

Глава 5. Создание запросов

5.1. Общая характеристика запросов

При работе с БД часто возникает необходимость выбрать из нее информацию, удовлетворяющую определенным условиям, или произвести различные вычисления. Например, нужно извлечь из учебной БД **Деканат** сведения о студентах, сдавших сессию на «отлично», либо определить, каков средний балл экзамена по английскому языку в данной группе.

Для решения таких задач предназначены *запросы*. Запрос сообщает Access, какая именно информация интересует пользователя в настоящий момент. В нем можно указать, какие

- таблицы содержат нужную информацию;
- записи нужно выбрать из таблиц БД и порядок их сортировки;
- поля должны быть выданы на экран;
- вычисления следует выполнить над выбранными данными.

Запросы могут быть использованы для выделения в таблицах различных групп данных и выполнения над ними вычислений и сравнений. С их помощью можно также выполнить следующие операции:

- добавление новых записей в таблицу;
- удаление записей из таблицы;
- изменение содержимого полей таблицы.

Запрос часто используется в качестве источника данных при создании формы или отчета. Открытие такой формы или отчета автоматически приводит к выполнению соответствующего запроса. Поэтому на экране всегда присутствует самая «свежая» информация об объекте.

Результат запроса можно распечатать или передать с помощью буфера обмена в другое приложение Windows, например, вставить в виде таблицы в документ Word.

Для создания запросов обычно используется метод, суть которого заключается в следующем. Заполняется специальная форма *бланк запроса*. В нее включаются нужные поля и выражения, а также указываются условия отбора записей и порядок их сортировки. Тем самым создается образец желаемого результата запроса.

Access анализирует этот образец и сохраняет в виде инструкций *языка структурированных запросов SQL* (Structured Query Language). Именно в таком виде запрос затем используется в качестве источника данных для форм и отчетов. Отметим, что некоторые важные виды запросов нельзя получить путем заполнения бланка запроса. Для их создания необходимо написать инструкцию на SQL.

5.1.1. Типы запросов в Access

Access позволяет создавать следующие типы запросов:

Запрос-выборка. Используется для отбора информации из таблиц и других запросов БД. При выполнении этого запроса Access создает *динамический набор* записей, содержащий выбранные данные. С этим набором в большинстве случаев можно работать как с обычной таблицей. Его можно просматривать и редактировать, выбирать нужную информацию и т.д. Однако в отличие от таблицы динамический набор записей физически в базе данных не существует и хранится в памяти ЭВМ лишь во время выполнения запроса. При изменении данных в полях динамического набора Access обычно автоматически вносит сделанные изменения в таблицы, на базе которых был построен запрос.

Перекрестный запрос. Представляет данные в формате, подобном электронной таблице, на основе условий, определенных в запросе. С его помощью можно сгруппировать большой объем информации и представить его в виде, удобном для восприятия. Этот вид запроса часто применяется при построении диаграмм.

Запрос на изменение. За одну операцию выбирает записи на основе указанных условий отбора и вносит в них изменения. Существует четыре типа запросов на изменение:

- *запрос на добавление* — добавляет группу записей из одной таблицы в другую таблицу.
- *запрос на удаление* — позволяет удалить группу записей из одной или нескольких таблиц.
- *запрос на обновление* — вносит изменения в группу записей одной или нескольких таблиц.
- *запрос на создание таблицы* — позволяет создать таблицу на основе данных, содержащихся в других таблицах БД.

Запрос SQL — запрос, который может быть создан только с помощью SQL. Существует три типа таких запросов:

- *запрос-объединение* — позволяет объединить поля из нескольких таблиц или запросов в один набор данных;
- *запрос к серверу* — передает инструкции SQL удаленной базе данных;
- *управляющий запрос* — создает, изменяет или удаляет таблицы или индексы базы данных Access.

Запросы являются гибким, интерактивным и итеративным средством. Если запрос сформулирован неточно, его можно легко изменить и выполнить еще раз. Составляя запросы, построенные на результатах предыдущих запросов, вы можете работать с базой данных, задавая ей вопросы типа «А что, если ...?».

5.1.2. Режимы окна запроса

Окно запроса может находиться в одном из трех режимов: конструктора, таблицы и SQL.

Режим конструктора предназначен для создания новых и изменения существующих запросов. Для открытия существующего запроса в режиме конструктора нужно открыть вкладку **Запросы** окна БД, выбрать нужный запрос из списка запросов и щелкнуть по кнопке **Конструктор**. В окне конструктора запросов появится выбранный запрос.

Режим таблицы служит для предварительного просмотра данных, отобранных запросом, или, в случае запроса на изменение, для просмотра данных, которые могут измениться в результате выполнения запроса.

Окно запроса может также находиться в *режиме SQL*, который используют для ввода или просмотра инструкций SQL при создании или изменении запроса. При создании запроса в режиме конструктора Access автоматически создает в режиме SQL эквивалентную инструкцию SQL.

Для переключения между этими режимами следует использовать кнопку **Вид** , расположенную слева на панели инструментов. Нажатие этой кнопки переводит окно в режим, значок которого выведен на ней. Если нажать стрелку рядом с кнопкой, то раскроется список доступных режимов и можно выбрать нужный режим.

5.1.3. Создание запроса

Прежде чем приступить к созданию запроса, следует продумать ответы на следующие вопросы:

- какая таблица или таблицы содержит нужную информацию,
- как связать эти таблицы,
- какой тип запроса использовать,
- каким условиям должны удовлетворять отбираемые записи,
- каким должен быть порядок их сортировки,
- какие вычисления нужно выполнить над отобранными данными,
- какое имя должен получить создаваемый запрос.

Для создания запроса нужно щелкнуть по корешку **Запросы** окна БД, а затем по кнопке **Создать** над списком запросов. На экране появится диалоговое окно **Новый запрос**, и Access предоставит вам выбор: создать запрос самому в режиме конструктора или воспользоваться помощью одного из мастеров по разработке запросов.

С помощью мастеров можно создать запросы следующих типов:

- *Простой запрос*. Создается запрос на выборку из указанных полей.

- *Перекрестный запрос.* Выводит данные в формате электронной таблицы.
- *Поиск повторяющихся записей.* Осуществляется поиск повторяющихся записей в указанной таблице или запросе.
- *Поиск записей, не имеющих подчиненных.* Находит все записи в главной таблице, не имеющие связанных с ними записей в подчиненной таблице.

Использование мастера Простой запрос

Для быстрого создания запроса можно воспользоваться мастером **Простой запрос**. Этот мастер позволяет не только отобрать нужные поля из таблиц или ранее созданных запросов, но также суммировать, вычислять средние значения и находить другие статистические характеристики определенных групп записей.

После выбора этого варианта появится диалоговое окно **Создание простых запросов**. В раскрывающемся списке **Таблицы/запросы** нужно выбрать таблицу или запрос и с помощью двойного щелчка отобрать поля, которые будут содержаться в создаваемом запросе. Если в запросе используются поля из нескольких таблиц или запросов, то эту операцию отбора полей следует повторить нужное число раз. Отметим, что мастер не разрешит использовать таблицы, между которыми не были установлены связи.

Затем нужно выбрать тип создаваемого запроса. Если нужно, чтобы отображались все записи, щелкните на переключателе **Подробный**. Для вывода только общей информации о записях щелкните на переключателе **Итоговый** и затем по кнопке **Итоги**. После этого укажите, какие итоговые значения необходимо вычислять, и щелкните по кнопке **ОК**.

В последнем диалоговом окне нужно ввести имя запроса и выбрать дальнейшие действия: просмотр результатов выполнения запроса в режиме таблицы или изменение его структуры в режиме конструктора. Затем следует щелкнуть по кнопке **Готово**.

Использование режима Конструктор

Больше возможностей для создания запроса предоставляет конструктор запросов. Чтобы создать запрос с его помощью, нужно выбрать пункт **Конструктор** в диалоговом окне **Новый запрос**. На экране появится окно **Запрос на выборку**, а поверх него окно **Добавление таблицы** со списком таблиц, хранящихся в текущей БД. По мере того как вы будете выбирать нужные таблицы, Access помещает списки их полей в верхней части окна запроса. Для выбора таблицы достаточно дважды

щелкнуть мышью по ее имени или щелкнуть сначала по ее имени, а затем по кнопке **Добавить**. Если в создаваемом запросе вы хотите использовать поля из ранее созданного запроса, то нужно щелкнуть по ярлычку **Запросы** или **Запросы и таблицы**. В этом случае список таблиц будет заменен или, соответственно, расширен списком уже существующих запросов, из которого можно выбрать нужный запрос.

После завершения отбора нужных таблиц и запросов следует нажать кнопку **Закреть**. Чтобы снова открыть окно **Добавление таблицы**, нужно щелкнуть по кнопке **Добавить таблицу** , расположенной на панели инструментов, или выбрать команду **Добавить таблицу** в меню **Запрос**.

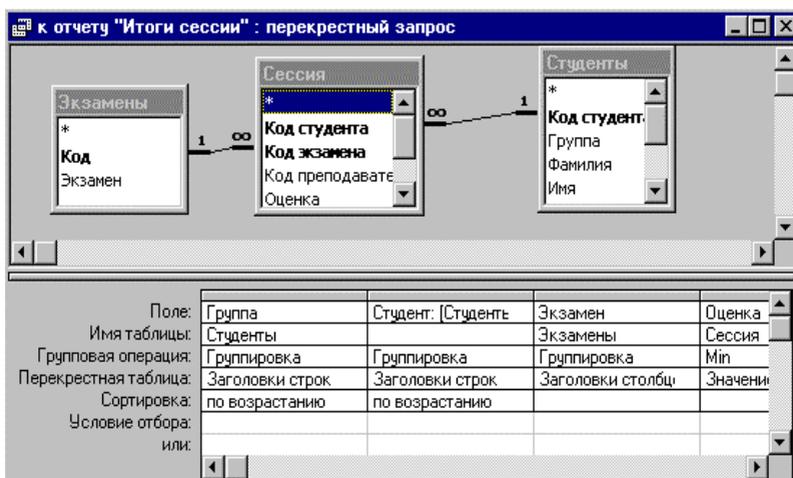


Рис. 5.1. Окно запроса в режиме конструктора

Если вы включили в запрос несколько таблиц, и между ними были установлены связи, то Access автоматически показывает их, рисуя *линии объединения* между связанными полями. Линию объединения можно создать самостоятельно, переместив ключевое поле главной таблицы на связываемое с ним поле подчиненной таблицы.

5.2. Работа в окне конструктора запросов

Окно конструктора запросов разделено на две части (см. рис. 5.1). Верхняя часть содержит списки полей таблиц или запросов, включенных в данный запрос. В нижней части находится бланк запроса, заполняя который вы указываете параметры создаваемого запроса. Каждый столбец бланка запроса соответствует одному из полей запроса, которое может быть полем из списка, а также вычисляемым или итоговым полем.

Для каждого поля можно указать в соответствующих строках бланка порядок сортировки записей по этому полю; будет ли оно включено в динамический набор — результат запроса, а также можно определить условия отбора его значений. Внизу окна находится полоса прокрутки полей запроса.

Бланк запроса содержит еще две строки, которые изначально обычно не видны на экране. Это строки **Имя таблицы** и **Групповая операция**. Чтобы отобразить их на экране, нужно выбрать в меню **Вид** соответствующие команды.

5.2.1. Добавление или удаление таблицы/запроса

Чтобы добавить таблицу или запрос в окно конструктора, достаточно щелкнуть правой кнопкой мыши в пустом месте верхней части окна запроса и выбрать в контекстном меню пункт **Добавить таблицу**. Затем выбрать в появившемся окне **Добавление таблицы** нужную таблицу или запрос.

В окне конструктора запросов не должны находиться посторонние (не используемые в создаваемом запросе) таблицы. Даже если такая таблица не связана с другими таблицами, само ее наличие в окне конструктора приведет к созданию неверного запроса.

Для удаления таблицы или запроса из окна конструктора нужно выделить удаляемую таблицу или запрос, щелкнув по списку полей и нажать клавишу **Delete**. Поля, добавленные в бланк путем переноса с помощью мыши из списка полей, также будут удалены из запроса.

5.2.2. Включение полей в бланк запроса

В бланк запроса следует включать только те поля, данные из которых выводятся на просмотр, обновляются или используются для задания условий отбора, группировки и сортировки.

Чтобы включить поле из списка полей таблицы или запроса в бланк запроса, достаточно дважды щелкнуть мышью по этому полю. Можно также перетащить его в нужное место бланка.

Для добавления сразу нескольких полей нужно выделить в списке эти поля, держа нажатой клавишу **Ctrl**, и затем перенести всю выделенную группу с помощью мыши в бланк запроса. Если добавляемые поля в списке идут подряд, то для их выделения можно, держа нажатой клавишу **Shift**, щелкнуть мышью по первому полю, а затем по последнему полю.

Если в запрос нужно включить все поля из списка, то в этом случае можно просто перенести в бланк запроса символ звездочки (*), расположенный над списком полей. Преимущество этого способа состоит в том, что при использовании * изменения в структуре полей базовой

таблицы автоматически отражаются в запросе, созданном таким способом. В частности, все поля, удаленные из базовой таблицы, будут автоматически удалены из запроса.

Для полей, добавленных в запрос с помощью *, нельзя задавать условия отбора или сортировки. В этом случае следует явно добавить требуемые поля в запрос, отменив для них вывод на экран.

5.2.3. Добавление столбца в бланк запроса

Чтобы вставить пустой столбец в бланк запроса, нужно выбрать любое место в столбце, слева от которого требуется добавить новый столбец, и выбрать пункт **Столбцы** в меню **Вставка**.

5.2.4. Удаление столбца из бланка запроса

Нужно выделить удаляемый столбец. Для этого следует установить курсор мыши в область выделения (над заголовком поля), добиться, чтобы он принял форму жирной направленной вниз стрелки, и щелкнуть мышью. Удаляемый столбец окрасится в черный цвет. Затем нужно нажать клавишу **Delete**. Для удаления нескольких идущих подряд столбцов достаточно распространить на них выделение перемещением мыши с нажатой левой кнопкой и затем нажать клавишу **Delete**. Можно также использовать команду **Удалить столбцы** в меню **Правка**.

5.2.5. Перемещение поля в бланке запроса

Включите режим **Сдвиг** одновременным нажатием клавиши **Ctrl** и **F8**. Затем выделите поле, которое требуется переместить. Для перемещения выделенного поля используются клавиши ← и →. Чтобы выключить режим **Сдвиг**, нужно нажать на клавишу **Esc**. Аналогично можно переместить группу идущих подряд полей.

5.2.6. Добавление вычисляемого поля

В бланк запроса можно добавить вычисляемые поля. *Вычисляемые поля* — это временные поля, создаваемые в динамическом наборе записей при выполнении запроса. В них заносятся результаты вычислений над данными из полей таблиц, включенных в запрос. Правила вычисления значения поля задаются с помощью выражения.

Для создания вычисляемого поля нужно щелкнуть на пустой ячейке строки **Поле**. Затем следует ввести в эту ячейку имя создаваемого поля, двоеточие и выражение, вычисляемое в этом поле. Если элементом выражения является поле, его имя нужно заключить в квадратные скобки. Арифметические операторы записываются обычным образом, а для объединения текстовых значений нужно использовать оператор **&**.

После ввода выражения следует щелкнуть за пределами ячейки. Если Access обнаружит ошибку, то будет выведено соответствующее сообщение.

Для модификации выражения нужно щелкнуть по содержащей его ячейке и внести изменения. Чтобы просмотреть введенное выражение целиком, нужно нажать клавиши **Shift+F2** или щелкнуть правой кнопкой мыши по содержащей его ячейке и выбрать в контекстном меню пункт **Масштаб**. Содержимое ячейки появится в отдельном окне **Область ввода**.

Примеры вычисляемых полей:

ФИО: [Фамилия] & " " & [Имя] & " " & [Отчество];

*Стоимость заказа: [Цена] * [Количество];*

Год рождения: Year([Дата рождения]);

*Цена со скидкой: [Цена]*0,9.*

Если при создании вычисляемого поля не указать его имя, а ограничиться только вводом выражения, то Access сам даст имя новому полю. Это имя можно легко изменить (см. п. 5.2.7).

При создании сложных выражений следует использовать построитель выражений. Для вызова построителя нужно щелкнуть правой кнопкой мыши на ячейке Поле, куда необходимо поместить вычисляемое поле и выбрать пункт **Построить**, или щелкнуть по ячейке Поле, а затем по кнопке **Построить**  на панели инструментов. Работа в окне построителя выражений описана в главе 4.

5.2.7. Изменение имени поля в запросе

Поле запроса можно переименовать, чтобы точнее описать его содержимое. Обычно эта операция производится при определении нового вычисляемого поля, а также при расчетах различных итоговых значений по данным существующего поля. В таких случаях Access по умолчанию использует имена типа «*ВыражениеN*» или «*Sum_Количество*», которые можно легко заменить более содержательными именами.

Чтобы переименовать поле, включенное в запрос, нужно установить указатель слева от первой буквы имени этого поля и ввести новое имя, а затем двоеточие. Например, для замены старого имени поля **Дата рождения** на новое имя **Год рождения** нужно ввести в строку **Поле** соответствующего столбца бланка запроса выражение *Год рождения: Дата рождения*.

Для изменения имени вычисляемого или итогового поля нужно просто заменить расположенное левее двоеточия старое имя на новое имя.

Другим способом изменения имени поля является указание нового имени в свойстве **Подпись** данного поля.

При изменении имени поля Access покажет новое имя как заголовок столбца запроса в режиме таблицы. Новое имя будет также использовано в любой форме (отчете), созданной на основе этого запроса.

5.2.8. Отображение/скрытие поля в результатах запроса

В бланк запроса можно добавить поле, используемое для задания условия отбора или порядка сортировки, без отображения данных из этого поля в результатах запроса. Для этого нужно установить флажки в строке *«Вывод на экран»* бланка для тех полей, которые требуется отображать, и снять флажки для неотображаемых полей. Последние поля не будут включены в результирующий динамический набор.

5.2.9. Изменение ширины столбца в бланке запроса

Перемещайте указатель к границе области выделения столбца, ширину которого нужно изменить, до тех пор пока указатель не примет вид двусторонней стрелки. Затем переместите границу столбца в нужном направлении.

Чтобы быстро установить оптимальную ширину столбца (по размеру данных), достаточно установить указатель на правую границу изменяемого столбца и дважды щелкнуть мышью. Если было выделено несколько столбцов, то после выполнения этой операции для каждого из них будет установлена оптимальная ширина.

5.2.10. Сортировка результатов запроса

Обычно результаты запроса появляются на экране в том же порядке, в котором записи находились в базе данных. Для удобства просмотра их можно отсортировать по алфавиту или в числовом порядке в зависимости от типа поля сортировки.

Чтобы выбрать порядок сортировки, установите указатель мыши в ячейку **Сортировка** под полем, значения в котором нужно отсортировать, щелкните мышью и раскройте появившийся список. Вы можете выбрать из этого списка тип сортировки *«По возрастанию»* (0 – 9, А – Я) или *«По убыванию»* (9 – 0, Я – А).

Сортировку можно задать сразу для нескольких полей. В этом случае порядок сортировки определяется их положением в бланке запроса. Сначала сортируются данные в крайнем левом поле и далее слева направо.

5.2.11. Вставка или удаление строки условий отбора

Для того чтобы вставить строку условий отбора в бланк запроса, нужно щелкнуть мышью по строке, над которой должна появиться новая строка, а затем выбрать в меню **Вставка** пункт **Строки**.

Для удаления строки условий отбора нужно щелкнуть мышью по удаляемой строке, а затем выбрать в меню **Правка** пункт **Удалить строки**.

5.2.12. Установка свойств запроса

Каждый запрос имеет свойства, которые определяют его поведение и вид полученного результата. Чтобы задать свойства запроса, нужно вызвать *бланк свойств*. Для этого следует щелкнуть правой кнопкой мыши, установив ее указатель в окне запроса вне бланка запроса и списков полей, и выбрать в контекстном меню пункт **Свойства**. На экране появится бланк свойств запроса. Ниже описаны некоторые из них.

а) *Вывод всех полей*

Указывает, какие поля должны быть включены в результат запроса: все поля из базовых таблиц или только те, для которых установлен флажок **Вывод на экран** в бланке запроса. Если вы создаете запрос для формы и хотите, чтобы все поля из всех таблиц были в ней доступны, то установите для свойства **Вывод всех полей** значение «Да».

б) *Набор значений*

Указывает, сколько записей должно быть возвращено запросом. Значением свойства **Набор значений** является целое число или число процентов. Например, для вывода 10 первых записей в качестве значения свойства **Набор значений** следует ввести число 10, а для вывода 10 процентов от общего числа записей — 10%.

Обычно это свойство используется, если нужен не весь результат запроса, а лишь записи, имеющие самые большие или самые маленькие значения в данном поле. В этом случае в ячейке **Сортировка**, соответствующей этому полю, следует выбрать «по убыванию», если нужно отобразить наибольшие значения, и «по возрастанию» — при выводе на экран наименьших значений.

в) *Уникальные значения*

Указывает, следует ли возвращать повторяющиеся значения полей. Если значение этого свойства «Да», то Access возвращает только те записи, у которых значения всех полей, изображаемых в режиме таблицы, являются уникальными. Если значение «Нет», то возвращаются все записи (используется по умолчанию).

5.2.13. Установка свойств полей запроса

Кроме свойств запроса можно задать свойства любого из его полей, кроме * и полей, для которых не установлен флажок **Вывод на экран**. Для вызова бланка свойств поля следует щелкнуть по нему правой кнопкой мыши, а затем выбрать пункт **Свойства** в контекстном меню.

Чаще всего бланк свойств поля запроса используется для изменения формата изображения данных в запросе. По умолчанию поле в запросе наследует формат соответствующего ему поля базовой таблицы. Чтобы

изменить этот формат, нужно щелкнуть по ячейке **Формат поля**, открыть список стандартных форматов и выбрать нужный формат или задать пользовательский формат.

В качестве формата для текстового поля можно ввести символ > или <, чтобы его содержимое выводилось прописными или строчными буквами. Для вывода дополнительных символов в ячейку **Формат поля** следует ввести @ и нужные символы.

С помощью задания соответствующего формата можно обеспечить вывод сообщения в том случае, когда в поле отсутствует значение. Например, чтобы при выводе сведений о преподавателях в случае пустого (*Null*) значения в поле **Домашний телефон** появлялось слово *неизвестно*, а в случае пустой строки ("") слово *нет*, следует задать формат: @; "неизвестно"; "нет".

Для поля типа *Дата/время* можно, в частности, использовать следующие форматы вывода его значений:

<i>ddd</i>	—	сокращенное название дня недели;
<i>dddd</i>	—	полное название дня недели;
<i>mmm</i>	—	сокращенное название месяца;
<i>mmmm</i>	—	полное название месяца;
<i>q</i>	—	номер квартала в году;
<i>yy</i>	—	последние две цифры номера года;
<i>yyyy</i>	—	полный номер года.

Свойство **Подпись** позволяет определить новое имя поля. Оно заменит при просмотре результатов запроса старое имя. Новое имя будет также использовано при создании формы (отчета), основанной на этом запросе.

Пример 5.1. Для того чтобы при отборе записей из таблицы **Студенты** появлялась информация не о дате, а о годе рождения, достаточно внести следующие изменения в свойства поля **День рождения**: в ячейку **Формат поля** ввести формат даты «*yyuu*», а в ячейку **Подпись** — текст «*Год рождения*».

5.2.14. Предварительный просмотр результатов запроса

Чтобы просмотреть результаты запроса, находясь в режиме конструктора, нужно выбрать в меню **Вид** пункт **Режим таблицы** или нажать кнопку **Вид**  на панели инструментов. На экране появится созданный динамический набор данных.

Чтобы снова вернуться в режим конструктора, нужно нажать кнопку **Вид** на панели инструментов или выбрать в меню **Вид** пункт **Конструктор**.

5.2.15. Выполнение запроса

Чтобы выполнить ранее созданный запрос, нужно найти его в списке запросов, находящихся во вкладке **Запросы** окна БД, и сделать двойной щелчок мышью по его значку или имени.

Для выполнения запроса, находящегося в окне конструктора, нужно нажать кнопку **Запуск**  на панели инструментов. Чтобы остановить выполнение запроса, следует нажать клавиши **Ctrl+Break**.

5.2.16. Сохранение запроса

Для сохранения запроса нужно выбрать команду **Сохранить** в меню **Файл** или нажать кнопку **Сохранить**  на панели инструментов. Если сохраняется новый запрос, Access автоматически предлагает для него стандартное имя. Лучше заменить его более содержательным именем и нажать кнопку **ОК**. Сохраненный запрос становится частью БД. Его имя не должно совпадать с именем какой-либо таблицы БД.

5.3. Отбор записей в однотабличных запросах

Обычно в результате выполнения запроса нужно найти только те записи, которые удовлетворяют некоторым условиям отбора. Ввод условия отбора в запросе похож на ввод условия на значение для поля (см. соответствующий пункт главы 2). Сначала рассмотрим случай, когда в создании запроса участвует одна таблица.

5.3.1. Точное совпадение значений полей

Простейший критерий отбора предполагает извлечение из исходной таблицы всех записей, имеющих одинаковые значения в одном или нескольких полях. В этом случае для создания соответствующего условия отбора нужно включить в бланк запроса поля, значения в которых должны совпадать, и напечатать в них значения-шаблоны в строке **Условие отбора**. Тем самым будет создан образец, с которым Access при выполнении запроса будет сравнивать все записи исходной таблицы. В результирующий динамический набор данных попадут лишь те записи, значения соответствующих полей которых совпали со значениями полей этого образца.

При вводе текстовых значений регистр символов несущественен. После завершения ввода текста Access автоматически заключает его в кавычки.

Пример 5.2. Требуется извлечь из таблицы **Студенты** сведения о девушках группы 9701. Для этого достаточно перенести в первое поле бланка запроса звездочку *, задающую вывод всех полей. В поле **Группа**

бланка запроса нужно ввести в строке **Условия отбора** значение-шаблон «9701», а в поле **Пол** — значение-шаблон "ж" и отменить вывод на экран содержимого этих полей. Созданный бланк запроса будет иметь следующий вид (см. рис. 5.2).

Поле:	Студенты.*	Группа	Пол
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора: или:		"9701"	"ж"

Рис. 5.2. Пример бланка запроса с использованием шаблонов

5.3.2. Шаблоны общего вида. Оператор Like

Часто условием отбора записей является не полное, а частичное совпадение значений в указанных полях исходной таблицы, например, если значения в текстовом поле содержат заданные символы. В этом случае при записи в строке условий соответствующего шаблона используются уже известные спецсимволы ?, # и *.

Условие отбора имеет следующий вид: *Like* <шаблон>. Как правило, вводить оператор *Like* не нужно. В большинстве случаев Access самостоятельно добавит его после завершения ввода строки, содержащей любой из вышеприведенных символов шаблона.

Таблица 5.5. Примеры условий с шаблонами

Условие	Комментарий
Like "п*в"	любой текст, начинающийся с буквы «п» и заканчивающийся буквой «в»;
Like "*запрос*"	любой текст, содержащий слово «запрос»;
Like "1?.05.72"	любая дата между 10 и 19 мая 72 года;
Like "*?.01.98"	любая дата в январе 98 года;
Like "*?.03.*"	любая дата в марте;
Like "*?.*.98"	любая дата в 98 году;
Like "*?.0[4-6].00"	любая дата во втором квартале 2000 года.

Например, чтобы получить информацию о студентах, фамилия которых начинается с буквы М, нужно ввести в поле **Фамилия** два символа: "м" и "*". После нажатия клавиши **Enter** Access заключит введенные символы в кавычки и добавит слева оператор *Like*. В итоге условие отбора нужных записей примет следующий вид: *Like "м*"*.

Оператор *Like* обычно используется при задании шаблонов для текстовых или Мемо полей. Однако с его помощью можно создавать шаблоны для отбора записей по содержимому поля даты (см. табл. 5.5).

5.3.3. Диапазон значений. Операторы *And* и *Between*

Иногда возникает необходимость отобрать записи, у которых значения в указанном поле лежат в некотором диапазоне значений. Диапазон значений можно определить, если перед значением указать один из операторов сравнения (см. гл. 4), например, > 100 или ≤ 10 .

Операторы сравнения могут использоваться в текстовых и числовых полях, а также в полях дат. Если диапазон имеет две границы, то оба граничных условия должны быть помещены на одной строке и разделены логическим оператором *And*.

Пример 5.3. Допустим, что нужно получить информацию о студентах, родившихся в первой половине августа 1972 года. В этом случае в поле **Дата рождения** бланка запроса нужно ввести условие $\geq \#01.08.72\#$ *And* $\leq \#15.08.72\#$.

Другим способом задания диапазона значений является использование оператора *Between*. Например, с его помощью приведенное выше условие отбора можно записать так: *Between* $\#01.08.72\#$ *And* $\#15.08.72\#$.

5.3.4. Список значений. Операторы *Or* и *In*

Для проверки содержится ли значение в заданном списке, нужно использовать логический оператор *Or* или оператор *In*. Если список невелик (не более 9 элементов), то требуемый результат можно получить, введя каждое значение из списка в отдельной ячейке строки условий **или**.

Пример 5.4. Для отбора записей таблицы **Студенты**, содержащих сведения о студентах групп 9701, 9702 и 9705, нужно в поле **Группа** бланка запроса ввести условие "9701" *Or* "9702" *Or* "9705". Это условие можно заменить условием *In* ("9701";"9702";"9705") либо ввести каждое из названий групп в отдельной строке условий **или**.

5.3.5. Отрицание образца. Оператор *NOT*

Чтобы найти записи, которые не удовлетворяют определенному условию, перед условием выбора нужно поставить логический оператор *Not*. Этот оператор можно использовать вместе со всеми другими рассмотренными выше условиями выбора.

Пример 5.5. Чтобы получить информацию обо всех студентах кроме тех, кто учится в группе 9701, в поле **Группа** бланка запроса к таблице **Студенты** надо ввести текст *not 9701*.

5.3.6. Отбор записей с Null значениями и пустыми строками

Иногда могут понадобиться записи, в полях которых отсутствуют значения. Вы можете найти записи в таблице, не содержащие значения в определенном поле, если наберете слово *null* в этом поле в бланке запроса. Access создаст условие отбора вида *Is Null*, и в таблицу попадут только те записи, которые не содержат значения (имеют пустое значение) в этом поле. Соответственно, вы можете найти только те записи, которые содержат значения в определенных полях, если наберете *not null (Is Not Null)* в этих полях в бланке запроса.

Пример 5.6. Чтобы определить, какие студенты получили (не получили) стипендию в сентябре, нужно создать запрос к таблице **Стипендия** и ввести в поле **Сентябрь** бланка запроса в качестве условия *Is Not Null (Is Null)*.

В том случае, когда поле таблицы может содержать пустые строки (""), для выборки записей, содержащих в этом поле пустую строку, нужно ввести в качестве условия отбора две идущих подряд кавычки "".

Пример 5.7. Описанные ниже варианты условий относятся к запросам, извлекающим информацию из таблицы **Преподаватели**.

1. Для получения списка преподавателей, не имеющих домашнего телефона (см. пример 2.1 на стр. 21), нужно ввести в поле **Домашний Телефон** в качестве условия отбора пустую строку ("").

2. Для получения списка преподавателей, о которых неизвестно, имеют ли они домашний телефон, нужно ввести в поле **Домашний Телефон** в качестве условия отбора слово *null*.

3. Для получения списка преподавателей, имеющих домашний телефон, нужно ввести в поле **Домашний Телефон** в качестве условия отбора текст *Is Not Null And <>""*.

5.3.7. Использование нескольких строк условий

При создании сложных запросов может возникнуть ситуация, когда описание критерия отбора занимает не одну, а несколько строк в бланке запроса. В этом случае Access последовательно анализирует каждую из строк и включает в итоговый набор те записи, для которых эти условия выполняются. Если строка содержит несколько ячеек с условиями отбора, то в набор попадают лишь те записи, которые удовлетворяют одновременно всем условиям в этой строке. В результате в итоговый набор будут включены все записи, удовлетворяющие хотя бы одной из строк условий.

Пример 5.8. Предположим, что нам нужно получить списки студентов групп 9701 и 9703, родившихся в 1972 году. Это можно сделать, создав запрос следующего вида:

Первая строка условия отбора включает в результат запроса список студентов 9701 группы, родившихся в 1972 году, а вторая строка — список студентов 9703 группы того же года рождения. Задание сортировки позволяет получить итоговый список, отсортированный сначала по группам, а внутри каждой группы — по фамилиям студентов. Отметим, что условие *Like* "*"*.72" необходимо указать в обеих строках, так как, если его не повторить во второй строке, то в итоговый список попадут все студенты 9703 группы.

Поле:	Группа	Фамилия	Имя	Дата рождения
Имя таблицы:	Студенты	Студенты	Студенты	Студенты
Сортировка:	по возрастанию	по возрастанию		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	"9701"			Like "*"*.72"
или:	"9703"			Like "*"*.72"

Рис. 5.3. Запрос, использующий две строки условий

В данном примере получить такой же результат можно, используя лишь одну строку условия отбора и один из операторов *Or* или *In*. Например, в поле **Группа** можно ввести условие "9701" *Or* "9703", а в поле **Дата рождения** — условие *Like* "*"*.72". Однако если поставлена задача получить списки студентов этих групп, родившихся в разные годы, то соответствующий запрос к таблице **Студенты** обязательно будет содержать более одной строки условий.

5.3.8. Использование вычисляемых полей

В ряде случаев для отбора записей необходимо включить в запрос вычисляемое поле и ввести условие, использующее значения этого поля.

Пример 5.9. Таблица **Заказы** содержит сведения о заказанных товарах. В частности, поле **Количество** содержит информацию о числе заказанных товаров, а поле **Цена** — цену товара. Чтобы выбрать из таблицы сведения о заказах, стоимость которых не ниже 500 руб., следует включить в запрос нужные поля, а также создать вычисляемое поле *Стоимость: [Количество]*[Цена]* и ввести в него условие ≥ 500 .

5.3.9. Использование ссылки на имя поля

Иногда необходимо использовать значение одного поля таблицы в качестве условия отбора для другого поля. При создании соответствующего выражения нужно заключить имя поля в квадратные скобки.

Пример 5.10. Таблица **Заказы** содержит поля **Дата получения** и **Дата отправки**. Чтобы найти все заказы, в которых от получения до отправки заказа прошло не более трех дней, следует в поле **Дата отправки** ввести условие $\leq [Дата\ получения] + 3$.

5.3.10. Использование ссылки на элемент управления

В условии отбора можно сослаться на текущее значение элемента управления открытой формы.

Пример 5.11. Пусть форма **Список жильцов** содержит поле с именем **Номер комнаты**, значениями которого являются номера комнат в студенческом общежитии. Чтобы создать запрос, выбирающий из таблицы **Общежитие** коды студентов, живущих в данной комнате, можно поступить так: поместить в бланк запроса поля: **Код студента** и **Комната**, затем ввести в поле **Комната** условие

Forms![Список жильцов]![Номер комнаты].

Если открыть форму **Список жильцов** и выбрать номер какой-либо комнаты, а затем запустить на выполнение созданный запрос, то на экране появятся коды студентов, живущих в этой комнате.

Если информация для запроса содержится в поле со списком **Номер комнаты**, то условие отбора запроса должно иметь следующий вид:

Forms![Список жильцов]![Номер комнаты].[Text].

5.3.11. Использование параметров в условиях отбора

В условиях отбора можно использовать не только конкретные значения, но и параметры. Перед выполнением запроса, содержащего один или несколько параметров, Access каждый раз будет запрашивать конкретные условия отбора и затем произведет отбор нужной информации. Используя параметры, можно заменить несколько однотипных запросов к данным одним запросом.

Чтобы задать параметр, нужно ввести в строку **Условие отбора** вместо конкретного значения произвольный текст, заключенный в квадратные скобки []. Этот текст Access рассматривает как имя параметра и выводит его в специальном диалоговом окне при выполнении запроса. Поэтому в качестве имени параметра лучше использовать содержательную фразу, причем имя каждого параметра должно быть уникальным.

Для каждого параметра запроса можно указать тип данных. Эту информацию Access использует для проверки введенного значения. Например, если параметр определен как числовой, то Access не разрешит

ввести значение, содержащее буквы. По умолчанию все параметры запроса имеют текстовый тип.

Для задания типа данных параметра нужно выбрать пункт меню **Запрос**, а затем **Параметры** или вызвать щелчком правой кнопки мыши контекстное меню запроса и выбрать в нем пункт **Параметры**. **Access выведет на экран** диалоговое окно **Параметры запроса**. В этом окне нужно ввести в столбце **Параметр** имя того параметра, тип которого следует изменить, точно в том же виде, в каком оно было введено в бланк запроса, но без квадратных скобок. В столбце **Тип данных** следует выбрать из раскрывающегося списка нужный тип данных параметра. После того как определены типы всех параметров, нужно нажать кнопку **ОК**.

При выполнении запроса Access попросит ввести одно за другим значения для каждого из параметров, используя для ввода диалоговые окна **Введите значение параметра**. После ввода всех значений запрос будет выполнен и на экране появится результирующий набор записей.

Поле:	Группа	Фамилия	Имя
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора: или:	[Введите номер группы]		

Рис. 5.4. Пример запроса с параметром

Пример 5.12. Чтобы получить список студентов любой учебной группы, используем запрос с параметром. Для его создания нужно включить в бланк запроса к таблице **Студенты** поля **Группа**, **Фамилия** и **Имя**. В строке **Условие отбора** в поле **Группа** следует ввести текст *Введите номер группы* и заключить его в квадратные скобки (**см. рис. 5.4**).

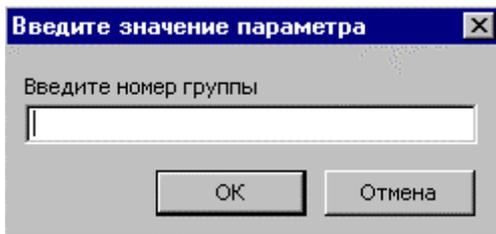


Рис. 5.5. Окно для ввода параметра

При выполнении запроса появится диалоговое окно с приглашением ввести номер группы (**см. рис. 5.5**). После ввода номера и нажатия кнопки **ОК** Access выполнит запрос и создаст требуемый список группы.

Параметры могут использоваться в условиях отбора совместно с шаблонами. Например, для нахождения всех студентов,

фамилия которых начинается с данной буквы, нужно включить в бланк запроса к таблице **Студенты** в поле **Фамилия** условие *Like [введите букву] & "*"'*.

5.3.12. Использование в условиях отбора подчиненного запроса

Иногда условие отбора использует значение, которое можно получить в результате выполнения другого запроса, называемого *подчиненным*. В этом случае можно поступить следующим образом:

1. Создать подчиненный запрос.
2. Перейти в режим SQL и скопировать инструкцию SQL, описывающую подчиненный запрос в буфер обмена.
3. Создать основной запрос и перейти в нужную ячейку строки **Условие отбора**.
4. Ввести требуемый условный оператор (например, > или <), а затем вставить содержимое буфера обмена, заключив его в круглые скобки.

Отметим, что подчиненный запрос должен иметь в качестве результата значение, т.е. одну запись, содержащую одно поле. Подчиненный запрос может использоваться в выражении, определяющем вычисляемое поле.

Пример 5.13. Пусть нужно найти студентов, которые не старше студента 9702 группы Иванова, причем известно, что у него в группе нет однофамильцев. В этом случае можно создать подчиненный запрос (см. рис. 5.6) к таблице **Студенты**, результат которого — дата рождения Иванова (27 марта 72 года)¹.

Поле:	Группа	Фамилия	Дата рождения
Сортировка:			
Вывод на экран:	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	"9702"	"иванов"	

Рис. 5.6. Подчиненный запрос, дающий дату рождения Иванова

Соответствующая инструкция SQL SELECT выглядит так:

```
SELECT Студенты.[Дата рождения] FROM Студенты WHERE
(((Студенты.Группа) = «9702») AND ((Студенты.Фамилия) = «иванов»));
```

Она содержит следующую информацию:

- список выбираемых полей (находится после ключевого слова SELECT);

¹ Для нахождения даты рождения Иванова можно также использовать функцию *DLookup*.

- таблицу, используемую в запросе (указывается в предложении FROM);
- условие отбора записей (указывается в предложении WHERE).

Поле:	Фамилия	Имя	Дата рождения
Сортировка:	по возрастанию		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			>={SELECT Студенты.[Дата
или:			

Рис. 5.7. Основной запрос, использующий инструкцию SQL

Эту инструкцию нужно скопировать в буфер обмена и создать еще один запрос к таблице **Студенты**, содержащий поля **Фамилия**, **Имя** и **Дата рождения**. В поле **Дата рождения** следует вставить из буфера обмена сохраненную инструкцию SELECT, заключив ее в круглые скобки и поставив в начале условия знак >=.

Условие отбора в поле **Дата рождения** будет выглядеть так:

```
>= (SELECT Студенты.[Дата рождения] FROM Студенты WHERE
(((Студенты.Группа)="9702") AND ((Студенты.Фамилия) ="иванов"));
```

Фамилия	Имя	Дата рождения
Амосов	Дмитрий	26 сентября 1974 г.
Болотов	Константин	24 июля 1974 г.
Бродников	Александр	24 июля 1974 г.
Брянская	Надежда	11 мая 1974 г.
Бутаков	Андрей	9 августа 1974 г.
Вохмякова	Любовь	19 сентября 1974 г.

Запись: 1 из 19

Рис. 5.8. Итоговый список студентов

В результате выполнения этого запроса будет получен требуемый список студентов (см. рис. 5.8). Отметим, что этот запрос даст правильный результат лишь в том случае, если у Иванова нет однофамильцев. Если это условие не выполнено, то необходимо использовать для отбора даты рождения не фамилию, а код студента.

5.4. Многотабличные запросы

Часто возникает необходимость в создании запроса, отбирающего информацию из нескольких таблиц или ранее сохраненных запросов. В этом случае списки полей всех используемых таблиц/запросов должны быть включены в окно конструктора создаваемого запроса. Процедура добавления таблиц/запросов в запрос, создаваемый в режиме конструктора, была описана выше (см. стр. 72).

Далее рассматриваются различные типы установления связи (объединения) таблиц/запросов, находящихся в окне конструктора, и соответствующие им принципы отбора записей в запросе. Задание связей между таблицами — важный этап конструирования запроса и сделанная на этом этапе ошибка, как правило, приводит к тому, что запрос дает неверный результат.

Отметим, что обычно Access устанавливает связи добавленной таблицы с другими таблицами автоматически. Это происходит в следующих случаях:

- таблицы были связаны между собой в окне **Схема данных**; информация о связи между ними берется из схемы данных;
- связь (внутреннее объединение) устанавливается между таблицами, имеющими поля с одинаковыми именами, если одно из них ключевое.

Однако нередко возникает ситуация, когда приходится либо задавать связь самостоятельно, либо переопределять тип связи, которую установил Access.

A : таблица	
	a1
	aa
▶	bb
*	

B : таблица	
	b1
	aa
▶	cc
*	

Рис. 5.9. Содержимое таблицы А и В

Рассмотрим основные принципы отбора записей, соответствующие различным типам объединения таблиц, на примере двух таблиц А и В. Каждая из них содержит по одному текстовому полю и две записи (см. рис. 5.9).

5.4.1. Случай несвязанных таблиц

Начнем с рассмотрения ситуации, когда таблицы А и В включены в запрос, но между ними не установлена связь (см. рис. 5.10). В этом случае Access формирует динамический набор данных, записи которого являются декартовым произведением¹ записей исходных таблиц. Его можно вывести

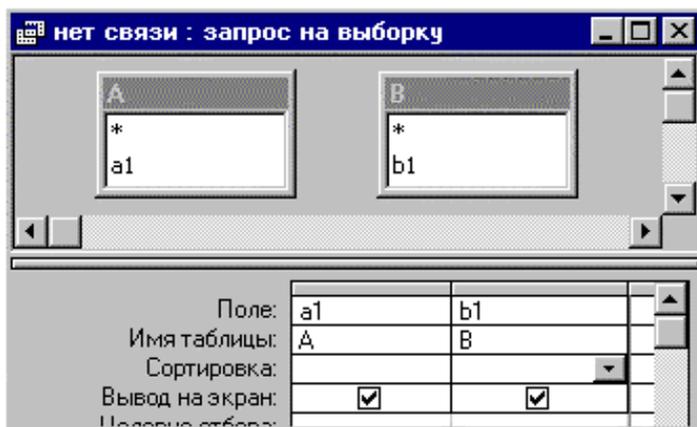


Рис. 5.10. Запрос с несвязанными таблицами

на экран, включив поля обеих таблиц в бланк запроса (см. рис. 5.11). Инструкция SQL, соответствующая этому запросу, имеет вид

SELECT A.a1, B.b1 FROM A, B;

	a1	b1
▶	aa	aa
	bb	aa
	aa	cc
	bb	cc

Рис. 5.11. Результат выполнения запроса

Если включить в бланк запроса поле только одной таблицы, например А (см. рис. 5.12), то результат выполнения запроса будет следующим: итоговый набор содержит записи таблицы А, но они продублированы: их не две, как можно было бы ожидать, а четыре (см. рис. 5.13).

Таким образом, сам факт присутствия в окне конструктора посторонней таблицы (В), не связанной с основной таблицей (А), используемой в запросе, приводит к дублированию отбираемых записей.

Появление лишних записей вызвано тем, что Access в этом случае создает такой же динамический набор, как и в предыдущем запросе (декартово произведение таблиц А и В). Однако на этот раз выводятся на экран не оба

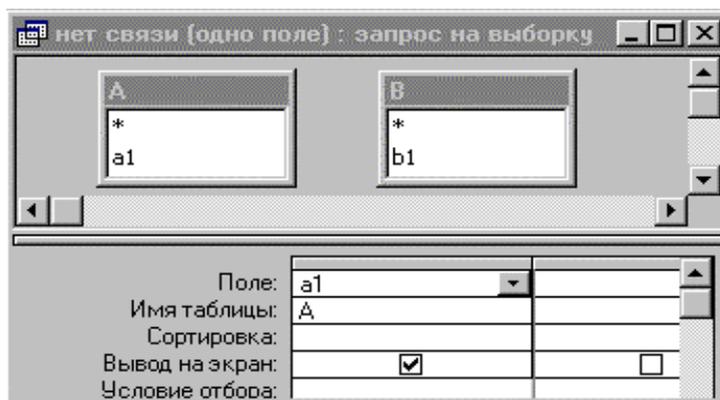


Рис. 5.12. Еще один запрос с несвязанными таблицами

столбца набора, а лишь первый столбец. Ясно, что число дубликатов записей таблицы А в итоговом наборе равно количеству записей, содержащихся в таблице В.

Инструкция SQL для запроса, представленного на рис. 5.12, имеет вид

SELECT A.a1 FROM A, B;

Она отличается от инструкции SQL предыдущего запроса лишь списком выводимых полей.



Рис. 5.13. Результат запроса

Итак, если в окне конструктора находятся две несвязанные таблицы, то Access рассматривает в качестве исходной таблицы их декартово произведение и число обрабатываемых им логических записей равно числу строк первой таблицы, умноженному на число строк второй таблицы.

¹ Декартовым произведением множеств X и Y называется множество Z, состоящее из элементов $z=(x,y)$, где x — элемент множества X, а y — элемент множества Y.

Обычно наличие несвязанных таблиц в окне конструктора — следствие невнимательности пользователя, создающего запрос. Однако можно добавить таблицу в запрос, не связывая ее с другими таблицами, для того чтобы иметь возможность ссылаться на значения ее полей.

Например, для получения списка студентов младше студента Иванова можно поступить следующим образом (см. также пример 5.13):

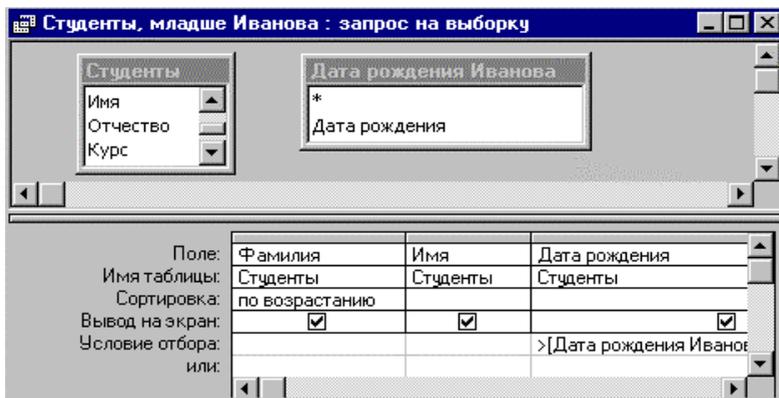


Рис. 5.14. Использование несвязанных таблиц

1. Создать и сохранить запрос, дающий дату рождения Иванова.
2. Поместить в окно конструктора таблицу **Студенты** и сохраненный запрос, не связывая их.
3. Поместить в бланк запроса поля **Фамилия**, **Имя**, **Дата рождения** и в поле **Дата рождения** задать условие отбора:

>[Дата рождения Иванова]![Дата рождения]

4. Выполнить запрос.

Присутствие в окне конструктора запроса **Дата рождения Иванова** позволяет использовать его значение в условии отбора (см. рис. 5.14).

5.4.2. Внутреннее объединение таблиц

Чтобы установить связь между таблицами А и В, нужно переместить поле одной таблицы на связываемое с ним поле другой таблицы. Access покажет установленную связь в виде тонкой линии.

Включим оба связанных поля в бланк запроса и выполним созданный запрос (см. рис. 5.15). Результирующий набор данных содержит лишь одну запись, причем значения полей совпадают (см. рис. 5.16).

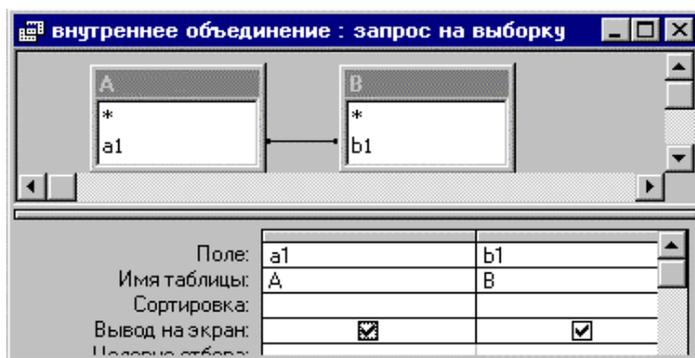


Рис. 5.15. Запрос со связанными таблицами

Мы видим, что при создании связи между таблицами А и В Access объединяет в динамическом наборе записи, имеющие одинаковые значения в связанных полях. Такой тип объединения называется *внутренним*. Конечно, если запрос содержит какие-либо условия на отбор записей, то в итоговый набор попадут лишь те записи, которые удовлетворяют этим условиям.

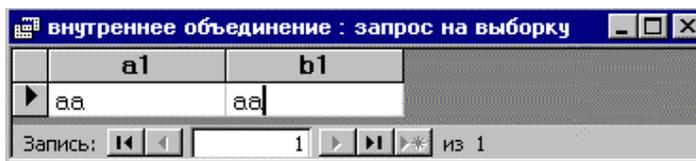


Рис. 5.16. Результат запроса со связанными таблицами

Инструкция SQL для созданного нами запроса имеет вид

```
SELECT A.a1, B.b1 FROM A INNER JOIN B ON A.a1 = B.b1;
```

Предложение FROM в ней выглядит не так, как в случае несвязанных таблиц. В нем добавилась операция INNER JOIN (внутреннее объединение), указаны объединяемые таблицы А и В и правило объединения записей $A.a1 = B.b1$ — совпадение значений в связанных полях.

Итак, для установления связи между таблицами в режиме конструктора запроса нужно переместить с помощью мыши поле из списка полей одной таблицы в соответствующее поле из списка полей другой таблицы. Обычно

«перетаскивается» ключевое поле главной таблицы, выделенное жирным шрифтом, на поле внешнего ключа подчиненной таблицы. Связываемые поля должны иметь одинаковый или совместимый тип данных.

По умолчанию Access использует *внутреннее объединение*, при котором из обеих таблиц отбираются лишь записи, содержащие одинаковые значения в полях связи. Если значения в этих полях одинаковы, запрос объединяет две соответствующие записи из обеих таблиц и отображает их как одну запись в динамическом наборе данных. Если для записей одной таблицы нет соответствующих записей из другой таблицы, то динамический набор будет пустым.

Пример 5.14. Нужно узнать, как студенты 9701 группы сдали экзамен по информатике.

Фамилии студентов хранятся в таблице **Студенты**, а их оценки — в таблице **Сессия**. Поэтому создание запроса нужно начать с включения в окно конструктора этих таблиц. После того как списки полей таблиц появятся в окне конструктора, Access автоматически соединит их линией по полю **Код студента**, так как связь между этими таблицами типа «один-ко-многим» содержится в схеме данных.

Добавим в бланк запроса поле **Группа** из таблицы **Студенты** и поля **Оценка** и **Код экзамена** из таблицы **Сессия**. Для отбора нужных записей введем условие на значения в поле **Группа: 9701** и условие на значения в поле **Код экзамена: 1** (информатика). Отменим вывод этих полей на экран снятием флажков в строке **Вывод на экран**.

Для вывода фамилий и имен студентов создадим вычисляемое поле, введя в любую пустую ячейку строки **Поле** выражение:

Студент: [Студенты].[Фамилия] & " " & [Студенты].[Имя].

Для его создания удобно использовать построитель выражений.

Чтобы список появлялся на экране в отсортированном виде, установим для поля **Студент** в строке **Сортировка** значение *по возрастанию*. На этом создание запроса завершено (см. рис. 5.17).

В том случае, когда две таблицы непосредственно связать нельзя, для их объединения нужно использовать дополнительные таблицы или запросы.

Пример 5.15. Требуется создать запрос, позволяющий определять, у каких студентов принял экзамен тот или иной преподаватель.

Так как таблицы **Студенты** и **Преподаватели** не имеют общих полей, то для установления связи между ними нужно добавить в окно конструктора таблицу **Сессия**, имеющую общие поля с обеими таблицами.

Связь между таблицами **Студенты** и **Сессия** устанавливается по полю Код студента, а между таблицами **Преподаватели** и **Сессия** — по полю **Код преподавателя**.

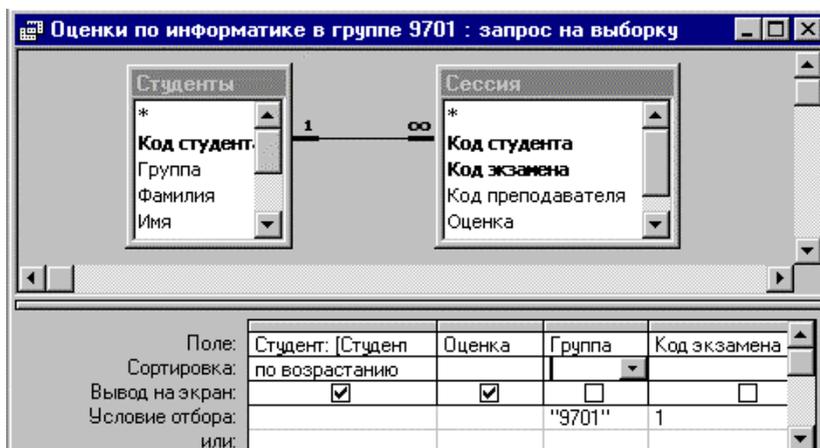


Рис. 5.17. Внутреннее объединение таблиц **Студенты** и **Сессия**

Запрос должен выдавать список студентов для любого преподавателя. Поэтому следует использовать запрос с параметром. Параметр лучше задать в поле **Код преподавателя**, чтобы обеспечить получение правильного результата и в том случае, когда у экзаменатора есть однофамилец. Для этого нужно ввести в этом поле в строке **Условие отбора** текст в квадратных скобках примерно такого содержания: *Введите код преподавателя*. В бланк запроса следует также добавить поле **Фамилия** из таблицы **Преподаватели** и поля **Группа** и **Фамилия** из таблицы **Студенты**. Так как в запрос включены два поля с одинаковым именем, имеет смысл их переименовать, введя перед старым именем новое имя и отделив имена двоеточием. Затем созданный запрос (см. рис. 5.18) нужно сохранить. При его запуске Access попросит ввести код преподавателя и после того как это будет сделано, выдаст на экран фамилию преподавателя и список его студентов.

Созданный запрос обычно можно легко изменить. Например, если возникло желание узнать, какие оценки получили студенты, то нужно открыть запрос в режиме конструктора, добавить в бланк поле **Оценка** и сохранить сделанные изменения.

Access позволяет также объединять таблицы по совпадению значений в нескольких полях связи (см. пример 5.19).

5.4.3. Внешнее объединение таблиц

При внутреннем объединении запись таблицы включается в динамический набор лишь в том случае, если в связанной с ней таблице найдется запись, имеющая совпадающее значение в поле связи. Часто возникает необходимость включить в результат запроса и те (а иногда только те) записи таблицы, для которых в таблице, связанной с ней, отсутствуют соответствующие им записи.

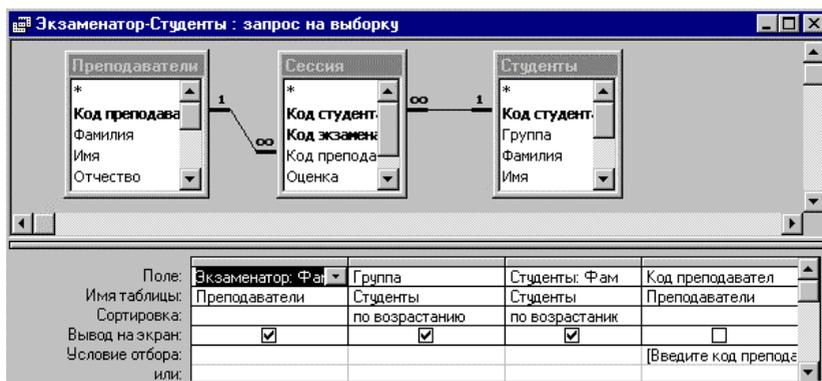


Рис. 5.18. Запрос к примеру 5.15

Пусть, например, нужно создать запрос, содержащий результаты сдачи экзамена в группе, причем итоговый список должен включать и фамилии студентов, по каким-то причинам не сдававших экзамен. Если использовать для связи таблиц **Студенты** и **Сессия** внутреннее объединение, то в итоговый список попадут лишь студенты, сдававшие экзамен (см. пример 5.14). Поэтому нужно использовать другой тип объединения таблиц.

Access позволяет изменить тип объединения таким образом, что из одной таблицы/запроса будут выбраны все записи, независимо от того, содержит ли вторая таблица/запрос соответствующие им записи. Для этого нужно сделать двойной щелчок мышью на линии объединения между таблицами, а в появившемся окне **Параметры объединения** выбрать подходящий тип объединения и нажать кнопку **ОК**.

Например, если мы хотим выбрать все записи таблицы А и те записи таблицы В, которые имеют совпадающие значения в полях связи, нужно выполнить следующие действия:

1. Создать связь между таблицами А и В путем «перетаскивания» поля *a1* на поле *b1*.

2. Двойным щелчком по линии объединения открыть окно **Параметры объединения** и установить нужный тип объединения (см. рис. 5.19).
3. Щелкнуть по кнопке **ОК** и добавить поля *a1* и *b1* в окно запроса (см. рис. 5.20).
4. Щелкнуть по кнопке **Запуск**  и выполнить запрос (см. рис. 5.21).

Для внешнего объединения Access добавляет стрелку в конце линии объединения. В нашем примере стрелка направлена слева направо от таблицы А к таблице В, а выбранный второй тип объединения называется *левым внешним объединением*.

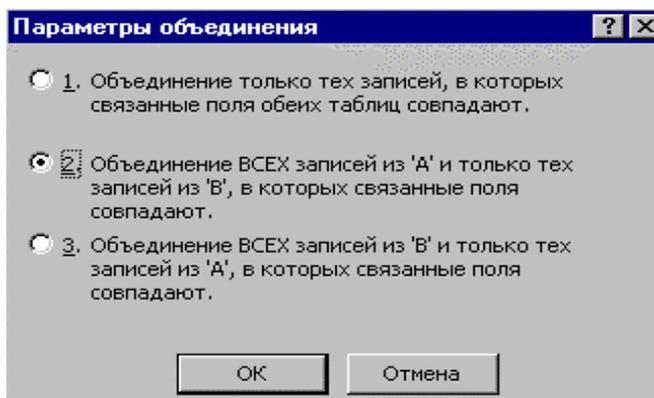


Рис. 5.19. Окно *Параметры объединения*

Если нужно отобразить все записи таблицы В и записи таблицы А, имеющие совпадающие значения в полях связи, то следует выбрать в окне **Параметры объединения** третий тип, называемый *правым внешним объединением*. При выборе этого типа стрелка в конце линии объединения будет направлена справа налево от таблицы В к таблице А.

Левому внешнему объединению (LEFT JOIN) в нашем примере соответствует инструкция SQL:

```
SELECT A.a1, B.b1 FROM A LEFT JOIN B ON A.a1 = B.b1;
```

а правому внешнему объединению (RIGHT JOIN) — инструкция SQL:

```
SELECT A.a1, B.b1 FROM A RIGHT JOIN B ON A.a1 = B.b1;
```

Отметим, что любой из описанных выше типов объединения таблиц может быть задан в окне **Связи** (см. рис. 2.6 на стр. 33) при определении связей между таблицами. Для этого следует щелкнуть по кнопке **Объединение**. Установленный тип связи будет использован по умолчанию Access при объединении таблиц в окне конструктора запросов.

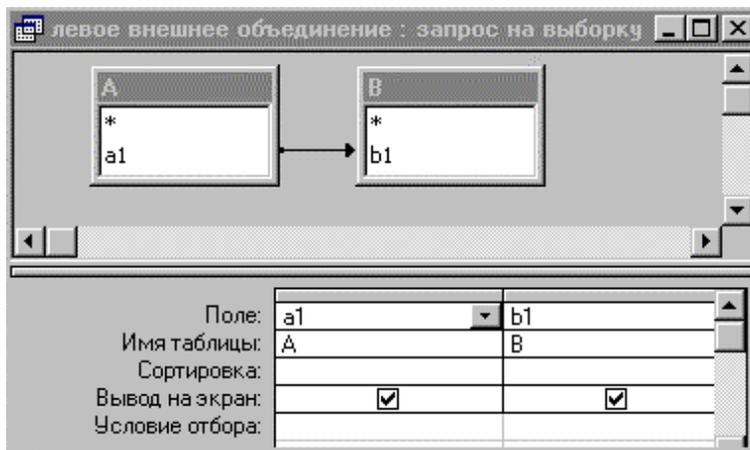


Рис. 5.20. Запрос с левым внешним объединением

	a1	b1
▶	aa	aa
	bb	

Рис. 5.21. Его результат

Пример 5.16. Нужно создать запрос **Адреса студентов**, содержащий список студентов курса, причем для проживающих в общежитии должен быть указан их адрес: номер общежития и номер комнаты.

Для этого следует добавить в окно конструктора две таблицы: **Студенты** и **Общежитие**, содержащие нужную информацию. Так как поле **Код студента** является ключевым в таблице **Студенты**, между ними будет автоматически установлена связь по этому полю, имеющая внутренний тип объединения. Если оставить эту связь без изменения, то Access отберет лишь записи с информацией о студентах, живущих в общежитии. Нам же нужен полный список студентов, поэтому этот тип объединения следует изменить на тип объединения, отбирающий все записи из таблицы **Студенты** и соответствующие им записи из таблицы **Общежитие**.

Затем нужно включить в бланк запроса поля с нужной информацией и сохранить в БД. После выполнения запроса появится список всех студентов курса с их адресами (см. рис. 5.22). Для студентов, не живущих в общежитии, поля **Общежитие** и **Адрес** остаются пустыми (содержат значение *Null*).

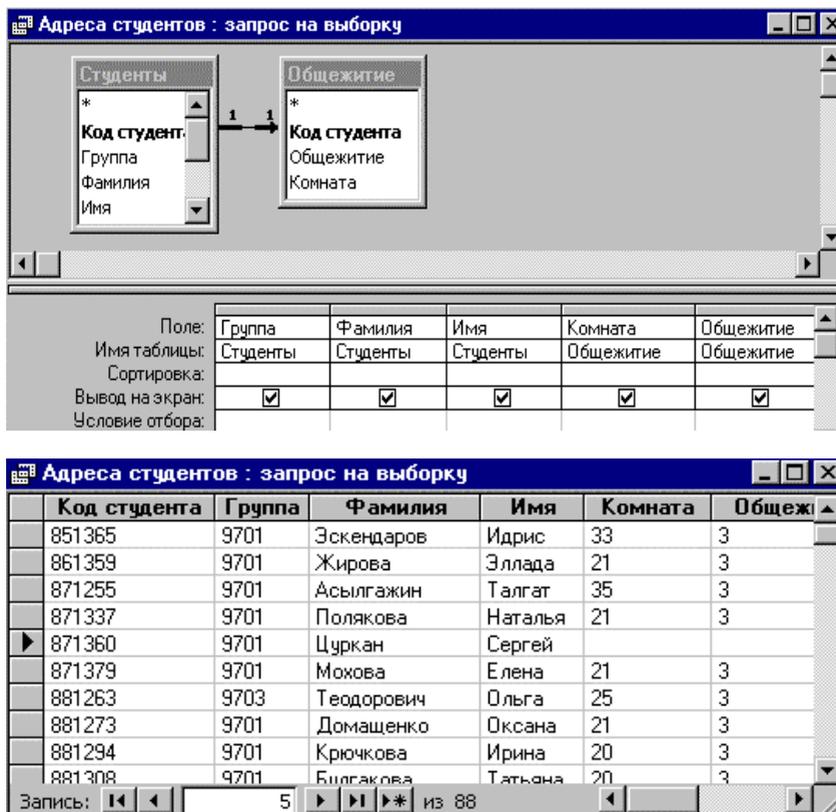


Рис. 5.22. Запрос Адреса студентов

Использование мастера Записи без подчиненных

Созданный запрос можно легко превратить в запрос, который находит студентов, не живущих в общежитии. Для этого нужно лишь добавить в качестве условия отбора в поле **Комната** или **Общежитие** выражение *Is Null*. Для создания такого рода запросов можно также использовать мастера **Записи без подчиненных**.

Пример 5.17. Создадим с помощью этого мастера запрос, который находит студентов, не сдававших сессию.

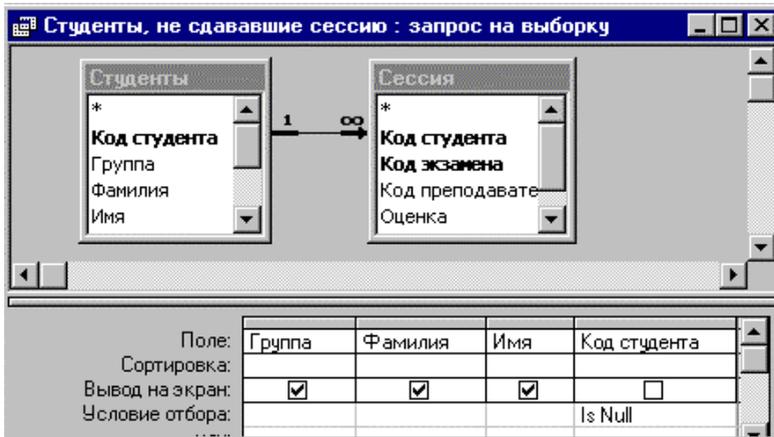


Рис. 5.23. Запрос, созданный мастером Записи без подчиненных

Таким студентам в таблице **Студенты** соответствуют записи, не имеющие связанных с ними записей в подчиненной таблице **Сессия**. После вызова мастера путем выбора соответствующего пункта в окне **Новый запрос** нужно последовательно ответить на следующие вопросы (ответ набран курсивом):

1. Указать таблицу, не имеющую связанных записей с таблицей, указываемой в следующем пункте: *Студенты*.
2. Указать таблицу с подчиненными записями: *Сессия*.
3. Указать поля, используемые для связи таблиц: в обеих таблицах *Код студента*.
4. Выбрать поля, отображаемые в результате выполнения запроса: *Группа, Фамилия, Имя*.
5. Задать имя запроса: *Студенты, не сдававшие сессию*.

	Группа	Фамилия	Имя
▶	9703	Теодорович	Ольга
	9704	Марычева	Ольга

Рис. 5.24. Студенты, не сдававшие сессию

В результате выполнения созданного мастером запроса (см. рис. 5.23) на экране появятся сведения о студентах, не сдававших сессию (рис. 5.24).

5.4.4. Самообъединение

Иногда для получения нужной информации следует объединить таблицу саму с собой, создавая тем самым *самообъединение* таблицы. Такая ситуация, в частности, возникает, если в таблице есть поле, которое ссылается на другое поле той же таблицы.



Код	Сотрудник	Подчиняется
1	Иванов	3
2	Петрова	4
3	Смирнов	4
4	Сидоров	3

Рис. 5.25. Таблица *Служащие*

начальника, этого можно добиться, создав запрос, использующий копию таблицы **Служащие** (см. рис. 5.26).

Для создания этого запроса нужно включить в окно конструктора таблицу **Служащие** дважды. Второй раз она будет добавлена под именем **Служащие-1**, которое во избежание ошибок лучше заменить другим. Для

Пример 5.18. Пусть имеется таблица **Служащие**, содержащая поле **Подчиняется** с идентификационными номерами начальников, причем сведения о них хранятся в той же таблице. Если нужно, чтобы вместо номера стояла фамилия

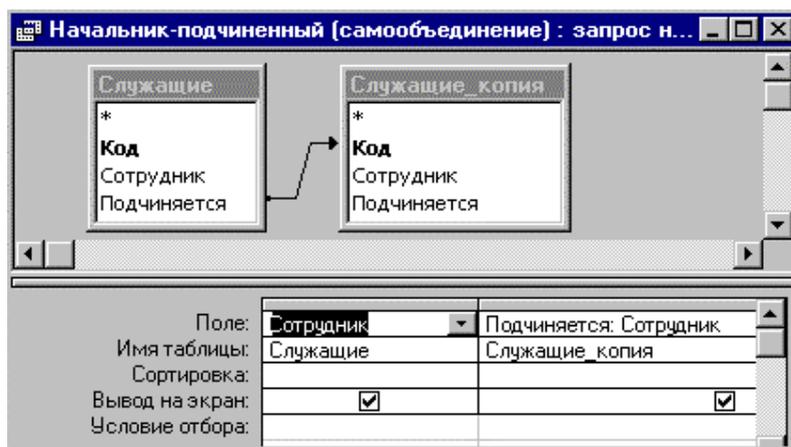


Рис. 5.26. Запрос, использующий самообъединение таблицы

этого нужно щелкнуть по копии таблицы правой кнопкой мыши, выбрать пункт **Свойства**, а затем в поле **Псевдоним** ввести текст *Служащие_копия*.

Затем следует установить связь между таблицами, соединив поле **Подчиняется** таблицы **Служащие** с полем **Код** ее копии.

	Сотрудник	Подчиняется
	Иванов	Смирнов
	Петрова	Сидоров
	Смирнов	главный начальник
	Сидоров	Смирнов

Рис. 5.27. Структура подчиненности

В качестве типа соединения лучше выбрать левое внешнее объединение, так как при внутреннем объединении в результате запроса не будет включена информация о самом главном начальнике. После этого следует включить в бланк запроса поле **Сотрудник** из обеих таблиц, изменив название этого поля из таблицы **Служащие_копия** на **Подчиняется**. Результат запроса представлен на рис. 5.27¹.

В качестве типа соединения лучше выбрать левое внешнее объединение, так как при внутреннем объединении в результате запроса не будет включена информация о самом

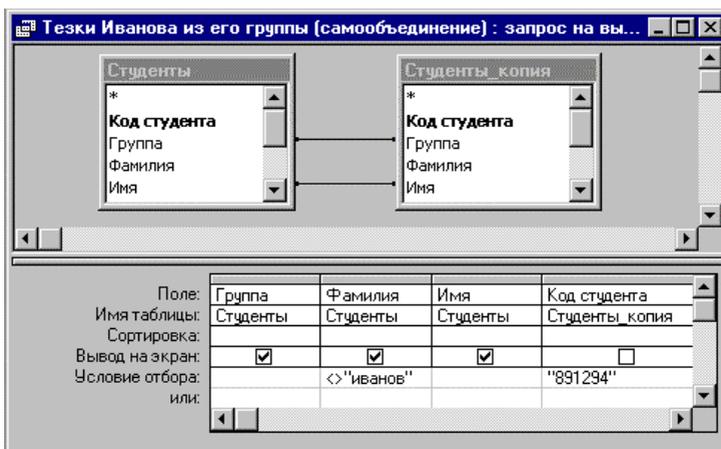


Рис. 5.28. Еще один пример на самообъединение

Самообъединение можно использовать и в других случаях.

Пример 5.19. Пусть нужно найти студентов, которые учатся в той же группе, что и студент Иванов (код 891294), и являются его тезками. Одним из возможных путей решения этой задачи является создание запроса, использующего самообъединение таблицы **Студенты** (см. рис. 5.28)².

¹ Чтобы в пустом поле напротив фамилии *Смирнов* появился текст, достаточно задать в свойствах поля *Подчиняется* такой формат: @;"главный начальник".

² Можно также использовать функцию *DLookup*.

Группа	Фамилия	Имя
9702	Кривоносов	Евгений
9702	Мелентьев	Евгений
▶ 9702	Пентегов	Евгений

Рис. 5.29. Тезки Иванова

В этом запросе Access объединяет пары записей, имеющие одинаковые значения в полях **Группа** и **Имя**. После выполнения запроса в динамическом наборе

остаются лишь три записи (см. рис. 5.29).

5.4.5. Объединение таблиц по отношению неравенства

В большинстве случаев таблицы объединяются по условию равенства значений в полях связи, но допустимо объединение таблиц и по условию «неравенства». В этом случае условием связи может быть любой из операторов сравнения: $>$, $>=$, $<$, $<=$, $<>$.

Для создания запроса, использующего такие условия связи, можно поступить следующим образом:

1. Создать в окне конструктора связь типа равенства.
2. Перейти в режим SQL и заменить знак "=" другим оператором сравнения. Однако такой запрос уже нельзя представить в режиме конструктора.

Пример 5.20. Заменяя знак "=" на " $<>$ ", в приведенной на стр. 93 инструкции SQL, можно создать запрос, объединяющий записи таблиц А и В с несовпадающими значениями в связанных полях. Его результат виден на рис. 5.30. Соответствующая инструкция SQL имеет вид

```
SELECT A.a1, B.b1 FROM A INNER JOIN B ON A.a1 <> B.b1;
```

	a1	b1
	bb	aa
	aa	cc
▶	bb	cc

Рис. 5.30. Связь по неравенству

Можно поступить иначе: не связывать таблицы в окне конструктора, а записать условие на несовпадение значений полей в строке условий. В нашем примере это условие накладывается на значения в поле *a1* и выглядит так: $<> [b1]$.

Соответствующая инструкция SQL имеет вид

```
SELECT A.a1, B.b1 FROM A, B WHERE (((A.a1)<>[b1]));
```

5.5. Подведение итогов

Иногда требуется, чтобы результатом выполнения запроса были не значения отдельных записей таблиц БД, а итоговые значения по группам записей. Например, может понадобиться определить средние баллы по экзаменам в каждой учебной группе или найти суммарную стипендию, полученную студентами в каком-то месяце. Получить ответ на такие вопросы можно с помощью *итогового запроса*.

5.5.1. Группировка записей в итоговом запросе

Для вычисления в запросе итоговых значений нужно щелкнуть по кнопке **Групповые операции** Σ на панели инструментов. В бланке запроса появится строка **Групповая операция** и для всех полей в этой строке будет установлено значение **Группировка**. Это же значение будет по умолчанию устанавливаться и для добавляемых в бланк запроса полей. Необходимо помнить, что в итоговом запросе Access группирует (объединяет в одну группу) записи с одинаковыми значениями в полях, имеющих в строке **Групповая операция** значение **Группировка**. Поэтому, если выполнить запрос, в бланке которого все поля имеют значения **Группировка**, то будет создан набор записей, включающий по одной записи из каждой группы.

Например, если включить в бланк запроса поле **Группа** таблицы **Студенты**, щелкнуть по кнопке **Групповые операции** и выполнить запрос, то Access выдаст на экран список номеров групп¹.

Однако никаких итогов Access не подведет. Для выполнения этой операции ему нужна следующая информация:

- по каким полям нужно подводить итоги;
- какие итоги нужно подводить по данному полю.

Поэтому в бланке запроса наряду с *групповыми полями*, по которым производится группировка, должны находиться *итоговые поля*, причем для каждого итогового поля должен быть указан тип подводимых итогов (сумма, среднее, максимум и т.п.).

Итоговым полем может быть как поле таблицы (запроса), находящейся в окне конструктора, так и специально созданное для этой цели вычисляемое поле. Для каждого итогового поля в строке **Групповая операция** должна быть указана статистическая функция, с помощью которой будет проводиться групповая операция — вычисление итогов в

¹ Фактически использование группировки по полю — быстрее способ создания списка уникальных значений в этом поле.

этом поле. Имя нужной функции (*Sum*, *Avg*, *Max* и т.д.) можно ввести в соответствующую ячейку в строке **Групповая операция** или выбрать из раскрывающегося списка.

Пример 5.21. Найти число студентов в каждой учебной группе.

Добавим в окно конструктора таблицу **Студенты**, затем щелкнем по кнопке **Групповые операции** и включим в бланк запроса поле **Группа**. Оно будет использовано для группировки записей.

Поле:	Группа	Число студентов: Код студента
Имя таблицы:	Студенты	Студенты
Групповая операция:	Группировка	Count
Сортировка:	по возрастанию	
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		

Рис. 5.31. Бланк итогового запроса (пример 5.21)

В качестве итогового поля выберем поле **Код студента**, а в качестве итоговой функции — функцию *Count*. Дадим итоговому полю имя **Число студентов**.

	Группа	Число студентов
▶	9701	19
	9702	16
	9703	19
	9704	18
	9705	15

Рис. 5.32. Его результат

Результат выполнения запроса представлен на рис. 5.32.

Для подсчета числа студентов можно было бы взять в качестве итогового поля любое другое поле таблицы **Студенты**. Единственное условие, которому оно должно удовлетворять, — отсутствие

пустых значений, так как функция *Count* игнорирует записи со значением *Null* в итоговом поле. Это условие заведомо выполнено для поля **Код студента**, так как оно является ключевым. Если же ни для одного поля таблицы нельзя гарантировать отсутствие пустых значений, то в этом случае для подсчета числа всех записей в каждой группе нужно создать вычисляемое поле, введя в строку **Поле** выражение *Count(*)* (см. рис.5.37 на стр. 108).

Часто для подведения итогов используется информация, содержащаяся в разных таблицах. В этом случае Access группирует записи результирующего набора.

Пример 5.22. Найти суммарные баллы, полученные студентами в сессию. Итоговый набор должен содержать следующие сведения: код студента, его фамилию, номер группы и итоговую сумму оценок по сданным экзаменам. Информация должна быть отсортирована по сумме набранных баллов.

Поле:	Код студента	Группа	Фамилия	Оценка
Имя таблицы:	Студенты	Студенты	Студенты	Сессия
Групповая операция:	Группировка	Группировка	Группировка	Sum
Сортировка:				по убыванию
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				

Рис 5.33. Итоги сдачи сессии

Для построения этого запроса добавим в окно конструктора таблицы **Студенты** и **Сессия** и щелкнем по кнопке **Групповые операции**. Затем включим в бланк запроса поля **Код студента**, **Фамилия**, **Группа** и **Оценка**. У первых трех полей оставим тип операции **Группировка**, а в поле **Оценка** выберем в качестве групповой операции функцию **Sum** и зададим тип сортировки *по убыванию* (см. рис. 5.33). Построенный запрос сохраним под именем **Итоги сдачи сессии**.

Группировку записей Access производит для динамического набора, полученного в результате внутреннего объединения таблиц **Студенты** и **Сессия**. Фактически в качестве группового поля используется поле **Код студента**, задающее самое дробное разбиение на группы: в одну группу объединяются записи об оценках одного студента. Отметим, что даже если бы нам не нужно было выводить на экран код студента, это поле все равно необходимо включить в запрос в качестве группового. Иначе Access объединит в одну группу записи с оценками однофамильцев, и запрос выдаст неверный результат.

5.5.2. Отбор записей, формирующих группы

В том случае, когда из групп должны быть исключены некоторые записи, нужно добавить в бланк запроса поле или поля, которые будут использованы для отбора нужных записей. Для создания условия отбора по данному полю нужно выбрать в строке **Групповая операция** значение **Условие** и ввести условие в строку **Условие отбора**. Access автоматически отключит вывод на экран значений этого поля.

Пример 5.23. Определить число студентов в каждой учебной группе, родившихся в 1973 году или позднее.

Для этого нужно в бланк запроса из предыдущего примера (см. рис. 5.31) включить дополнительно поле **День рождения**, выбрать в строке **Групповая операция** значение **Условие** и ввести условие отбора >

31.12.72. Access добавит к дате с двух сторон # и бланк запроса будет иметь вид, представленный на рис. 5.34.

Поле:	Группа	Число студентов: 1	Дата рождения
Имя таблицы:	Студенты	Студенты	Студенты
Групповая операция:	Группировка	Count	Условие
Сортировка:	по возрастанию		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:			>#31.12.72#

Рис 5.34. Отбор записей в группах

После добавления этого условия в Access включит в формируемые группы лишь те записи, которые относятся к студентам, родившимся позднее 31 декабря 1972 года.

Поле:	Группа	Оценка	Пол	Код экзамена
Имя таблицы:	Студенты	Сессия	Студенты	Сессия
Групповая операция:	Группировка	Avg	Условие	Условие
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора:			"ж"	"1"

Рис. 5.35. Бланк запроса к примеру 5.23

Пример 5.24. Определить средние баллы по информатике девушек в каждой учебной группе.

Этот запрос использует данные из двух таблиц: **Студенты** и **Сессия**. В его бланк следует включить следующие поля (см. рис. 5.35):

	Группа	Средний балл
	9701	4,11
	9702	3,67
▶	9703	4,17
	9704	4,00
	9705	3,91

Рис 5.36. Итоговый результат

- **Группа** — используется для группировки записей;
- **Пол** и **Код экзамена** — служит для отбора нужных записей в группах;
- **Оценка** — по этому полю производится групповая операция

Avg (вычисление среднего значения среди оценок по информатике).

Результат выполнения запроса представлен на рис. 5.36. Чтобы получить значения средних баллов в таком (отформатированном) виде необходимо настроить свойства итогового поля Оценка. Для этого нужно щелкнуть правой кнопкой мыши по полю **Оценка**, открыть список его

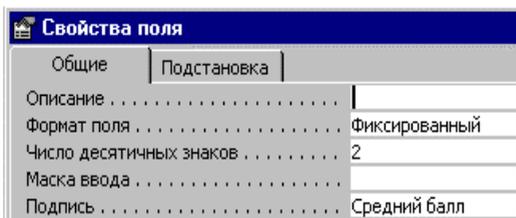


Рис 5.37. Установка свойств поля Оценка

свойств и установить для свойства **Формат** поля значение *Фиксированный*, для свойства **Число десятичных знаков** — 2, а для свойства **Подпись** — *Средний балл* (см. рис. 5.37).

5.5.3. Отбор групп

В итоговый динамический набор можно включать не все группы. Для того чтобы исключить ненужные группы, следует ввести условия отбора в любое поле, содержащее в строке **Групповая операция** значение *Группировка* или одну из статистических функций.

Пример 5.25. Определить группы, в которых учится более 16 студентов.

Для этого достаточно добавить в поле **Число студентов** бланка запроса из примера 5.21 (см. рис. 5.31) условие отбора: > 16 . В результате выполнения этого запроса будут найдены номера учебных групп.

Поле:	Группа	Число студентов: Count(*)
Имя таблицы:	Студенты	
Групповая операция:	Группировка	Выражение
Сортировка:	по возрастанию	
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		> 16

Рис. 5.38. Использование функции Count(*)

Для подсчета числа студентов в учебной группе можно также использовать функцию *Count(*)*. В этом случае бланк запроса будет иметь вид, представленный на рис. 5.38.

Строка условий может содержать выражение, в состав которого входит результат некоторой статистической операции. Для вычисления этого результата должна использоваться не обычная статистическая функция, а статистическая функция по подмножеству.

Пример 5.26. Определить учебные группы, имеющие по информатике более высокий средний балл, чем на курсе в целом.

Запрос использует информацию из таблиц **Студенты** (номера групп) и **Сессия** (оценки). В бланк итогового запроса следует включить следующие поля (см. рис. 5.39):

Поле:	Группа	Оценка	Код экзамена
Имя таблицы:	Студенты	Сессия	Сессия
Групповая операция:	Группировка	Avg	Условие
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		>DAvg("[Оценка]";"Сессия"; "1")	

Рис. 5.39. Использование функции DAvg в условии отбора

- **Группа** — используется для группировки записей;
- **Оценка** — используется как итоговое поле для подсчета средних оценок в группах, а также для отбора нужных групп записей;
- **Код экзамена** — это поле служит для отбора записей в группах (код экзамена по информатике равен 1).

Условие отбора групп выглядит так:

`>DAvg("[Оценка]";"Сессия";"[Код экзамена]='1'")`

Результат вычисления значения функции *DAvg* — средний балл по информатике на курсе.

Для нахождения среднего балла на курсе можно также использовать подчиненный запрос. В этом случае условие отбора в поле **Оценка** будет таким:

`>(SELECT DISTINCTROW Avg(Сессия.Оценка) AS Avg_Оценка
FROM Сессия WHERE (((Сессия.[Код экзамена])="1")));`

Пример 5.27. Определить общую сумму стипендии, полученной студентами групп 9701-9703 в сентябре, октябре и за оба месяца.

Так как информация о распределении студентов по группам содержится в таблице **Студенты**, а информация о стипендии — в таблице **Стипендия**, то создание запроса начнем с включения этих таблиц в окно конструктора. Они связываются по полю **Код студента**. Затем нужно щелкнуть по кнопке **Групповые операции** и включить в бланк запроса поле **Группа**, которое будет использоваться для группировки записей. Для отбора нужных групп следует ввести условие `< 9704` или другое равносильное ему условие, например, `9701 Or 9702 Or 9703`.

Следующий шаг — создание итоговых полей, подсчитывающих суммарную стипендию за каждый месяц. Для этого нужно включить соответствующие поля в бланк запроса, выбрать в качестве групповой

функции *Sum* и заменить названия, которое Access дает итоговым полям (*Sum_ <месяц>*) на более подходящие.

На завершающем этапе построения запроса добавим итоговое поле, подсчитывающее суммарную стипендию за оба месяца. Для этого создадим вычисляемое поле, введя в пустую ячейку строки **Поле** выражение

$$Nz([Стипендия]![Сентябрь])+Nz([Стипендия]![Октябрь])¹$$

и выберем в качестве групповой функции *Sum*. Это поле тоже нужно переименовать и, вызвав его свойства, установить ему в качестве формата вывода значений *Денежный*. Результаты запроса представлены на рис. 5.40.

Группа	Сентябрь	Октябрь	Всего
9701	280 000р.	295 000р.	575 000р.
9702	235 000р.	250 000р.	485 000р.
9703	305 000р.	305 000р.	610 000р.

Рис. 5.40. Стипендия за два месяца

Интересно, что Access модифицировал созданный нами запрос, внося следующие изменения в итоговое поле **Всего** (см. рис. 5.41):

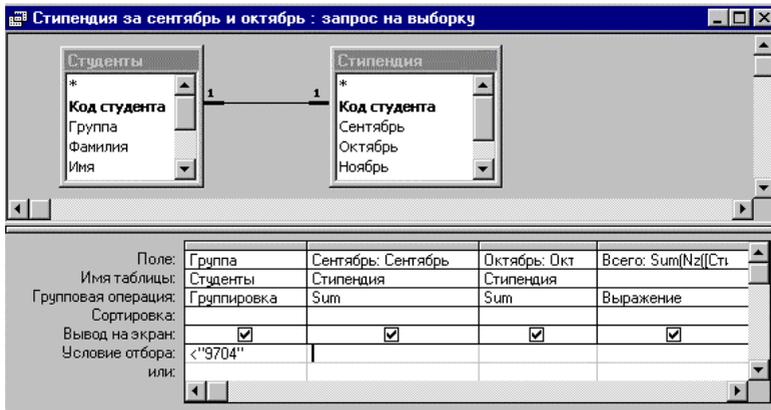


Рис 5.41. Отбор групп в итоговом запросе

- выражение, которое вычисляется в этом поле, заменено выражением $Sum(Nz([Стипендия]![Сентябрь])+Nz([Стипендия]![Октябрь]))$
- в строке **Групповая операция** групповая функция *Sum* заменена групповой операцией *Выражение*².

¹ О том, как это сделать с помощью построителя выражений, и почему использована функция *Nz*, см. пример 4.1 и замечание к нему.

² Эта групповая операция обычно выбирается, если в выражении в строке **Поле** используются одна или несколько групповых функций.

В запрос можно включить дополнительные итоговые поля. Например, для подсчета числа студентов, получающих стипендию в сентябре, нужно добавить в бланк поле **Сентябрь** и выбрать групповую функцию *Count*.

5.6. Перекрестные запросы

Перекрестный запрос — это итоговый запрос специального типа, выводящий результат в виде *перекрестной таблицы*, похожей на электронную таблицу.

Для создания перекрестной таблицы нужно указать:

- *заголовки строк* — одно или несколько полей, значения которых будут использованы в качестве заголовков строк таблицы;
- *заголовки столбцов* — поле, значения которого будут использованы в качестве заголовков столбцов;
- *значения* — числовое поле, значения которого будут использованы для проведения итоговых расчетов;
- *групповую операцию* — групповую функцию, задающую тип итоговых расчетов, или выражение, содержащее групповые функции.

Построение перекрестной таблицы производится примерно по такой схеме. Access производит группировку данных по групповым полям, которые также служат заголовками ее строк и столбцов. Групповая операция проводится над числами, содержащимися в поле значений. Итоговое значение для данной группы помещается в ячейку таблицы, находящуюся на пересечении строк и столбца, названия которых берутся из соответствующих групповых полей. Часть ячеек перекрестной таблицы могут оказаться пустыми (содержащими значение *Null*). Это обстоятельство необходимо учитывать при создании выражений, включающих поля перекрестной таблицы.

Перекрестные запросы обычно используются при создании диаграмм и отчетов. Сохранить перекрестную таблицу в виде обычной таблицы Access с помощью запроса на создание таблицы (см. стр. 124) нельзя.

5.6.1. Мастер Создание перекрестных таблиц

Для создания перекрестного запроса можно воспользоваться соответствующим мастером. При этом нужно иметь в виду, что он создает перекрестный запрос на базе полей одной таблицы/запроса, причем группировка проводится по всем записям. Если для создаваемого перекрестного запроса такая таблица/запрос не существует, то следует сначала создать запрос, содержащий всю необходимую информацию, и

лишь затем воспользоваться услугами мастера. Рассмотрим работу мастера на следующем примере из БД **Книги**.

Пример 5.28. Требуется создать запрос, содержащий сведения о количестве книг, проданных продавцами в каждом квартале и за весь 1997 год.

Так как нужная информация находится в разных таблицах, создадим запрос, который будет использован мастером в качестве источника данных при построении перекрестной таблицы. Для этого поместим в окно конструктора таблицы **Продавцы** и **Заказы**. Access свяжет их по полю **Код продавца**. В бланк запроса включим поле **Код продавца**, а также поля **Количество** и **Дата отправки** со сведениями о количестве книг в заказе и дате его отправки. Создадим вычисляемое поле **Продавец**, содержащее фамилии и имена продавцов. Для этого в пустую ячейку строки **Поле** введем выражение *Продавец: [Фамилия] & " " & [Имя]*. В поле **Дата отправки** введем условие отбора заказов, выполненных в 1997 году, и сохраним запрос под именем «*Заказы 1997 года*» (см. рис. 5.42).

Поле:	Код продавца	Продавец: [Фамилия] & "	Дата отправки	Количество
Имя таблицы:	Продавцы		Заказы	Заказы
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			Like "*" : 97"	

Рис. 5.42. Бланк запроса с информацией для мастера

Для вызова мастера нужно выбрать пункт **Перекрестный запрос** в окне **Новый запрос**. Работа мастера состоит из следующих шагов.

1. Мастер предлагает указать таблицу или запрос, поля которых будут использованы в создаваемом запросе. В списке запросов нужно найти запрос «**Заказы 1997 года**» и щелкнуть по кнопке **Далее** >.
2. Мастер просит указать поля, значения которых будут использованы в качестве заголовков строк (не более трех). Выберем в левом окне, содержащем список доступных полей, поля **Код продавца** и **Продавец**, перебросим их с помощью кнопки > в правое окно и перейдем к следующему окну диалога.
3. Мастер просит указать поле, значения которого будут использованы в качестве заголовков столбцов. Выберем поле **Дата отправки**.
4. Так как выбранное поле имеет тип **Дата/Время**, то мастер предлагает выбрать временной интервал, с которым нужно сгруппировать данные в этом поле. Выберем *Квартал*.
5. На этом этапе мастер просит указать итоговое поле и групповую функцию. Выберем поле **Количество**, а в качестве групповой функции

укажем *Sum*. Здесь же предоставляется возможность указать, нужно ли включать в таблицу итоговое значение по каждой строке. Поскольку мы хотим включить в таблицу данные об общем числе книг, проданных каждым продавцом, нужно оставить «галочку» во флажке с подписью *Да*. В противном случае «галочку» нужно убрать.

- В заключение мастер предлагает дать имя созданному запросу и указать действия после его создания. Назовем запрос «**Работа продавцов в 97 году**» и выберем просмотр результатов запроса.

Созданная таблица имеет лишь один недостаток: столбец с итогами работы каждого продавца за год получил название **Итоговое значение Количество**. Для его устранения перейдем в режим конструктора и заменим имя соответствующего поля в бланке запроса на имя **За год**. Затем перейдем в режим таблицы и переместим этот столбец в конец таблицы. Сохраним все сделанные изменения. Теперь итоговая таблица имеет такой вид (см. рис. 5.43):

Работа продавцов в 97 году : перекрестный запрос							
	Код	Продавец	Кв 1	Кв 2	Кв 3	Кв 4	За год
	1	Ильина Елена	4	50	18	43	115
	2	Иванова Наталья	11	18	12	47	88
	3	Петров Андрей	14	17	15	56	102
	4	Костин Сергей	20	19	20	28	87
▶	5	Сидоров Андрей	9	16	20	16	61

Рис. 5.43. Итоги работы продавцов в 1997 году

При создании перекрестной таблицы на базе многотабличного запроса возможна ситуация, когда в него включены одноименные поля, принадлежащие различным таблицам, например, поля **Фамилия** из таблиц **Продавцы** и **Покупатели**. В этом случае перед использованием мастера такие поля следует переименовать. Если этого не сделать, то Access создаст неработоспособный перекрестный запрос.

5.6.2. Создание перекрестного запроса без помощи мастера

Для создания перекрестного запроса вручную необходимо:

- Выбрать в окне **Новый запрос** пункт **Конструктор**.
- Включить в верхнюю часть окна конструктора все таблицы и запросы, данные которых будут использованы в создаваемом запросе.

3. На панели инструментов нажать кнопку Тип запроса  и выбрать **Перекрестный**. В бланке запроса появятся две дополнительные строки: **Групповая операция** и **Перекрестная таблица**.
4. Добавить поля в строку **Поле** в бланке запроса и создать, если нужно, дополнительно вычисляемые поля. Все поля вначале будут получать в строке **Групповая операция** значение **Группировка**.
5. Для полей, значения которых будут использованы в качестве заголовков строк, нужно в строке **Перекрестная таблица** указать значение **Заголовки строк** и оставить в строке **Групповая операция** значение **Группировка**.
6. Для поля, значения которого будут использованы в качестве заголовков столбцов, нужно в строке **Перекрестная таблица** задать значение **Заголовки столбцов** и оставить в строке **Групповая операция** значение **Группировка**.
7. Для поля, значения которого будут использованы при создании перекрестной таблицы, нужно в строке **Перекрестная таблица** задать значение **Значения**, а в строке **Групповая операция** выбрать групповую функцию, используемую для вычисления значений перекрестной таблицы (например, *Sum* или *Count*).
8. Чтобы отобразить нужные заголовки строк или столбцов, следует ввести выражение, задающее условие отбора в строку **Условие отбора** для соответствующих полей-заголовков в ячейке строки **Перекрестная таблица**.

Если нужно исключить некоторые записи до вычисления значений перекрестной таблицы, то следует добавить в бланк запроса поле или поля, которые будут использованы для отбора нужных записей. Для создания условия отбора по данному полю нужно выбрать в строке **Групповая операция** значение **Условие** и ввести условие в строку **Условие отбора**. Ячейка в строке **Перекрестная таблица** должна быть пустой.

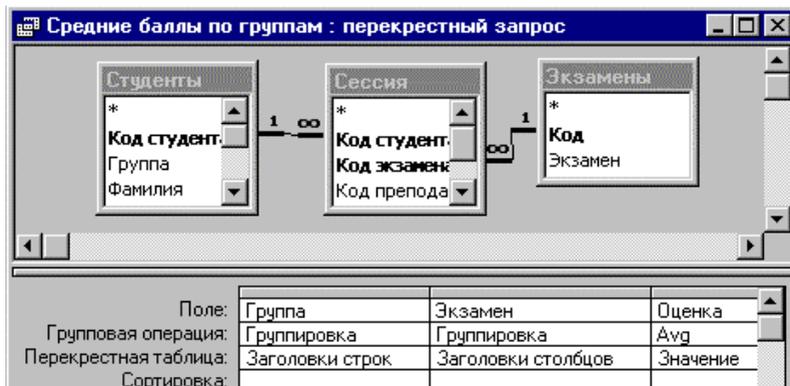
Если при отборе записей запроса используются параметры, то для них обязательно должен быть указан их тип.

Пример 5.29. Создадим перекрестный запрос **Средние баллы по группам**. Его результатом должна быть таблица, столбцами которой являются средние баллы в группах по данному экзамену, а заголовками столбцов служат названия экзаменов (см. рис. 5.44).

Группа	Английский	Информатика	История	Философия
9701	3,88	4,24	4,44	4,05
9702	3,94	4,12	4,81	4,31
9703	4,33	4,22	4,44	4,35
9704	3,81	4,00	4,06	4,47
9705	3,67	3,80	4,53	4,13

*Рис. 5.44. Перекрестная таблица **Средние баллы по группам***

В этом запросе используются данные из таблиц **Студенты**, **Сессия** и **Экзамены**. После их добавления в окно конструктора укажем тип создаваемого запроса: **Перекрестный**. Затем включим в бланк запроса следующие поля: **Группа**, **Экзамен** и **Оценка**. Каждое из них получит в строке **Групповая операция** значение **Группировка**. Так как значения из поля **Группа** будут служить заголовками строк, выберем для него в строке **Перекрестная таблица** значение **Заголовки строк**. Соответственно для



*Рис. 5.45. Бланк запроса **Средние баллы по группам***

поля **Экзамен** выберем для него в этой строке значение **Заголовки столбцов**. Поле **Оценка** используется для подсчета среднего балла. Поэтому выберем для него в строке **Групповая операция** значение **Avg**, а в строке **Перекрестная таблица** — **Значение**. Чтобы значения таблицы выдавались в отформатированном виде, зададим следующие свойства этого поля: для **Формат поля** — **Фиксированный** и для **Число десятичных знаков** — 2.

Сохраним созданный запрос под именем **Средние баллы по группам** (см. рис. 5.45). В дальнейшем (в главе 6) мы используем его для построения диаграммы.

Использование вычисляемых полей в перекрестном запросе

В перекрестном запросе наравне с обычными полями в качестве заголовков могут использоваться вычисляемые поля.

Пример 5.30. Создадим перекрестный запрос **Итоги сессии на курсе**. Итоговая таблица должна содержать ФИО студентов, номера их учебных групп, номера зачетных книжек и оценки по экзаменам. Информация в ней должна быть отсортирована по группам, а затем по фамилиям.

Группа	Код студента	ФИО: [Студенты]	Экзамен	Оценка
Студенты	Сессия		Экзамены	Сессия
Группировка	Группировка	Группировка	Группировка	First
Заголовки строк по возрастанию	Заголовки строк	Заголовки строк	Заголовки столбц	Значение

Рис. 5.46. Бланк запроса Итоги сессии на курсе

В этом запросе также используются данные из таблиц **Студенты**, **Сессия** и **Экзамены**. Добавим их в окно конструктора и укажем тип создаваемого запроса: *Перекрестный*. Затем включим в бланк запроса следующие поля: **Группа**, **Код студента**, **Экзамен** и **Оценка**. Зададим для первых двух полей в строке **Перекрестная таблица** значение *Заголовки строк*, а для поля **Экзамен** — *Заголовки столбцов*. Для поля **Оценка** укажем в качестве групповой операции *First*, а для строки **Перекрестная таблица** — *Значение*. Выбор групповой операции в данном случае особой роли не играет (можно взять *Sum* или *Avg*), так как после группировки записей в поле значений каждой группы содержится не более одного числа (оценки студента по экзамену).

Создадим вычисляемое поле **ФИО**, и также выберем для него в строке **Перекрестная таблица** значение *Заголовки строк*. Разместим это поле после поля **Группа** и зададим в полях **Группа** и **ФИО** сортировку по возрастанию. Для поля **Код студента** зададим в качестве свойства **Подпись** значение *№ зач_кн*. На этом создание запроса завершено. Сохраним его под именем **Итоги сессии на курсе**.

Добавление итогового столбца в перекрестную таблицу

В перекрестную таблицу можно добавить один или несколько столбцов, содержащих итоговые сведения по строке. Для добавления итогового столбца следует еще раз включить в бланк запроса поле, значения которого используются при создании перекрестной таблицы. В

строке **Групповая операция** нужно выбрать подходящую итоговую функцию, а в строке **Перекрестная таблица** — значение *Заголовки строк*.

Группа	№ зач. кл.	ФИО	Английский	Информатика
9701	891302	Касымбеков Мурат Сарсе	4	5
9701	891303	Кенжебаева Шолпан Саду	4	5
9701	891307	Коняев Сергей Николаевич	3	5
9701	881294	Крючкова Ирина Алексеев	4	4
9701	891311	Куликов Сергей Владимир	4	5
9701	891314	Лукьянов Александр Ген	5	5
9701	871379	Мохова Елена Иосифовна	2	4
9701	871337	Полякова Наталья Геннад	3	
9701	871360	Цуркан Сергей Павлович	4	4
9701	851365	Эскендаров Идрис Язулов	4	4
9702	891255	Анурина Наталья Альберт	4	2
9702	891266	Борисов Дмитрий Юрьевич	4	5

Рис. 5.47. Итоги сессии на курсе

Например, чтобы включить в перекрестную таблицу **Итоги сессии на курсе** столбец со средними баллами студентов и сделать его последним в таблице, достаточно выполнить следующие действия:

Поле:	Средний балл: Оценка
Имя таблицы:	Сессия
Групповая операция:	Avg
Перекрестная таблица:	Заголовки строк
Сортировка:	

Рис. 5.48. Создание итогового столбца

1. Включить в бланк запроса поле **Оценка** из таблицы **Сессия**.
2. Выбрать в строке **Групповая операция** значение *Avg*, а в строке **Перекрестная таблица** — значение *Заголовки строк*.

3. Дать этому столбцу новое название **Средний балл** и установить

в его свойствах нужный формат вывода значений.

4. Перейти в режим просмотра и сделать столбец **Средний балл** последним в итоговой таблице.

Отметим, что добавление итоговой строки в перекрестную таблицу невозможно, так как результат запроса может содержать строки либо с детализированной информацией, либо с агрегированной. Однако в формах и отчетах это вполне допустимо.

Использование свойства «Заголовки столбцов»

Иногда требуется изменить порядок следования столбцов в итоговой таблице. Для этого нужно, находясь в режиме конструктора, вызвать щелчком мыши по кнопке **Свойства**  на панели инструментов свойства запроса. Затем щелкнуть на свойстве **Заголовки столбцов** и ввести текст заголовков в нужном порядке, отделяя их друг от друга точкой с запятой или запятой в зависимости от установок Windows.

1. В перекрестной таблице появятся лишь те столбцы, чьи имена указаны в свойстве **Заголовки столбцов**. Это обстоятельство можно использовать для отбора нужных столбцов.
2. Если какое-то имя введено с ошибкой, то в таблице появится пустой столбец с этим именем.

Пример 5.31. Нужно создать перекрестный запрос с информацией о заказах покупателей, живущих в данном городе, за последние три месяца 1997 года. Название города — параметр запроса. Заголовками строк должны быть фамилии и инициалы покупателей, заголовками столбцов — названия месяцев в формате «мм-гг», а значениями таблицы — суммарная стоимость заказов, сделанных покупателем в данном месяце.

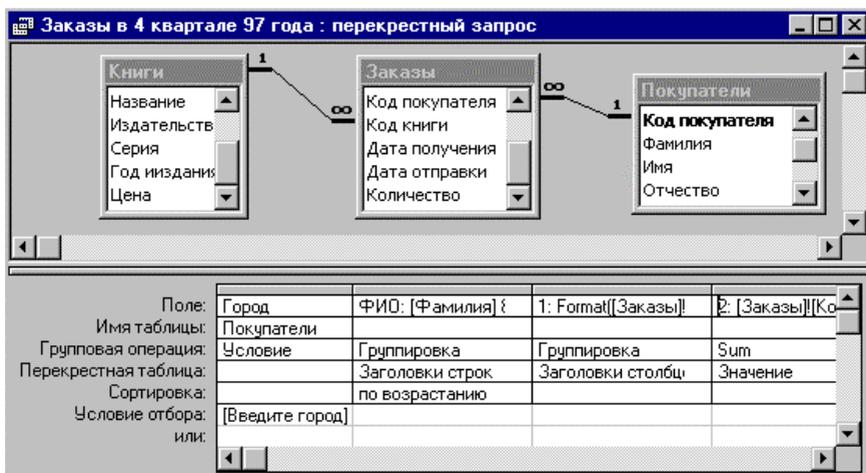


Рис. 5.49. Перекрестный запрос с параметром

В этом запросе используется информация из таблиц **Покупатели**, **Заказы** и **Книги**. После их добавления в окно конструктора укажем тип создаваемого запроса: *Перекрестный*.

Затем включим в бланк запроса поле **Город** и введем в строке **Условие отбора** в качестве имени параметра текст *Введите город* и заключим его в

квадратные скобки. Выберем пункт меню **Запрос**, а затем **Параметры**. В окне **Параметры запроса** введем имя этого параметра и укажем его тип: *Текстовый*. В строке **Групповая операция заменим** значение *Группировка* на *Условие*.

Создадим вычисляемое поле ФИО, которое будет использоваться в качестве заголовков строк. Для этого введем в строку **Поле** выражение *ФИО: [Фамилия] & " " & Left([Имя];1) & ". " & Left([Отчество];1) & ". "*, в строке **Перекрестная таблица** укажем значение *Заголовки строк*, а в строке **Сортировка** — *по возрастанию*.

Для заголовков столбцов также создадим вычисляемое поле *Format ([Заказы] ![Дата отправки];«ттт-уу»)*. В строке **Перекрестная таблица** укажем значение *Заголовки столбцов*. Затем вызовем свойства запроса и в **свойстве Заголовки столбцов** введем текст заголовков *окт-97; ноя-97; дек-97* (см. рис. 5.50).

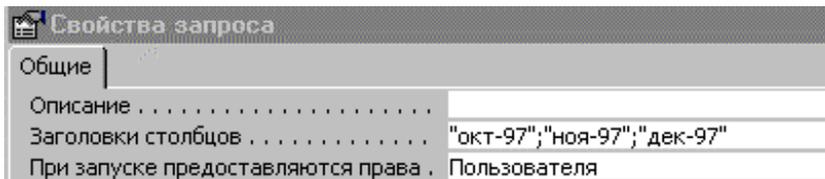


Рис. 5.50. Ввод заголовков столбцов

Для вычисления значений итоговой таблицы создадим еще одно поле, введя в строку **Поле** выражение *[Заказы]![Количество]*[Книги]![Цена]*. Для этого поля нужно в строке **Перекрестная таблица** выбрать *Значение*, а в строке **Групповая операция** — *Sum*.

На этом создание запроса завершено. При его выполнении Access попросит указать город, в котором живут покупатели. После ввода текста *новосибирск* появится таблица со сведениями о заказах покупателей, живущих в Новосибирске (см. рис. 5.51).

	ФИО	окт-97	ноя-97	дек-97
	Буланов П. С.		225р.	43р.
	Конев С. Л.	186р.		
	Кузнецов С. И.		124р.	381р.
	Лукьянов Д. В.			
▶	Смирнов И. В.			115р.

Рис. 5.51. Заказы покупателей из Новосибирска

Задание фиксированных заголовков столбцов позволяет уменьшить время выполнения запроса. Если перекрестный запрос используется как

базовый для подчиненной формы, то явное указание имен столбцов в этом свойстве необходимо для установления связи между подчиненной формой и запросом.

5.7. Запросы на изменение

Запросы этого типа позволяют автоматизировать изменения данных в таблицах, а также сохранить результат запроса в виде таблицы Access.

5.7.1. Запрос на добавление

С помощью запроса на добавление можно скопировать данные из одной таблицы (источника) и поместить их в другую таблицу (получатель). Причем можно добавить в таблицу-получатель как записи целиком, так и содержимое отдельных полей таблицы-источника.

Для создания запроса на добавление нужно выполнить следующие действия:

1. Включить в окно конструктора таблицу-источник. Затем выбрать пункт меню **Запрос**, а потом **Добавление**. На экране появится диалоговое окно **Добавление** (см. рис 5.52).
2. В поле **Имя таблицы** указать имя таблицы-получателя и затем нажать кнопку **ОК**. Если таблица находится в другой БД, то нужно перед нажатием кнопки выбрать параметр **В другой базе данных** и ввести полное имя этой БД.

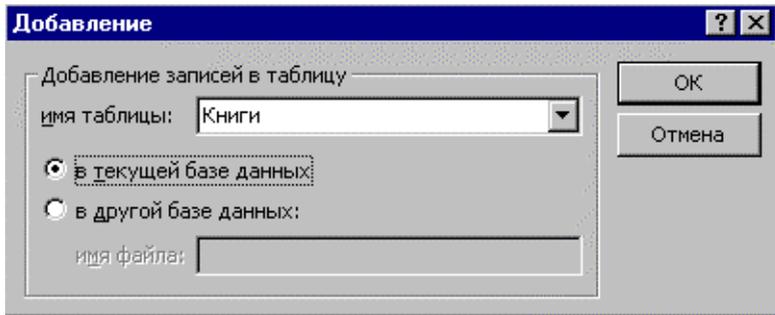


Рис. 5.52. Диалоговое окно **Добавление**

3. В бланк запроса следует включить:
 - поля, участвующие в добавлении;
 - поля, для которых задаются условия отбора;
 - поле, соответствующее ключевому полю таблицы-получателя¹.

¹Если ключевое поле в таблице-получателе имеет тип *Счетчик*, то этого делать не нужно.

Если все поля в обеих таблицах имеют одинаковые имена, то можно просто переместить символ «звездочка» (*) в бланк запроса. Если имя добавляемого поля совпадает с именем соответствующего поля таблицы-получателя, то имя поля таблицы-получателя будет автоматически введено в строку **Добавление**. Если же имена полей таблиц отличны друг от друга, то в строке **Добавление** нужно указать имя поля таблицы-получателя, которое соответствует добавляемому полю таблицы-источника. Затем следует ввести условия отбора записей из таблицы-источника в строку **Условие отбора**.

4. Для просмотра добавляемых записей нужно щелкнуть по кнопке **Вид**  на панели инструментов. Затем следует вернуться в режим конструктора, повторно щелкнув по этой кнопке.

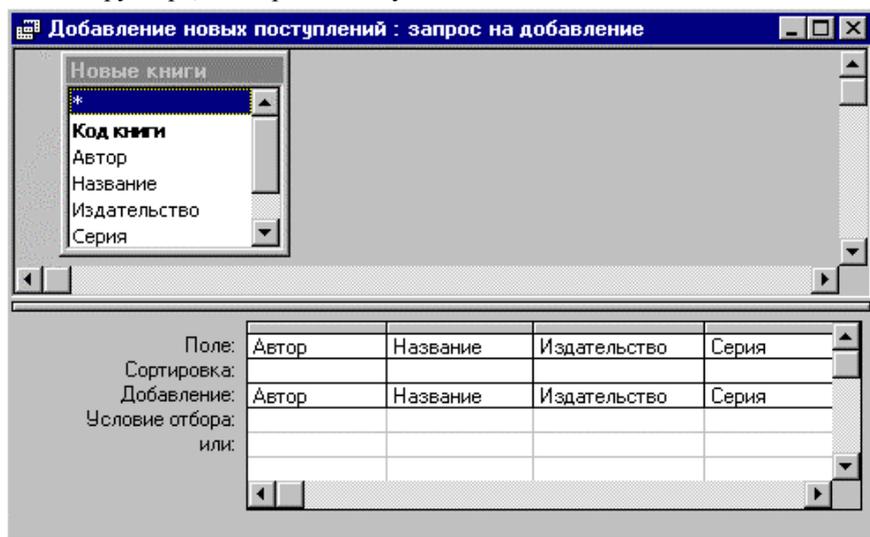


Рис. 5.53. Запрос на добавление в режиме конструктора

5. Для добавления записей нужно щелкнуть по кнопке **Запуск**  на панели инструментов. Access сообщит число добавляемых записей и попросит подтвердить выполнение операции. После подтверждения записи будут добавлены в таблицу-получатель.

Пример 5.32. Нужно добавить в таблицу **Книги** сведения о новых поступлениях книг. Эти сведения содержатся в таблице **Новые книги**, имеющей такую же структуру, что и таблица **Книги**, кроме поля **Цена**.

Создание запроса начнем с включения в окно конструктора таблицы-источника **Новые книги**. Затем вызовем контекстное меню запроса и

выберем сначала **Тип запроса**, а потом пункт **Добавление**. В появившемся окне **Добавление** укажем имя таблицы-получателя *Книги*.

В бланк запроса поместим все поля таблицы **Новые книги** за исключением ключевого поля **Код книги**, так как оно имеет тип *Счетчик*. Поскольку имена полей в обеих таблицах совпадают, Access сам сопоставляет с каждым полем таблицы-источника соответствующее поле таблицы-получателя (см. рис. 5.53).

После выполнения запроса в конец таблицы **Книги** будут добавлены записи с информацией о новых книгах. Access автоматически введет значения в поле **Код книги**. Поле **Цена** в добавленных записях будет пустым, так как сведения о ценах книг отсутствовали в таблице-источнике.

5.7.2. Запрос на удаление

Запрос на удаление позволяет удалить ненужные записи из таблицы. С его помощью можно удалять только всю запись целиком, а не отдельные поля внутри нее.

Для создания запроса на удаление нужно выполнить следующие действия:

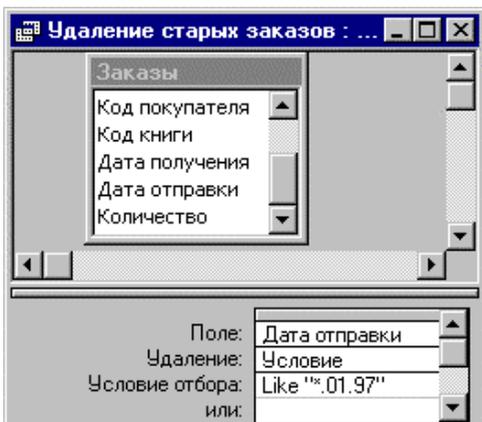


Рис. 5.54. Удаление заказов за январь 97 года

1. Включить в окно конструктора таблицу, из которой предстоит удалить записи. Затем **выбрать пункт меню Запрос**, а потом **Удаление**.
2. Ввести условия отбора удаляемых записей в строку **Условие отбора**. Эти условия появятся под словом **Условие** строки **Удаление**.
3. Для просмотра удаляемых записей нужно щелкнуть по кнопке **Вид**  на панели инструментов. Затем следует вернуться в режим

конструктора, повторно нажав эту кнопку.

4. Для удаления записей нужно нажать кнопку **Запуск**  на панели инструментов. Access сообщит число удаляемых записей и попросит подтвердить выполнение операции. После подтверждения записи будут удалены из таблицы.

Пример 5.33. Нужно удалить из таблицы **Заказы** сведения о заказах, отправленных в январе 1997 года.

Для создания запроса нужно включить в окно конструктора таблицу **Заказы** и выбрать в меню **Запрос** тип запроса **Удаление**. Затем следует добавить в бланк запроса поле **Дата отправки** и ввести условие отбора: **.01.97* (см. рис. 5.54).

Чтобы просмотреть удаляемые записи, щелкнем по кнопке **Вид**. На экране появятся четыре записи (см. рис. 5.55). После выполнения запроса информация об этих заказах будет удалена из БД.

При удалении записей с помощью запроса на удаление отменить эту операцию нельзя. Поэтому рекомендуется перед выполнением запроса сделать резервную копию таблицы. Тогда, если в результате выполнения запроса будут удалены не те записи, то их можно будет восстановить из резервной копии.

	Дата отправки
	24.01.97
	25.01.97
	26.01.97
	29.01.97

Рис. 5.55. Удаляемые записи

При выполнении запроса на удаление могут быть удалены некоторые записи из таблиц, не включенных в запрос. Это произойдет, если в запрос включена таблица, находящаяся на стороне «один» отношения «один-ко-многим», и для этого отношения установлен режим

каскадного удаления. При удалении записей из этой таблицы будут также удаляться связанные с ними записи из подчиненных таблиц.

Например, если в связи «один-ко-многим» между таблицами **Студенты** и **Сессия** установлен режим каскадного удаления, то при удалении из таблицы **Студенты** сведений об исключенном студенте с помощью соответствующего запроса Access автоматически удалит из таблицы **Сессия** информацию о его оценках.

Если режим каскадного удаления не включен, то прежде чем удалить записи из главной таблицы, нужно выполнить запрос на удаление связанных с ними записей из подчиненных таблиц.

Пример 5.34. Нужно удалить из таблицы **Покупатели** записи, относящиеся к покупателям из Омска. Известно, что эта таблица является главной в связи «один-ко-многим» между таблицами **Покупатели** и **Заказы**, для которой установлен режим обеспечения целостности данных, но не включен режим каскадного удаления.

Для выполнения этой операции с помощью запроса на удаление нужно включить в окно конструктора обе связанные таблицы и указать тип запроса — **Удаление**. Затем следует добавить в бланк запроса поле **Город** из таблицы **Покупатели** и ввести в него условие отбора: *омск*. После этого

нужно перенести в бланк запроса звездочку (*) из таблицы **Заказы**. В ячейке **Удаление** в этом поле появляется значение **Из** (см. рис. 5.56).

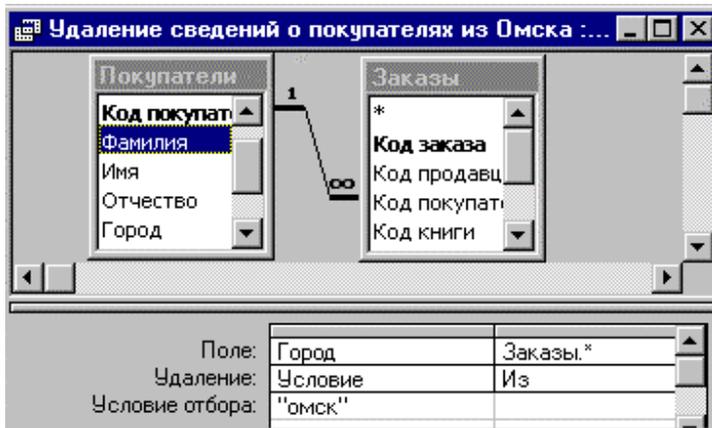


Рис. 5.56. Удаление заказов из Омска

Если выполнить этот запрос, то из подчиненной таблицы **Заказы** будут удалены записи, относящиеся к омским покупателям. Для их предварительного просмотра достаточно щелкнуть по кнопке **Вид**.

После удаления записей о заказах омичей нужно удалить саму таблицу **Заказы** из окна конструктора (см. рис. 5.57). Для этого следует выделить щелчком мыши список ее полей и нажать клавишу **Delete**. Затем нужно еще раз выполнить запрос. Теперь будут удалены записи из главной таблицы **Покупатели**, указанные в условии отбора.

5.7.3. Запрос на создание таблицы

Запрос на создание таблицы позволяет создать таблицу на основе данных, содержащихся в других таблицах БД. Он часто используется для создания таблицы, предназначенной для экспорта в другую БД Access или архивной копии таблицы. Например, можно создать таблицу со сведениями обо всех старых заказах на книги прежде чем удалить их из таблицы **Заказы**.

Для создания запроса этого типа нужно выполнить следующие действия:

1. Включить в окно конструктора таблицы или запросы, содержащие записи, которые следует поместить в новую таблицу.

2. Поместить в бланк запроса поля, которые должна содержать новая таблица. Добавить к ним поля, для которых задаются условия отбора. В строке **Условие отбора** ввести условия выбора нужных записей.

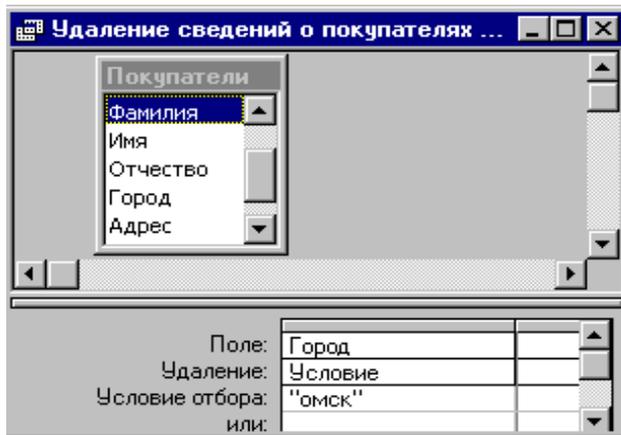


Рис. 5.57. Удаление сведений о покупателях из Омска

3. Выбрать пункт меню **Запрос**, а затем **Создание таблицы**. На экране появится диалоговое **Создание таблицы** (см. рис. 5.58).

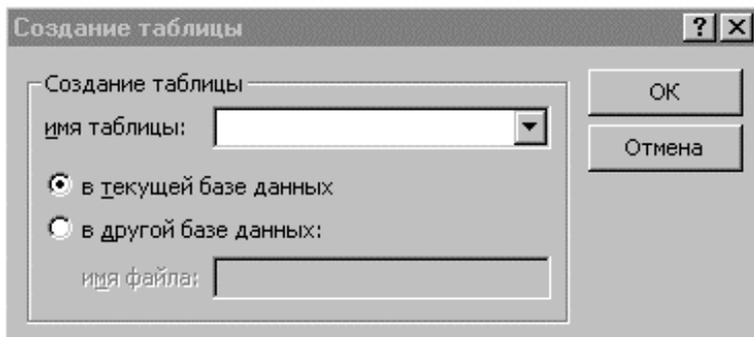


Рис. 5.58. Диалоговое окно *Создание таблицы*

4. В поле **Имя таблицы** ввести имя создаваемой таблицы и затем щелкнуть по кнопке **ОК**. Если таблица находится в другой БД, то нужно перед нажатием кнопки выбрать параметр **В другой базе данных** и ввести полное имя этой БД.

5. Для предварительного просмотра новой таблицы нужно щелкнуть по кнопке **Вид**  на панели инструментов. Затем следует вернуться в режим конструктора, повторно щелкнув по этой кнопке.
6. Для создания таблицы нужно щелкнуть по кнопке **Запуск**  на панели инструментов. Если в БД уже существует таблица с таким именем, то Access попросит разрешения удалить ее перед началом операции. Затем он сообщит число записей и попросит подтвердить выполнение операции. После подтверждения будет создана таблица.

Пример 5.35. Построим запрос, создающий таблицу, в которой будет храниться информация о «старых» заказах.

Для этого добавим в окно конструктора таблицу **Заказы** и перенесем в бланк запроса звездочку (*) и поле **Дата отправки**. Введем в это поле

Поле:	Заказы.*	Дата отправки
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		<[конечная дата ?]

условие в виде параметра (см. рис. 5.59) и установим для него тип *Дата/время*. Затем укажем тип запроса

Рис. 5.59. Отбор «старых» заказов

Создание таблицы и в

окне **Создание таблицы** введем имя создаваемой таблицы: **Старые заказы**.

Выполним созданный запрос. Access попросит ввести значение параметра и после ввода даты создаст таблицу **Старые заказы**, содержащую заказы, дата отправки которых не превосходит введенной даты. Ее структура идентична структуре таблицы **Заказы**. При повторном выполнении запроса старая таблица будет удалена и ее заменит новая таблица с тем же именем.

5.7.4. Запрос на обновление

Этот запрос позволяет внести изменения в группу записей одной или нескольких таблиц. Для создания запроса на обновление нужно выполнить следующие действия:

1. Включить в окно конструктора таблицу или запрос, содержащие записи, которые следует обновить.
2. Выбрать пункт меню **Запрос**, а затем **Обновление**. В бланке запроса появится дополнительная строка **Обновление**.
3. Поместить в бланк запроса обновляемые поля и добавить к ним поля, для которых задаются условия отбора.

4. В строке **Условие отбора** ввести условия выбора нужных записей, а в строке **Обновление** ввести выражения, которые должны быть использованы для изменения полей.
5. Для предварительного просмотра обновляемых записей щелкнуть по кнопке **Вид**  на панели инструментов. Выводимый список будет содержать старые значения. Затем следует вернуться в режим конструктора, повторно щелкнув по этой кнопке.
6. Для обновления записей щелкнуть по кнопке **Запуск**  на панели инструментов. Access сообщит число обновляемых записей и попросит подтвердить выполнение операции. После подтверждения в записи будут внесены новые значения.

Пример 5.36. Построим запрос, вносящий изменения в цены книг издательства «Питер». Предполагается, что цена всех книг этого издательства должна увеличиться на 20%.

Поле:	Цена	Издательство
Имя таблицы:	Книги	Книги
Обновление:	[Цена]*1,2	
Условие отбора:		"Питер"

Для создания запроса добавим в окно конструктора таблицу **Книги** и выберем пункт меню **Запрос**, а

затем пункт **Обновление**. Поместим в бланк запроса обновляемое поле **Цена** и поле **Издательство**, в котором зададим условие отбора *Питер* (см. рис. 5.60). После выполнения запроса Access обновит значения в шести записях.

Рис. 5.60. Изменение цен на книги

5.7.5. Ошибки при выполнении запросов на изменение

Перед внесением изменений в БД Access анализирует запрос и изменяемые данные. При обнаружении ошибки он выдает соответствующее сообщение и предоставляет пользователю возможность отменить операцию. Access различает следующие типы ошибок:

1. *Ошибки преобразования типа* возникают при добавлении данных в существующую таблицу, если тип данных полей-получателей не совпадает с типом данных полей-источников и добавляемые данные не могут быть преобразованы к типу данных полей-получателей.
2. *Нарушение уникальности ключа*. Такие ошибки возникают при попытке добавления или изменения записи в таблице, в результате чего появляется дубликат первичного ключа или уникального индекса.

Access не будет добавлять или изменять такие записи. Поэтому перед выполнением такой операции нужно изменить значения первичного ключа в исходной таблице.

3. *Нарушение блокировки* возникает при работе с таблицей, к которой имеют доступ другие пользователи в сети. Access не может выполнить операцию обновления записи, если в нее в это же время вносит изменения другой пользователь.
4. *Нарушение условий на значение*. Если добавляемая или обновляемая запись не удовлетворяет условию на значение для одного из полей или таблицы, то Access не произведет операцию добавления или обновления для этой записи.