

Проектирование таблиц в среде Access

2.1. Введение

Таблицы являются основным объектом всех БД, в том числе и Access. Они служат для хранения данных и представляют собой совокупность записей. Каждая запись включает в себя одно или более полей. Поле определяется его типом и набором свойств. Насколько продуманы содержание и взаимосвязи таблиц перед их проектированием и реализацией, в конечном итоге определяет количественные и качественные характеристики БД. В Access, в отличие от многих СУБД, можно перепроектировать даже заполненные таблицы, хотя это удастся сделать далеко не всегда, поэтому состав и структуру таблиц необходимо тщательно проанализировать до создания других объектов Access, а тем более до эксплуатации БД. При проектировании БД в среде Access и, в частности, ее таблиц, следует помнить некоторые характеристики: максимальный объем БД может составлять 2 Гб, максимальная длина имени поля – 64 символа, максимальное количество полей – 255, максимальная длина записи – 2 Кбайта, теоретически в таблице может содержаться 2 миллиарда записей.

В СУБД Access возможны три способа создания таблиц: путем ввода данных, с помощью мастера и в режиме конструктора, к изучению которых мы и перейдем.

2.2. Создание таблиц путем ввода данных

Самым простым, но не самым лучшим способом создания таблиц, является построение таблиц путем ввода данных.

Окно, в котором представлены ярлыки для выбора способа создания таблиц, приведено на рисунке 2.1:

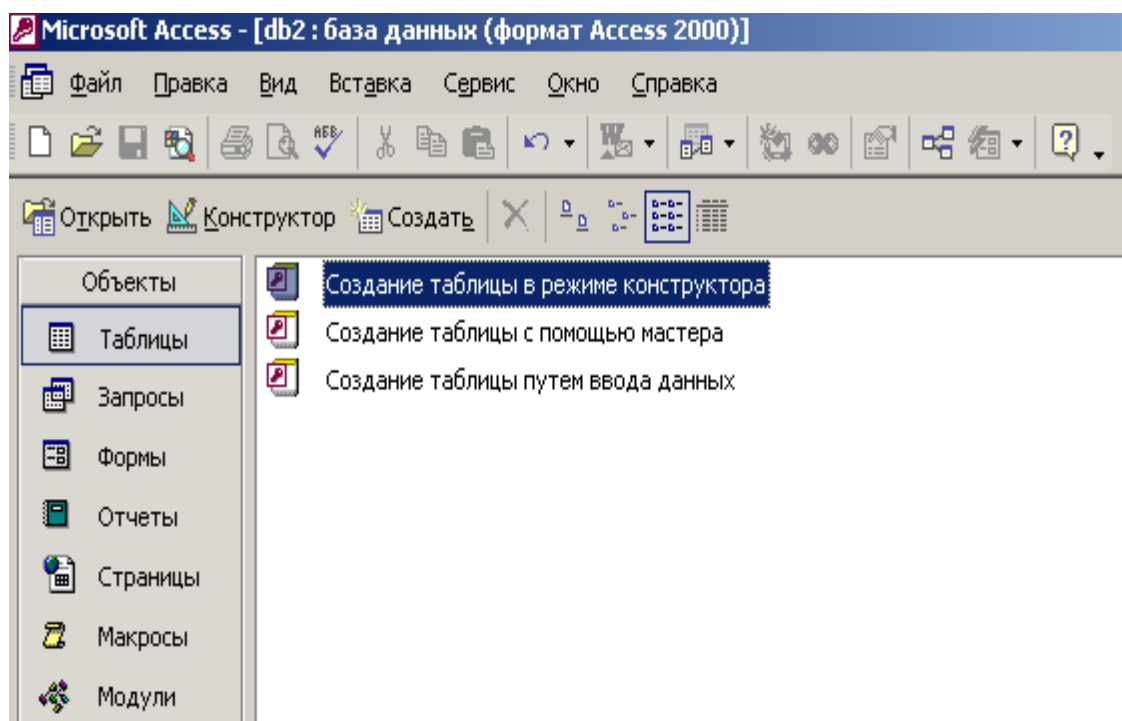


Рис. 2.1.Окно выбора способа создания таблиц

Для формирования таблиц путем ввода данных необходимо выбрать соответствующий ярлык, после чего загрузится следующее окно (рисунок 2.2):

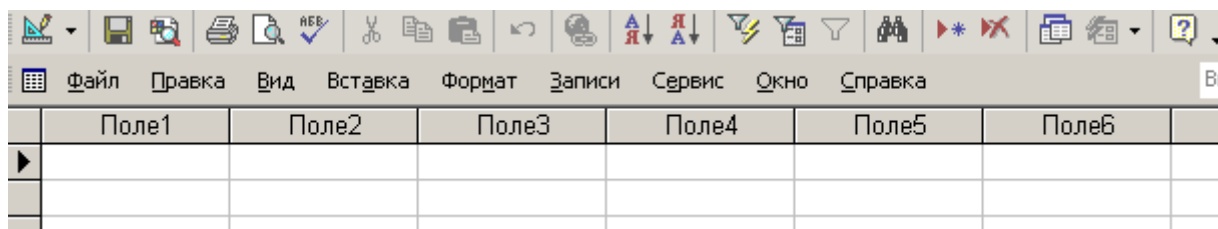


Рис. 2.2.Окно ввода данных в таблицу

В поля представленной таблицы, а всего их шесть, вводятся данные. При этом надо следить за тем, чтобы типы данных, вводимых в одноименные столбцы, были одинаковыми.

Заголовки полей можно поменять, для чего необходимо щелкнуть правой кнопкой мыши по заголовку, выбрать из контекстного меню команду «Переименовать столбец» и ввести новое имя столбца. С помощью контекстного меню можно также вставить, удалить столбец и выполнить другие полезные команды, характерные для текстового редактора Word и табличного процессора 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 этого мастера активное поле преобразуется в поле со списком.

По окончании ввода данных необходимо закрыть и сохранить таблицу, при этом система предложит создать ключевой столбец. Если согласиться с этим предложением, то сформируется дополнительное поле с типом данных Счетчик. На рисунке 2.3 приведен пример заполненной таблицы:

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

Рис. 2.3.Таблица, созданная путем ввода данных

Эта же таблица, открытая в режиме конструктора, выглядит следующим образом (рисунок 2.4):

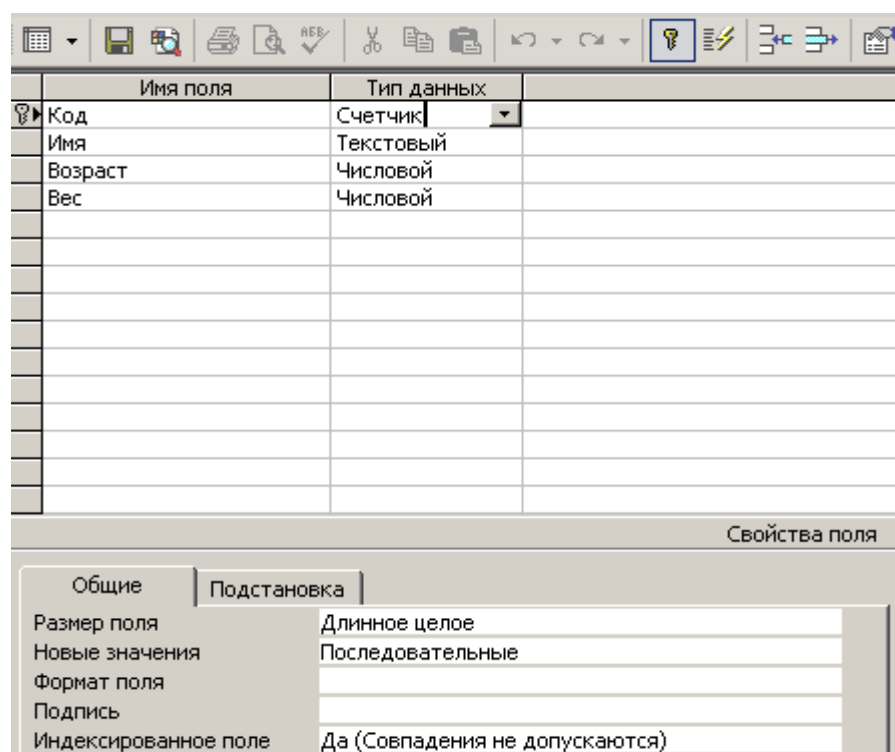


Рис. 2.4. Таблица, открытая в режиме конструктора

Как видно из этого рисунка, сформировано дополнительное поле типа Счетчик, а для других полей автоматически определены их типы и свойства, которые сформированы, исходя из контекста введенных данных. Несмотря на то, что поля «Возраст» и «Вес» – числовые, свойства у них различны, так как их размер равен 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 в последнем случае возможны ошибки преобразования. Конечно, можно удалить данные из таблицы, а затем переопределить ее структуру.

2.3. Создание таблиц с помощью мастера

Для формирования таблиц с помощью мастера необходимо выбрать соответствующий ярлык, после чего загрузится следующее окно (рисунок 2.5):

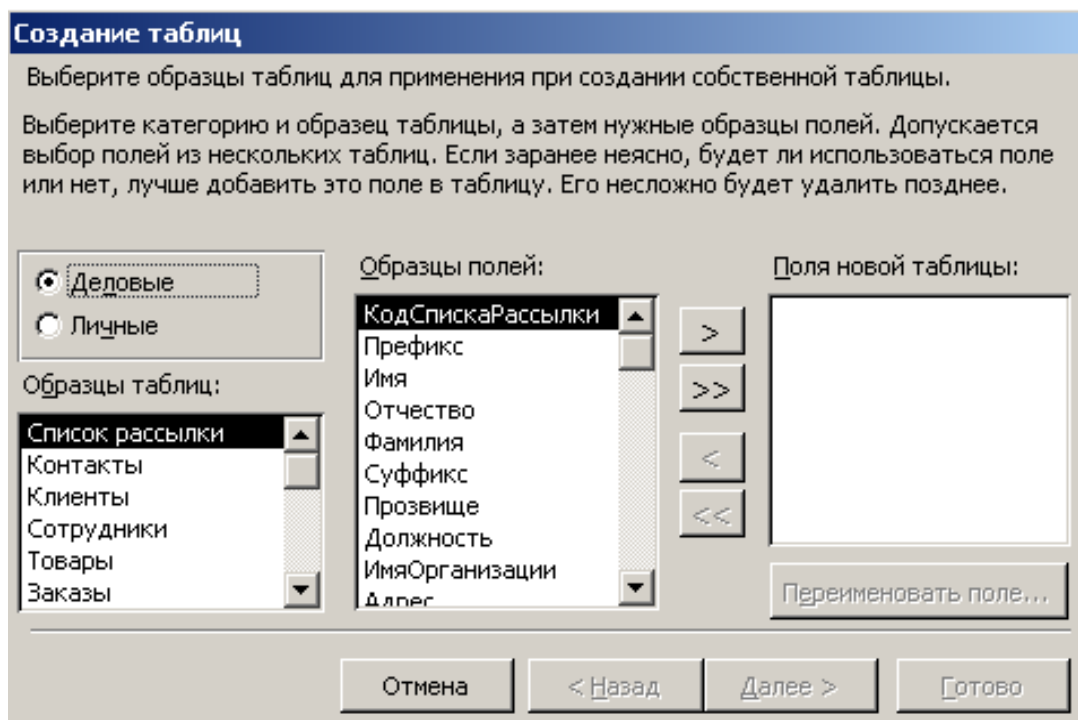


Рис. 2.5.Первое окно мастера создания таблиц

Как видно из рисунка, в левой части окна представлен список спроектированных таблиц, в центральной части – их поля, в правой части будут отображаться поля новой таблицы.

Всего представлено несколько десятков таблиц, причем примерно половина из них служат для делового применения и половина – для личного использования. 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  и  поля удаляются из новой таблицы. Новую таблицу можно также 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 (рисунок 2.6):

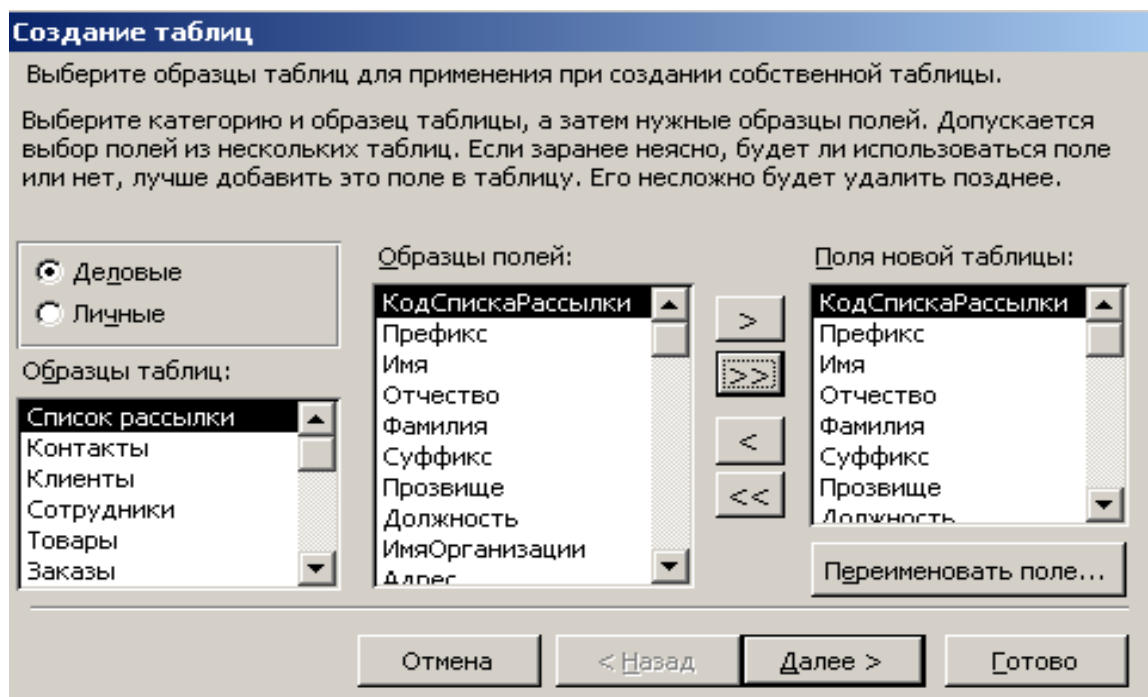


Рис. 2.6. Пример формирования таблицы

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

На следующем шаге мастера, представленном на рисунке 2.7, вводится имя таблицы и выбирается один из трех режимов работы с таблицей:

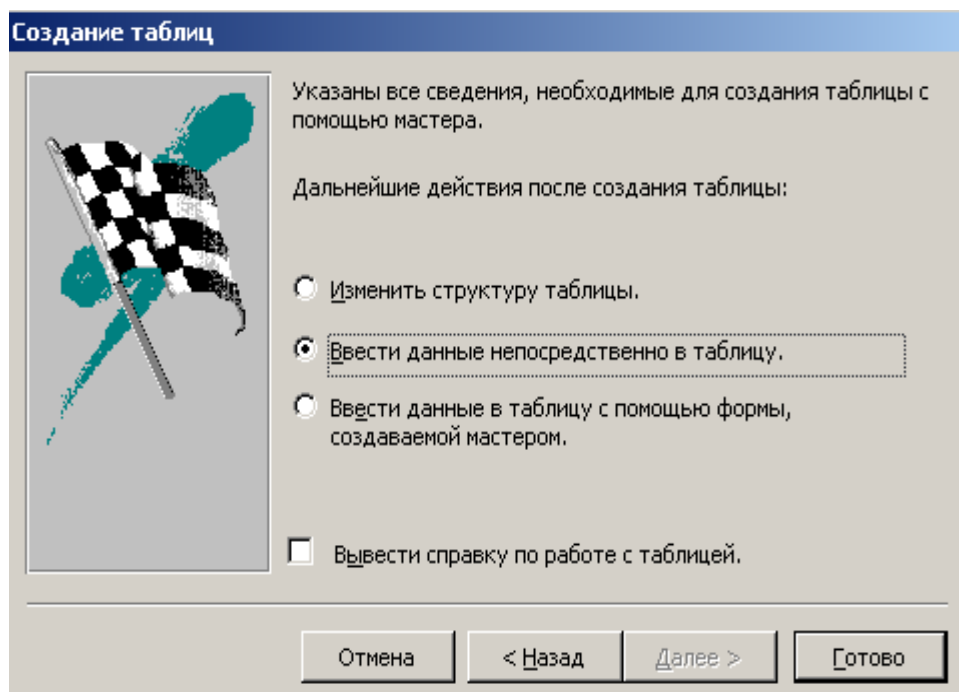


Рис. 2.7. Выбор режимов работы с таблицей

2.4. 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 «Тип данных» описываются их типы, в области «Описание» вводятся необязательные примечания к полям, в области «Свойства поля» назначаются свойства полей.

В подавляющем большинстве случаев таблицы следует проектировать так, чтобы они удовлетворяли требованиям нормализации. Эти требования связаны с совокупным составом полей таблиц, причем обычно 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.

2.4.1. Описание имен полей

Имена полей должны удовлетворять следующим требованиям: они должны быть уникальны, длина имени не должна превышать 64 символа, в них не должны содержаться специальные символы «.», «!», «[», «]». Наконец, имена не должны начинаться с пробела, хотя в самом имени пробелы допустимы. Однако надо иметь в виду, что в этом случае при обращении к таким именам их необходимо заключать в квадратные скобки. Эти же правила действительны для имен всех объектов среды Access.

2.4.2. Назначение типов полей

Типы полей выбираются из предлагаемого списка, приведенного на рисунке 2.9:

Имя поля	Тип данных
Код издательства	Текстовый
	Текстовый
	Поле МЕМО
	Числовой
	Дата/время
	Денежный
	Счетчик
	Логический
	Поле объекта OLE
	Гиперссылка
	Мастер подстановки

Свойства поля

Общие Подстановка

Размер поля 50

Формат поля

Рис. 2.9.Окно выбора типа данных

Ниже кратко охарактеризованы типы полей Access.

Текстовый тип – символьные данные до 255 символов, причем независимо от назначенной длины в поле данного типа хранится только введенное количество символов.

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

Числовой тип используется для хранения числовых данных. Он имеет такие подтипы: байт, целое, длинное целое, одианное 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 соответственно. В системе возможны даты от 100 г. до 9999 г. Формат этого поля выбирается посредством свойства «Формат». На рисунке 2.10 приведены различные форматы:

Формат поля	Полный формат даты	19.06.1994 17:34:23
Маска ввода	Длинный формат даты	19 июня 1994 г.
Подпись	Средний формат даты	19-июн-1994
Значение по умолчанию	Краткий формат даты	19.06.1994
Условие на значение	Длинный формат времени	17:34:23
Сообщение об ошибке	Средний формат времени	5:34
Обязательное поле	Краткий формат времени	17:34
Индексированное поле	Нет	
Режим IME	Нет контроля	
Режим предложений IME	Нет	

Рис. 2.10. Форматы даты и времени

Денежный тип данных предназначен для хранения данных, точность представления которых составляет до 4-х знаков после запятой, а целая часть содержит до 15-ти знаков.

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

Логический тип данных используется для хранения одного из двух значений – Истина/Ложь.

Поле объекта OLE содержит ссылку на OLE-объект, объем которого ограничен только имеющимся в наличии дисковым пространством. Ссылка на конкретный объект реализуется в процессе заполнения этого поля, причем поле данного типа не может быть ключевым или индексным.

Гиперссылка позволяет ввести в поле гиперссылку.

Мастер подстановок позволяет запустить одноименный мастер, который даст возможность определить текущее поле как поле со списком и указать список для выбора.

2.4.3. Назначение свойств полей

Многие свойства полей для различных типов полей совпадают, но есть и оригинальные свойства, поэтому ниже кратко охарактеризованы свойства полей среды Access.

Подпись может быть введена почти для всех типов полей. То, что введено в это поле, 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 это поле.

Формат поля указывает формат вывода данных в режиме Таблицы, причем 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; конечно, при необходимости его можно изменить.

Условие на значение позволяет ввести выражение, в соответствии с которым будут проверяться данные, введенные в поле. Например, для поля «Возраст» может быть введено такое 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.

Свойство «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.

Все перечисленные 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 какую-либо группу полей назначить ключом, необходимо, удерживая клавишу Ctrl, выделить нужные поля и щелкнуть по тому же инструменту.

2.5. Связывание таблиц на схеме данных

2.5.1. Введение

Важнейшими свойствами реальных БД являются возможность получения информации из нескольких таблиц одновременно, а также целостность и непротиворечивость данных. Таблицы отображают реальные объекты, кроме того, они формируются в результате работы по нормализации исходных таблиц. После нормализации исходная таблица может быть преобразована в две или более таблицы. Связи между таблицами позволяют вывести совокупную информацию и автоматически отслеживать целостность данных.

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

В качестве примера связывания таблиц спроектированы две таблицы – Издательства и Книги, представленные на рисунке 2.11:

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

Рис. 2.11.Таблицы Издательства и Книги

При проектировании таблиц задействованы все типы полей, тип поля определяется его назначением. В качестве ключевых полей назначены счетчики, так как счетчики автоматически изменяют свое значение, хотя назначение счетчиков ключевыми полями совсем 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 СУБД Access, так и разработчика БД.

Для формирования, отображения и редактирования схемы данных используется инструмент



после выбора которого загрузится окно, представленное на рисунке 2.12:

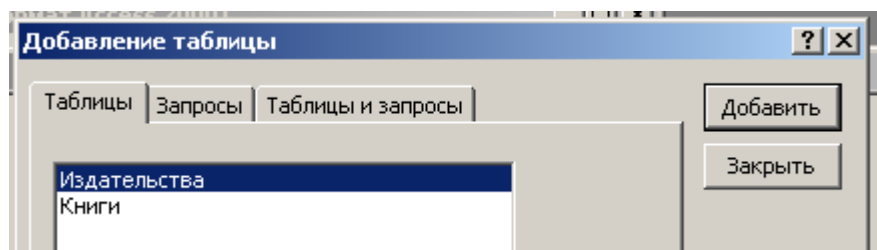


Рис. 2.12.Окно добавления таблиц и запросов

В этом окне выбираются нужные таблицы, а также запросы, которые добавляются в схему данных.

На рисунке 2.13 приведена исходная схема данных для двух таблиц, которые теперь надо связать между собой:

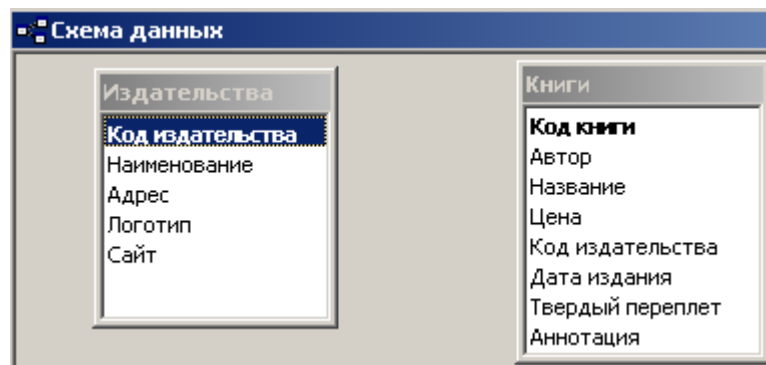


Рис. 2.13. Исходная схема данных

Существуют 4 типа связей между таблицами: один к одному, один ко многим, многие к одному и многие ко многим. Второй и третий типы связей различаются только тем, как по отношению друг к другу располагаются таблицы.

2.5.2. Связь один к одному

Для рассматриваемого случая связь один к одному можно создать, связав первичные ключи таблиц – Код издательства и Код книги. Такая связь будет означать, что каждому издательству ставится в соответствие только одна книга, хотя это и маловероятно. Чтобы связать таблицы, необходимо щелкнуть на ключевом поле одной таблицы и протащить мышью до ключевого поля другой таблицы. В результате сформируется окно, представленное на рисунке 2.14, в котором описывается связь:

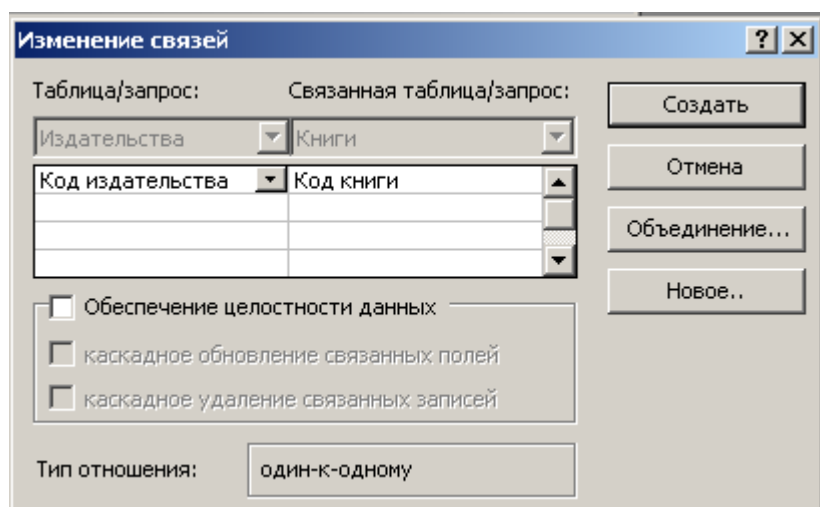


Рис. 2.14. Окно описания связи между таблицами

Если задействовать флажок «Обеспечение целостности данных», 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, хотя значение поля типа Счетчик изменить невозможно.

Если задействовать флажок «Каскадное удаление связанных записей», то при удалении какой-либо записи в одной таблице автоматически выполняется удаление связанной записи в другой таблице.

При формировании связей необходимо правильно установить параметры объединения. От этого зависит последующая выдача информации при запросе к связанным таблицам. Для выбора параметров объединения необходимо щелкнуть по кнопке «Объединение...», после чего сформируется окно (рисунок 2.15):

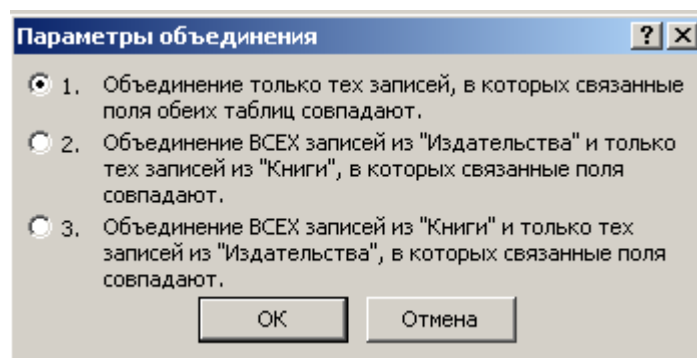


Рис. 2.15.Окно выбора параметров объединения

Эти объединения называются соответственно внутреннее, левое и правое. Кроме того, существует еще и внешнее объединение, но в СУБД Access для формирования такого объединения необходимо выполнить специальные действия.

Внешний вид связи зависит от значения флажка «Обеспечение целостности данных» и выбранных параметров объединения. На рисунках 2.16 – 2.18 представлены все три варианта.

Флажок установлен, объединение внутреннее:



Рис. 2.16.Внутреннее объединение

Флажок установлен, объединение левое:

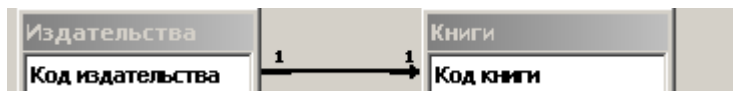


Рис. 2.17.Левое объединение

Флажок установлен, объединение правое:



Рис. 2.18.Правое объединение

Свойства этой связи можно редактировать, для чего по ней надо дважды щелкнуть. Кроме того, связь можно удалить традиционным способом.

2.5.3. Связь один ко многим

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

В нашем случае более логично сформировать связь один ко многим: ведь каждое издательство может издавать множество книг.

Формирование и описание такой связи реализуется по аналогии с предыдущим примером. Только теперь связываются первичный ключ «Код издательства» таблицы «Издательства» и внешний ключ «Код издательства» таблицы «Книги» (рисунок 2.19):

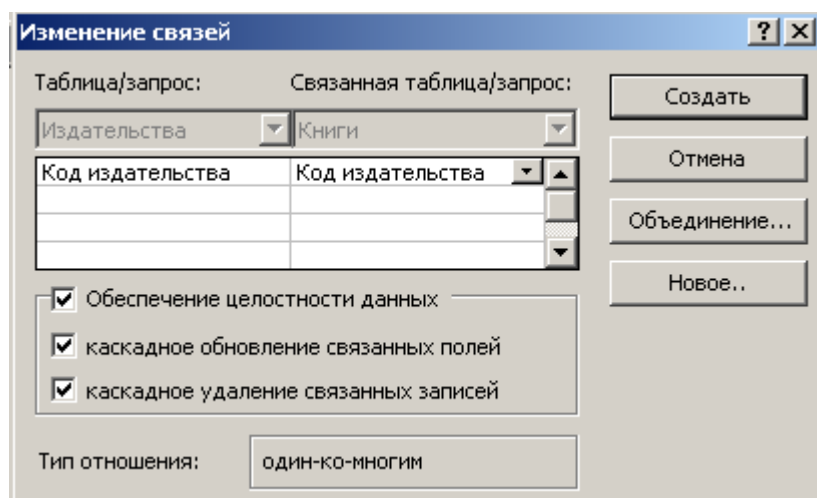


Рис. 2.19.Формирование связи один ко многим

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

Установленные флажки имеют тот же смысл, что и в предыдущем примере. Они означают, 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 обо всех книгах, которые изданы в данном издательстве, из таблицы «Книги».

Внешний вид связи, как и в предыдущем случае, зависит от флажка «Обеспечение целостности данных» и параметров объединения. Единственное отличие заключается в появлении условного знака бесконечности на конце связи со стороны «многие», что видно из рисунка 2.20:

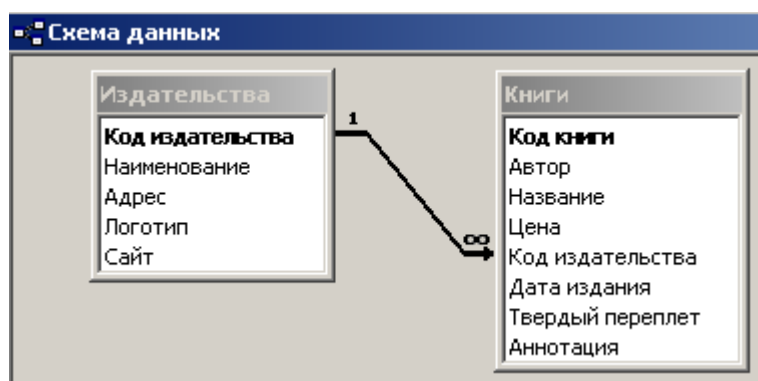


Рис. 2.20.Связь между таблицами один ко многим

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

2.5.4. 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 таблицу вводятся и другие поля. На рисунке 2.21 представлена такая связующая таблица в режиме конструктора.

Поля лучше назвать так же, как и в связываемых таблицах. Типы данных полей в нашем случае должны быть числовыми, их размер – длинное целое (как у счетчика). В свойствах полей «Индексированное поле» необходимо выбрать «Да (Допускаются совпадения)». Это связано с тем, что в таблице «Связи» значения данных полей могут повторяться – связь многие ко многим.

Имя поля	Тип данных
Код издательства	Числовой
Код книги	Числовой

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)

Рис. 2.21. Таблица «Связи» в режиме конструктора

После описания таблицы «Связи» ее следует включить в схему данных. Для этого необходимо загрузить схему данных, выбрать из контекстного меню команду «Добавить таблицу» и добавить таблицу «Связи» в схему данных. В результате появится окно, представленное на рисунке 2.22:

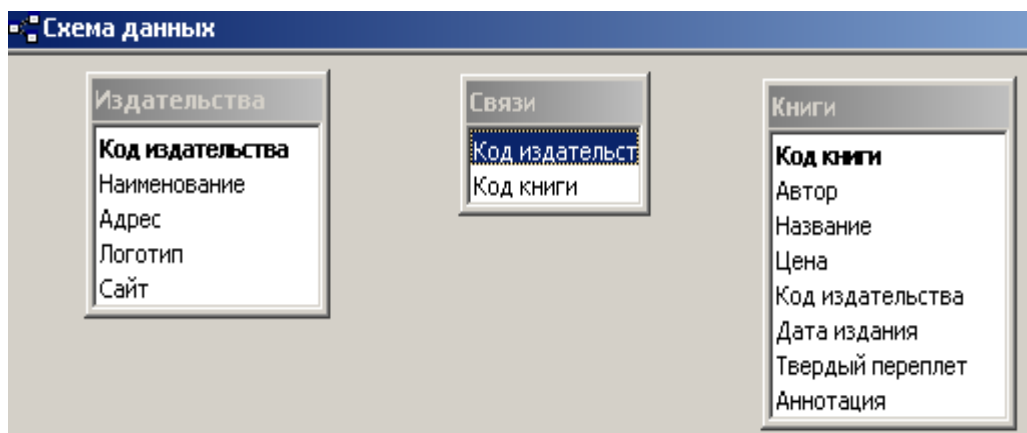


Рис. 2.22. Три таблицы для формирования связи многие ко многим

Теперь осталось сформировать связь один ко многим между таблицами «Издательства» и «Связи» и такую же связь – между таблицами «Книги» и «Связи». В результате получится схема данных, приведенная на рисунке 2.23:

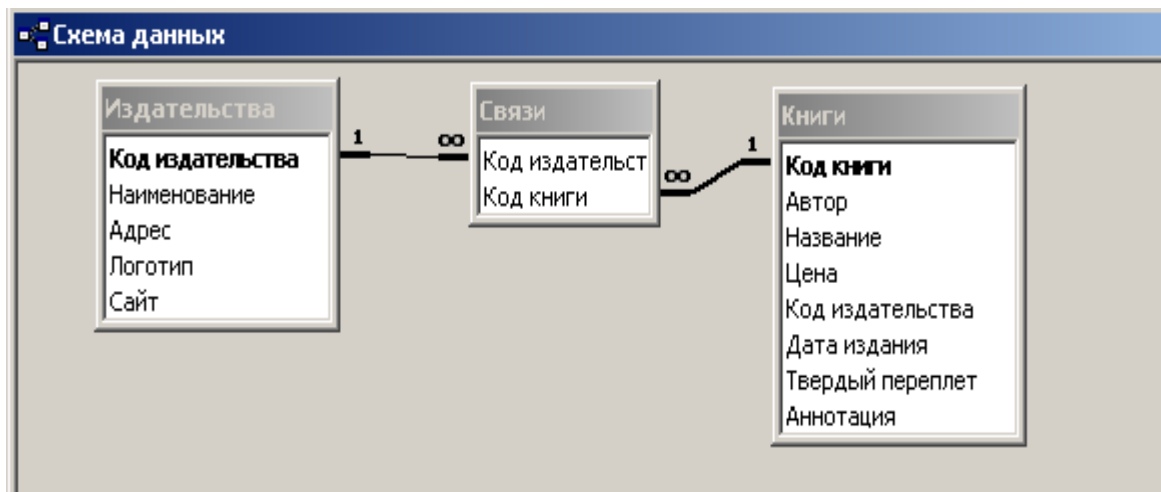


Рис. 2.23.Схема данных со связью многие ко многим
между таблицами «Издательства» и «Книги»

Итак, между таблицами «Издательства» и «Книги» сформирована связь многие ко многим. При построении запроса на основе этой схемы данных можно просмотреть все издательства и книги, которые ими изданы.