ленточная

Для создания **Автоформы ленточной** необходимо выбрать соответствующую позицию из списка типов новых форм, выбрать в поле со списком имя таблицы или запроса, для которых формируется форма, и щелкнуть по кнопке **ОК**.

После этого сформируется форма вида:

Код книги	Автор	Название	Цена	злыства	дания	Тверды	Аннотация
▶	(Счетчик)		0,00р.	0		<input type="checkbox"/>	

Ленточная форма, отображающая несколько записей, выглядит следующим образом:

Код книги	Автор	Название	Цена	злыства	дания	Тверды	Аннотация
▶	1 Шиллер	Разбойники	120,00р.	2	1.2002	<input checked="" type="checkbox"/>	Классика мировой литературы
2	Шекспир	Отелло	55,00р.	2	1.2003	<input checked="" type="checkbox"/>	Классика мировой литературы
3	Мильтон	Потерянный	77,00р.	3	2.2002	<input checked="" type="checkbox"/>	
4	Данте	Божественн-	66,00р.	3	2.2002	<input type="checkbox"/>	
5	Гомер	Эллиада	55,00р.	3	2.2002	<input type="checkbox"/>	

Как видно из рисунка, в **ленточной форме**, в отличие от **формы в столбец**, видны одновременно несколько записей. Нередко такой формат вывода данных предпочтительней.

5. Автоформа табличная

Для создания **Автоформы табличной** необходимо выбрать соответствующую позицию из списка типов новых форм, выбрать в поле со списком имя таблицы или запроса, для которых формируется форма, и щелкнуть по кнопке **ОК**.

После этого сформируется форма вида:

Код книги	Автор	Название	Цена	Код изда	Дата издан	Твердый пе
▶	(Счетчик)		0,00р.	0		<input type="checkbox"/>

Табличная форма еще более компактная, чем ленточная, но менее наглядна. По сути, это таблица в режиме просмотра. Кроме того, в форме такого типа нельзя разместить элементы управления, о которых речь пойдет ниже.

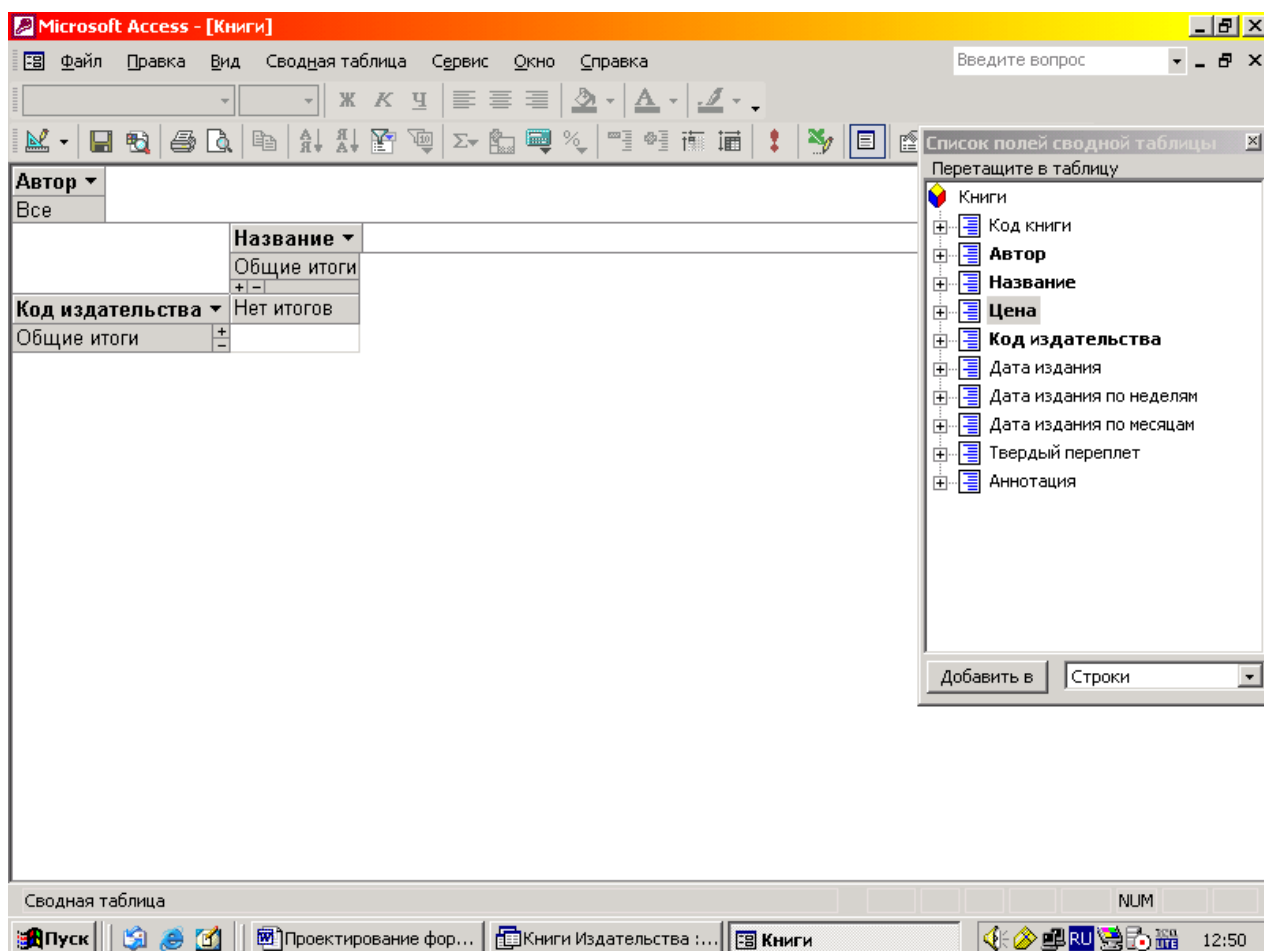
Табличная форма, отображающая несколько записей, выглядит следующим образом:

	Код книги	Автор	Название	Цена	Код изда	Дата изда
▶	1	Шиллер	Разбойники	120,00р.	2	01.01.2002
	2	Шекспир	Отелло	55,00р.	2	01.01.2003
	3	Мильтон	Потерянный рай	77,00р.	3	02.02.2002
	4	Данте	Божественная комедия	66,00р.	3	02.02.2002
	5	Гомер	Эллиада	55,00р.	3	02.02.2002
	6	Гомер	Одиссея	55,00р.	3	02.02.2003
	7	Грин	Алые паруса	77,00р.	1	01.01.2004
	8	Жюль Верн	Машина времени	90,00р.	1	01.01.2004
*	(Счетчик)			0,00р.	0	

6. Автоформа Сводная таблица

Для создания **Автоформы Сводная таблица** необходимо выбрать соответствующую позицию из списка типов новых форм, выбрать в поле со списком имя таблицы или запроса, для которых формируется форма, и щелкнуть по кнопке **ОК**.

После этого сформируется форма вида:



В правой части окна представлен список полей таблицы или запроса, на базе которых строится форма. Любое поле из списка можно переместить в соответствующую область сводной таблицы, формируя, таким образом, вид сводной таблицы.

Вид сводной таблицы, отображающей данные, представлен на рисунке:

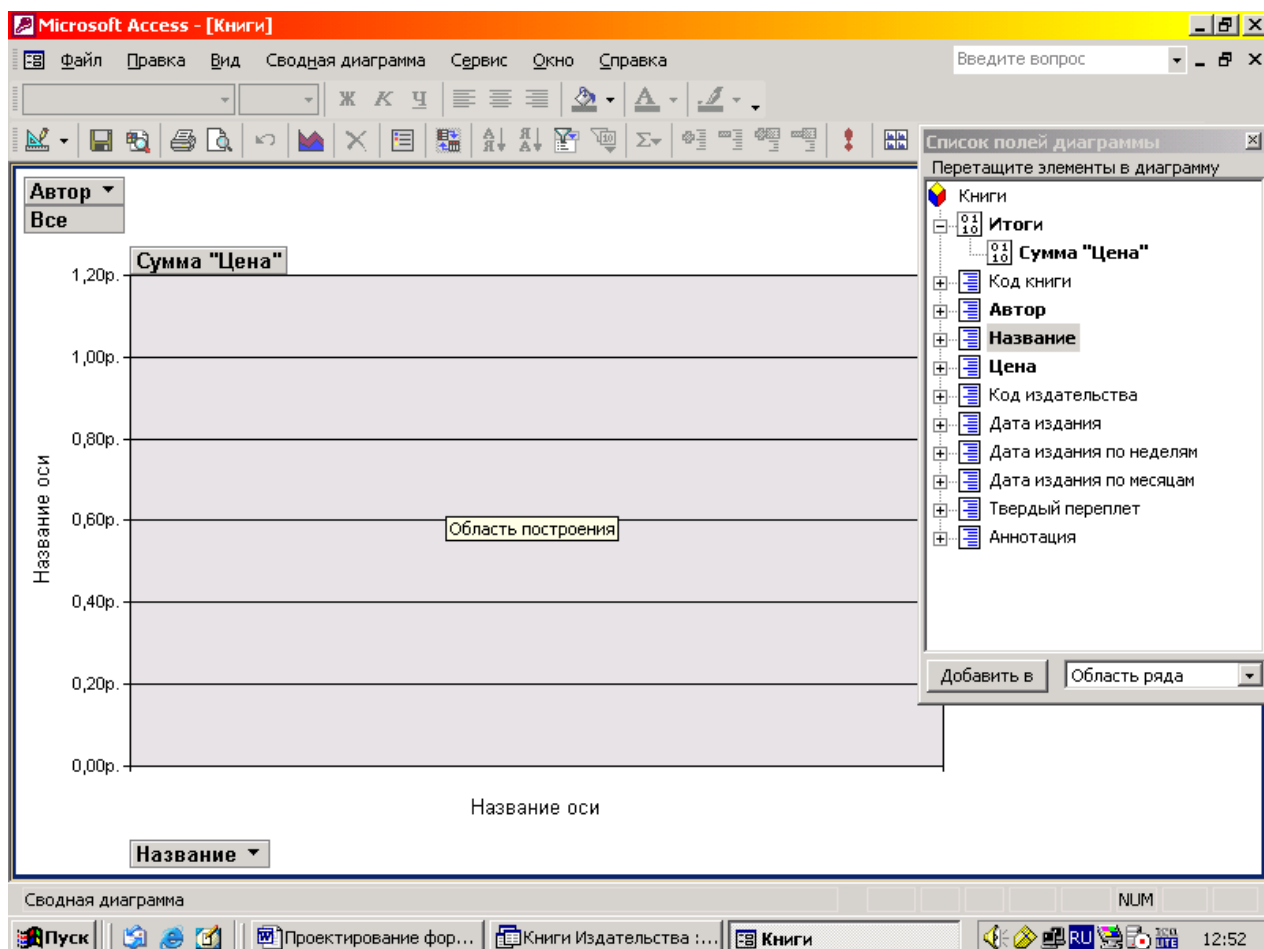
Автор ▾		Название ▾				
Все		Алые паруса		Божественная комедия	Машина времени	Одиссея
		+ -		+ -	+ -	+ -
Код издательства ▾	Цена ▾	Цена ▾	Цена ▾	Цена ▾	Цена ▾	
1	77,00р.			90,00р.		
2						
3		66,00р.			55,00р.	
Общие итоги						

Смысл сводной таблицы и технология работы с ней аналогичны соответствующим понятиям для сводных таблиц в Excel.

7. Автоформа Сводная диаграмма.

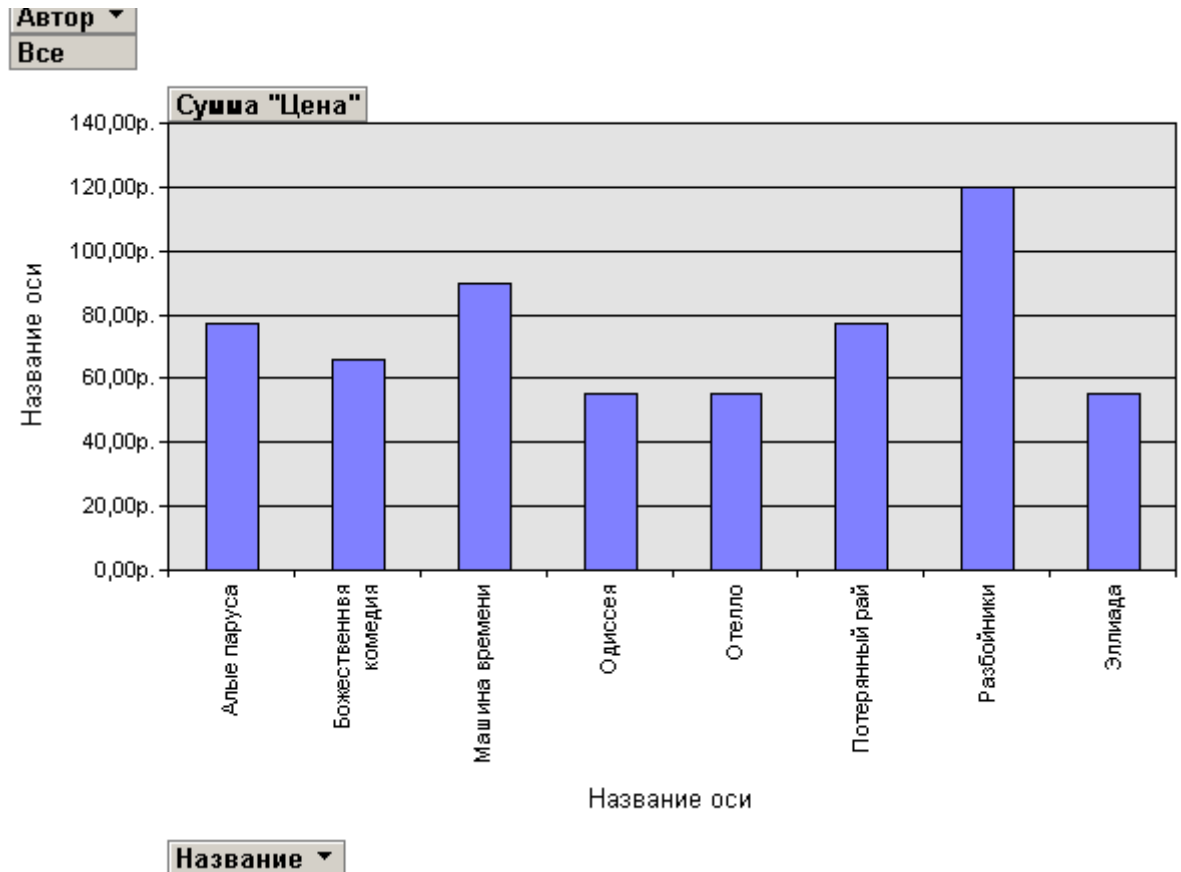
Для создания Автоформы Сводная диаграмма необходимо выбрать соответствующую позицию из списка типов новых форм, выбрать в поле со списком имя таблицы или запроса, для которых формируется форма, и щелкнуть по кнопке **ОК**.

После этого сформируется форма вида:

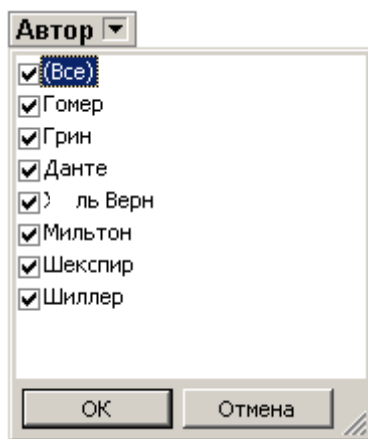


В правой части окна представлен список полей таблицы или запроса, на базе которых строится сводная диаграмма. Любое поле из списка можно переместить в соответствующую область сводной диаграммы, формируя, таким образом, вид сводной диаграммы.

Вид сводной диаграммы отображающей данные представлен на рисунке:



Списковое поле Автор в верхней части окна позволяет указать авторов, для которых отображается сводная диаграмма. Открытый список авторов выглядит следующим образом:



В этом списке можно выбрать тех авторов, для которых необходимо построить диаграмму. Например, если установить флажки следующим образом,

Автор

(Все)

Гомер

Грин

Данте

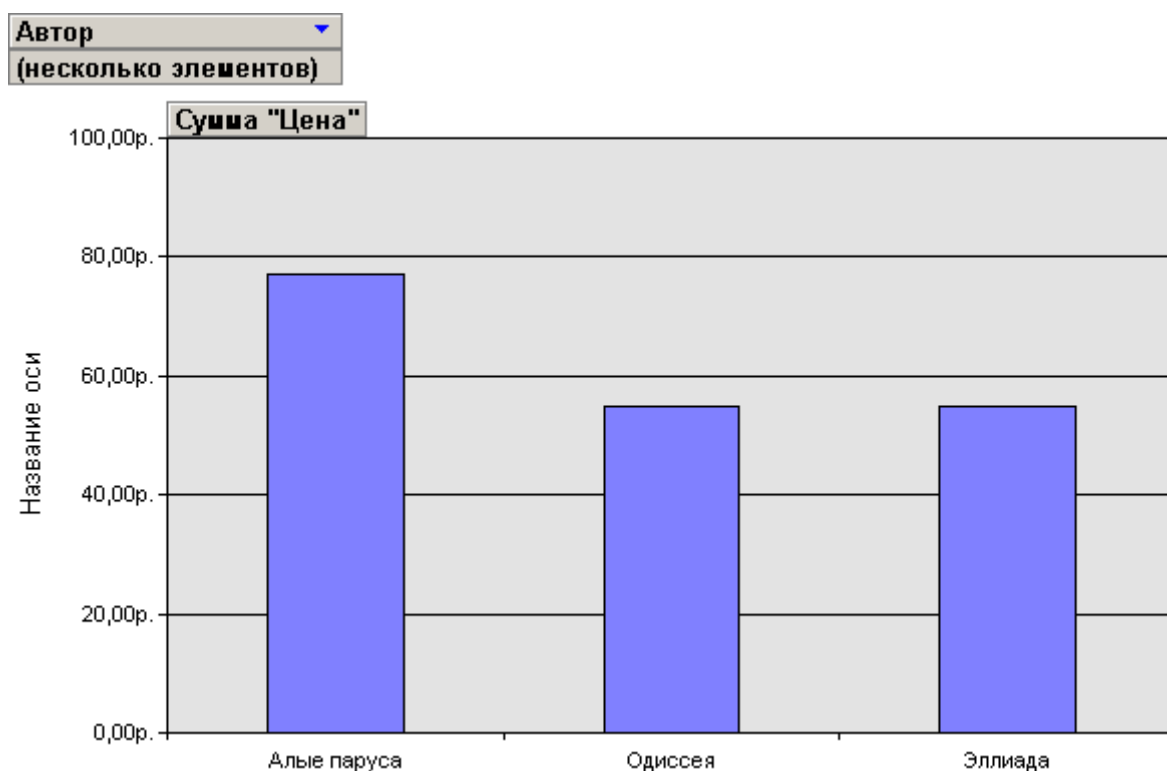
...ль Верн

Мильтон

Шекспир

Шиллер

сводная диаграмма будет представлена в следующем виде:

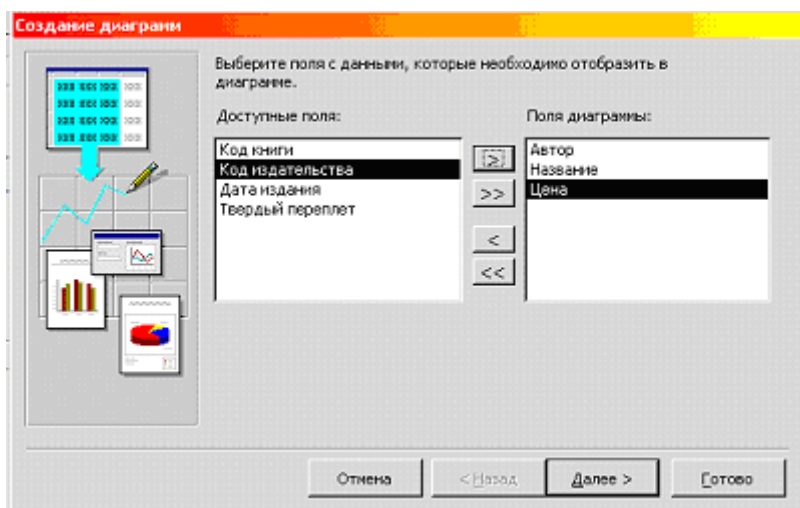


Аналогичным образом можно манипулировать со списковым полем Название.

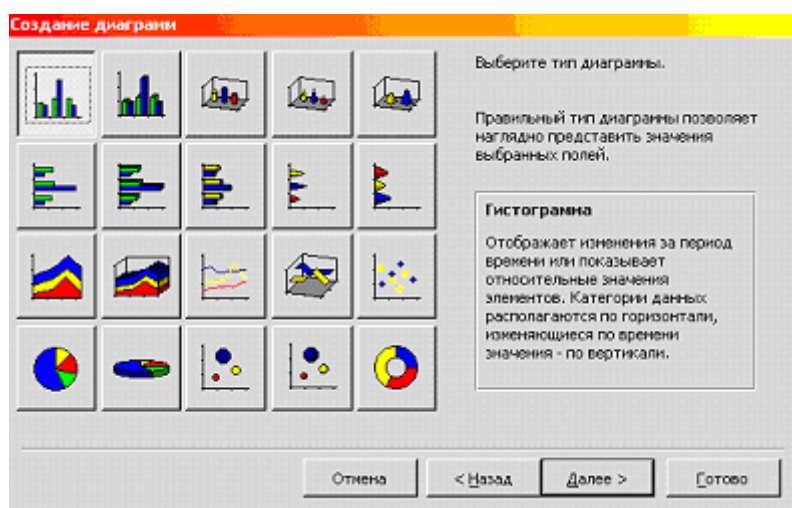
8. Диаграмма

Для создания **Автоформы Диаграмма** необходимо выбрать соответствующую позицию из списка типов новых форм, выбрать в поле со списком имя таблицы или запроса, для которых формируется форма, и щелкнуть по кнопке **ОК**.

После этого сформируется форма вида:



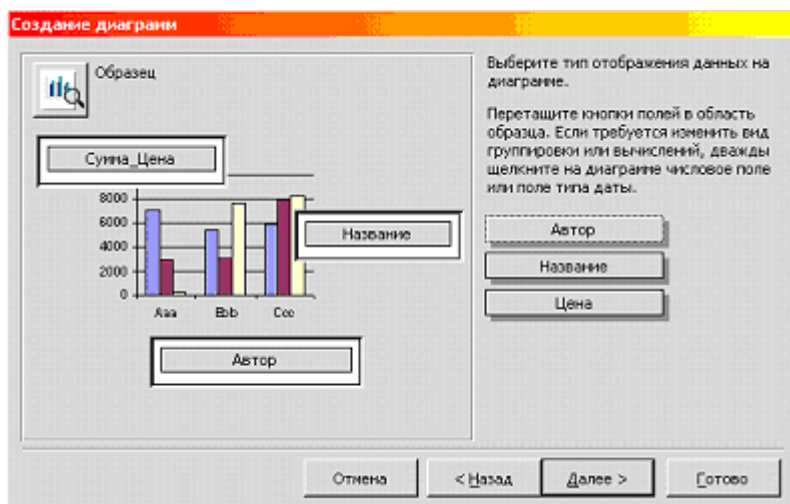
Затем в этом окне необходимо выбрать поля, для которых необходимо построить диаграмму и нажать кнопку **Далее**. В результате сформируется следующее окно:



В этом окне выбирается тип диаграммы. Следует помнить, что различные типы диаграмм требуют различного числа полей. В частности, для **круговой диаграммы** необходимы данные одного поля с числовым значением, для **гистограммы** необходимы данные из 2-х или 3-х полей, одно из которых должно быть числового типа.

После выбора типа диаграммы, допустимой для используемого состава полей, необходимо нажать кнопку **Далее**. После чего загрузится окно для расположения полей на диаграмме. Нередко мастер построения диаграмм сам размещает поля на макете диаграммы.

Эта форма представлена ниже:



В случае если разработчик диаграммы предполагает расположить поля на диаграмме, то он должен перетащить поля из правой части окна в нужное место на макет диаграммы.

После нажатия кнопки **Готово** сформируется диаграмма следующего вида.

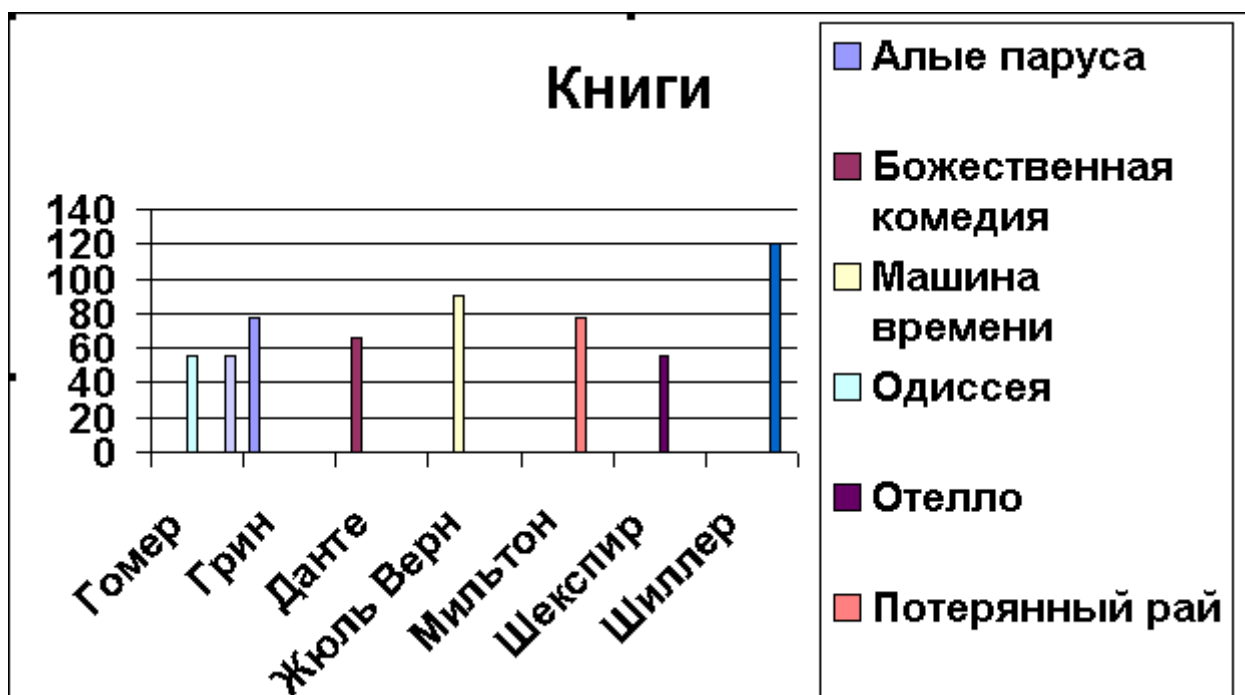


Диаграмма для рассматриваемой таблицы мало информативна и мало полезна. Однако можно назвать множество случаев, когда диаграммы могут быть очень полезными. Достаточно полистать экономические и технические журналы, чтобы в этом убедиться.

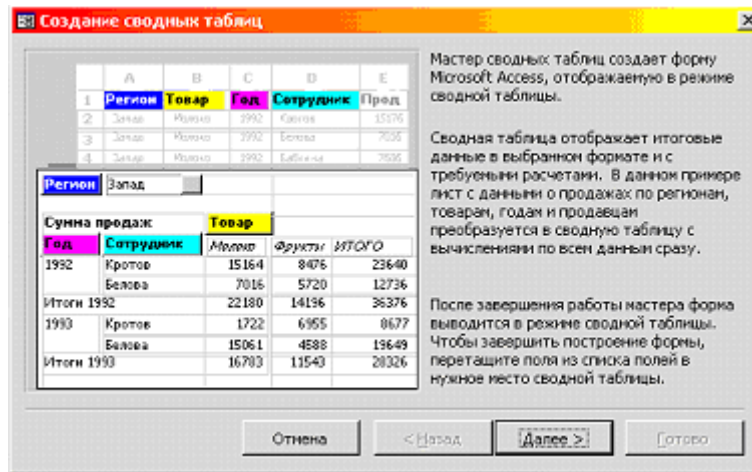
9. Сводная таблица

Выше рассмотрено, как **сводная таблица** строится посредством режима **Автоформа Сводная таблица**. Цель использования режима **Сводная таблица** аналогична. Процесс

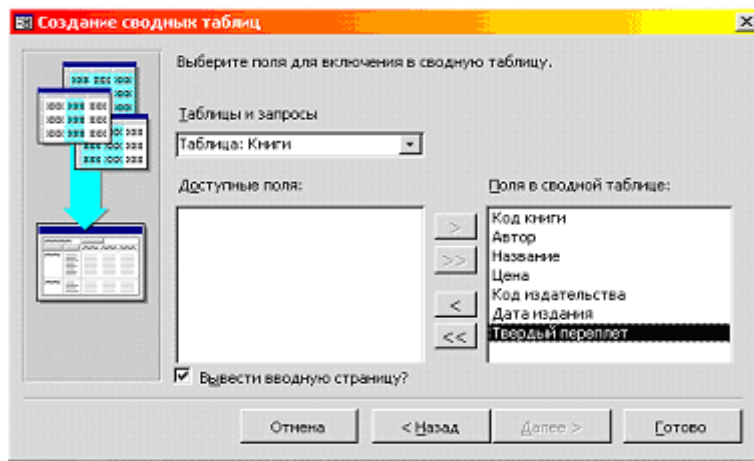
формирования несколько отличается. Этот режим задействован для совместимости с предыдущими версиями Access.

Для создания **Сводной таблицы** необходимо выбрать соответствующую позицию из списка типов новых форм, выбрать в поле со списком имя таблицы или запроса, для которых формируется форма, и щелкнуть по кнопке **ОК**.

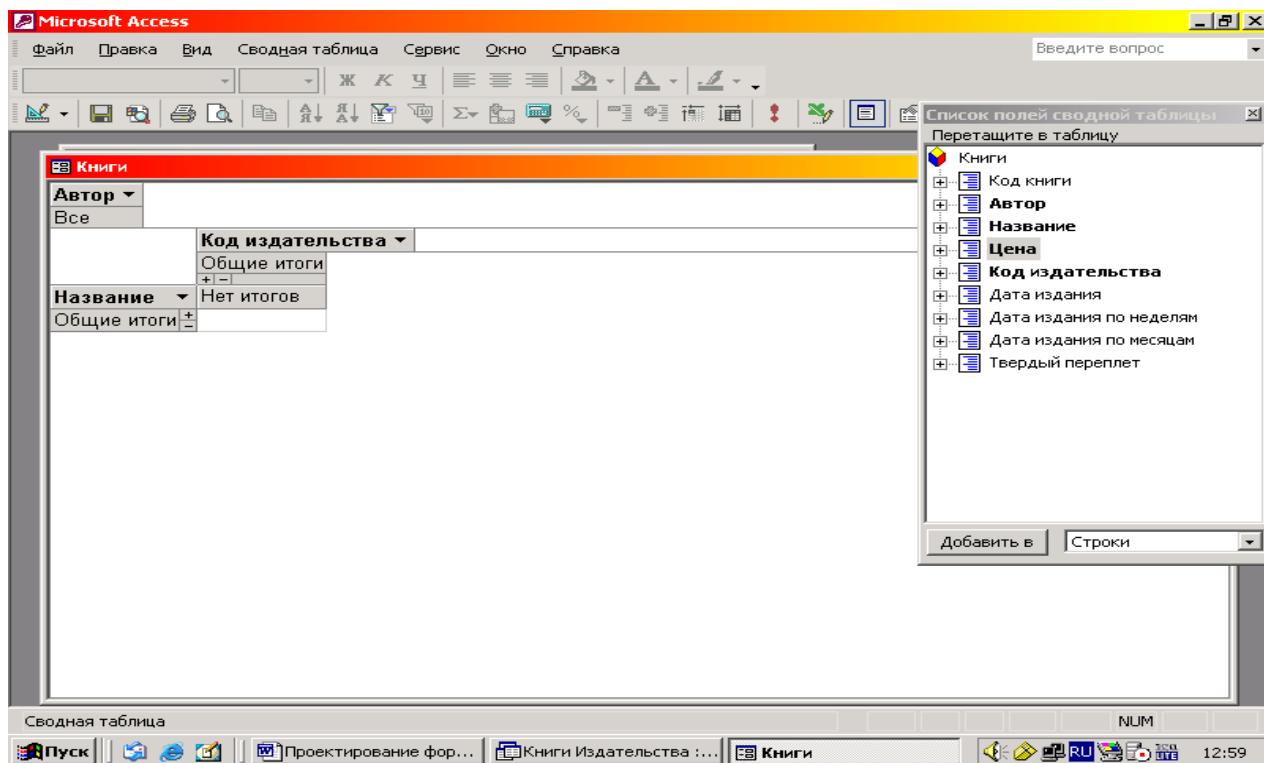
После этого сформируется форма вида:



В данной форме представлена информация справочного характера. После ее прочтения следует нажать кнопку **Далее**. Это приведет к загрузке окна:



В данном окне необходимо выбрать поля, для которых предполагается сформировать сводную таблицу. После нажатия кнопки **Готово** отобразится уже знакомое окно.

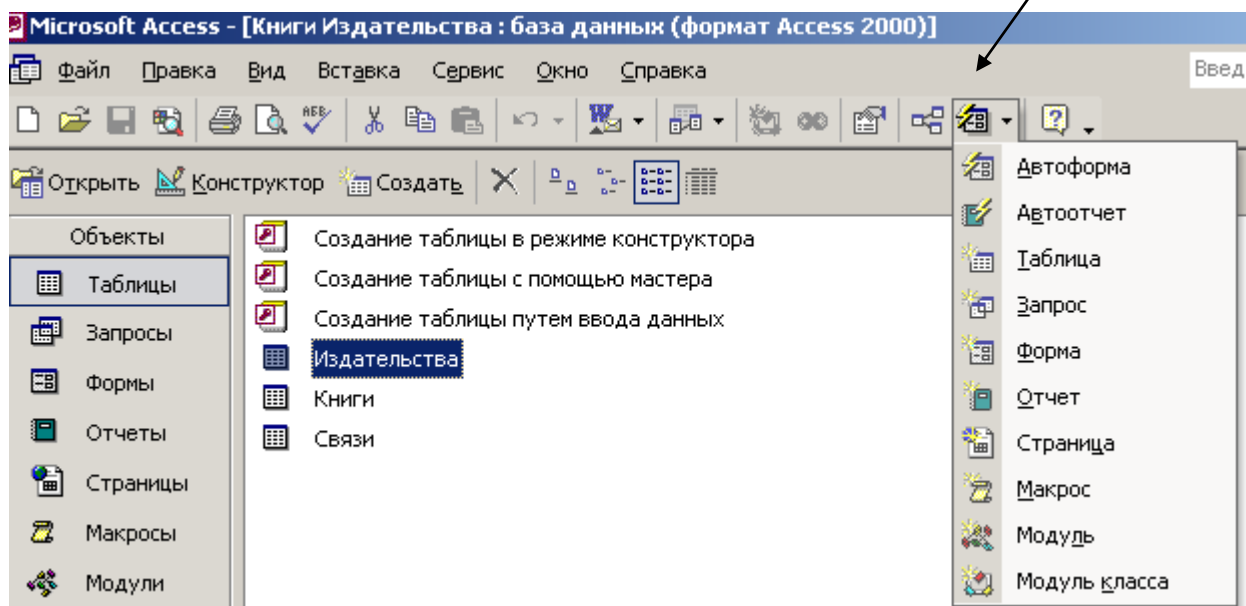


Возможные действия в этом окне аналогичны действиям, которые описаны ранее.

Выше описаны режимы, которые становятся доступными, если в базовом окне Access выбрать объекты **Формы** и щелкнуть по инструменту **Создать**. Не рассмотрен только первый режим Конструктор. Этот режим представляет особый интерес и будет рассмотрен отдельно.

9. Создание формы посредством инструмента Новый объект

В базовом окне Access на основной панели инструментов задействован инструмент **Новый объект**. На рисунке представлен этот инструмент в развернутом виде.



С помощью режима **Автоформа** можно за один шаг построить форму в столбец. Часто результат данного способа построения формы ничем не отличается от результата применения ранее рассмотренного способа. Однако, если таблицы связаны между собой, то в рамках основной формы формируется и подчиненная форма. В этом и есть принципиальное отличие рассматриваемого режима.

Рассматривается случай, когда таблицы связаны следующим образом:



Для инициации режима **Автоформа** необходимо в базовом окне Access выбрать объекты **Таблицы**. Затем выделить таблицу, для которой строится форма. И после этого с помощью инструмента **Новый объект** выбрать режим **Автоформа**.

Форма, сформированная описанным путем, для таблицы **Издательства** выглядит следующим образом:

Код книги	Автор	Название
7	Грин	Алые паруса
8	Жюль Верн	Машина времени
*	(Счетчик)	

Как видно из рисунка, в форме представлена запись таблицы **Издательства** и все, связанные с этой записью записи таблицы **Книги**. Такое представление данных является

чрезвычайно удобным. В частности, в такой форме можно вводить данные об издательстве, а также вводить данные обо всех книгах, изданных в данном издательстве.

С помощью навигационных кнопок, которые расположены в нижней части формы



можно перемещаться по записям таблицы Издательства и связанным записям таблицы Книги.

Такая форма удобна не только для ввода связанных данных, но и для их просмотра.


Однако, видны не все связанные поля таблицы Издательства. Для просмотра полей, которые не поместились в форму необходимо задействовать кнопки прокрутки.

Не всегда удобно постоянно использовать кнопки прокрутки. Для улучшения ситуации можно отредактировать форму с помощью конструктора форм. В результате редактирования форма может принять следующий вид:

Код издательства

Наименование

Адрес

Логотип 

Сайт

	Код книги	Автор	Название	Цена	Дата издания	Твердый п
▶	7	Грин	Алые паруса	77,00р.	01.01.2004	<input checked="" type="checkbox"/>
	8	Жюль Верн	Машина времени	90,00р.	01.01.2004	<input checked="" type="checkbox"/>
*	(Счетчик)			0,00р.		<input type="checkbox"/>

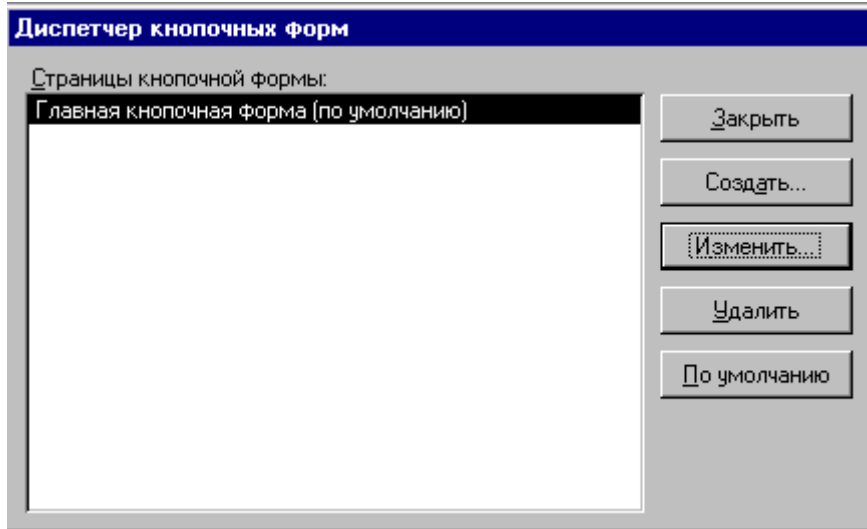
Как видно из рисунка использование **Конструктора** форм значительно улучшило наглядность формы. Это только малая часть возможностей **конструктора форм**.

10. Диспетчер кнопочных форм.

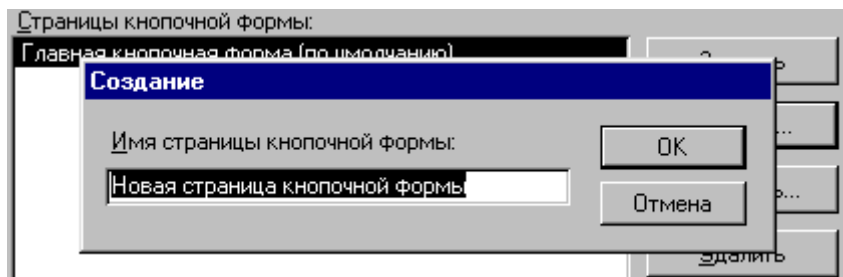
Кнопочные формы представляют собой формы, на которых расположены кнопки. Эти кнопки могут быть использованы для открытия других кнопочных форм или выполнения конкретных действий. Примерами конкретных действий могут быть: открыть отчет, выполнить программу, открыть форму, открыть запрос.

Кнопки можно создать и в режиме конструктора. Однако использование **Диспетчера кнопочных форм** нередко оказывается более удобным для разработчика. В частности, при использовании **Диспетчера** не нужно задумываться о стандартизации интерфейса.

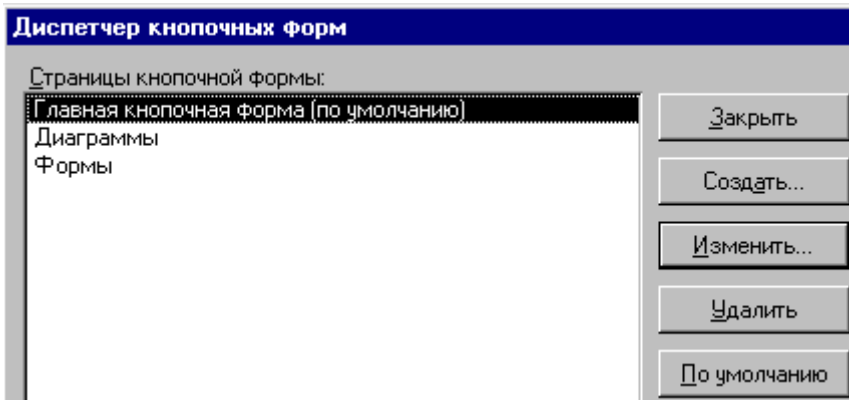
Для запуска **Диспетчера кнопочных форм** необходимо обратиться к меню **Сервис/Настройки/Диспетчер кнопочных форм**. В результате загрузится окно:



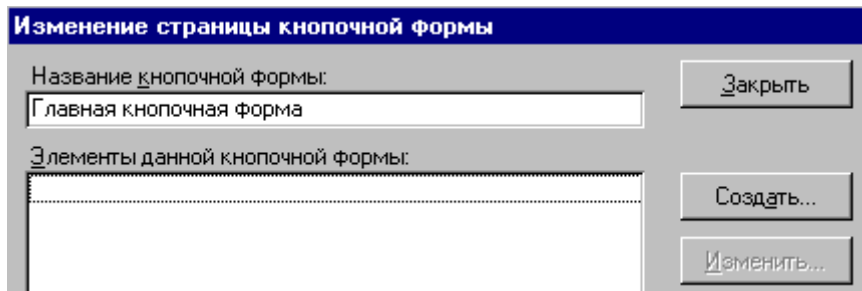
Главная кнопочная форма уже создана, но она пока пустая. Если разработчик предполагает задействовать несколько кнопочных форм, то он должен щелкнуть по кнопке **Создать**. В результате сформируется окно, в котором можно задать имя новой кнопочной формы:



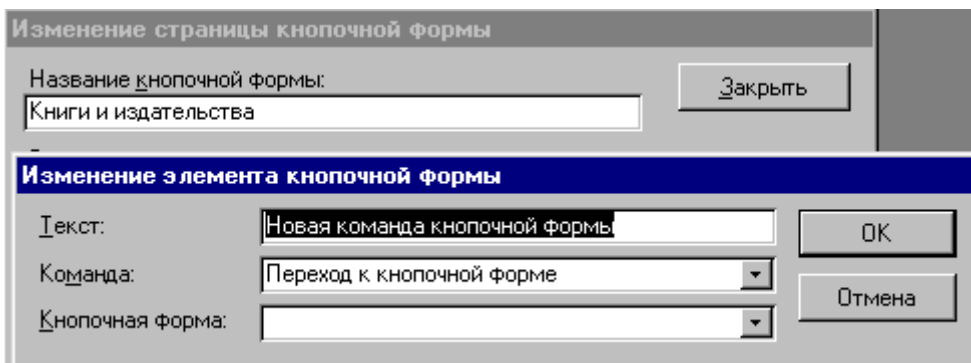
В результате применения этого режима дважды и назначения информативных имен сформируется окно:



Таким образом, спроектированы 3-и кнопочных формы. Теперь на них нужно разместить кнопки и закрепить за этими кнопками действия. Для этого нужно выделить нужную страницу кнопочной формы и щелкнуть по кнопке Изменить. В результате выполнения этих действий для главной кнопочной формы сформируется окно:

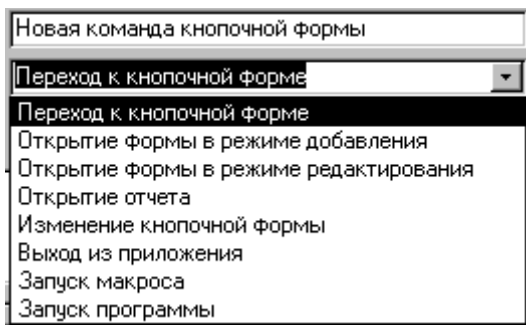


Имеет смысл изменить имя **Главная кнопочная форма** на более информативное, например **Книги и издательства**. Для создания кнопки необходимо щелкнуть по кнопке **Создать**. После этого сформируется окно:

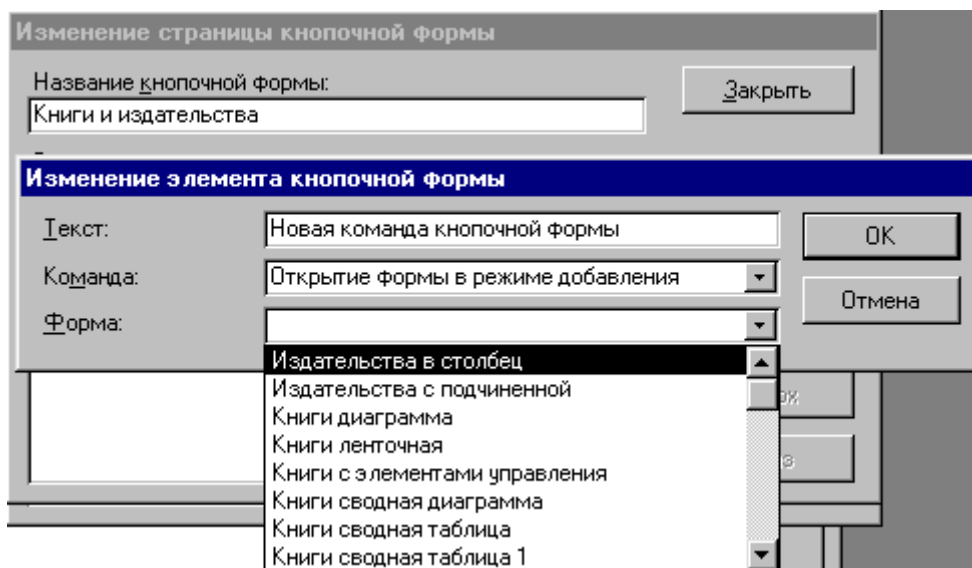


В поле **Текст** необходимо набрать текст для создаваемой кнопки. В поле со списком **Команда** необходимо выбрать из раскрытого списка команду, которая будет выполняться при нажатии проектируемой кнопки. В поле со списком **Кнопочная форма** необходимо выбрать из раскрытого списка конкретный объект, который будет загружаться при нажатии проектируемой кнопки.

Список команд следующий:

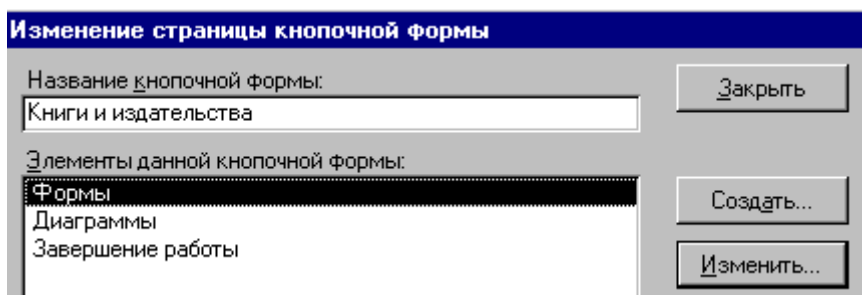


В зависимости от выбранной команды нижнее поле может именоваться различным образом: **Кнопочная форма**, **Форма**, **Отчет** и др. В случае команды **Открытие формы** в режиме добавления этот список будет иметь вид:



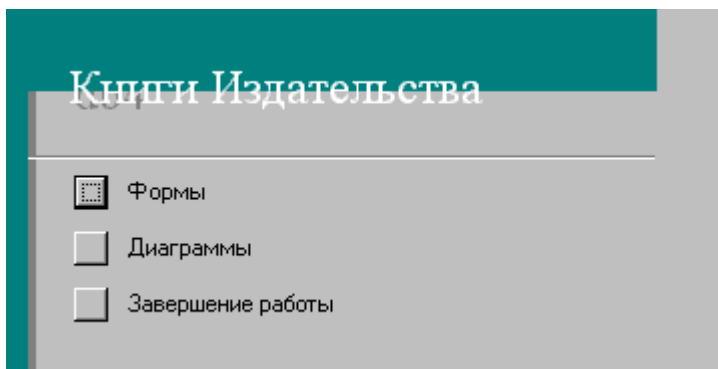
В списке приведены все объекты типа **Форма**.

Использована дважды команда **Переход к кнопочной форме** и объекты кнопочные формы **Диаграмма** и **Форма**. Использована команда **Выход из приложения**. Это позволило получить следующую картину:

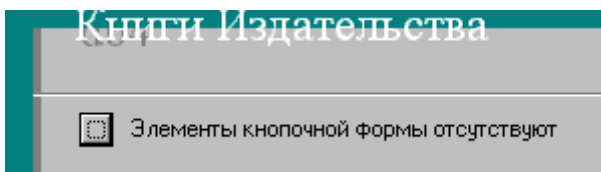


Таким образом, Создана Главная кнопочная форма (**Книги издательства**), в которой размещены 3- кнопки. Две из них служат для открытия других кнопочных форм, одна для выхода из приложения.

После закрытия **Диспетчера кнопочных форм** и открытия **Кнопочной формы** в режиме просмотра сформируется форма следующего вида:



При нажатии на кнопку **Формы** загрузится соответствующая форма:

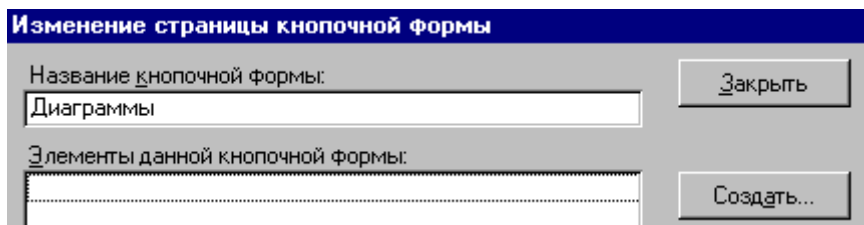


То же самое будет при нажатии на кнопку **Диаграммы**.

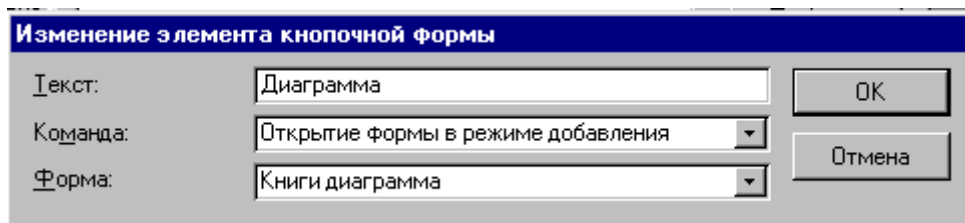
Причина отсутствия элементов кнопочной формы в том, что для данной формы не сформированы кнопки.

При нажатии на кнопку **Завершение работы** приложение завершит работу.

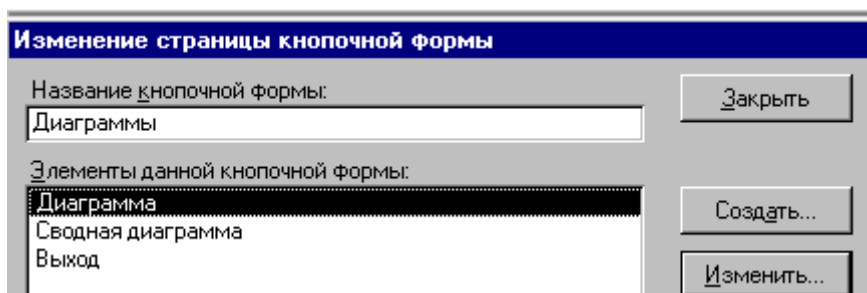
Для определения кнопочных форм необходимо вновь обратиться к **Диспетчеру кнопочных форм**, выбрать из списка нужную кнопочную форму и нажать кнопку **Изменить**. После чего откроется окно:



В этом окне нужно щелкнуть по кнопке создать, сформировать элемент кнопочной формы, как это делалось ранее. Например:

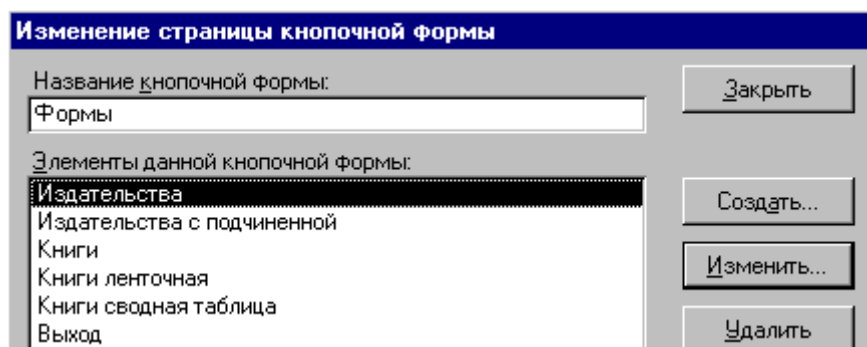


По аналогии можно создать и другие кнопки. Одна из кнопочных форм в режиме **Диспетчера** в результате может выглядеть следующим образом:

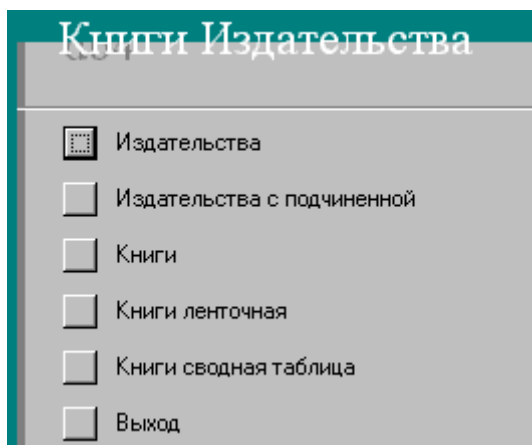


Здесь за кнопкой с подписью **Выход** закреплена команда **Переход к кнопочной форме**, а в качестве объекта - главная кнопочная форма **Книги и издательства**. Таким образом, посредством данной кнопки осуществляется возврат из подчиненной формы в главную форму.

По аналогии можно изменить кнопочную форму **Формы**. В результате кнопочная форма **Форма** в режиме **Диспетчера** может принять вид:

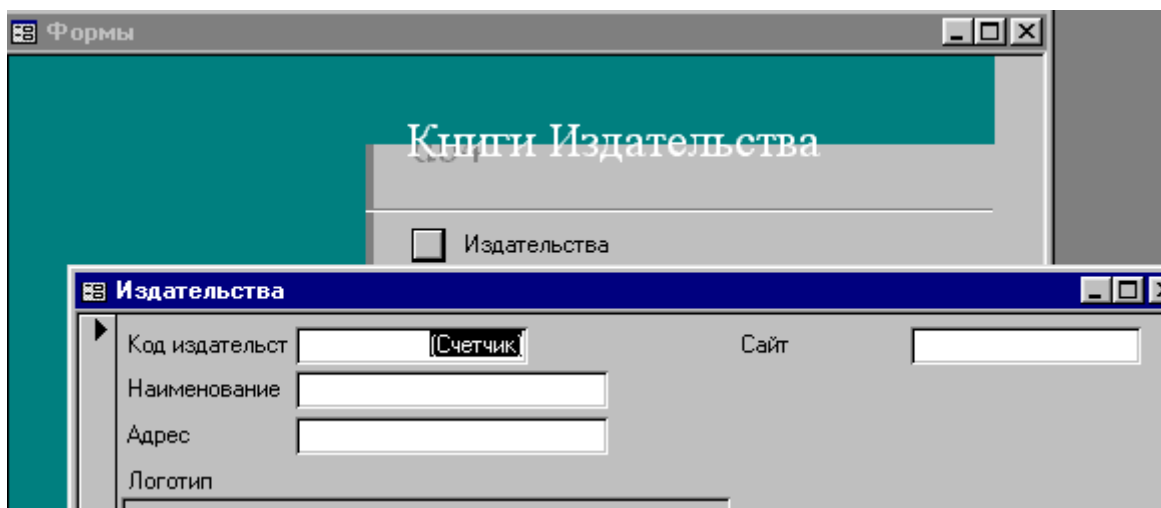


Эта же форма в режиме просмотра примет вид:



Таким образом, реализован наглядный однотипный интерфейс пользователя - из главной кнопочной формы вызываются 2-е подчиненные кнопочные формы, в этих подчиненных кнопочных формах реализовано открытие других форм и возврат в главную кнопочную форму.

Например, если щелкнуть по кнопке **Издательства**, то откроется форма **Издательства** в режиме добавления:



Следует помнить, что редактировать кнопочную форму нужно крайне осторожно. Можно сформировать надпись или рисунок вне области кнопок. Но в случае редактирования в области кнопок, кнопочная форма может быть разрушена.

Следует обратить внимание на то, что после создания кнопочной формы автоматически формируется вспомогательная таблица Элементы кнопочной формы. Эту таблицу нельзя удалять и редактировать.

Задание на лабораторную работу.

34. Спроектировать форму с помощью Мастера форм.
35. Открыть спроектированную форму в режиме просмотра и проанализировать результат проектирования.
36. Спроектировать Автоформу в столбец.
37. Открыть спроектированную форму в режиме просмотра и проанализировать результат проектирования.
38. Спроектировать Автоформу ленточную.
39. Открыть спроектированную форму в режиме просмотра и проанализировать результат проектирования
40. Спроектировать Автоформу табличную.
41. Открыть спроектированную форму в режиме просмотра и проанализировать результат проектирования.
42. Спроектировать Автоформу сводная таблица.
43. Открыть спроектированную форму в режиме просмотра и проанализировать результат проектирования.
44. Спроектировать Автоформу сводная диаграмма.

45. Открыть спроектированную форму в режиме просмотра и проанализировать результат проектирования.
46. Спроектировать Автоформу диаграмма.
47. Открыть спроектированную форму в режиме просмотра и проанализировать результат проектирования.
48. Спроектировать сводную таблицу.
49. Открыть спроектированную форму в режиме просмотра и проанализировать результат проектирования.
50. Реализовать связь Один ко Многим для 2-х таблиц.
51. Создать форму посредством инструмента Новый объект для таблицы, расположенной в связи со стороны Один.
52. Открыть спроектированную форму в режиме просмотра и проанализировать результат проектирования.
53. Разработать кнопочную форму посредством Диспетчера кнопочных форм.
54. Открыть спроектированную форму в режиме просмотра и проанализировать результат проектирования.

Проектирование отчетов в Access 2002 с помощью Конструктора.

Введение

Отчеты - это средство формирования твердых копий документов. Далеко не всегда удобно просматривать данные посредством форм или запросов, в особенности большие объемы данных. Кроме того, электронный документ юридически не является документом.

Отчеты базируются на основе таблиц или запросов. Для формирования отчетов могут использоваться данные из нескольких таблиц.

В отчете можно группировать данные по нескольким полям и, таким образом, выводить информацию в наглядной форме.

В отчетах можно выполнять вычисления с использованием в формулах данных, выводимых на печать. При этом вычисления могут выполняться для каждой записи, для каждой группы записей, для всего отчета.

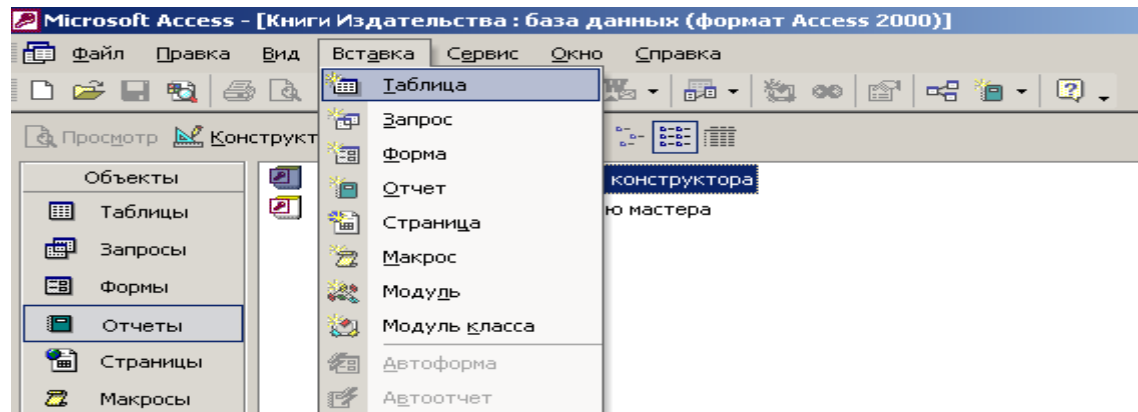
В отчетах можно строить различного рода диаграммы с целью наглядного отображения информации.

На базе информации, содержащейся в таблицах, с помощью средств проектирования отчетов, можно создавать почтовые наклейки.

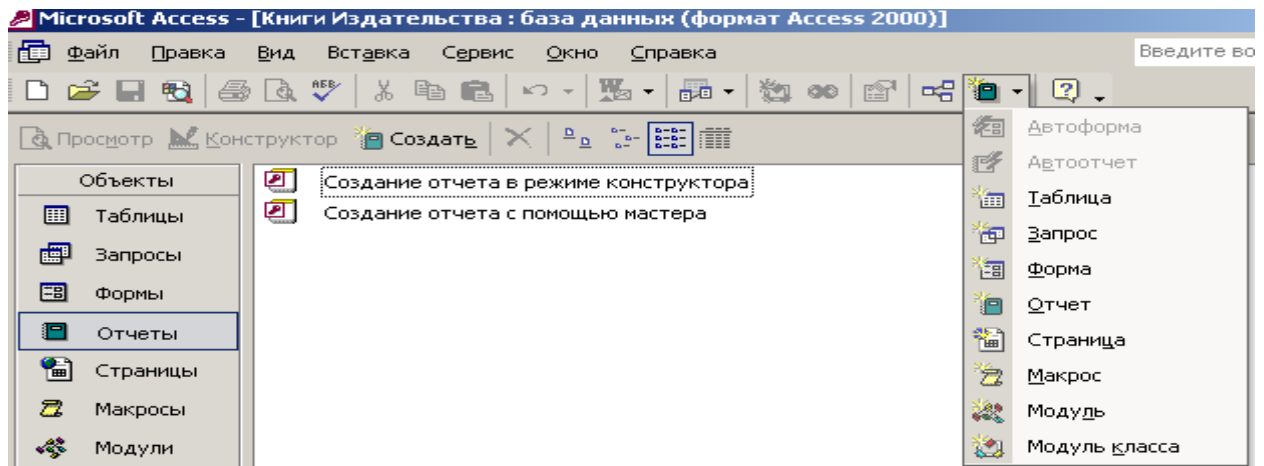
1. Запуск Конструктора отчетов

Запустить конструктор отчетов в Access 2002 можно 5-ю способами.

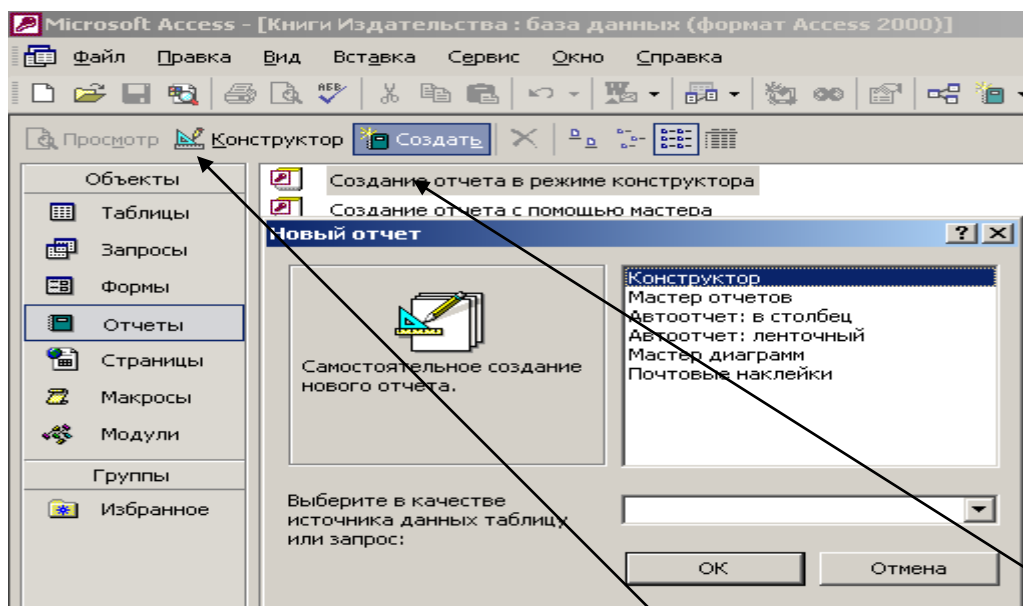
1. Можно использовать меню Вставка/Отчет:



2. Можно использовать инструмент Новый объект/Отчет:



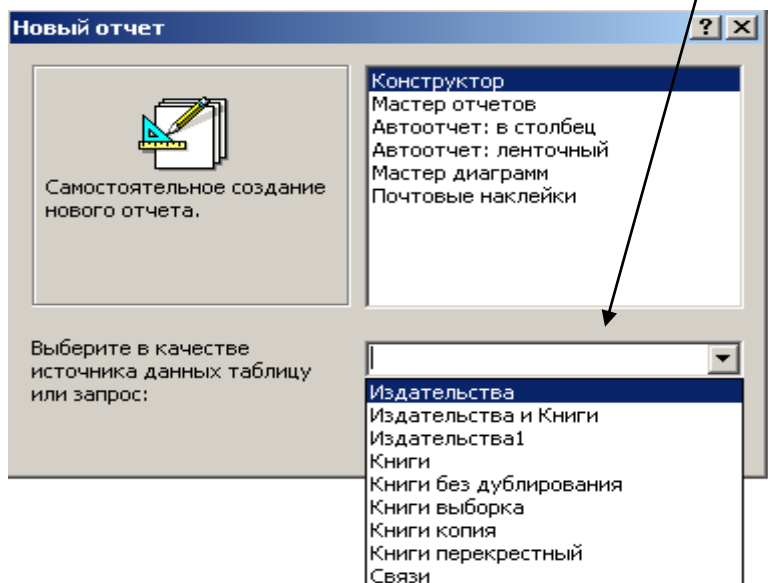
3. Можно использовать инструмент Создать/Конструктор:



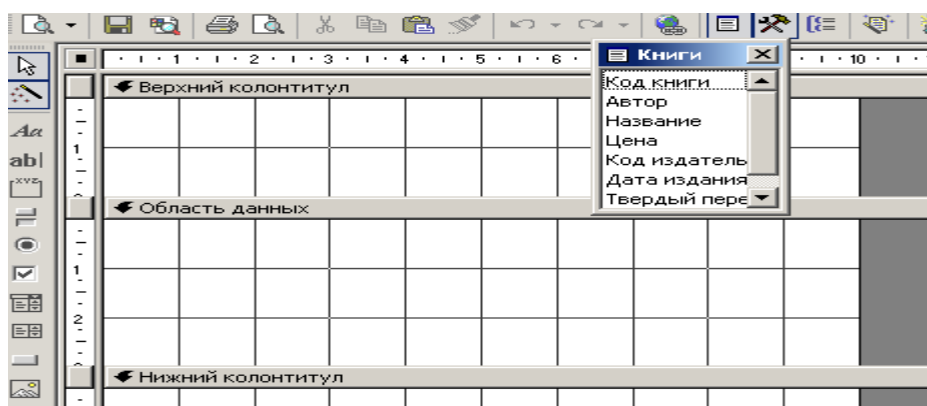
4. Можно использовать инструмент **Создание отчета в режиме конструктора**.


5. Можно использовать инструмент **Конструктор**, (если создается первый запрос).

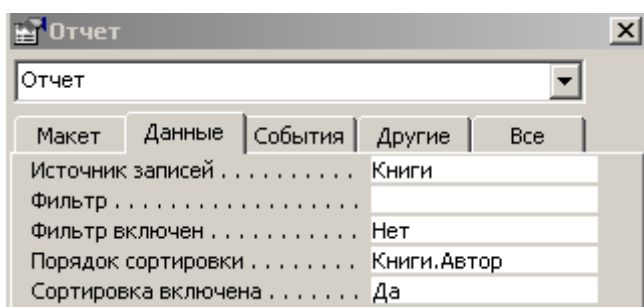
В последних 2-х случаях имя таблицы или запроса, который используется в качестве источника данных, не запрашивается. В первых 3-х случаях формируется окно **Новый отчет**, в котором с помощью поля со списком можно указать имя таблицы или запроса:



Использование любого из 5-и способов приводит к формированию пустого отчета:




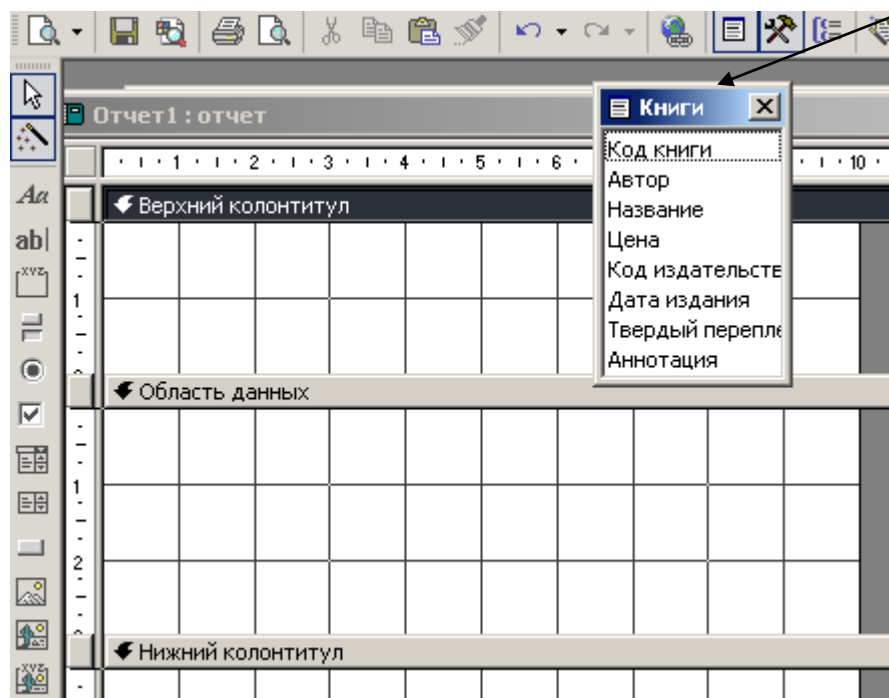
Если при запуске **Конструктора** имя таблицы или запроса, который используется в качестве источника данных не указано, то его можно указать с помощью свойств отчета. Для доступа к свойствам отчета необходимо дважды щелкнуть по маркеру выделения отчета , который расположен в левом верхнем углу. В результате отобразятся вкладки со свойствами отчета:



На вкладке **Данные** в строке **Источник записей** можно ввести или выбрать из списка таблицу или запрос, на котором предполагается проектировать отчет.

3. Формирование полей с данными в отчете

Если поля объекта, на базе которого строится отчет, не отображены, то необходимо щелкнуть по инструменту  (список полей) на панели инструментов **Конструктор отчетов**. После этого сформируется список полей, которые можно расположить в отчете.

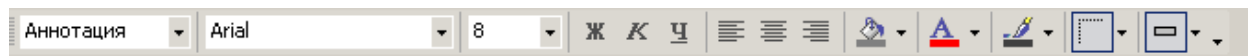


Как правило, поля из списка располагаются в **Области данных**. Для этого необходимо щелкнуть по нужному полю в списке и перетащить его в **Область данных**. Таким образом, в **Области данных** размещено несколько полей:





После перетаскивания поля автоматически формируется и его подпись. В полях располагаются данные, подписи используются как комментарии к полям с данными.

Как видно из рисунка, не все подписи, сформировались удачно, да и в некоторых полях данные могут не поместиться. Поля и подписи можно редактировать, изменять их размеры, перетаскивать. Для редактирования объектов отчета, в том числе и рассматриваемых, используется стандартная панель инструментов **Форматирование**:



После редактирования отчет в режиме конструктора может принять следующий вид:

Верхний колонтитул	
Область данных	
Автор:	Автор
Название:	Название
Цена:	Цена
Дата издания:	Дата издания
	<input checked="" type="checkbox"/> Твердый переплет
Аннотация:	Аннотация

Чтобы посмотреть, как будет выглядеть отчет в режиме просмотра, можно щелкнуть по инструменту  (вид) на панели инструментов **Конструктор отчетов**. Чтобы вернуться в режим конструктора, необходимо щелкнуть по инструменту  (вид) на панели инструментов **Конструктор отчетов**. В режиме просмотра фрагмент отчета примет вид:

Автор:	Грин
Название:	Алые паруса
Цена:	88,00р.
Дата издания:	
	<input type="checkbox"/> Твердый переплет
Аннотация:	

Автор:	Грин
Название:	Алые паруса
Цена:	82,00р.
Дата издания:	01.01.2004
	<input checked="" type="checkbox"/> Твердый переплет
Аннотация:	

Как видно из рисунка отчет не является компактным. Область листа используется нерационально, подписи повторяются. Отчет такого рода называется отчетом в столбец. Иногда он бывает полезен.

В данном случае оправданно поля с данными расположить в одну строку, а заголовки полей расположить также в одну строку в области **Верхнего колонтитула**. Для этого необходимо расширить область отчета, переместить поля, вырезать заголовки и разместить их в области Верхнего колонтитула. В результате названных манипуляций будет построен отчет вида:

Верхний колонтитул					
Автор:	Название:	Цена:	Дата издания:	Т.п.	Аннотация:
Область данных					
Автор	Название	Цена	Дата издания	<input checked="" type="checkbox"/>	Аннотация
Нижний колонтитул					

В режиме просмотра фрагмент отчета теперь примет вид:


Автор:	Название:	Цена:	Дата издания:	Т.п	Аннотация:
Грин	Алые паруса	88,00р.		<input type="checkbox"/>	Классика мировой литературы
Грин	Алые паруса	82,00р.	01.01.2004	<input checked="" type="checkbox"/>	
Данте	Божественная комедия	71,00р.	02.02.2002	<input type="checkbox"/>	

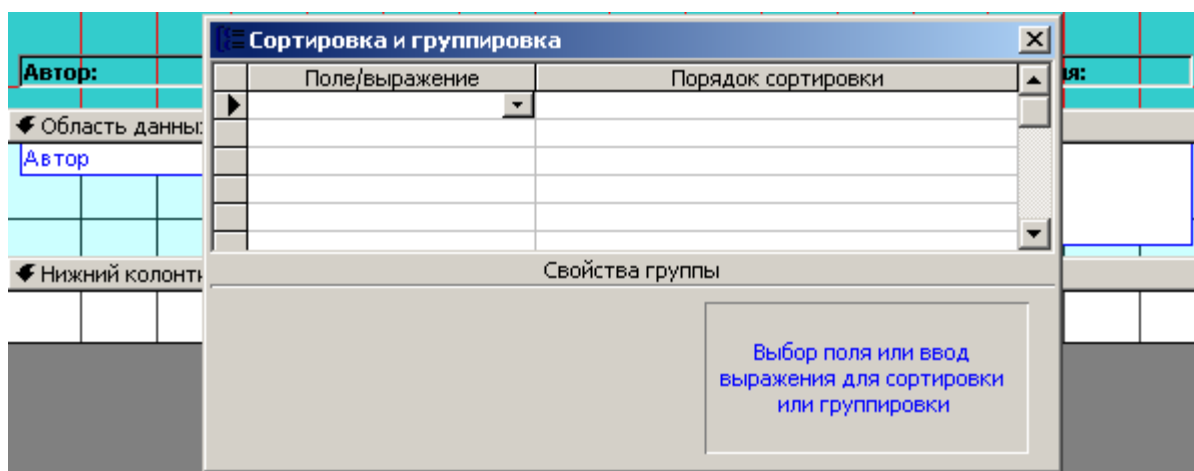
Отчет такого вида называется ленточным отчетом. Этот отчет получился более наглядным, компактным.

Чтобы сохранить спроектированный отчет можно выбрать меню **Файл/Сохранить**. Можно также сохранить отчет, щелкнув по кнопке закрытия окна, и подтвердить сохранение.

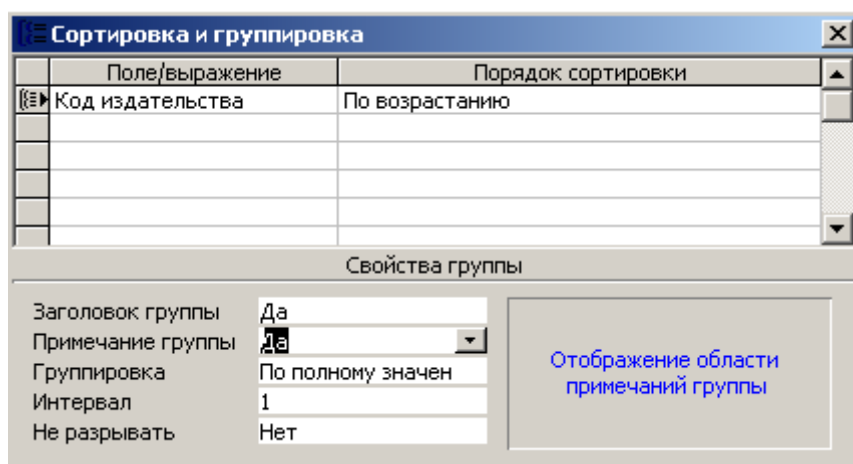
4. Проектирование отчетов с группировкой по полям

Группировка позволяет располагать поля группами, которые определяются значениями полей. Например, можно сгруппировать поля по **Коду издательства** и книги одного издательства будут расположены в одной группе. Можно также сгруппировать поля по авторам. Группировка в некоторых отчетах позволяет формировать печатные документы на качественно новом уровне.

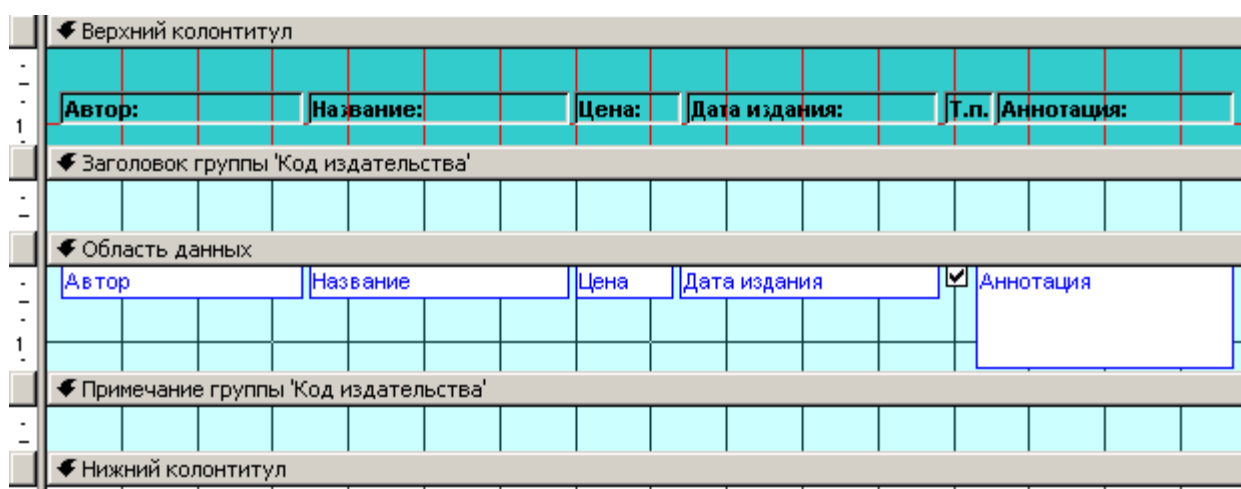
Для реализации группировки необходимо щелкнуть по инструменту  (**группировка**) на панели инструментов **Конструктор отчетов**. В результате сформируется бланк:



Для группировки по **Коду издательства** этот бланк необходимо заполнить следующим образом:



В результате отчет в режиме конструктора примет вид:



Как видно из рисунка, сформировались еще две области: **Область заголовка группы** и **Область Примечания группы**. В **Области заголовка группы** обычно располагают поле, по которому осуществляется группировка. В **Области примечания группы** обычно располагают суммирующие функции для полей группы. После добавления поля **Код издательства** в **Область заголовка группы** отчет в режиме просмотра примет вид:

Автор:	Название:	Цена:	Дата издания:	Т.п	Аннотация:
Код издательства: 1					
Грин	Алые паруса	82,00р.	01.01.2004	<input checked="" type="checkbox"/>	
Жюль Верн	Машина времени	95,00р.	01.01.2004	<input checked="" type="checkbox"/>	
Код издательства: 2					
Шекспир	Отелло	65,00р.	01.01.2003	<input checked="" type="checkbox"/>	Классика мировой литературы
Шиллер	Разбойники	125,00р.	01.01.2002	<input checked="" type="checkbox"/>	Классика мировой литературы
Код издательства: 3					
Грин	Алые паруса	88,00р.		<input type="checkbox"/>	Классика мировой

Информация о книгах сгруппирована по издательствам. **Код издательства** не является информативным, гораздо лучше, если бы на его месте располагалось **Наименование** издательства. Для реализации этого необходимо построить запрос на основе связанных таблиц **Издательства** и **Книги**. Из таблицы **Издательства** в запрос можно включить только **Наименование** издательства. Затем на базе этого запроса необходимо спроектировать отчет, а в качестве поля для группировки использовать поле **Наименование** издательства. Возможны и другие решения.

Группировать можно по нескольким полям. Например, в рассматриваемом случае еще полезно выполнить группировку по полю **Автор**. Действия, которые необходимо для этого выполнить, аналогичны действиям рассмотренным выше.


5. Использование панели элементов в Конструкторе отчетов


Панель элементов **Конструктора отчетов** обычно располагается в левой части окна Конструктора, но ее, как и другую панель, можно расположить в произвольном месте окна. Панель элементов имеет вид:

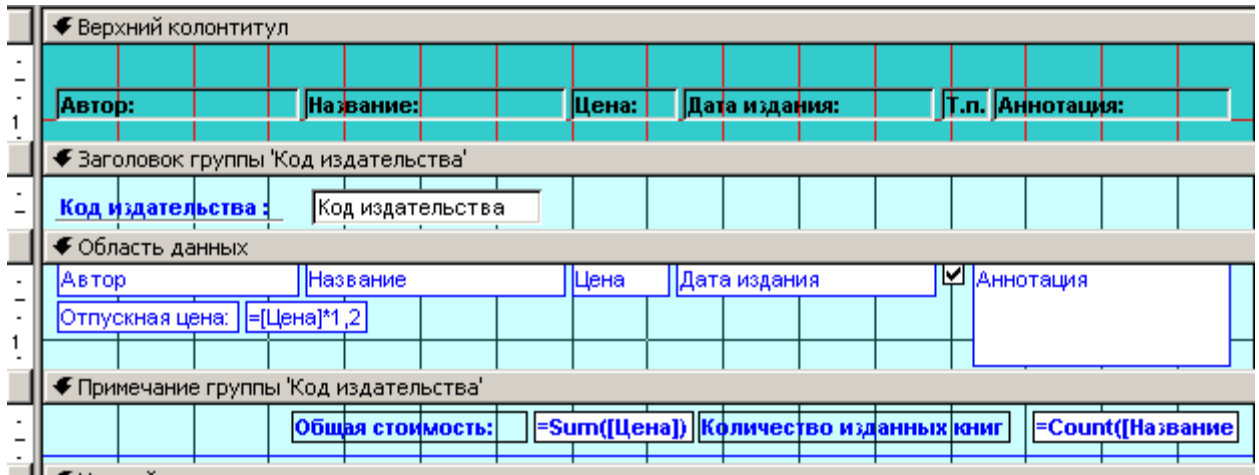


Нетрудно заметить, что она ничем не отличается от панели элементов **Конструктора форм**. Однако использование значительной части элементов в отчете не имеет смысла. В

отчете не нужны группа переключателей, выключатели, переключатели, флажки, поля со списком, списки, кнопки, вкладки.


С помощью элемента  (**Надпись**) в отчете можно ввести текст в любой из его областей. Например, можно ввести заголовок отчета, заголовки групп.


С помощью элемента  (**Поле**) в отчете можно сформировать поле в любой из его областей. В этом поле можно ввести выражение.





Верхний колонтитул											
Автор:		Название:			Цена:		Дата издания:		Т.п.	Аннотация:	
Заголовок группы 'Код издательства'											
Код издательства :		Код издательства									
Область данных											
Автор		Название			Цена		Дата издания		<input checked="" type="checkbox"/>	Аннотация	
Отпускная цена:		=[Цена]*1,2									
Примечание группы 'Код издательства'											
Общая стоимость:				=Sum([Цена])			Количество изданных книг		=Count([Название])		


В **Области данных** сформировано поле с формулой для расчета отпускной цены для каждой книги. В области **Примечание группы** сформировано поле с формулой для расчета общей стоимости книг издательства и поле с формулой для расчета общего количества книг, изданных в издательстве.

С помощью элемента  (**Линия**) в отчете проведена линия, разделяющая группы полей.

С помощью элемента  (**Прямоугольник**) в отчете при необходимости можно нарисовать прямоугольник.

С помощью элемента  (**Рисунок**) в отчет при необходимости можно вставить рисунок.

С помощью элемента  (**Свободная рамка объекта**) в отчет при необходимости можно вставить OLE объект.

С помощью элемента  (**Подчиненная форма**) можно вставить в отчет подчиненную форму или отчет. О них речь пойдет ниже.

Спроектированный отчет в режиме просмотра выглядит следующим образом:

Автор:	Название:	Цена:	Дата издания:	Т.п	Аннотация:
Код издательства: 1					
Грин	Алые паруса	82,00р.	01.01.2004	<input checked="" type="checkbox"/>	
Отпускная цена	98,40р.				
Жюль Верн	Машина времени	95,00р.	01.01.2004	<input checked="" type="checkbox"/>	
Отпускная цена	114,00р.				
Общая стоимость:		177,00р.	Количество изданных книг		2
Код издательства: 2					
Шекспир	Отелло	65,00р.	01.01.2003	<input checked="" type="checkbox"/>	Классика мировой литературы
Отпускная цена	78,00р.				
Шиллер	Разбойники	125,00р.	01.01.2002	<input checked="" type="checkbox"/>	Классика мировой литературы
Отпускная цена	150,00р.				
Общая стоимость:		190,00р.	Количество изданных книг		2

Вычисляемые поля после их формирования имели числовой формат. Для изменения формата на денежный следует обратиться к свойствам полей и выбрать нужный формат. Свойства областей отчета и свойства его объектов в основном совпадают соответственно со свойствами областей форм и свойствами ее объектов.

6. Внедрение в отчет подчиненной формы или отчета

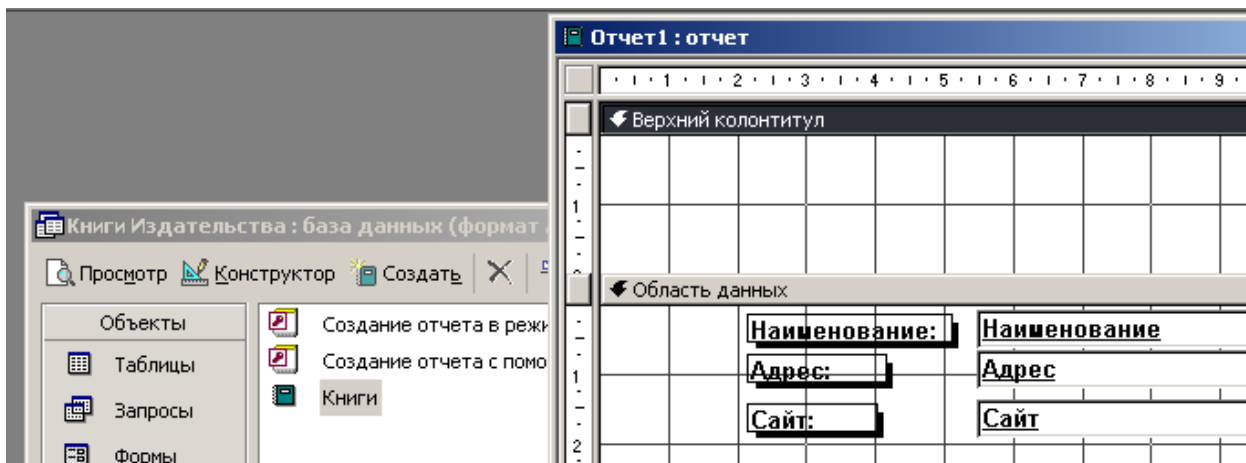
Также как у таблицы может быть подчиненная таблица, так и у отчета может быть подчиненный отчет. При этом главный отчет строится на базе главной таблицы, а подчиненный отчет строится на базе подчиненной таблицы. Такого рода отчеты очень напоминают отчеты с группировкой, но иногда являются предпочтительней.

Отчет, в который будет внедряться подчиненный, в данном случае строится на базе таблицы **Издательства**. Этот отчет в режиме конструктора имеет вид:

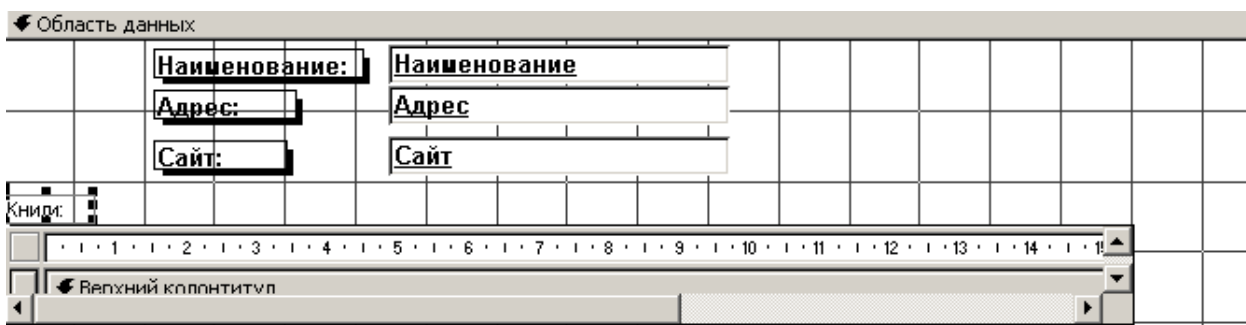
← Область данных	
Наименование:	Наименование
Адрес:	Адрес
Сайт:	Сайт

Отчет, который построен на базе подчиненной таблицы **Книги**, уже сформирован и называется **Книги**. Теперь для внедрения этого отчета в отчет, представленный на рисунке, достаточно перетащить его пиктограмму из базового окна таблицы в область данных

отчета. Для технической реализации этого действия необходимо уменьшить область окна **Конструктор отчета** и переместить окно так, чтобы была видна пиктограмма отчета **Книги**:



После перетаскивания пиктограммы отчета **Книги** в область данных создаваемого отчета окно конструктора примет вид:



В режиме просмотра фрагмент спроектированного отчета выглядит следующим образом:

Наименование:	Мир
Адрес:	Бауманская ,2
Сайт:	бббб@ааа.ру

Книги

Код издательства :	1		
Жюль Верн	Машина времени	95,00р.	01.01.2004 <input checked="" type="checkbox"/>
Отпускная цена:	114,00р.		
Грин	Алые паруса	82,00р.	01.01.2004 <input checked="" type="checkbox"/>
Отпускная цена:	98,40р.		
Общая стоимость:	177,00р.	Количество изданных книг	2

Наименование:	Прометей
Адрес:	Подбельского,7
Сайт:	

Книги

Код издательства :	2		
Шекспир	Отелло	65,00р.	01.01.2003 <input checked="" type="checkbox"/> Классика мировой литературы
Отпускная цена:	75,00р.		

Этот отчет можно отредактировать. Например, поле **Код издательства** в подчиненном отчете лишнее. Ему соответствует поле **Наименование** в главном отчете. Поэтому поле **Код издательства** следует удалить. Отчет станет более понятным и более компактным.

Так как отчеты и формы в Access по своей природе очень близки, то в отчет в качестве подчиненного отчета, можно включать форму. Из соображений наглядности лучше включать в качестве подчиненной формы ленточную форму. Манипуляции, связанные с включением в главный отчет подчиненной формы, аналогичны манипуляциям, рассмотренным выше. Фрагмент отчета с подчиненной формой в режиме просмотра выглядит следующим образом:

Наименование:	Мир
Адрес:	Бауманская ,2
Сайт:	бббб@aaa.ru

Книги

Автор	Название	Цена	Код издательства	Дата издания	Тв. переплет	Аннотация
Грин	Алые паруса	82,00р.	1	01.01.2004	<input checked="" type="checkbox"/>	
Жюль Верн	Машина времени	95,00р.	1	01.01.2004	<input checked="" type="checkbox"/>	
Суммарная цена:		177,00р.				

Наименование:	Прометей
Адрес:	Подбельского,7
Сайт:	

Книги

Автор	Название	Цена	Код издательства	Дата издания	Тв. переплет	Аннотация
Шиллер	Разбойники	125,00р.	2	01.01.2002	<input checked="" type="checkbox"/>	Классика ми...

В отчет, как и в форму, можно включить несколько подчиненных отчетов или форм.

Задание на лабораторную работу.

55. Выполнить анализ всех способов построения отчета с помощью Конструктора отчетов.
56. Сформировать с помощью Конструктора отчетов Отчет в столбец.
57. Выполнить форматирование сформированного отчета.
58. Вывести отчет в режиме просмотра.
59. Сформировать с помощью Конструктора отчетов Ленточный отчет.
60. Выполнить форматирование сформированного отчета.
61. Вывести отчет в режиме просмотра.
62. Выполнить проектирование отчета с группировкой по полям.
63. Выполнить форматирование сформированного отчета.
64. Вывести отчет в режиме просмотра.
65. Сформировать в отчете элементы: надписи, поля, рисунки, линии, прямоугольники.
66. Выполнить форматирование сформированного отчета.
67. Вывести отчет в режиме просмотра.
68. Внедрить в отчет подчиненный отчет.
69. Выполнить форматирование сформированного отчета.
70. Вывести отчет в режиме просмотра.

71. Внедрить в отчет подчиненную форму.
72. Выполнить форматирование сформированного отчета.
73. Вывести отчет в режиме просмотра.

Проектирование отчетов в Access 2002 с помощью Мастеров.

Введение

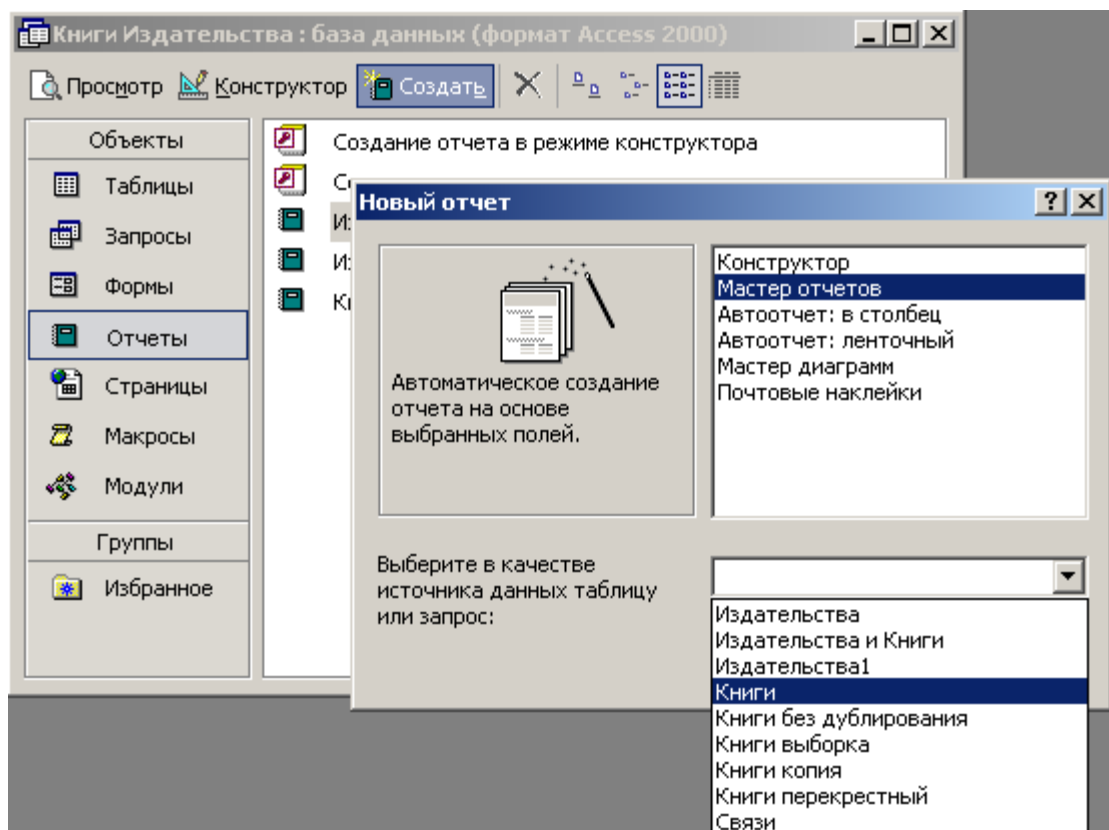
Access 2002 предоставляет разработчику БД широкие возможности для проектирования отчетов с помощью мастеров. Использование мастера, как правило, существенно упрощает процесс проектирования отчета.

При проектировании отчетов можно задействовать следующие мастера: **Мастер отчетов, Автоотчет в столбец, Автоотчет ленточный, Мастер диаграмм, Почтовые наклейки**. Эти отчеты можно построить и с помощью конструктора отчетов, однако, процесс проектирования отчета с помощью конструктора чаще всего более трудоемок.

Часто предпочтительно создать отчет с помощью мастера, а затем выполнить его доработку в режиме конструктора. В подавляющем большинстве случаев отчет, построенный посредством мастера, нуждается в модификации. В частности, поля в отчете могут не помещаться или наоборот занимать неоправданно много места. В диаграммах, построенных посредством мастера диаграмм, отдельные данные могут просто не отображаться.

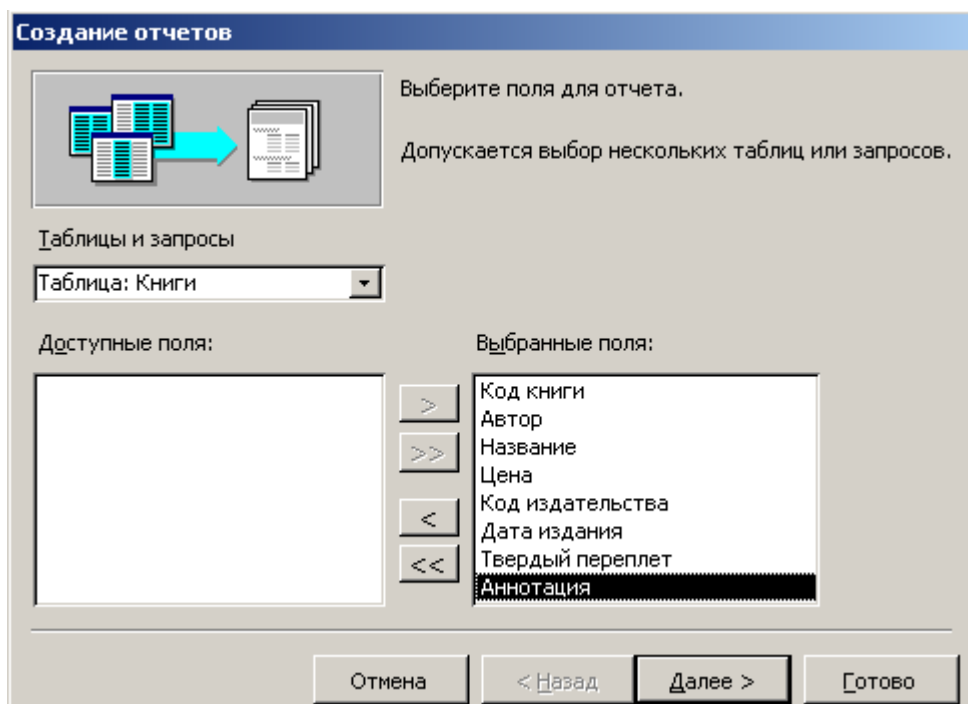
2. Мастер отчетов

Для доступа к мастерам отчетов необходимо в базовом окне Access обратиться к вкладке **Отчеты** и щелкнуть по кнопке **Создать**. В результате загрузится окно:



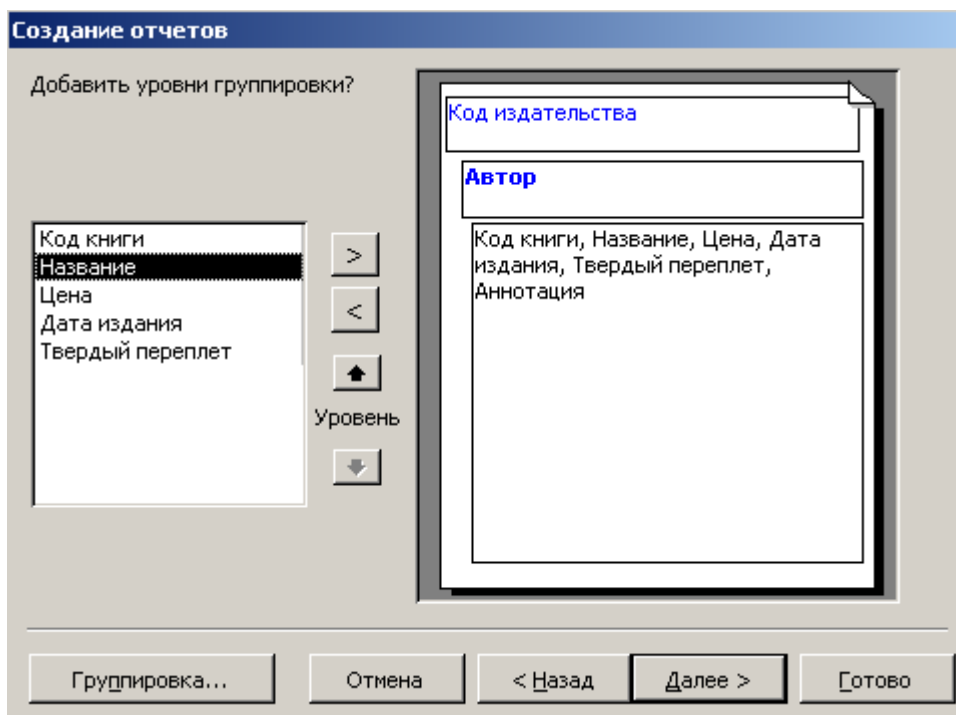
В этом окне надо выбрать нужный мастер из предлагаемого списка мастеров. Кроме того, в поле со списком надо выбрать таблицу или запрос, на базе которого предполагается строить отчет. В рассматриваемом случае в качестве мастера выбран **Мастер отчетов**, а в качестве базовой таблицы выбрана таблица **Книги**.

После щелчка по кнопке ОК сформируется следующее окно:







В данном окне необходимо выбрать список полей, которые необходимо включить в отчет. В рассматриваемом случае в отчет включаются все поля таблицы **Книги**.

Обращение к следующему шагу мастера приведет к загрузке окна:



В этом окне предлагается выбрать поля, по которым в отчете будет осуществляться группировка записей. Группировка предполагает размещение записей в отчете в соответствии с полями, по которым выполняется группировка. Например, если группировка осуществляется по полю **Код издательства**, то в отчете записи, относящиеся к конкретному издательству, будут размещаться в одной группе. Группировку можно осуществлять по нескольким полям, что и реализовано в данном случае. Здесь добавлена группировка записей по полю **Автор**. Таким образом, в рамках групп по полю **Издательства** будут формироваться группы по полю **Автор**.

Для добавления уровня группировки необходимо в левой части окна выделить нужное поле и щелкнуть по кнопке . Для исключения группировки необходимо щелкнуть по группирующему полю в правой части окна и щелкнуть по кнопке .

Уровень группировки можно понизить или повысить и таким образом назначить группу, в рамках которой будут располагаться другие группы. Для изменения уровня группировки необходимо выделить в правой части окна группирующее поле и щелкнуть по кнопке  или по кнопке . После назначения уровней группировки и обращения к следующему шагу мастера загрузится окно:

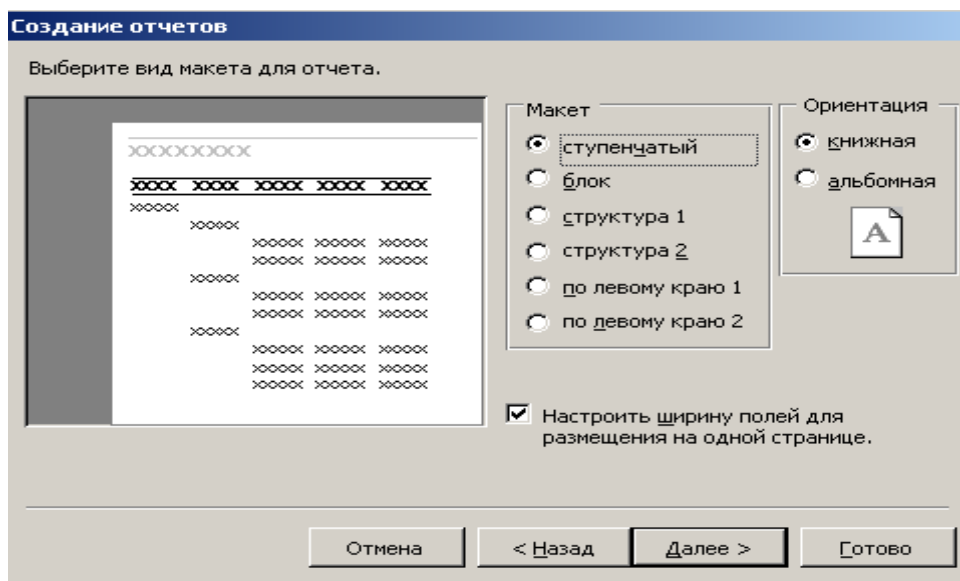
В данном окне выводится список полей, по которым в принципе могут быть подсчитаны итоги. Эти поля имеют числовой или логический тип. В рассматриваемом случае к таким полям относятся **Цена** и **Твердый переплет**.

Справа от этих полей расположены флажки, с помощью которых можно выбрать агрегирующие функции. Для каждого поля можно выбрать одну или более функций. В данном случае для поля **Цена** назначается подсчет суммарной цены. Суммарная цена будет подсчитываться для каждой группы и для отчета в целом.

Посредством радиокнопки **Только итоги** можно сформировать отчет, в котором не будут выводиться записи с данными, а будут выводиться только итоговые значения. Радиокнопка **Данные и итоги** обеспечивает вывод всех записей отчета и итоговых значений.

Флажок **Вычислять проценты** позволяет вычислять и выводить в отчет в данном случае процентное отношение цены конкретной группы к суммарной стоимости книг.

После нажатия кнопки **ОК** мастер вернется к предыдущему окну. После нажатия в этом окне кнопки **Далее** сформируется окно:

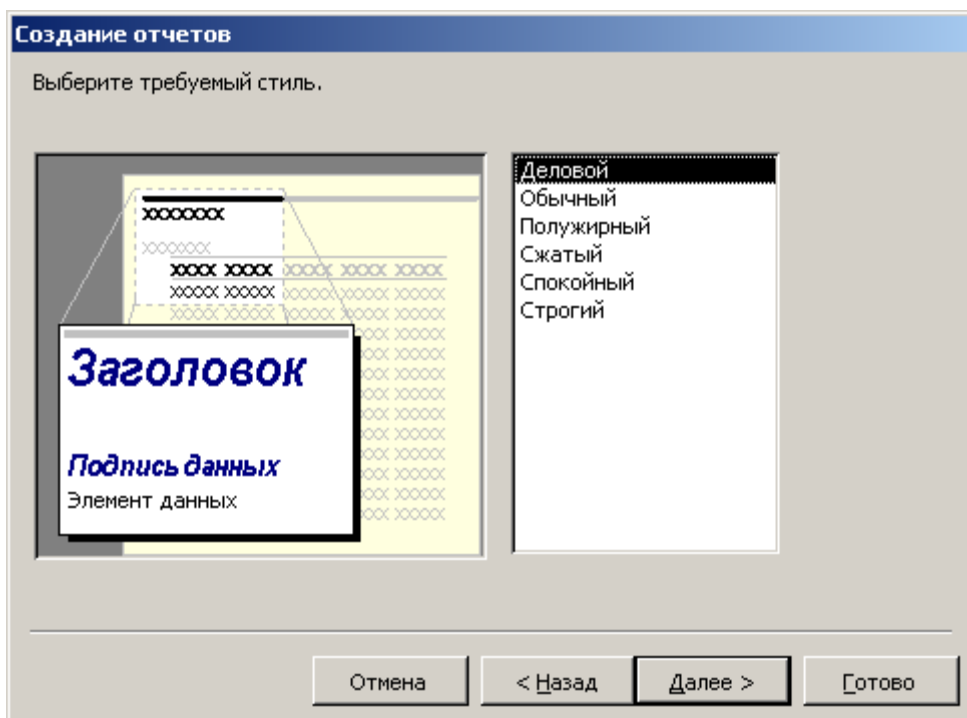


В данном окне можно выбрать макет отчета. При выборе макета в левой части окна отображается внешний вид отчета. При большом количестве полей и записей оправдано выбирать компактные отчеты: **структура, по левому краю**.

В случае большого числа полей следует выбирать **альбомную ориентацию**.

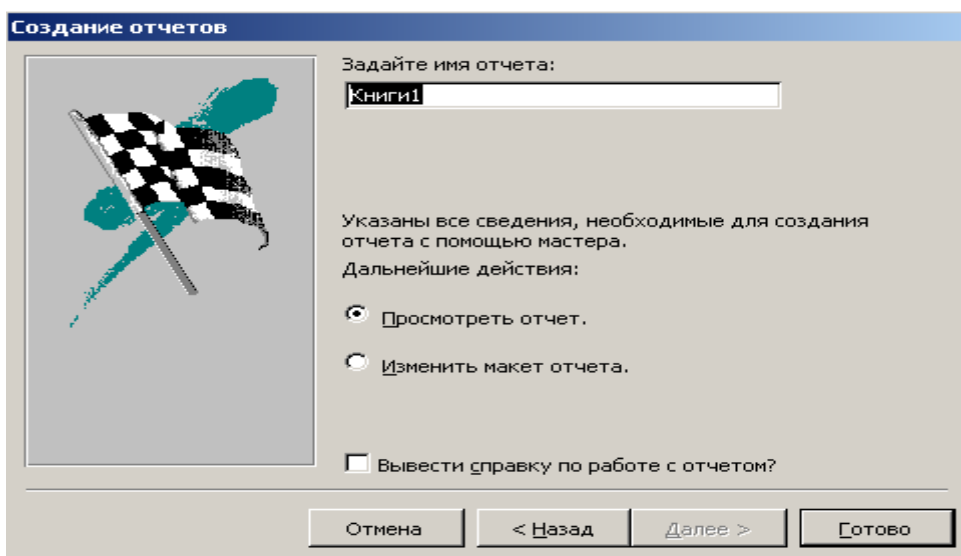
Флажок **настроить ширину полей для размещения на одной странице** хоть и обеспечивает данную функцию, но часто приводит к необходимости тщательного редактирования отчета в режиме конструктора.

На следующем шаге мастера загрузится окно:



В данном окне из предлагаемого списка стилей можно выбрать требуемый стиль. Стиль выбирается в зависимости от количества полей и записей, располагаемых в отчете, а также исходя из назначения отчета.

На следующем шаге мастера загрузится окно:



В этом окне можно ввести имя отчета, а затем вывести его в режиме конструктора или просмотреть. В режиме просмотра спроектированный отчет выглядит следующим образом:

Книги

Код издательства	Автор	Название
	1	
	Грин	Алые паруса
	Итого для 'Автор' = Грин (1 запись)	
	Sum	
	Жюль Верн	Машина времени
	Итого для 'Автор' = Жюль Верн (1 запись)	
	Sum	
Итого для 'Код издательства' = 1 (2 записей)		
Sum		
	2	
	Шекспир	Отелло
	Итого для 'Автор' = Шекспир (1 запись)	
	Sum	
	Шиллер	Разбойники
	Итого для 'Автор' = Шиллер (1 запись)	

Вероятно, данный отчет следует отредактировать в режиме конструктора. Код издательства, по которому осуществляется группировка, неинформативен. Поэтому отчет лучше было бы формировать на базе запроса, в котором включено поле **Наименование** издательства. Кроме того, в данном случае группировка по фамилиям авторов бессмысленна, так как таблица **Книги** практически не заполнена и каждый автор, в основном, представлен только одной книгой.

3. Автоотчет в столбец.

Для формирования **Автоотчета в столбец** необходимо в окне **Новый отчет** выбрать позицию **Автоотчет в столбец**. В поле **Выберите в качестве источника таблицу или запрос** необходимо указать имя объекта, на базе которого строится отчет.

В рассматриваемом случае в качестве источника выбрана таблица **Издательства**. После нажатия по кнопке **ОК** за один шаг сформируется отчет, который представлен на рисунке:

Издательств

Код издательства

Наименование

Адрес

Логотип



Сайт

Код издательства

Наименование

Адрес

Логотип



Сформированный отчет не очень нагляден и нуждается в редактировании. Кроме того, в отчете нерационально используется область вывода.

4. Автоотчет ленточный.

Ленточный отчет более рационально использует область вывода. Для формирования Автоотчета ленточный необходимо в окне **Новый отчет** выбрать позицию **Автоотчет ленточный**. В поле **Выберите в качестве источника таблицу или запрос** необходимо указать имя объекта, на базе которого строится отчет.

В рассматриваемом случае также в качестве источника выбрана таблица **Издательства**. После нажатия по кнопке **ОК** за один шаг сформируется отчет, который представлен на рисунке:

Издательство

<i>Код издательства</i>	<i>Наименование</i>	<i>Адрес</i>	<i>Логотип</i>
1	Мир	Бауманская,2	
2	Прометей	Подбельского,7	
3	Эксмо	11-я парковая,55	

В этом отчете более рационально используется область вывода. Однако в нем часть поля **Логотип** и все поле **Сайт** расположено на смежном листе. Запись с помощью конструктора вполне возможно разместить на одном листе. В частности, можно существенно уменьшить области вывода для полей **Код издательства**, **Наименования** и **Адреса**, а поля **Логотип** и **Сайт** можно сместить влево.

Как видно из результатов выполнения автоотчетов они далеко не всегда удовлетворяют потребностям пользователей БД. Их положительной стороной является то, что трудоемкость их формирования минимальна.

5. Мастер диаграмм.

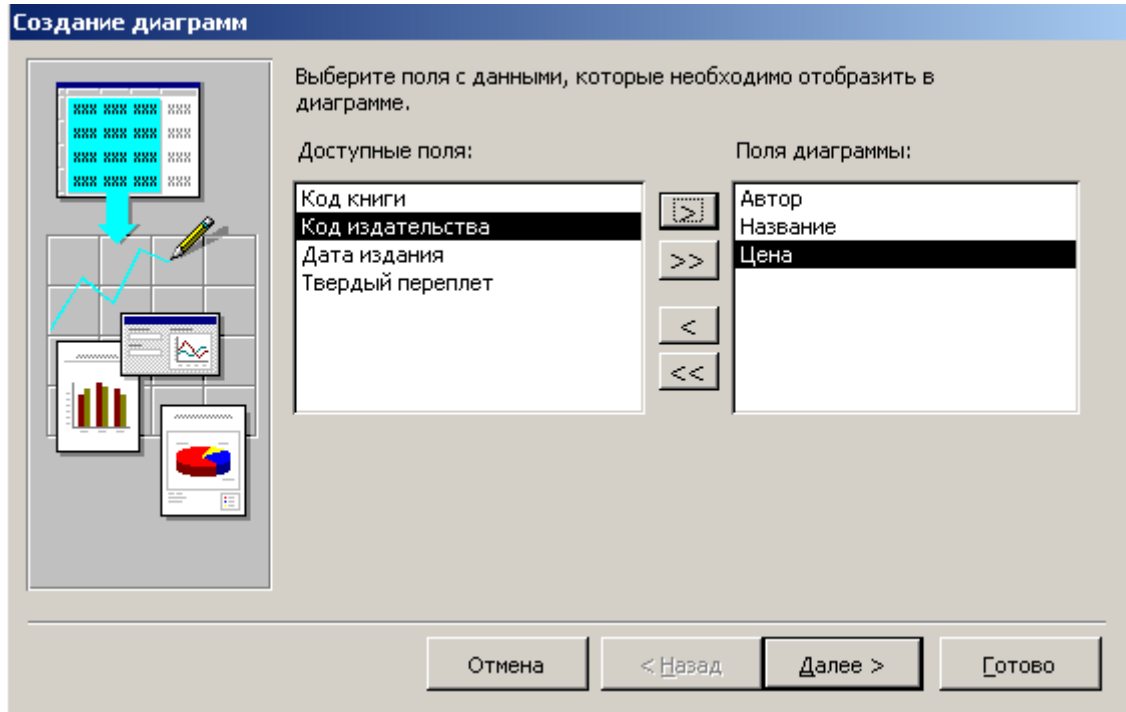
Использование диаграмм в отчетах существенно упрощает сравнительный анализ данных, упрощает анализ тенденций изменения данных в зависимости от времени или других факторов.

Access 2002 позволяет строить диаграммы 20-и типов. При этом в диаграмме может быть задействовано одно и более полей.

В связи с тем, что при построении диаграммы необходимо учесть несколько факторов, в мастере диаграмм задействовано несколько шагов.

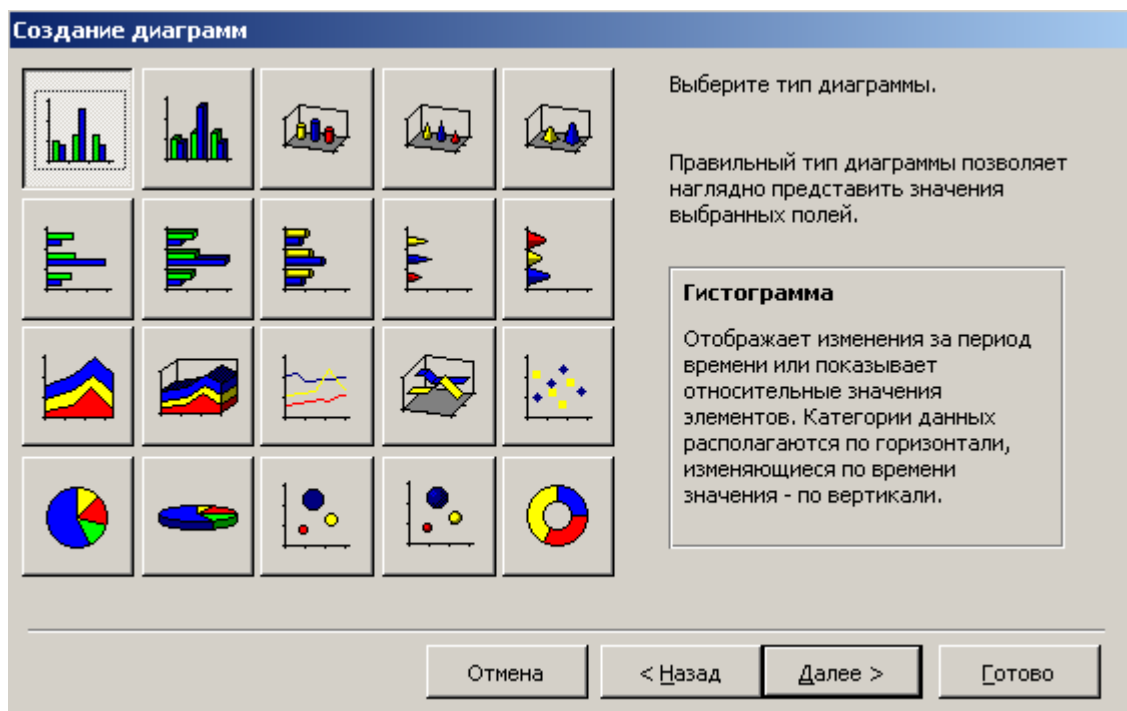
Для формирования диаграммы необходимо в окне **Новый отчет** выбрать позицию **Мастер диаграмм**. В поле **Выберите в качестве источника таблицу или запрос** необходимо указать имя объекта, на базе которого строится диаграмма.

В рассматриваемом случае также в качестве источника выбрана таблица **Книги**. После нажатия по кнопке **ОК** сформируется окно, которое представлено на рисунке:

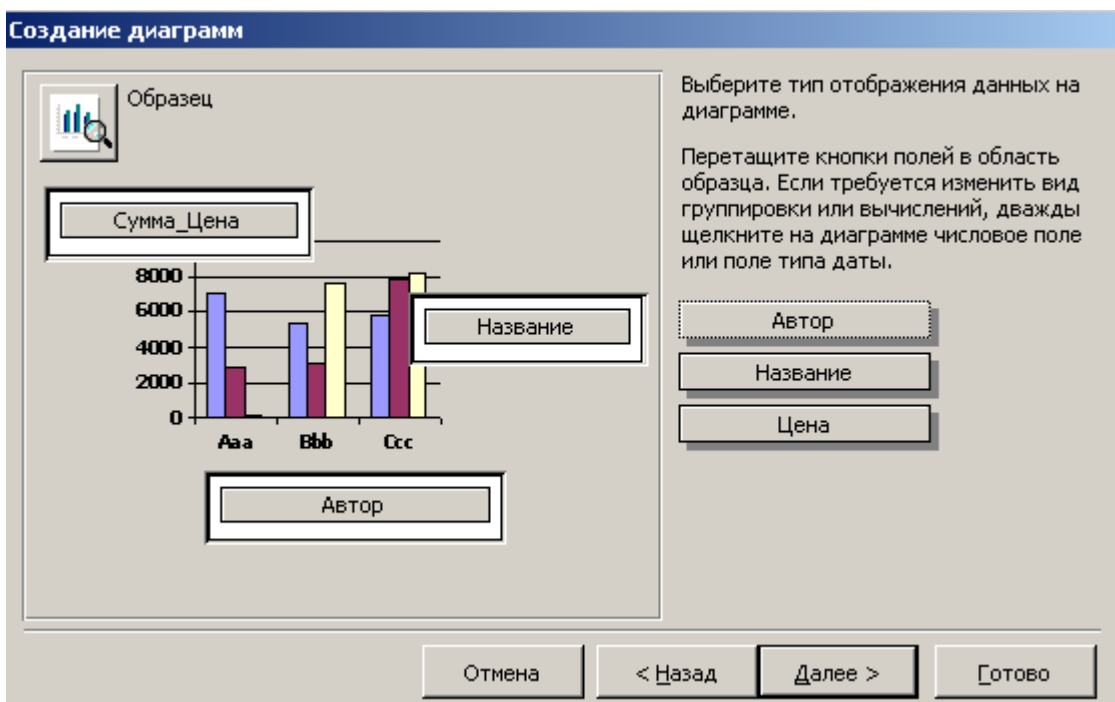


В данном окне выводится список полей объекта, на основе которого строится диаграмма. В нем выбираются поля диаграммы.

На следующем шаге мастера формируется окно вида:

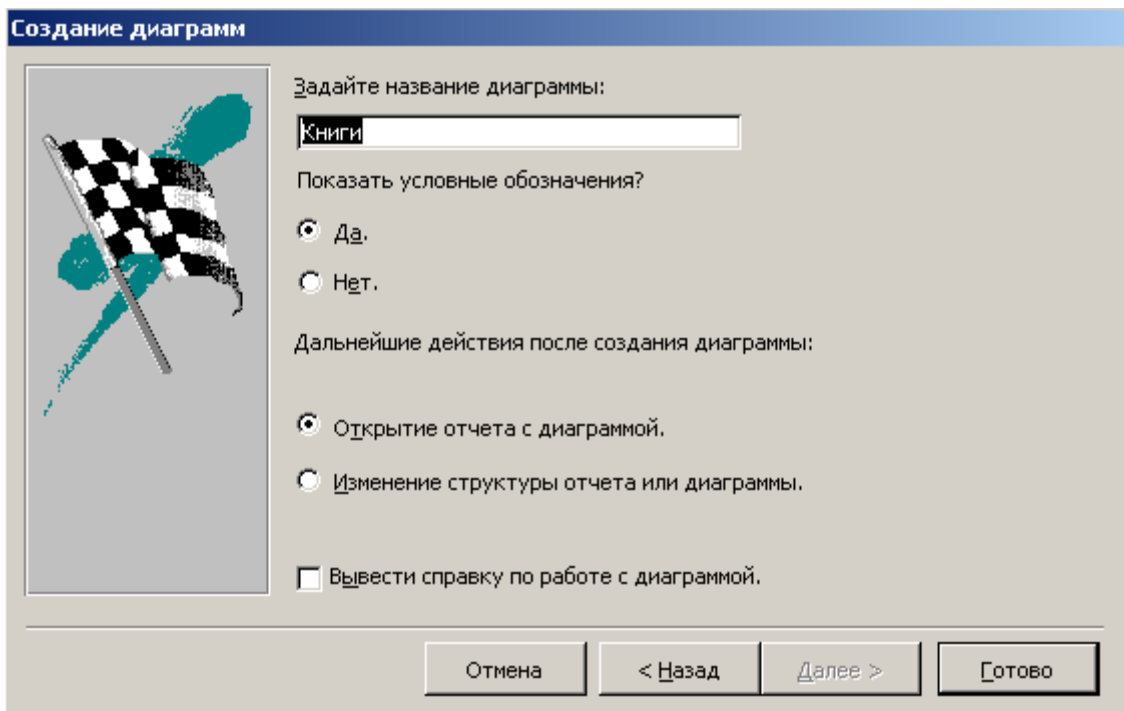


В данном окне следует выбрать тип диаграммы. Выбор типа диаграммы зависит от того, сколько полей и какие поля используются для построения диаграммы. Например, если это 3-и поля, одно из которых числовое, оправдано использование первых 10-и типов диаграмм. Для рассматриваемого примера выбран тип диаграммы **Гистограмма**. Вид окна, которое формируется на следующем шаге мастера, зависит от выбранного типа диаграммы. Для **Гистограммы** сформируется окно следующего вида:



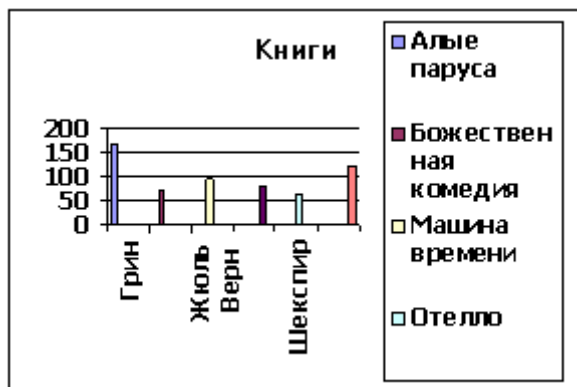
В этом окне необходимо поля разместить в области диаграммы в соответствии с тем, как их предполагается отображать. Не исключено, что поля в данном окне будут уже расположены приемлемым образом. Если это не так, то поля из левой части окна следует переместить в нужную область диаграммы.

На следующем шаге мастера сформируется окно:



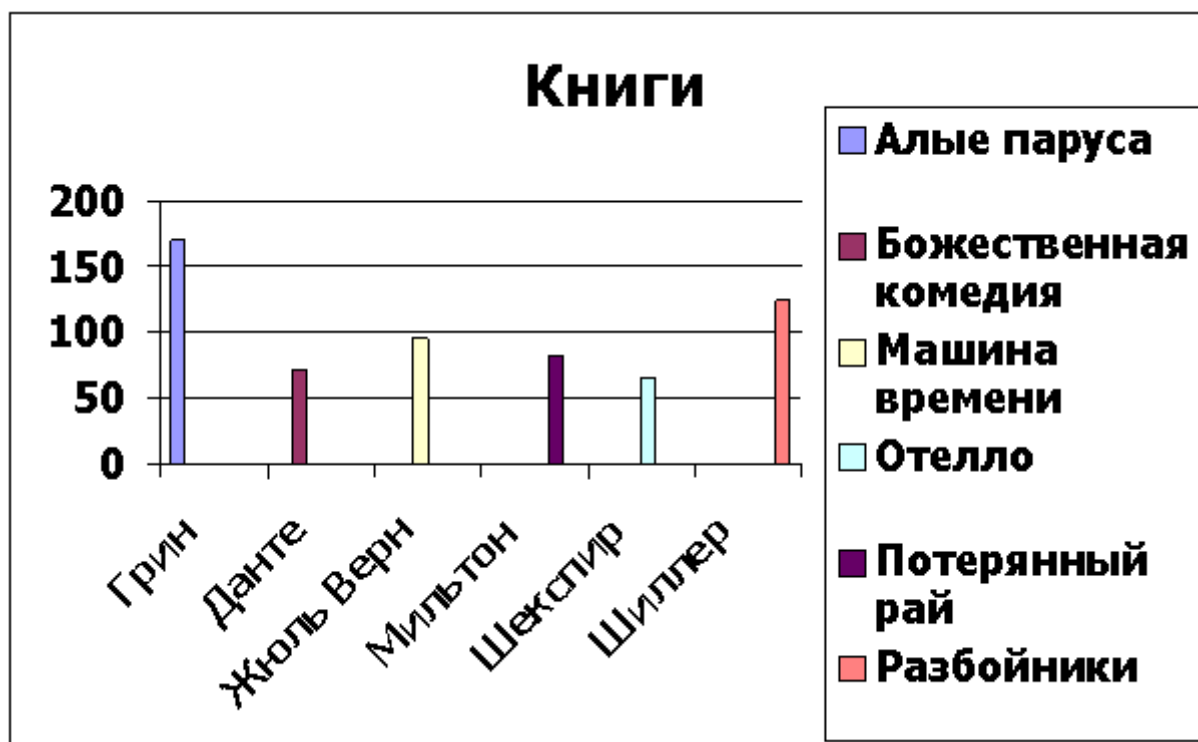
В нем можно предусмотреть вывод условных обозначений, а затем открыть диаграмму в режиме конструктора или открыть ее в режиме просмотра.

В режиме просмотра построенная диаграмма выглядит следующим образом:



Как видно из рисунка, на диаграмме поместились не все авторы, а также не все обозначения. При этом полностью не задействована вся область вывода.

Диаграмму следует сохранить, а затем загрузить в режиме конструктора. Затем увеличить область диаграммы. После этого диаграмма в режиме просмотра примет более приемлемый вид:

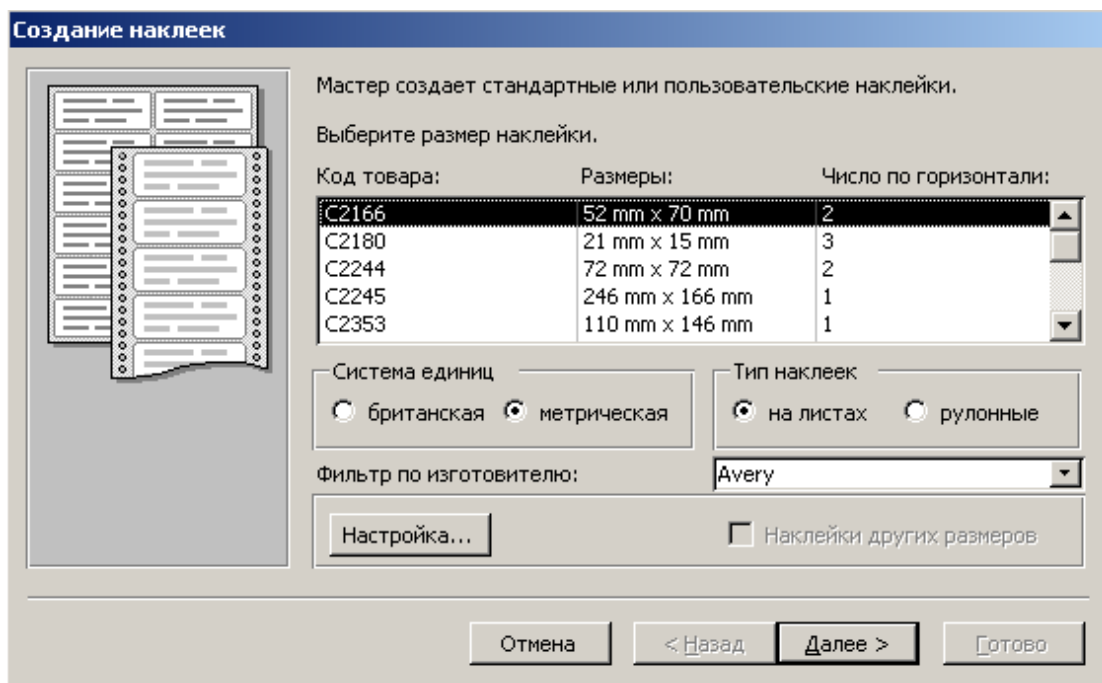


6. Мастер наклеек.

Access 2002 очень удобно использовать для формирования наклеек и визитных карточек. Для этого используется мастер наклеек. При удачном его использовании можно получить группу карточек, в которых будут расположены данные одной записи. Таким образом, например, можно сформировать картотеку книг или картотеку издательств.

Для формирования наклеек необходимо в окне **Новый отчет** выбрать позицию **Почтовые наклейки**. В поле **Выберите в качестве источника таблицу или запрос** необходимо указать имя объекта, на базе которого формируются наклейки.

В рассматриваемом случае в качестве источника выбрана таблица **Книги**. После нажатия по кнопке **ОК** сформируется окно, которое представлено на рисунке:



На следующем шаге мастера необходимо выбрать тип шрифта, цвет шрифта, его размер, насыщенность, начертание.

На очередном шаге работы мастера необходимо выбрать из предложенного списка полей объект, на базе которого формируются наклейки и список полей, включаемых в наклейки.

На следующем шаге мастера можно из предложенного списка полей объекта, на базе которого формируются наклейки, выбрать поля для сортировки.

В завершающем шаге мастера можно вывести отчет с наклейками для редактирования в режиме конструктора или вывести отчет с наклейками в режиме просмотра.

Результат проектирования наклеек может быть таким:

Гомер
Одиссея
2003-02-02

Гомер
Эллиада
2002-02-02

Грин
Алые паруса
2004-01-01

Данте
Божественная комедия
2002-02-02

Жюль Верн
Машина времени
2004-01-01

Мильтон
Потерянный рай
2002-02-02

В режиме конструктора отчет с наклейками можно отредактировать. В частности, можно нарисовать линии разреза.

Задание на лабораторную работу.

57. Спроектировать отчет с помощью Мастера отчетов.
58. Провести эксперименты с группировкой, подсчетом итоговых значений, многократно создавая отчет с помощью Мастера отчетов.
59. Провести эксперименты с различными макетами и различными стилями отчетов, многократно создавая отчет с помощью Мастера отчетов.
60. Просмотреть и отредактировать отчет в режиме конструктора.
61. Сформировать Автоотчет в столбец.
62. Просмотреть и отредактировать отчет в режиме конструктора.
63. Сформировать Автоотчет ленточный.
64. Просмотреть и отредактировать отчет в режиме конструктора.
65. Реализовать отчет с диаграммой.
66. Провести эксперименты с различными типами диаграмм, многократно создавая диаграммы с помощью Мастера диаграмм.
67. Просмотреть и отредактировать диаграмму в режиме конструктора.
68. Сформировать наклейки для разных таблиц.
69. Провести эксперименты с различными форматами наклеек, многократно создавая наклейки с помощью Мастера наклеек.
70. Просмотреть и отредактировать наклейки в режиме конструктора.

Использование макросов в Access 2002.

Введение

Макросы являются удобным средством выполнения команд в Access 2002. Макрос представляет собой одну или более макрокоманд, объединенных под одним именем. Каждой макрокоманде ставится в соответствие какое-либо действие. Например, открыть запрос, открыть форму, перейти к элементу управления, запустить приложение, вывести сообщение. Всего в Access 2002 задействовано более 50-и макрокоманд.

У макрокоманды могут быть аргументы. Число аргументов зависит от типа макрокоманды. Есть макрокоманды без аргументов. Посредством аргументов конкретизируется действие, которое выполняет макрокоманда. Например, в макрокоманде **Открыть таблицу** в качестве аргумента выступает имя таблицы. Некоторые аргументы имеют список возможных значений. Например, у макрокоманды **Выполнить команду** всего один аргумент, который имеет более 500 возможных значений.

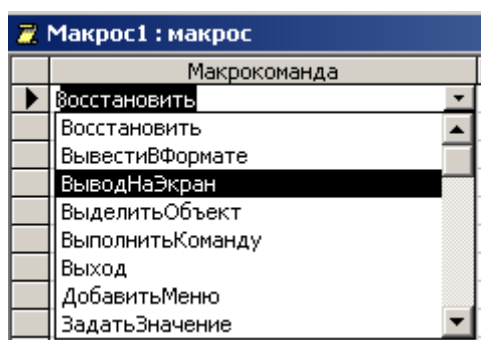
Для каждой макрокоманды может быть сформировано условие ее выполнения. В качестве аргументов условий можно задействовать значение полей форм и отчетов.

Макрос1 : макрос			
Имя макроса	Условие	Макрокоманда	Примечание

Имя макроса и условие используются не очень часто, поэтому данные области по умолчанию не выводятся. Чтобы их скрыть, используются соответственно меню **Вид/Имя макроса** и **Вид/Условия**.

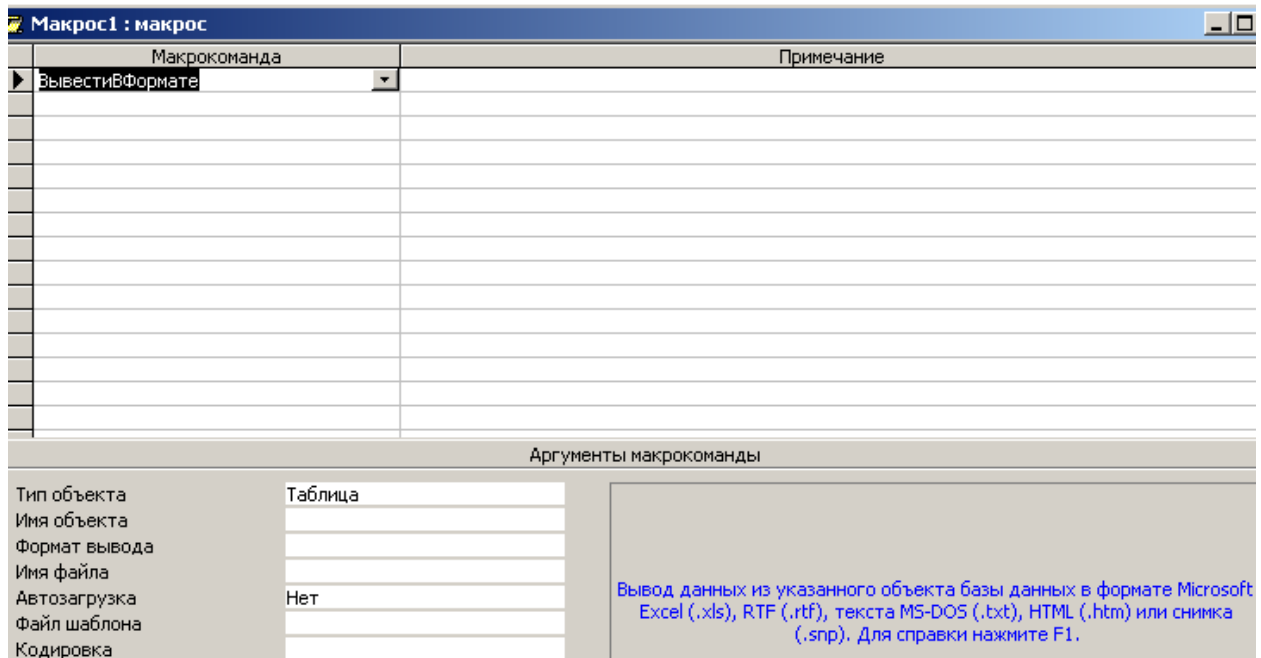
Чтобы сформировать макрокоманду, необходимо в области **Макрокоманда** в поле со списком выбрать нужную макрокоманду и в случае необходимости ввести аргументы этой макрокоманды.

Поле с раскрытым списком выглядит следующим образом:



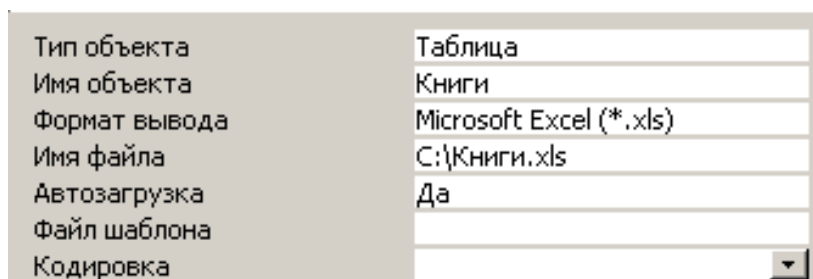
Здесь представлен список из 57 макрокоманд с аргументами и без аргументов. Например, макрокоманда **Восстановить** не имеет аргументов. Она служит для восстановления прежних размеров того окна, из которого макрос запускается и поэтому не нуждается в уточнениях.

Макрокоманда **ВывестиВФормате** имеет несколько аргументов, часть из которых необходимо обязательно заполнить. Ниже представлен вид окна макросов с выбранной макрокомандой **ВывестиВФормате** и соответствующим списком аргументов, который располагается в нижней части окна:



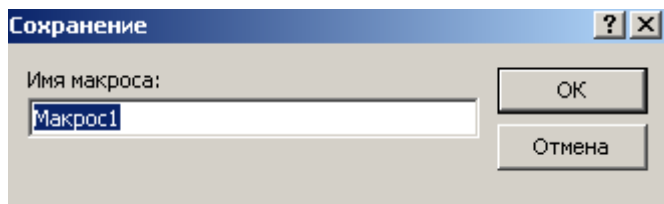
Как видно из рисунка, данная макрокоманда представлена списком из 7-и аргументов. В нижней правой части окна выводится краткая справка по выбранной макрокоманде или выбранному полю. Для получения более детальной справки, как и в любом приложении Microsoft Office, необходимо выбрать меню **Справка/Что это такое?** И щелкнуть по нужному объекту.

Рассматриваемая макрокоманда предназначена для вывода данных из объекта базы данных в другой формат вывода. В качестве объекта вывода могут быть задействованы практически все объекты Access (таблица, форма, отчет и другие). В качестве формата вывода могут использоваться форматы HTML (*.htm; *.html), текстовые файлы (*.txt), Microsoft Excel (*.xls), Rich Text Format (*.rtf) и другие. Ниже представлен заполненный список аргументов:

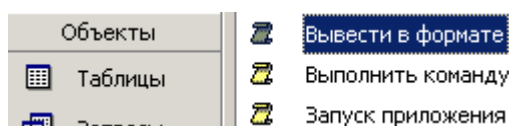


Посредством заполненных аргументов указано, что с помощью данной макрокоманды выводятся данные из таблицы **Книги** в формате Microsoft Excel. Имя создаваемого файла **Книги.xls**. Значение **Да** аргумента **Автозагрузка** указывает на то, что после запуска данного макроса будет загружен файл **Книги.xls**.

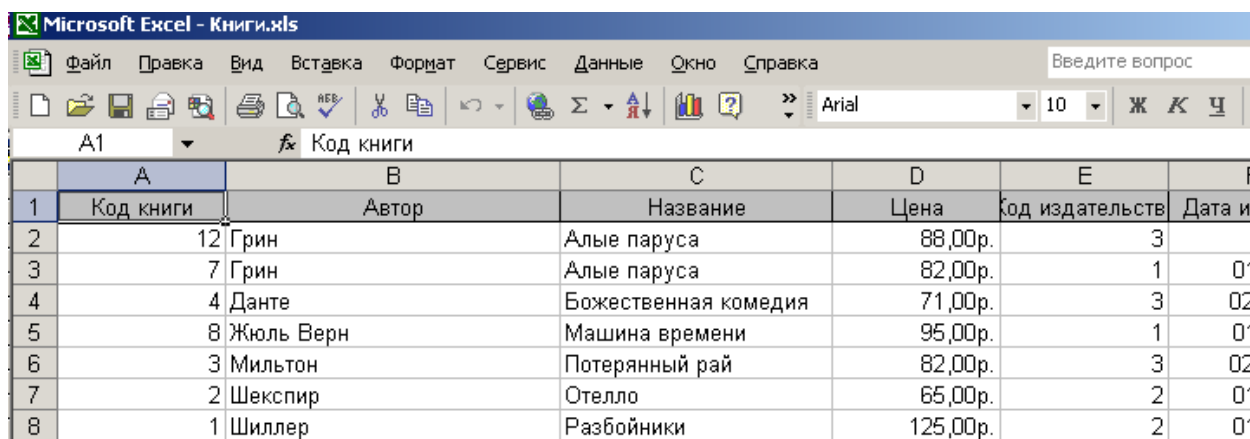
Для сохранения созданного макроса можно воспользоваться, как обычно, меню **Файл/Сохранить**. Можно также закрыть окно и подтвердить сохранение. При сохранении макроса будет запрошено его имя:



В рассматриваемом случае макрос сохранен с именем **Запуск приложения**. Значок данного макроса разместится на вкладке **Макросы** наряду с другими макросами:



Для выполнения макроса достаточно дважды щелкнуть по его значку или имени. Результат выполнения макроса **Запуск приложения**:



	A	B	C	D	E	F
1	Код книги	Автор	Название	Цена	Код издательств	Дата изд.
2	12	Грин	Алые паруса	88,00р.	3	
3	7	Грин	Алые паруса	82,00р.	1	01
4	4	Данте	Божественная комедия	71,00р.	3	02
5	8	Жюль Верн	Машина времени	95,00р.	1	01
6	3	Мильтон	Потерянный рай	82,00р.	3	02
7	2	Шекспир	Отелло	65,00р.	2	01
8	1	Шиллер	Разбойники	125,00р.	2	01

Как видно из рисунка, загрузился Microsoft Excel, а в нем в качестве таблицы представлена таблица базы данных. Файл **Книги.xls** будет храниться на диске вплоть до следующего вызова макроса и обновления.

Этот макрос, как и любой другой макрос, можно связать с каким-либо событием какого-либо объекта БД. Тогда он будет запускаться при наступлении этого события.

Рассмотренный макрос может оказаться очень полезным. В частности, полученную таблицу можно передать или переслать специалисту Excel для обработки в данной системе и использования ее возможностей.

3. Примеры макрокоманд.

Число макрокоманд довольно велико и рассмотреть их все не представляется возможным. Некоторые из макрокоманд используются исключительно редко. Назначение многих из них очевидно. В связи с этим вначале рассматриваются часто используемые простые макрокоманды, а затем часто используемые более сложные макрокоманды.

Макрокоманда **ВыводНаЭкран** позволяет отменить вывод информационных сообщений, сопровождающих выполнение некоторых макросов.

Макрокоманда **ВыделитьОбъект** позволяет сделать активным объект, который указан в аргументах макрокоманды. После этого имеется возможность выполнять макрокоманды для этого объекта.

Макрокоманда **Выход** позволяет завершить работу с Access.

Макрокоманда **Закреть** позволяет закрыть активный объект. Тип объекта задается в аргументах макрокоманды. Это может быть таблица, форма, отчет, запрос и другие объекты. Имя объекта также задается в аргументах макрокоманды.

Макрокоманда **ЗапускПрограммы** позволяет запустить программу. Ее имя задается в аргументах макрокоманды.

Макрокоманда **КЭлементуУправления** позволяет перейти к элементу управления в активной форме. Обычно это поле. Имя элемента управления вводится в качестве аргумента макрокоманды.

Макрокоманда **ПесочныеЧасы** позволяет вывести или отменить вывод изображения указателя мыши в виде песочных часов. Обычно часы выводятся до начала выполнения продолжительных действий БД и убираются по их завершению.

Макрокоманда **ПоказатьВсеЗаписи** позволяет отменить фильтр таблицы запроса или формы, если он использовался.

Макрокоманда **Развернуть** позволяет развернуть активное окно. Нередко это бывает полезно, т.к. бывает, что активное окно располагается на фоне окон открытых ранее.

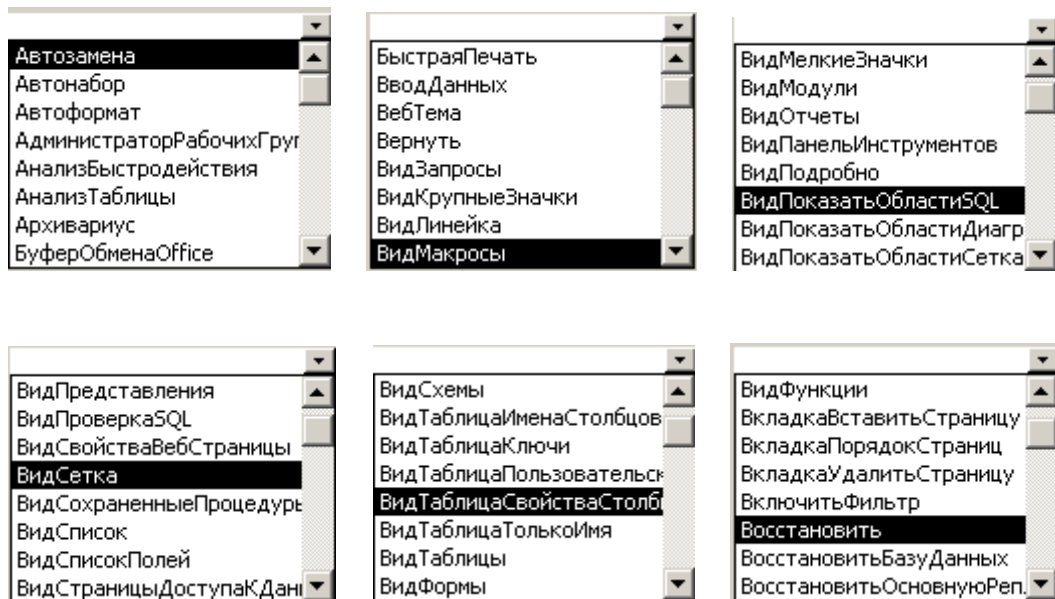
Макрокоманда **Сигнал** позволяет выдать звуковой сигнал. Обычно сигнал используется, чтобы обратить внимание пользователя на нестандартную ситуацию.

Макрокоманда **УстановитьСообщения** позволяет запретить или разрешить вывод режимных сообщений. Например, сообщений о том, что предполагается удаление из таблицы 23-х записей.

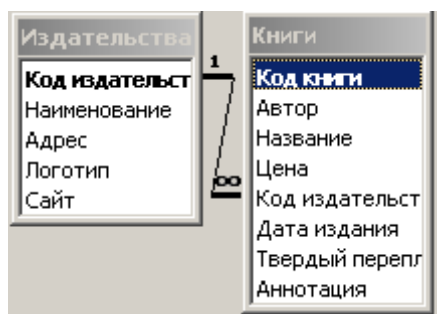
3.1. Макрокоманда **ВыполнитьКоманду**.

Данная макрокоманда позволяет выполнить одну из 500 команд. Эти команды, в основном, являются аналогами команд системы Access, которые задействованы в меню, подменю на панелях инструментов. Макрокоманда имеет всего один аргумент, который выбирается из предлагаемого списка. В качестве примера на рисунках представлена

примерно десятая часть команд, которые можно выполнить посредством макрокоманды **ВыполнитьКоманду**:



Данная макрокоманда является очень полезной и предоставляет пользователю в рамках своей БД использовать практически все возможности системы Access. Например, если использовать в качестве аргумента **СхемаДанных**, то после выполнения макрокоманды отобразится схема данных:



И эта команда может быть доступна пользователю БД. В принципе реальная БД должна быть освобождена от меню Access. Это делается из соображений невозможности доступа пользователя к модификации ее объектов, а также для избежания перегруженности окон приложения.

3.2. Макрокоманда ЗадатьЗначение.

Данная макрокоманда позволяет присвоить какому-либо элементу какое-либо значение. Элемент указывается с помощью аргумента **Элемент**, а значение указывается посредством аргумента **Выражение**. Например, в определенной ситуации может

возникнуть необходимость занесения в поле **Код издательства** формы **Книги** значение **Кода издательства** активной записи формы **Издательства**. Это необходимо для связывания записей. Тогда элементу **[Forms]![Книги]![Код издательства]** необходимо присвоить значение **[Forms]![Издательства]![Код издательства]**. Здесь использована нотация, принятая в Access. Ключевое слово **[Forms]** определяет то, что элемент принадлежит классу **Формы**. Далее следует имя формы и имя поля, разделенные знаком **“!”**. Все элементы конструкции необходимо окружать квадратными скобками.

В окне макрокоманды ее аргументы выглядят следующим образом:

Элемент	[Forms]![Книги]![Код издате
Выражение	[Forms]![Издательства]![Ко,

В качестве выражения может быть задействовано любое математическое выражение.

3.3. Макрокоманда **ЗапускМакроса**.

Данная макрокоманда используется для циклического запуска макроса.

Ее аргументы выглядят следующим образом:

Имя макроса	Вывести в формате
Число повторов	11
Условие повтора	[Forms]![Книги]![Цена]>70

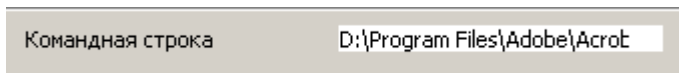
С помощью аргумента **Имя макроса** задается имя макроса, который необходимо выполнять в цикле.

Аргумент **Число повторов** задает максимальное число повторов выполнения макроса.

С помощью аргумента **Условие повтора** задается условие завершения циклического вызова макроса. Цикл будет прекращен, если условие примет значение **“Ложь”**.

3.4. Макрокоманда **ЗапускПриложения**

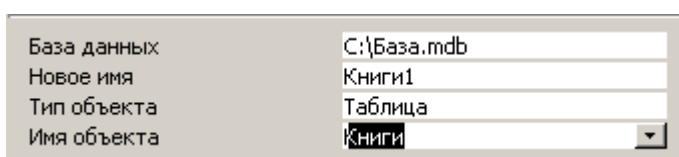
Данная макрокоманда позволяет запустить приложение Windows. Она имеет единственный аргумент, в котором необходимо ввести командную строку для запуска приложения. Например, если необходимо запустить из БД приложение Acrobat 5.0, то надо в качестве аргумента макрокоманды ввести командную строку вида **D:\Program Files\Adobe\Acrobat 5.0\Reader\AcroRd32.exe**. В окне макрокоманды это будет выглядеть следующим образом:



3.5. Макрокоманда КопироватьОбъект.

Данная макрокоманда позволяет копировать любой объект из одной БД в другую БД или в ту же БД, но с другим именем. Эта макрокоманда может быть полезна для создания архивных копий таблиц, для использования в новой БД объектов существующей БД.

Заполненный список аргументов данной макрокоманды выглядит следующим образом:



Здесь аргумент **База данных** позволяет задать полное имя БД, в которую будет копироваться объект. Эта БД должна существовать. Если копирование осуществляется в текущую БД, то аргумент **База данных** не заполняется.

Аргумент **Новое имя** задает новое имя объекта. Если имя объекта не меняется, данный аргумент не заполняется.

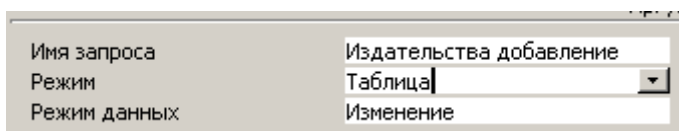
Аргумент **Тип объекта** позволяет задать тип копируемого объекта.

Аргумент **Имя объекта** позволяет задать имя копируемого объекта.

Создавать и использовать эту макрокоманду имеет смысл, когда часто выполняются однотипные копирования и копирование осуществляется пользователем. Гораздо проще открыть две БД и перетащить объект из одной БД в другую. В рамках одной БД можно скопировать объект, например, с помощью контекстного меню и вставить его с другим именем.

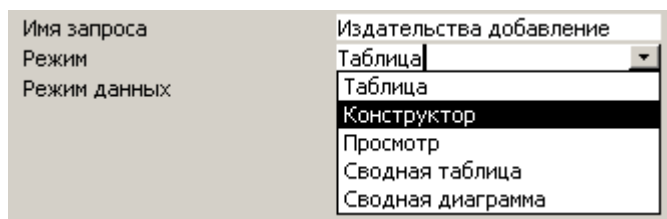
3.6. Макрокоманда ОткрытьЗапрос.

Данная макрокоманда позволяет открыть запрос в режиме просмотра. Ее аргументы выглядят следующим образом:

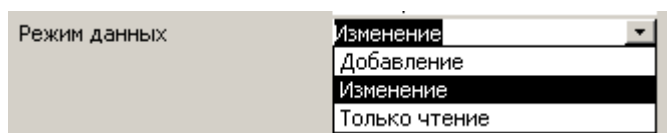


С помощью аргумента **Имя запроса** задается имя запроса, который предполагается открыть посредством макрокоманды.

Аргумент **Режим** позволяет выбрать режим, в котором будет открыт запрос. Список возможных режимов следующий:



Аргумент **Режим данных** позволяет указать, в каком режиме открывается запрос. Список возможных режимов следующий:

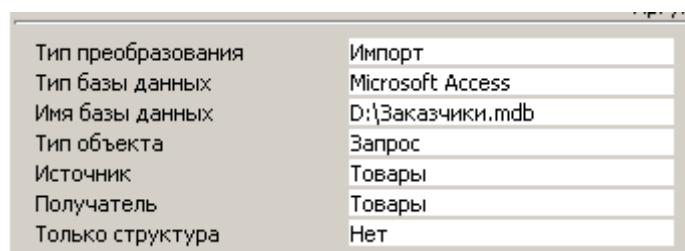


Существует несколько макрокоманд, подобных рассмотренной макрокоманде. Например, возможно использование макрокоманд **Открыть таблицу**, **Открыть форму**, **Открыть отчет**.

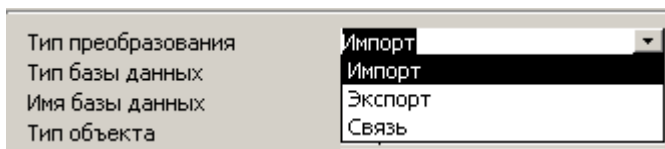
3.7. Макрокоманда ПреобразоватьБазуДанных.

Данная макрокоманда позволяет реализовать импорт или экспорт объектов БД. Кроме того, она позволяет реализовать связь с таблицами внешней БД. Со связанными таблицами можно работать, как и с любыми другими таблицами текущей БД.

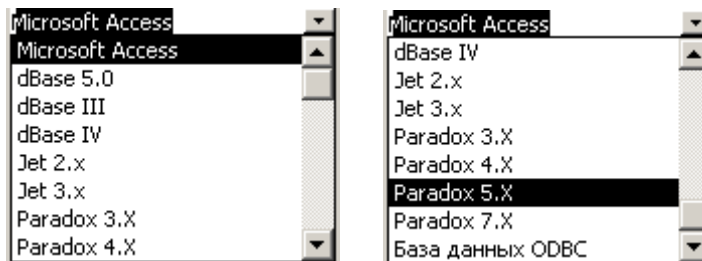
Список аргументов данной макрокоманды выглядит следующим образом:



Аргумент **Тип преобразования** может принимать следующие значения:

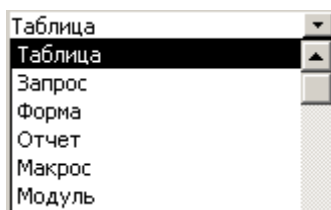


Аргумент **Тип базы** данных может принимать следующие значения:



С помощью аргумента **Имя базы** данных задается имя БД, из которой импортируется объект в активную БД. С помощью этого же аргумента можно задать имя БД, в которую экспортируется объект активной БД. Этот же аргумент позволяет задать имя БД, с которой связывается таблица текущей БД. Смысл этого аргумента определяется аргументом **Тип преобразования**.

Аргумент **Тип объекта** может принимать следующие значения:



Аргумент **Источник** позволяет указать имя объекта внешней БД.

Аргумент **Получатель** позволяет указать имя объекта активной БД.

Аргумент **Только структура** позволяет указать, что импортируется или экспортируется только структура объекта или структура вместе с данными.

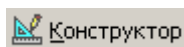
4. Связывание макросов с событиями.

Макрос можно запустить из базового окна Access, щелкнув по нему два раза. Кроме того, для выполнения макроса можно использовать меню **Сервис/Макрос/Выполнить** макрос. Эти способы запуска макросов удобно использовать при разработке БД и отладки макросов. Для пользователей БД такие средства неприемлемы.

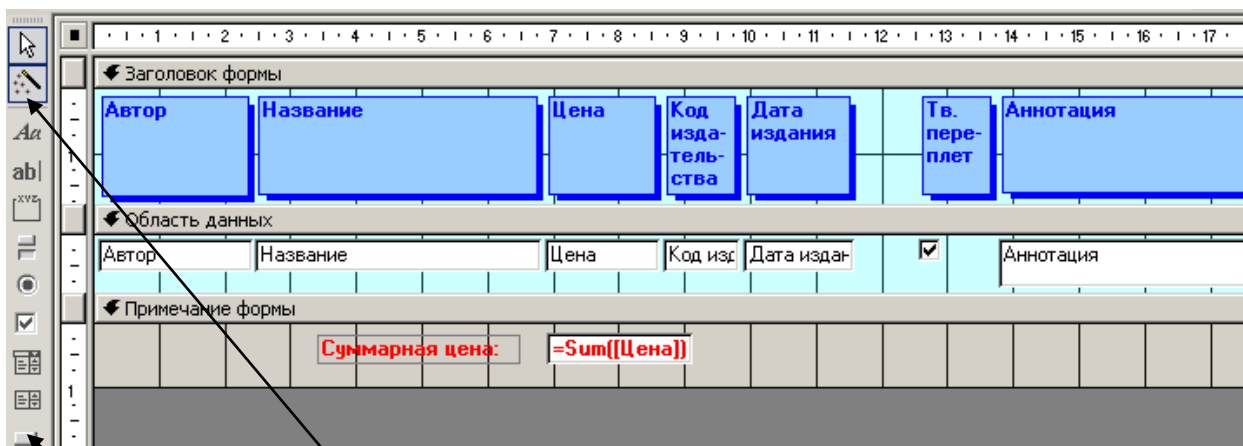
Для запуска макросов из интегрированной среды пользователя используется связь макросов с событиями. Например, макрос может быть выполнен при наступлении события **Нажатие кнопки**. Далее рассматривается этот пример.

Предполагается, что в форме **Книги ленточная** будет сформирована кнопка, нажатие которой позволит загрузить таблицу **Книги** в формате Excel.

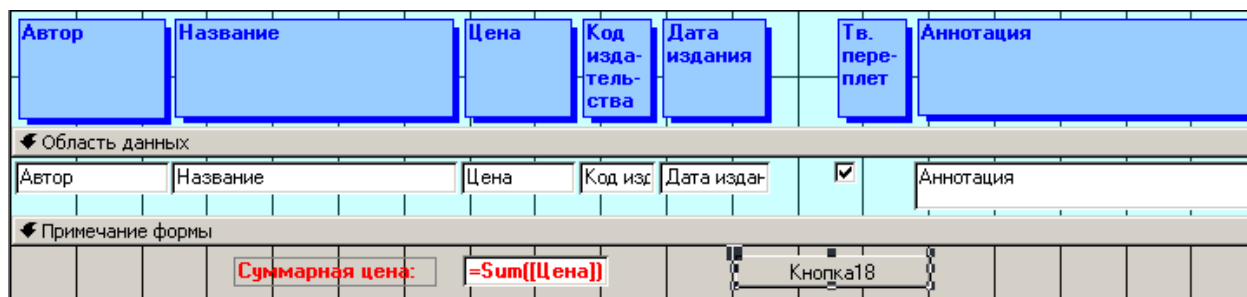
Для реализации этого, необходимо открыть форму **Книги ленточная** в режиме конструктора. Для этого необходимо в базовом окне Access выделить значок, соответствующий форме **Книги ленточная** и щелкнуть по инструменту **Конструктор**



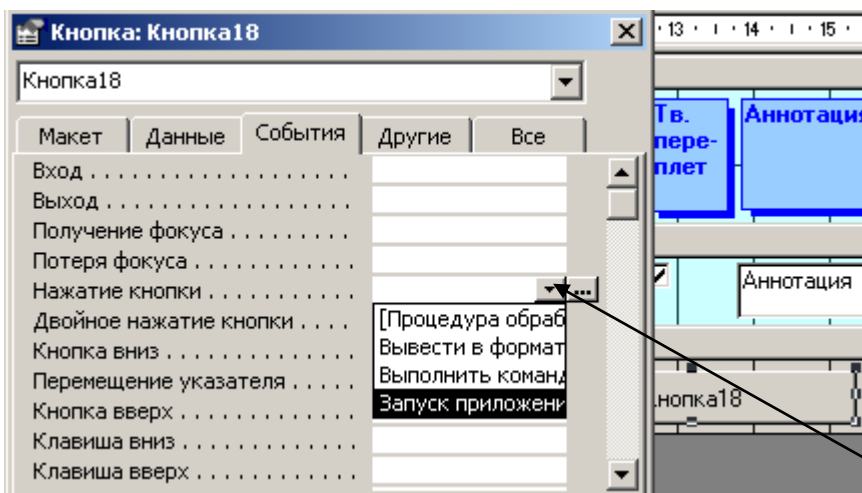
. В результате этих действий загрузится окно:




В этом окне необходимо отключить **Мастера**, для этого необходимо щелкнуть по соответствующему элементу. **Мастера** отключаются для того, чтобы при формировании кнопки не предлагалось за ней закрепить какие-либо функции. Затем надо щелкнуть по элементу **Кнопка** и расположить ее в нужном месте формы. В результате форма примет следующий вид:

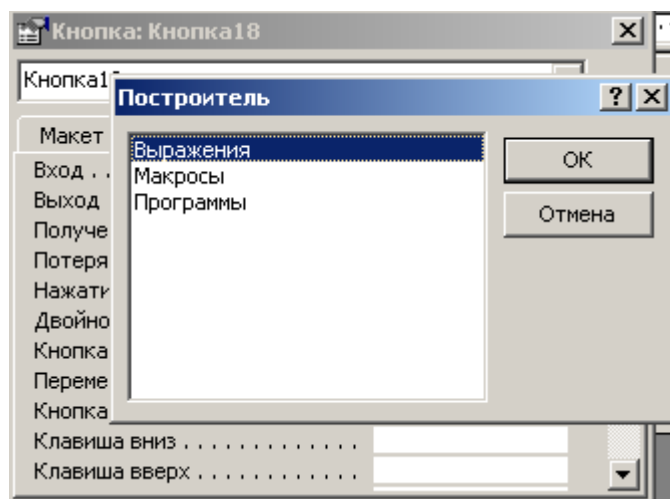


Кнопку можно увеличить, потащив за маркеры выделения, а затем обратиться к ее свойствам. Для того, чтобы обратиться к свойствам любого объекта можно дважды по нему щелкнуть или щелкнуть правой клавишей мыши и из контекстного меню выбрать **Свойства**. Свойства кнопки выглядят следующим образом:



Свойству **Нажатие кнопки** необходимо поставить в соответствие макрос, который загружает таблицу **Книги** в формате Excel. Этот макрос был предварительно сформирован и называется **Запуск приложения**. Поэтому необходимо щелкнуть по маркеру списка в свойстве **Нажатие кнопки** и выбрать позицию **Запуск приложения**.

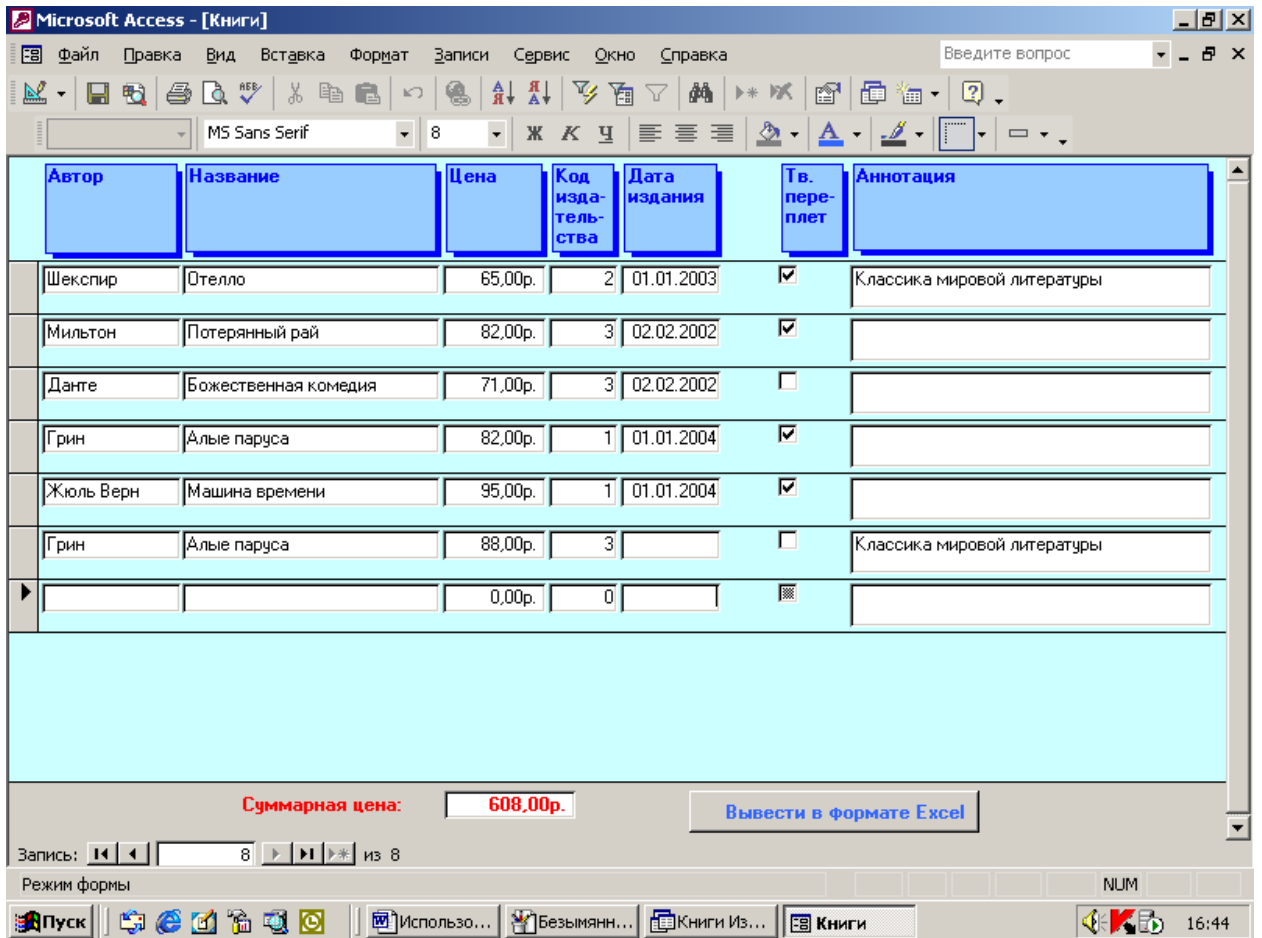
Если бы макрос не был предварительно создан, можно было бы в свойстве **Нажатие кнопки** щелкнуть по строителю выражения . После этого загружается окно, с помощью которого можно сформировать выражение, макрос или программу, не покидая **Конструктора формы**.



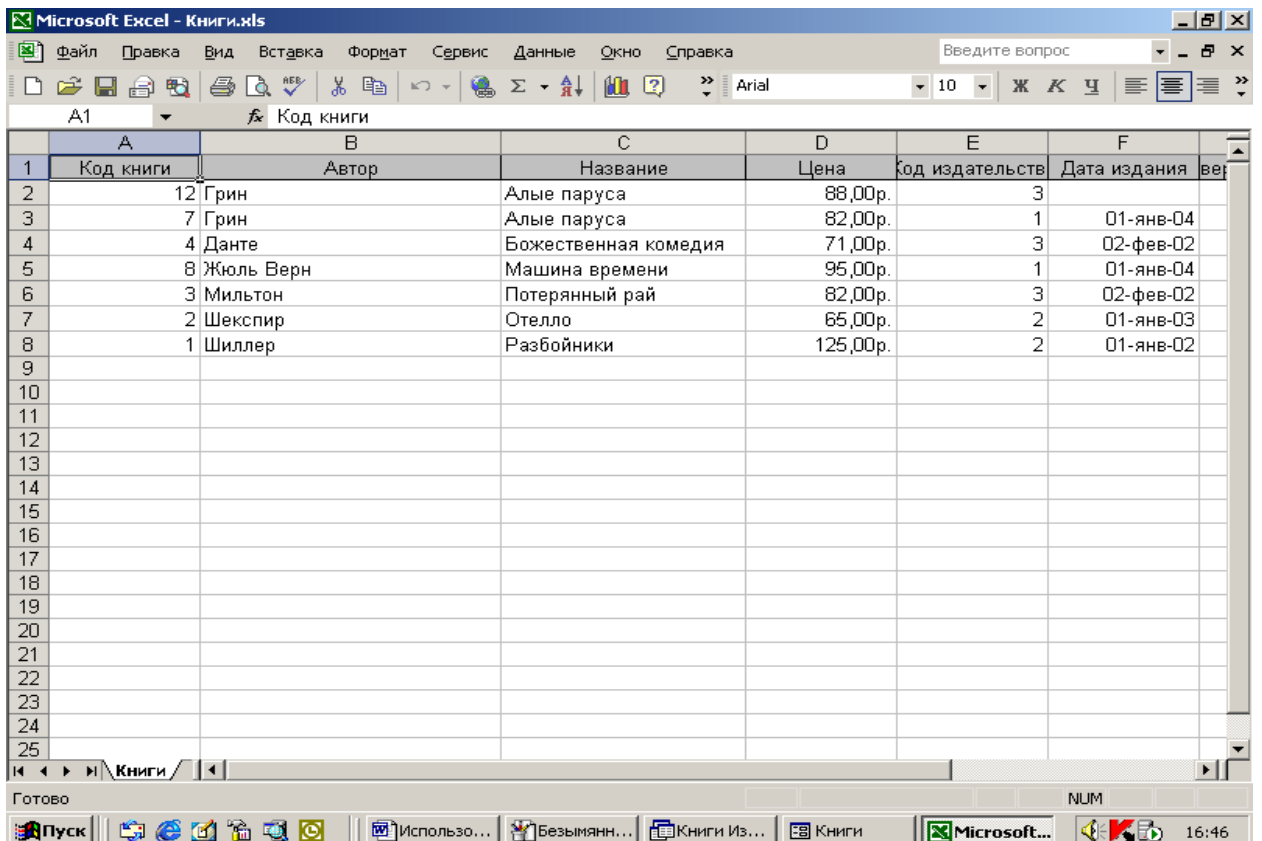
Для задания информативной надписи на кнопке используется свойство **Подпись** на вкладке **Макет**. Незначительное форматирование кнопки позволит привести ее к виду:



Теперь, после сохранения отредактированной формы и открытия ее в режиме просмотра она примет вид:



Если щелкнуть по сформированной кнопке, то загрузится окно:



Таким образом, сформирована и опробована связь макроса с событием.

5. Создания макросов с условиями.

В качестве условий выполнения макросов чаще всего используются значения полей формы. Например, при вводе цены в форме **Книги** необходимо осуществлять проверку ее значения. Для этого формируется макрос вида:

	Условие	Макрокоманда
▶	[Forms]![Книги]![Цена]<5	Сообщение
	[Forms]![Книги]![Цена]>10000	Сообщение

В первой макрокоманде макроса проверяется меньше ли значение поля **Цена** формы **Книги** чем 5. Если это так, то выдается сообщение.

Во второй макрокоманде макроса проверяется больше ли значение поля **Цена** формы **Книги** чем 10000. Если это так, то выдается сообщение.

Аргументы первой макрокоманды следующие:

Сообщение	Таких низких цен не бывает
Сигнал	Да
Тип	Критическое
Заголовок	Неправильная цена!

С помощью аргумента **Сообщение** задается сообщение, которое выводится на экран, в случае если условие выполняется.

С помощью аргумента **Сигнал** можно предусмотреть выдачу звукового сигнала.

Типы сообщений могут быть следующими:

Критическое
Отсутствует
Критическое
Предупреждающее?
Предупреждающее!
Информационное

С помощью аргумента **Заголовок** можно ввести заголовок выводимого сообщения.

Созданный макрос необходимо связать с каким-либо событием формы **Книги**. Для этого она открывается в режиме конструктора:

Область данных

КНИГИ

Автор: Автор

Название: Название

Цена: Цена Отпускная цена: $=[\text{Цена}] * 1,2$

Код издательства: Код издательства

Дата издания: Дата издания

Твердый переплет

Аннотация: Аннотация

Логичнее всего созданный макрос связать с событием **Выход** поля **Цена**:

Поле: Цена

Цена

Макет Данные События Другие Все

До обновления

После обновления

Внесены изменения

Отмена

Изменение

Выход

Выход Проверка цены

Получение фокуса

Потеря фокуса

Нажатие кнопки

Двойное нажатие кнопки

После редактирования формы и сохранения можно проверить правильность работы макроса. Для этого необходимо открыть форму **КНИГИ** в режиме просмотра. Затем ввести в поле **Цена** цену меньшую 5 или большую 10000 рублей.

Результат проверки очень низкой цены следующий:

КНИГИ

Автор: Шиллер

Название: Разбойники

Цена: 2,00р.

Код издательства: 2

Дата издания: 01.01.2002

Твердый переплет

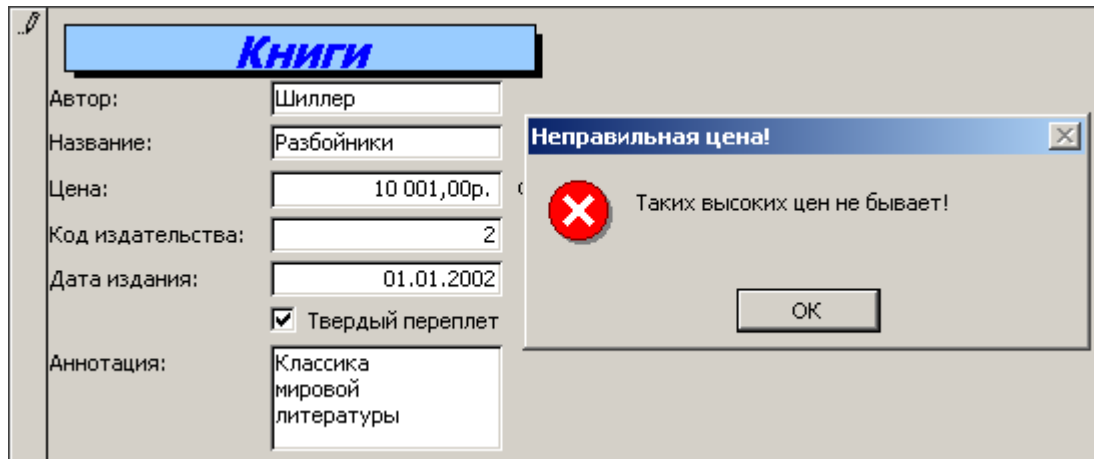
Аннотация: Классика мировой литературы

Неправильная цена!

Таких низких цен не бывает

OK

Результат проверки очень большой цены следующий:



Таким образом, макрос сформирован верно и он откликается на событие выхода из поля **Цена**.

Задание на лабораторную работу.

71. Открыть окно для создания макросов.
72. Выполнить анализ всех его областей.
73. Просмотреть весь список макросов, прочитать справки по этим макросам и справки по их аргументам.
74. Создать несколько простых макросов и проанализировать результаты их выполнения.
75. Создать несколько нетривиальных макросов и проанализировать результаты их выполнения.
76. Создать несколько макросов для связи их с событиями.
77. Отредактировать формы с целью реализации связи событий и созданных макросов.
78. Открыть формы в режиме просмотра и инициировать соответствующие события.
79. Создать макросы с условиями их выполнения.
80. Отредактировать формы с целью реализации связи событий и созданных макросов.
81. Открыть формы в режиме просмотра и инициировать соответствующие события.