

Проектирование запросов с помощью конструктора

3.1. Введение

Запросы, наряду с таблицами, являются основными объектами баз данных – ведь БД, по сути, и разрабатываются для того, чтобы пользователь имел возможность запрашивать хранимую в них информацию в нужном ему 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 или удаления данных, создания новых таблиц, добавления данных одной таблицы к записям другой. В запросах можно выполнять вычисления с использованием значений полей и сортировать выводимые данные по этим значениям.

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

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

3.2. Проектирование запросов на выборку

Базовое окно среды Access после выбора объектов «Запросы» выглядит следующим образом (рисунок 3.1):

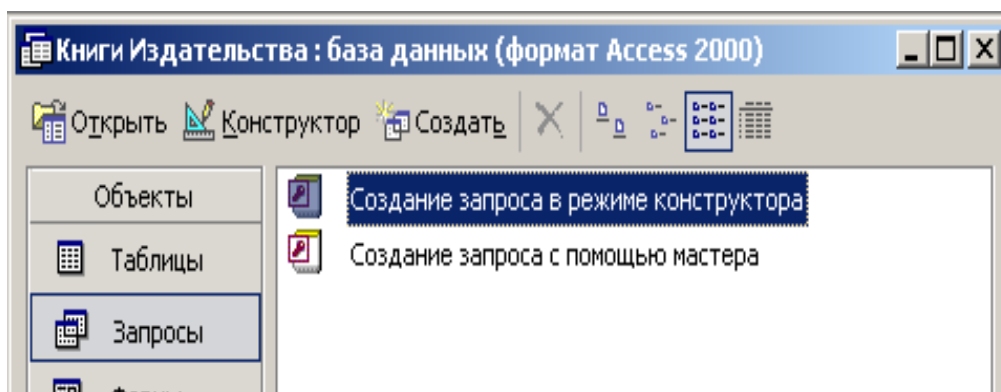


Рис. 3.1.Базовое окно создания запросов

Для того чтобы инициировать проектирование запроса в режиме конструктора, необходимо щелкнуть по соответствующей строке. Первый запрос можно также создать посредством инструмента «Конструктор», расположенного в верхней строке панели инструментов. После инициации формирования запроса загрузится окно (рисунок 3.2):

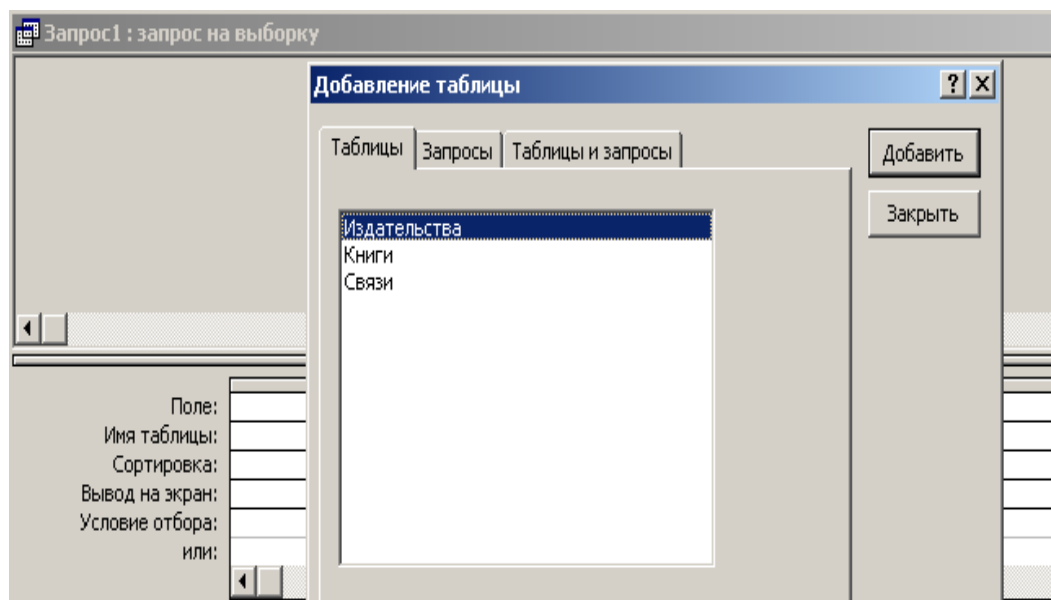


Рис. 3.2.Окно формирования запроса

Запрос можно формировать на основе таблицы, другого запроса или на базе того и другого. Для выбора объекта, на основе которого строится запрос, используются три вкладки. На вкладке «Таблицы» 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 3.3:

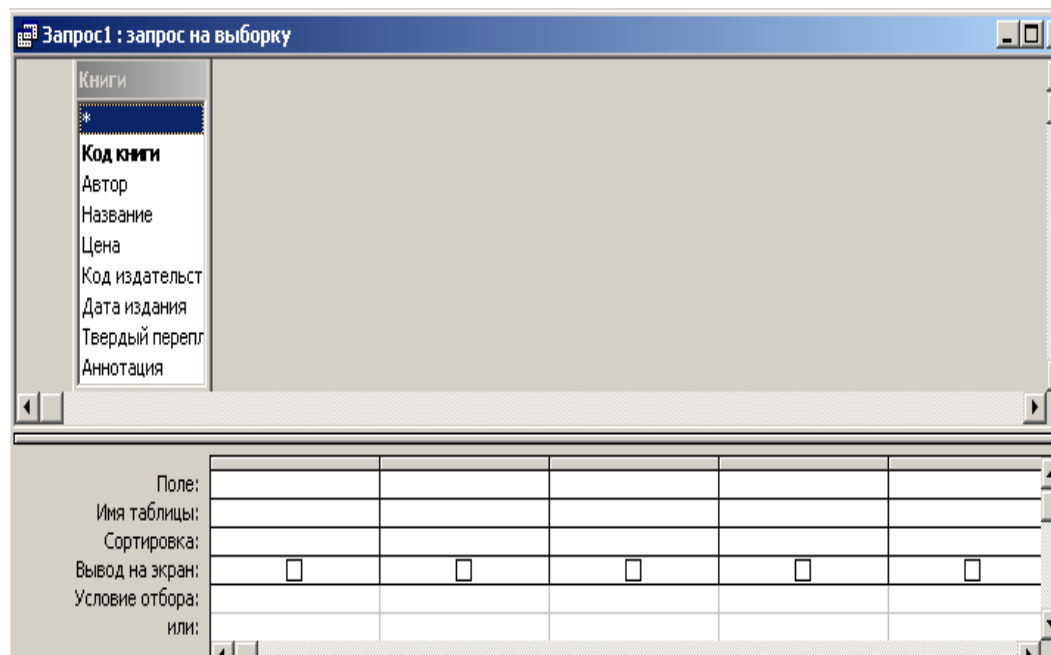


Рис. 3.3.Таблица «Книги» для формирования запроса

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», как это показано на рисунке 3.4:

Поле:	
Имя таблицы:	Книги.*
Сортировка:	Код книги
Вывод на экран:	Автор
Условие отбора:	Название
или:	Цена
	Код издательства
	Дата издания
	Твердый переплёт

Рис. 3.4.Список полей для формирования запроса

Если выбрать строку «*», то запрос будет формироваться для всех полей таблицы «Книги». В приведенном на рисунке 3.5 примере задействованы четыре поля этой таблицы:

Поле:	Автор	Название	Цена	Дата издания
Имя таблицы:	Книги	Книги	Книги	Книги
Сортировка:				
Вывод на экран:	☒	☒	☒	☒
Условие отбора:				
или:				

Рис. 3.5.Запрос по четырем полям таблицы «Книги»

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 выходя из конструктора. Для этого надо щелкнуть по инструменту «Запуск»:



После выполнения сформированного запроса получим следующую выборку (рисунок 3.6):

	Автор	Название	Цена	Дата издания
►	Шиллер	Разбойники	120,00р.	2002-01-01
	Шекспир	Отелло	55,00р.	2003-01-01
	Мильтон	Потерянный рай	77,00р.	2002-02-02
	Данте	Божественная комедия	66,00р.	2002-02-02
	Гомер	Илиада	55,00р.	2002-02-02
	Гомер	Одиссея	55,00р.	2003-02-02
	Грин	Алые паруса	77,00р.	2004-01-01
	Жюль Верн	Таинственный остров	90,00р.	2004-01-01

Рис. 3.6.Результат выполнения запроса

Как видно из рисунка, записи не отсортированы. Чтобы их отсортировать, в бланке запроса используется строка «Сортировка» (рисунок 3.7):

Поле:	Автор	Название	Цена	Дата издания
Имя таблицы:	Книги	Книги	Книги	Книги
Сортировка:	по возрастанию			
Вывод на экран:	☒	по возрастанию	☒	☒
Условие отбора:		по убыванию		
или:		(отсутствует)		

Рис. 3.7.Выполнение сортировки по возрастанию

После запуска этого запроса записи в выборке будут отсортированы по авторам в алфавитном порядке, как это видно из рисунка 3.8:

	Автор	Название	Цена	Дата издания
►	Грин	Алые паруса	88,00р.	
	Грин	Алые паруса	82,00р.	01.01.2004
	Данте	Божественная комедия	71,00р.	02.02.2002
	Жюль Верн	Таинственный остров	95,00р.	01.01.2004
	Мильтон	Потерянный рай	82,00р.	02.02.2002
	Шекспир	Отелло	65,00р.	01.01.2003
	Шиллер	Разбойники	125,00р.	01.01.2002
*			0,00р.	

Рис. 3.8.Результат выполнения запроса с сортировкой

Для фильтрации записей таблицы в бланке запроса используется строка «Условие отбора», которое может выглядеть следующим образом (рисунок 3.9):

Поле:	Автор	Название	Цена	Дата издания
Имя таблицы:	Книги	Книги	Книги	Книги
Сортировка:	по возрастанию			
Вывод на экран:	☒	☒	☒	☒
Условие отбора:			<90	>#01.09.2002#
или:				

Рис. 3.9.Примеры условий отбора

Практически для каждого поля можно использовать условия отбора, для чего они объединяются союзом И (AND). Есть возможность объединять условия и союзом ИЛИ (OR), для чего в бланке запроса задействуется последняя строка, причем таких строк можно использовать практически неограниченное количество. Для того чтобы отличить поле типа дата от других типов полей, его необходимо заключать в символы решетка «#».

Результат запуска запроса с условиями – на рисунке 3.10:

	Автор	Название	Цена	Дата издания
►	Грин	Алые паруса	77,00р.	2004-01-01
	Гомер	Одиссея	55,00р.	2003-02-02
	Шекспир	Отелло	55,00р.	2003-01-01
	Жюль Верн	Таинственный остров	90,00р.	2004-01-01

Рис. 3.10.Выборка по запросу с условиями отбора

Запросы на выборку нередко можно использовать для ввода данных, о чем свидетельствует последняя запись в полученной выборке, 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, поэтому этот запрос нельзя использовать для ввода данных. При попытке добавления записи возникнет сообщение об ошибке (рисунок 3.11):

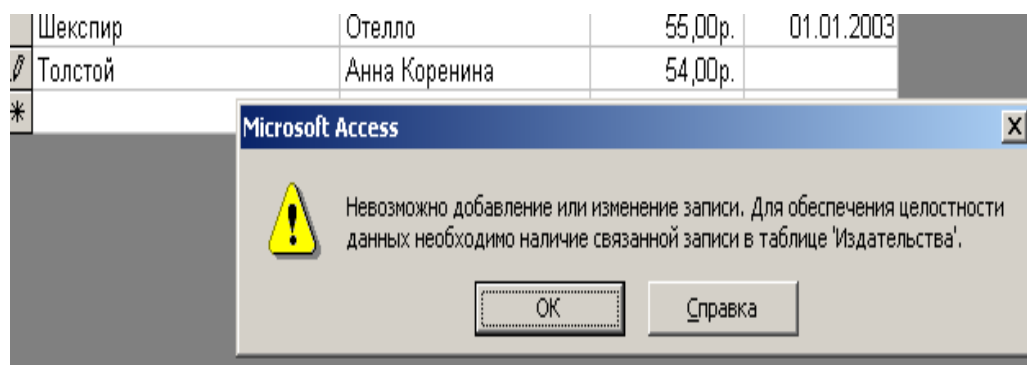


Рис. 3.11.Сообщение об ошибочной попытке добавления записи

Этот и другие запросы можно просмотреть в формате SQL, заодно освоив основные конструкции языка запросов, для чего необходимо выбрать меню «Вид/Режим SQL». В формате SQL запрос выглядит следующим образом (рисунок 3.12):

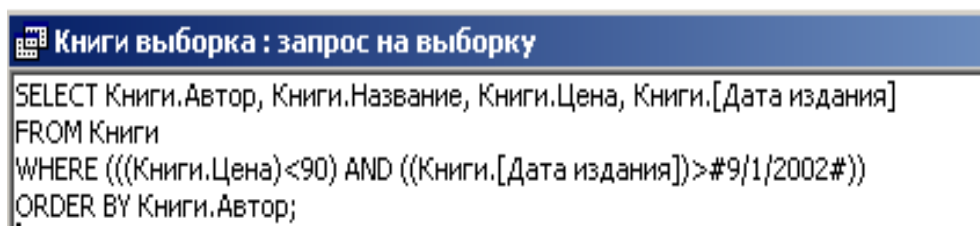


Рис. 3.12.Вид запроса в форматеSQL

Команда SELECT указывает на то, что формируется запрос на выборку данных. В ней перечисляются выводимые поля. Так как поля могут выводиться из нескольких таблиц, то перед именем поля указывается имя таблицы. Конструкция FROM указывает на 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 – «Код издательства» (рисунок 3.13):

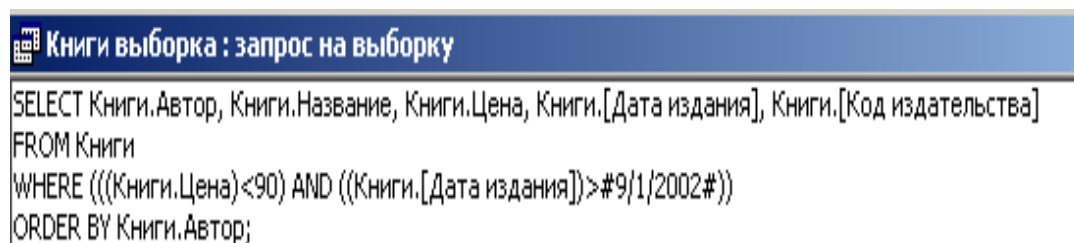


Рис. 3.13.Запрос с добавлением кода издательства

В запрос можно включить вычисляемое поле, например, поле «Наценка» (рисунок 3.14):

Поле:	Автор	Название	Цена	Дата издания	Наценка: [Цена]*0,2
Имя таблицы:	Книги	Книги	Книги	Книги	
Сортировка:	по возрастанию				
Вывод на экран:	☒	☒	☒	☒	☒
Условие отбора: или:			<90	>#01.09.2002#	

Рис. 3.14. Запрос с добавлением вычисляемого поля

Чтобы формат вывода поля «Наценка» был денежный, необходимо обратиться к общим свойствам этого поля и выбрать «Денежный» формат поля, для чего надо выделить столбец «Наценка», щелкнуть правой клавишей мыши и выбрать позицию контекстного меню «Свойства» (рисунок 3.15):

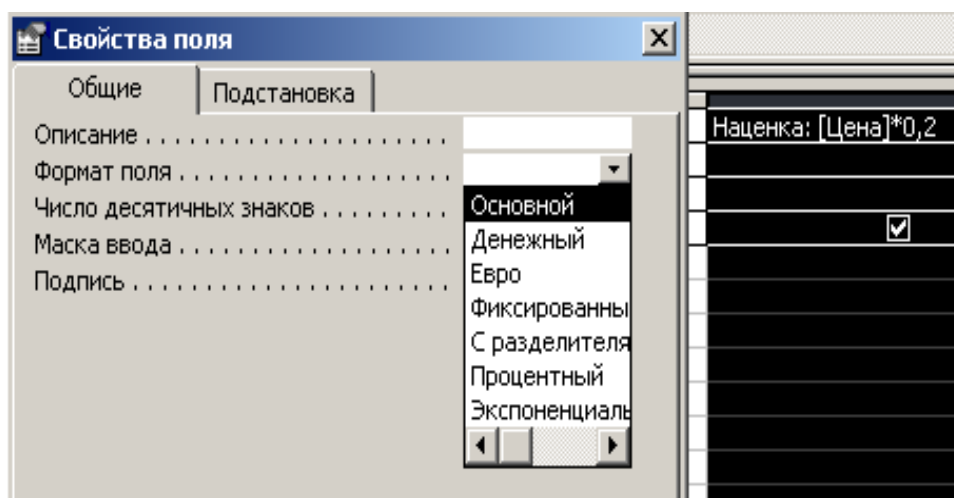


Рис. 3.15. Выбор денежного формата поля «Наценка»

Результат запуска этого запроса представлен на рисунке 3.16:

	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.

Рис. 3.16. Выборка по запросу с новым полем

3.3. Проектирование перекрестных запросов

Перекрестные запросы подобны структурам табличного редактора Excel, в которых присутствуют заголовки строк и заголовки столбцов. В ячейках таблицы располагаются числовые данные. Примером такой таблицы может быть ведомость зарплат. Заголовки строк – фамилии, заголовки столбцов – даты, в ячейках располагаются суммы. В СУБД Access обычную таблицу можно преобразовать в перекрестную таблицу, то есть сформировать запрос. В качестве примера такого преобразования используем следующий запрос на выборку, приведенный на рисунке 3.17:

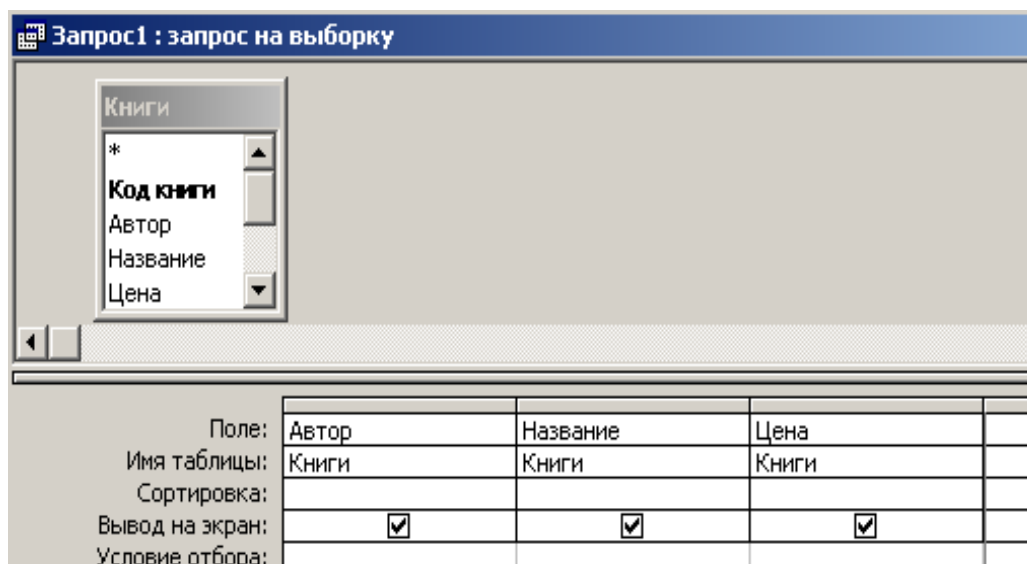


Рис. 3.17. Исходный запрос на выборку

Для преобразования этого запроса в перекрестный необходимо выбрать в меню, которое показано на рисунке 3.18, команду «Запрос/Перекрестный».

В результате исходный запрос преобразуется в следующий вид, приведенный на рисунке 3.19.

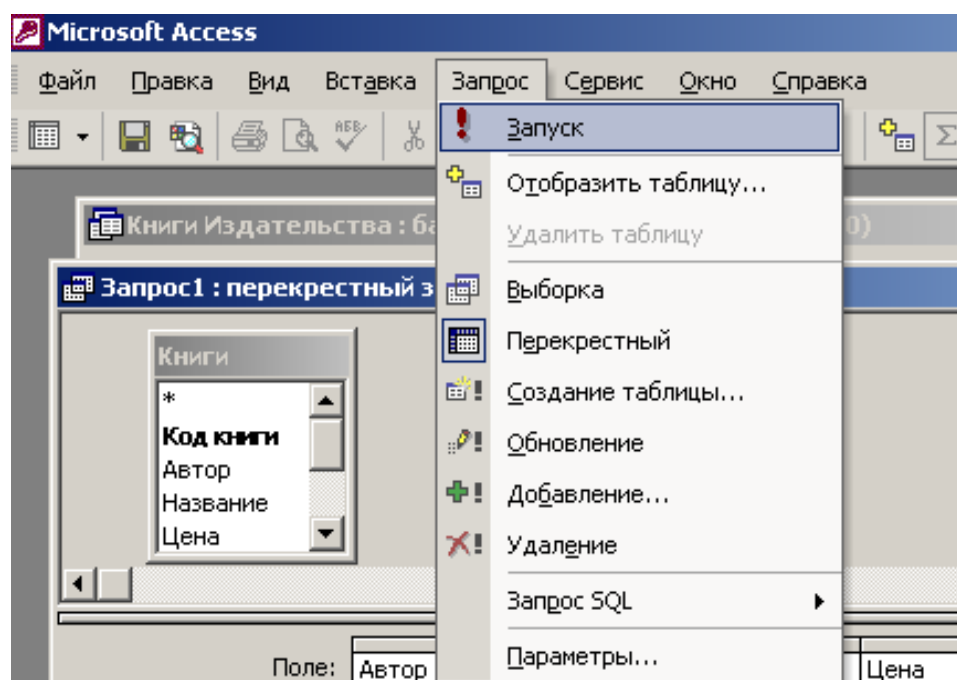


Рис. 3.18. Начало формирования перекрестного запроса

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

Рис. 3.19. Преобразование исходного запроса

Как видно из последнего рисунка, в бланке запроса возникли две новые строки «Групповая операция» и «Перекрестная таблица». По умолчанию в строке «Групповая операция» установлено значение «Группировка», которое означает, что записи группируются по одинаковым значениям. Например, если в таблице «Книги» есть несколько одноименных книг одного и того же автора, то при группировке по полям «Автор» и «Название» в перекрестном запросе эта одноименная книга будет представлена один раз. По полю «Цена» группировать данные не имеет смысла, для этого поля лучше выбрать из предлагаемого списка сумму (Sum), что отражено на рисунке 3.20:

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)		

Рис. 3.20. Назначение полям групповых операций

В строке бланка запроса «Перекрестная таблица» для всех полей необходимо выбрать из списка их назначение. Результат такого выбора показан на рисунке 3.21:

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

Рис. 3.21. Заполнение строки «Перекрестная таблица»

Для следующих данных, представленных в таблице «Книги» и на рисунке 3.22, результат выполнения спроектированного перекрестного запроса в режиме просмотра приведен на рисунке 3.23.

Код книги	Автор	Название	Цена	Код издателя	Дата изда
12	Грин	Алые паруса	88,00р.	3	
7	Грин	Алые паруса	82,00р.	1	01.01.
4	Данте	Божественная комедия	71,00р.	3	02.02.
8	Жюль Верн	Таинственный остров	95,00р.	1	01.01.
3	Мильтон	Потерянный рай	82,00р.	3	02.02.
2	Шекспир	Отелло	65,00р.	2	01.01.
1	Шиллер	Разбойники	125,00р.	2	01.01.

Рис. 3.22. Таблица «Книги»

	Автор	Алые паруса	Божественная	Отелло	Потерянный ра	Разбойники	Таинс
►	Грин	170,00р.					
	Данте		71,00р.				
	Жюль Верн						
	Мильтон				82,00р.		
	Шекспир			65,00р.			
	Шиллер					125,00р.	

Рис. 3.23.Результат выполнения перекрестного запроса

Как видно из результатов выдачи перекрестного запроса, суммарная стоимость книг Грина «Алые паруса» – 170 р. Можно привести множество примеров, когда перекрестные запросы оказываются очень полезными.

В режиме SQL полученный перекрестный запрос выглядит так:

```
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 необходимость выполнения сложения 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 показывает, 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 значение поля «Автор» таблицы «Книги». Конструкция PIVOT Книги.Название показывает, что в качестве привязки к столбцам запроса используется поле «Название» таблицы «Книги».

3.4. 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, также бывает полезно сформировать таблицу, которую можно использовать в качестве временного хранилища данных.

Запрос на создание таблицы, как и перекрестный, строится на основе запроса на выборку. В качестве примера используем запрос, приведенный на рисунке 3.24:

Поле:	Автор	Название
Имя таблицы:	Книги	Книги
Сортировка:		
Вывод на экран:	☒	☒
Условие отбора:		

Рис. 3.24. Исходный запрос на выборку

Здесь, как и во многих других типах запросов, можно задействовать строки бланка «Сортировка» и «Условие отбора». Предполагается, что в новой таблице будет храниться информация о названиях книг и их авторах. Однако в исходной таблице встречаются повторения названных сочетаний, которые можно исключить в свойствах запроса. Для доступа к свойствам запроса необходимо дважды щелкнуть по серой области бланка и в строке «Уникальные значения» выбрать слово «Да» (рисунок 3.25):

Свойство	Значение
Описание	
Режим по умолчанию	Режим та
Вывод всех полей	Нет
Набор значений	Все
Уникальные значения	Да
Уникальные записи	Нет
При запуске предоставляются права	Пользоват
База данных-источник	(текущая)
Строка подключения-источник	
Блокировка записей	Отсутстве
Тип набора записей	Динамиче
Время ожидания ODBC	60
Фильтр	

Рис. 3.25. Общие свойства запроса

Для преобразования исходного запроса в запрос на создание таблицы необходимо обратиться к меню «Запрос/Создание таблицы», после чего будет запрошено имя создаваемой таблицы (рисунок 3.26):

Создание таблицы
имя таблицы:
☒ в текущей базе данных ☐ в другой базе данных:
имя файла:
Обзор...
ОК
Отмена

Рис. 3.26.Окно ввода имени создаваемой таблицы

Как видно из рисунка, таблицу можно создать и в другой базе данных. После ввода имени создаваемой таблицы и выполнения запроса сформируется новая таблица, содержимое которой представлено на рисунке 3.27:

	Автор	Название
►	Шиллер	Разбойники
	Шекспир	Отелло
	Мильтон	Потерянный рай
	Данте	Божественная комедия
	Грин	Алые паруса
	Жюль Верн	Таинственный остров
	Грин	Алые паруса

Рис. 3.27.Сформированная таблица

Для сохранения запроса, сформированного в режиме конструктора, так же 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 СУБД Access.

Спроектированный запрос на создание таблицы в формате SQL выглядит следующим образом:

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

FROM Книги;

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

3.5. Проектирование запроса на добавление

Запрос на добавление позволяет добавить записи одной таблицы к записям другой. Запрос такого рода нередко может оказаться полезным, например, в следующей ситуации. По электронной почте поступают данные в виде таблиц о вновь открывшихся издательствах, которые нужно добавлять в обобщенную таблицу «Издательства».

Запрос на добавление, как и другие запросы, строится на основе запроса на выборку. Такой исходный запрос с использованием таблицы «Издательства1» приведен на рисунке 3.28:

Поле:	Код издательства	Наименование	Адрес	Логотип	Сайт
Имя таблицы:	Издательства1	Издательства1	Издательства1	Издательства1	Издательства1
Сортировка:					
Вывод на экран:	☒	☒	☒	☒	☒
Условие отбора:					

Рис. 3.28.Исходный запрос на выборку

Для формирования запроса на добавление необходимо выбрать меню «Запрос/Добавление», после чего откроется уже знакомое окно, приведенное на рисунке 3.29, в котором нужно указать имя таблицы – приемника информации:

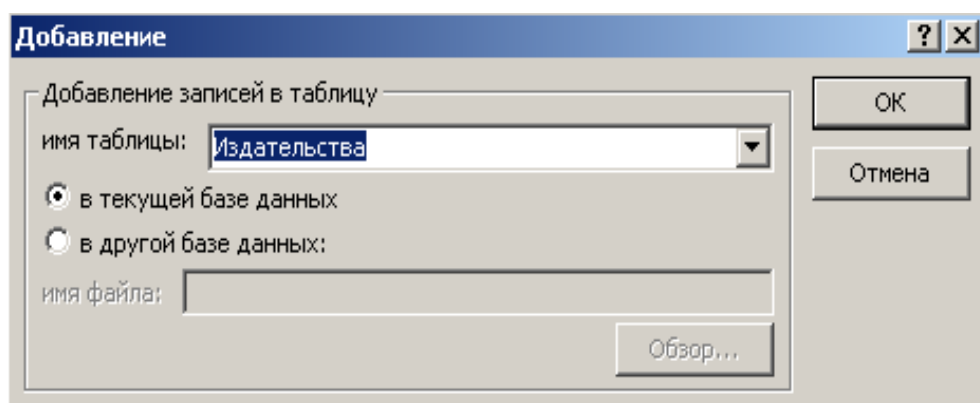


Рис. 3.29.Окно ввода имени таблицы-приемника

После щелчка по кнопке ОК получившийся запрос примет вид, приведенный на рисунке 3.30:

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

Рис. 3.30.Сформированный запрос на добавление

В бланке запроса появилось новая строка «Добавление», в которой представлены поля таблицы-приемника. В рассматриваемом случае структуры обеих таблиц совпадают, поэтому в этой строке исправлять ничего не надо, в противном случае могут потребоваться исправления.

Содержимое таблицы «Издательства» до выполнения запроса представлено на рисунке 3.31:

	Код издательс	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

Рис. 3.31.Исходная таблица «Издательства»

Содержимое таблицы «Издательства1» до выполнения запроса представлено на рисунке 3.32:

	Код издательс	Наименование	Адрес	Логотип	Сайт
►	4	Машиностроен	Королева,15		
	5	Энергия	Уральская,77		

Рис. 3.32.Исходная таблица «Издательства1»

После выполнения запроса на добавление содержимое таблицы «Издательства» примет вид, приведенный на рисунке 3.33:

	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		

Рис. 3.33.Таблица «Издательства» с добавленными записями

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;

Конструкция SELECT Издательства1.[Код издательства], Издательства1.Наименование, Издательства1.Адрес, Издательства1.Логотип, Издательства1.Сайт FROM Издательства1 позволяет выбрать из таблицы «Издательства1» 5 полей. Конструкция INSERT INTO Издательства ([Код издательства], Наименование, Адрес, Логотип, Сайт) позволяет добавить значения выбранных полей в таблицу «Издательства» в соответствии с перечисленными полями этой таблицы.

3.6. Проектирование запроса на обновление

Нередко требуется изменить значения каких-либо полей. Например, нужно увеличить цену всех книг на определенную величину или уменьшить стоимость дорогих книг. Для реализации операций над данными такого рода используются запросы на обновление.

Запрос на обновление также формируется на основе запроса на выборку. В качестве примера рассмотрим следующий запрос, приведенный на рисунке 3.34:

The screenshot shows a query builder window. On the left, a list of fields from the 'Книги' table is shown: Код книги, Автор, Название, and Цена. The 'Цена' field is selected. Below this, a table summarizes the query parameters:

Поле:	Цена	
Имя таблицы:	Книги	
Сортировка:		
Вывод на экран:	☒	☐
Условие отбора:		

Рис. 3.34.Исходный запрос на выборку

Для преобразования запроса на выборку в запрос на обновление необходимо выбрать меню «Запрос/Обновление», после чего бланк запроса примет такой вид (рисунок.3.35):

Рис. 3.35. Бланк запроса со строкой «Обновление»

Как видно из этого рисунка, в бланке запроса появилась строка «Обновление», в которой можно ввести выражение для обновления. Например (рисунок 3.36):

Рис. 3.36. Запрос с выражением для обновления

Имя поля необходимо заключать в квадратные скобки, чтобы отличать его от текста в выражении. После выполнения этого запроса стоимость всех книг увеличится на 10 рублей.

В результате содержимое таблицы «Книги» примет вид, представленный на рисунке 3.37:

	Код книги	Автор	Название	Цена
	12	Грин	Алые паруса	93,00р.
	7	Грин	Алые паруса	87,00р.
	4	Данте	Божественная комедия	76,00р.
	8	Жюль Верн	Таинственный остров	100,00р.
	3	Мильтон	Потерянный рай	82,00р.
	2	Шекспир	Отелло	65,00р.
	1	Шиллер	Разбойники	130,00р.

Рис. 3.37. Результат выполнения запроса на обновление

В запросе на обновление нередко используют условие отбора. Например, требуется уценить на 5 рублей книги, которые стоят больше 70-ти рублей. Для этого необходим запрос вида, приведенного на рисунке 3.38:

Рис. 3.38. Запрос на обновление с условием отбора

В результате выполнения этого запроса содержимое таблицы «Книги» примет вид, представленный на рисунке 3.39:

	Код книги	Автор	Название	Цена
▶	12	Грин	Алые паруса	88,00р.
	7	Грин	Алые паруса	82,00р.
	4	Данте	Божественная комедия	71,00р.
	8	Жюль Верн	Таинственный остров	95,00р.
	3	Мильтон	Потерянный рай	77,00р.
	2	Шекспир	Отелло	65,00р.
	1	Шиллер	Разбойники	125,00р.

Рис. 3.39.Содержимое таблицы «Книги»

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, так как они сформированы автоматически на случай использования сложного выражения.

3.7. Проектирование запроса на удаление

В процессе работы с БД иногда возникает необходимость удаления некоторых записей таблицы. Например, нужно удалить книги какого-либо автора или издательства. Для реализации операций над данными такого рода используются запросы на удаление.

Запросы на удаление также формируются на основе запроса на выборку. В качестве примера рассмотрим следующий запрос, который приведен на рисунке 3.40:

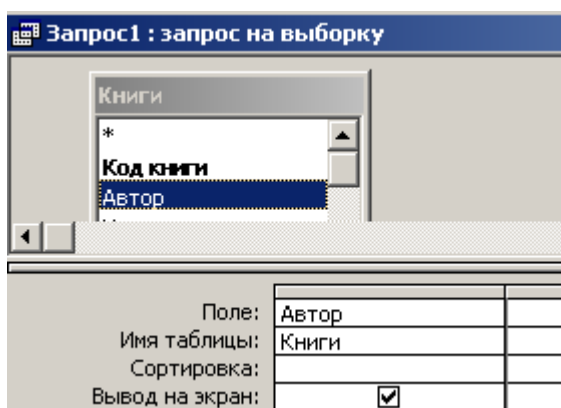


Рис. 3.40.Исходный запрос на выборку

Для преобразования запроса на выборку в запрос на удаление необходимо обратиться к меню «Запрос/Удаление». В результате запрос преобразуется к следующему виду (рисунок 3.41):

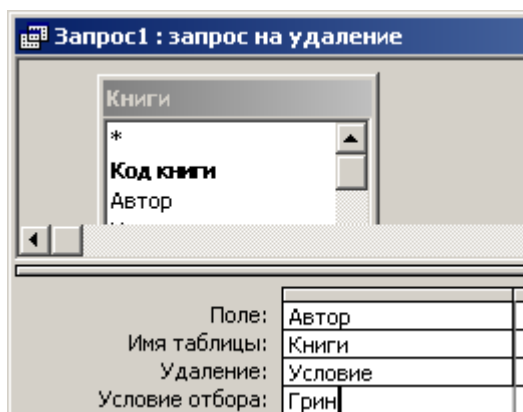


Рис. 3.41. Запрос на удаление

Как видно из этого рисунка, бланк запроса изменился. Если в строку «Условие отбора» ничего не вводить, то при выполнении запроса очистится вся таблица. Бывают такие случаи, когда это нужно сделать. 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 только те, которые соответствуют введенным критериям. В данном случае в качестве критерия используется автор Грин. Следует помнить, что удаление записей – необратимая операция, поэтому проектировать и использовать запросы такого рода нужно очень аккуратно. Правда стандартные средства среды Access, если они не отменены разработчиком БД, предупреждают пользователя о результатах, к которым приведет выполнение запросов на создание таблицы, добавление, обновление, удаление. Например, при попытке выполнить сформированный запрос отобразится окно (рисунок 3.42):

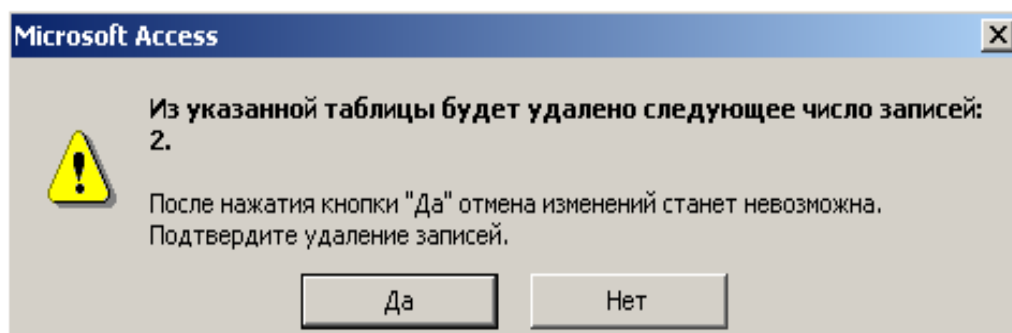


Рис. 3.42. Сообщение о результате выполнения запроса

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 вводить его в процессе работы. Для этого применяются параметрические запросы, для обозначения которых условие отбора берется в квадратные скобки. Например (рисунок 3.43):

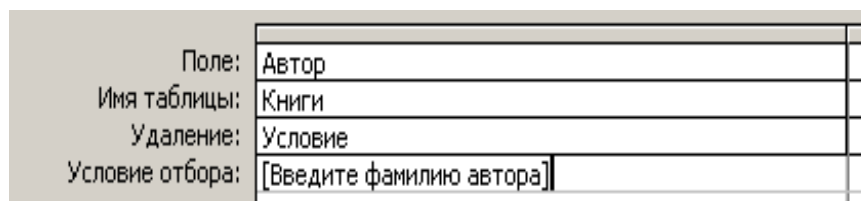


Рис. 3.43. Формирование параметрического запроса

В качестве параметра используется предложение «Введите фамилию автора». Такое сложное имя параметра выбрано в связи с тем, что это имя выводится при запрашивании значения параметра.

При обращении к данному запросу сформируется форма, в которой пользователь должен указать фамилию автора (рисунок 3.44):

Рис. 3.44.Форма для ввода значения параметра

В результате выполнения запроса содержимое таблицы «Книги» примет вид, приведенный на рисунке 3.45:

Код книги	Автор	Название	Цена	Код издательс	Д
12	Грин	Алые паруса	88,00р.	3	
7	Грин	Алые паруса	82,00р.	1	
4	Данте	Божественная комедия	71,00р.	3	
8	Жюль Верн	Таинственный остров	95,00р.	1	
2	Шекспир	Отелло	65,00р.	2	
1	Шиллер	Разбойники	125,00р.	2	

Рис. 3.45.Результат выполнения запроса на удаление

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

В формате SQL сформированный 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 система Access сама назначает им имена по умолчанию. Первому запросу приписывается имя «Запрос1». Если же такое имя уже есть, то запросу приписывается имя «Запрос2» и так далее, что характерно и для других объектов. При сохранении запроса его имя можно изменить в предлагаемой форме, которая приведена на рисунке 3.46:

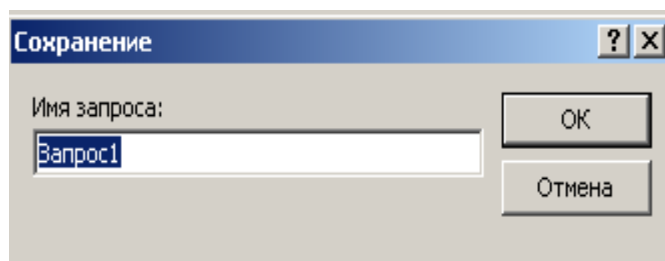


Рис. 3.46.Форма для ввода имени запроса

Имеет смысл присваивать запросам информативные имена, чтобы при проектировании объектов БД на основе сформированных запросов было легче разобраться в назначении последних.

Спроектированные запросы в базовом окне СУБД Access выглядят следующим образом (рисунок 3.47):

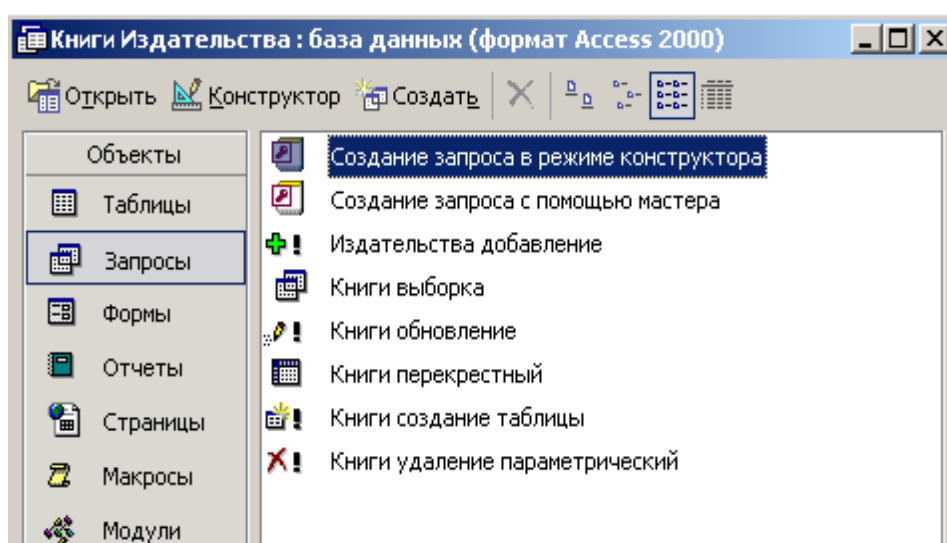


Рис. 3.47.Представление запросов в базовом окне

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

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

3.8. 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 следующего примера в бланке запроса добавлены таблицы «Издательства» и «Книги», что показано на рисунке 3.48.

Между таблицами автоматически образовалась связь. Это произошло потому, что она была сформирована при построении схемы данных. Эту связь можно удалить. А если в схеме данных связь не установлена или схема данных отсутствует, то такую связь можно сформировать в бланке запроса.

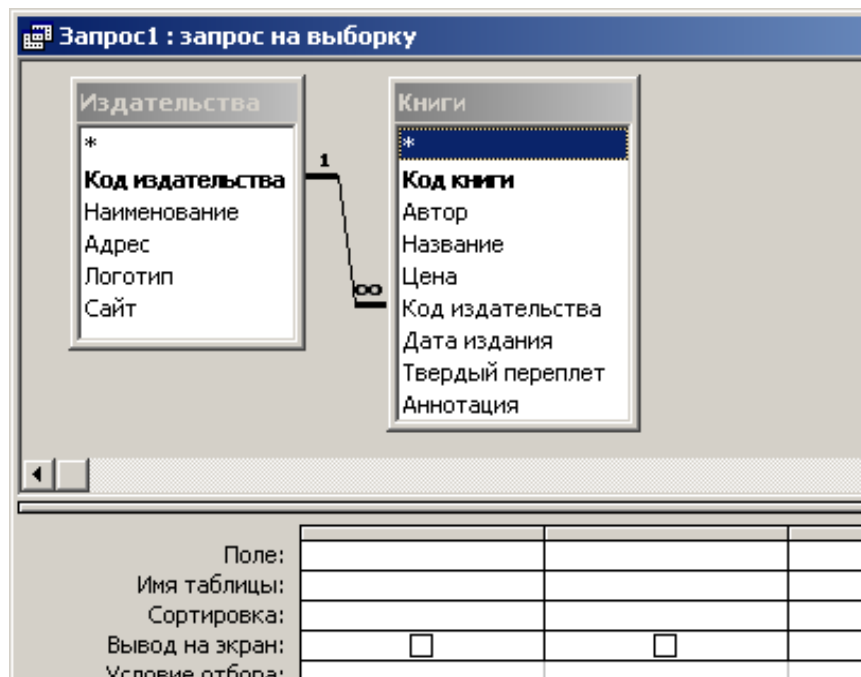


Рис. 3.48. Формирование запроса для двух таблиц

Для удаления связи необходимо щелкнуть по ней и нажать клавишу «Delete». На рисунке 3.49 приведен бланк запроса с удаленной связью и двумя выводимыми полями из разных таблиц:

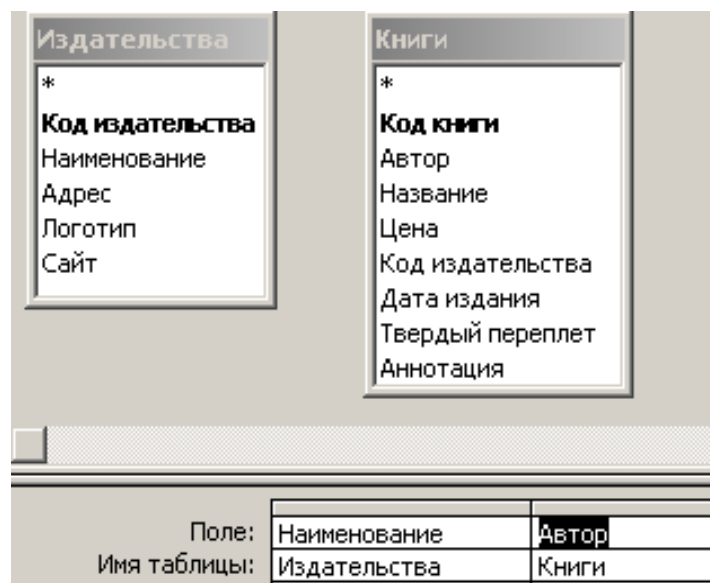


Рис. 3.49. Запрос с удаленной связью между таблицами

Результатом выполнения этого запроса будет декартово произведение двух полей, другими словами, будет выведено каждое издательство со всеми авторами. Часть записей, отображаемых при выполнении данного запроса, приведена на рисунке 3.50:

	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

Рис. 3.50.Результат выполнения запроса по двум таблицам

Всего сформировано 35 записей, количество которых равно произведению количества книг в таблице «Книги» (7) на количество издательств в таблице «Издательства» (5).

Проектировать такой запрос для этих таблиц не имеет большого 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 протащить курсор до внешнего ключа «Код издательства» таблицы «Книги». В среде Access можно задействовать один из трех типов связей, о чем ранее уже шла речь. Для назначения типа связи необходимо дважды щелкнуть по связи, в результате чего отобразится форма «Параметры объединения» (рисунок 3.51):

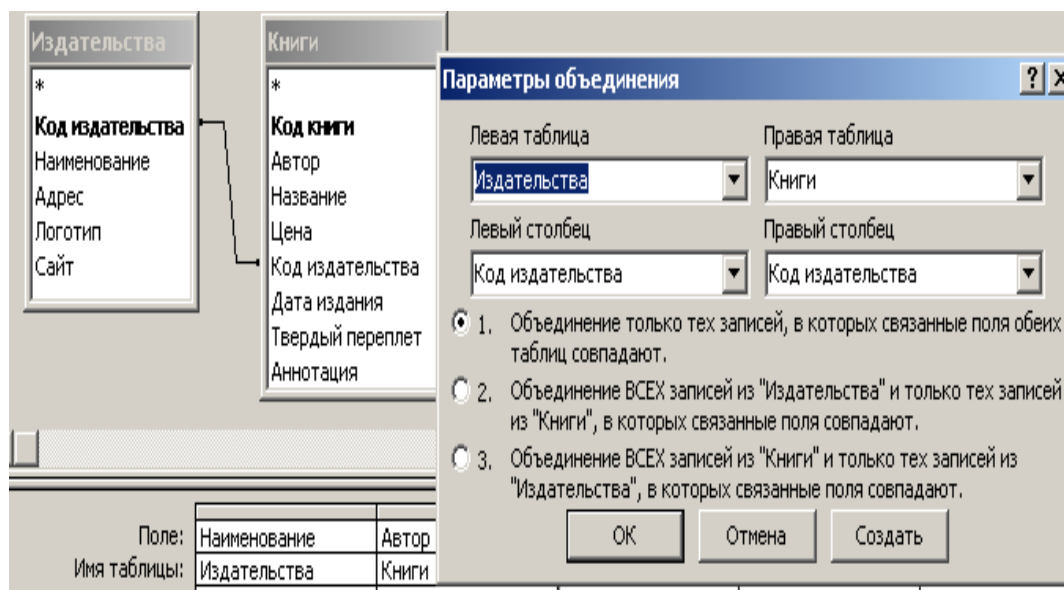


Рис. 3.51.Выбор параметров объединения

Если выбрать 1-й тип объединения, то результат выполнения запроса будет иметь вид, представленный на рисунке 3.52:

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

Рис. 3.52.Первый результат выполнения запроса

В формате SQL этот запрос имеет вид:

SELECT Издательства.Наименование, Книги.Автор

FROM Издательства INNER JOIN Книги ON Издательства.[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 двух 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, представленный на рисунке 3.53:

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

Рис. 3.53.Второй результат выполнения запроса

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 этом запросе используется левое соединение (LEFT JOIN), то есть 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, представленный на рисунке 3.54:

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

Рис. 3.54.Третий результат выполнения запроса

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 этом запросе используется правое соединение (RIGHT JOIN), то есть такое соединение, при котором выбираются все записи правой таблицы и только те записи левой таблицы, в которых все связанные поля совпадают.

Из соответствующих рисунков видно, что результаты выполнения запросов в первом и третьем случаях совпадают. Это произошло в связи с тем, что при вводе книг в таблицы «Книги» всегда вводился «Код издательства», поэтому в таблице «Книги» нет книг, с которыми нет связанных записей в таблице «Издательства». В рассматриваемом случае просто невозможно сформировать запись в таблице «Книги», не указав «Код издательства». Это обеспечено схемой данных, приведенной на рисунке 3.55:

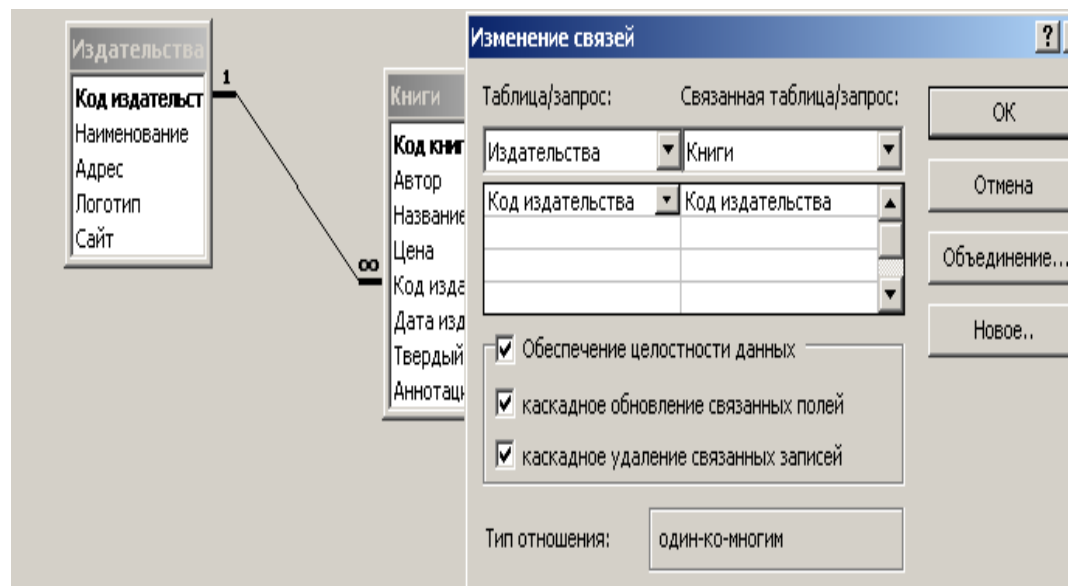


Рис. 3.55.Окно «Изменение связей» между двумя таблицами

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