

Федеральное агентство по образованию
Государственное образовательное учреждение
высшего профессионального образования
ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Е. К. Белый

Десять шагов в Microsoft Access

Петрозаводск
Издательство ПетрГУ
2005

ББК 32.97
УДК 681.3
Б 439

Рецензенты:

Н. А. Будникова, доцент кафедры Информатики и математического обеспечения ПетрГУ, канд. физ.-мат. наук;
Н. В. Коцобан, начальник Центра информационных технологий и телемедицины Республиканской больницы МЗ РК

Печатается по решению
редакционно-издательского совета
Петрозаводского государственного университета

Белый Е. К.

Б 439 Десять шагов в Microsoft Access: Учеб. пособие. – Петрозаводск: Изд-во ПетрГУ, 2005. – 92 с.
ISBN 5-8021-0505-4

Учебное пособие предназначено для студентов первых курсов технических факультетов, изучающих курс «Информатика», и может быть использовано как для самостоятельного изучения Microsoft Access, так и при проведении лабораторных работ. Пособие позволяет в течение десяти академических часов, проведенных за компьютером, получить первые необходимые навыки разработки приложений Access.

ББК 32.97
УДК 681.3

ISBN 5-8021-0505-4

© Белый Е. К., 2005
© Петрозаводский государственный университет, 2005

Предисловие

Результатом можно только восхищаться, а процесс можно научиться применять.

Эдвард де Боно

Подготовка квалифицированного инженера предполагает формирование навыков эффективной работы с данными. Традиционно в курсе «Информатики» для технических факультетов значительное внимание уделяется вопросам обработки данных, а вопросы разработки и построения баз данных отходят на задний план. В какой-то мере, учитывая специфику профессиональной деятельности инженера, это оправданно. Однако обработке данных всегда предшествует построение информационной модели исследуемого процесса, разработка структуры таблиц, организация сбора и обновления данных. Таким образом, без должной подготовки данных до этапа обработки можно просто не дойти. Привить необходимые для построения баз данных навыки, выполняя серию разрозненных заданий и упражнений по созданию отдельных объектов Microsoft Access, нереально, тем более в жестких временных рамках учебного процесса. Поэтому мы пошли по пути обучения в процессе разработки приложения. При таком подходе свойства таблиц, запросов, форм, отчетов и других объектов Microsoft Access будут раскрываться в их взаимосвязи. Известный французский психолог Эдвард де Боно, слова которого взяты в качестве эпиграфа, считал, что нельзя научить человека творчески мыслить, заставляя его делать последовательность правильных, безукоризненно логичных шагов. В процессе разработки приложения мы иногда будем принимать не самые лучшие решения, наблюдать, что в результате получилось, и снова возвращаться для доработки «неудачных» фрагментов приложения. В литературе большое внимание уделяется этапу проектирования базы данных, однако применить эти рекомендации на практике, в полной мере, может только человек, разработавший не одну базу. Мы будем смотреть на процесс разработки приложения Microsoft Access глазами «новичка».

Поскольку пособие отражает процесс разработки приложения, естественно было разбить материал на шаги, соответствующие этапам работы. Всего таких шагов десять. Средний студент в состоянии преодолеть один шаг за академический час. Если вначале читатель практически работает под диктовку автора, то далее с каждым шагом ему предоставляется все большая и большая самостоятельность.

В качестве объекта для разработки базы данных мы выбрали склад. Почему именно склад? Во-первых, эта задача позволяет достаточно полно раскрыть возможности Microsoft Access. Во-вторых, задача имеет практическую направленность, поскольку ни одно предприятие не обходится без склада. На складе завода хранятся запасы сырья, которое закупается большими партиями, а затем в процессе производства распределяется по цехам, на складе хранятся запасы готовой продукции. В любом медицинском учреждении есть фармацевтический отдел, который обеспечивает закупку медикаментов и предметов медицинского назначения у поставщиков и распределение их по отделениям. Бумага, на которой подготовлено настоящее пособие, также поступила на кафедру с университетского склада. Во всех приведенных выше примерах необходим учет прихода, расхода и наличия товаров, и все перечисленные примеры можно объединить одной информационной моделью. Приложение, которое мы собираемся разработать, предназначено для работы в среде Microsoft Access версии 97 и выше.

Пособие можно использовать как для самостоятельного изучения Microsoft Access, так и при проведении лабораторных работ по курсу «Информатика». В последнем случае задания внутри группы студентов легко можно сделать индивидуальными, предоставив каждому студенту возможность выбрать свой набор хранящихся на складе товаров.

Замечания и предложения вы можете направлять по адресу: **belyi@petsu.ru**.

1. Шаг первый. Работа с таблицами и запросами

1.1. Создание новой базы данных. Для начала создайте папку. Если пособие используется в лабораторных работах, студент может включить в название папки номер своей зачетной книжки или фамилию. Например: «Склад_0350268». После первого запуска Microsoft Access в окне приложения появится диалоговое окно, в котором Вам будет предложено открыть одну из созданных ранее баз данных (если таковые имеются) либо создать новую, пока пустую.

Сохраните новую базу данных в только что созданной папке, присвоив файлу название, например, «Петров и компания» (рис. 1). Нажмите кнопку «Создать» (Create). На экране появится окно базы данных. В окне базы данных представлены основные объекты базы данных: таблицы, запросы, формы и другие. Если Вы располагаете версией Microsoft Access 97, интересующий Вас класс объектов можно выбрать, щелкнув левой клавишей мыши по соответствующей заставке в верхней части окна базы данных. В версиях, начиная с 2000-й, классы объектов располагаются в левой части окна базы данных (рис. 2).

1.2. Создание таблицы. Таблица – объект базы данных, предназначенный для хранения данных. Каждая таблица содержит информацию о предметах определенного типа. Перед созданием таблицы следует определить набор важных для пользователя характеристик (свойств, параметров) предмета.

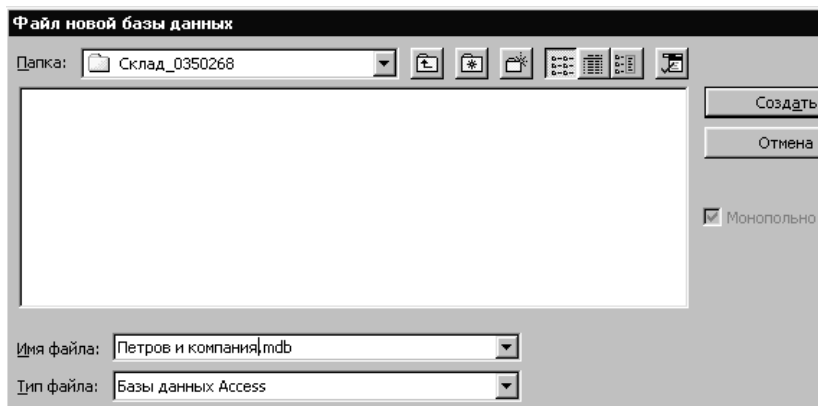


Рис. 1. Создание новой базы данных

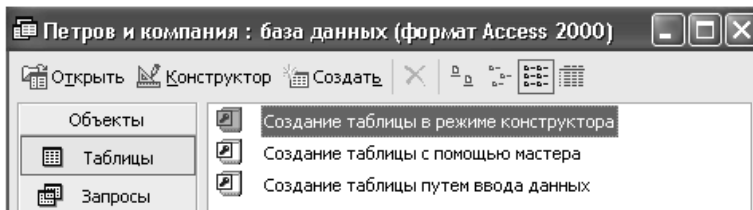
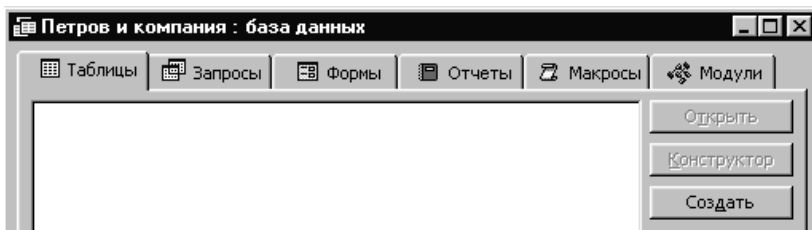


Рис. 2. Окна баз данных Access 97 и Access 2000

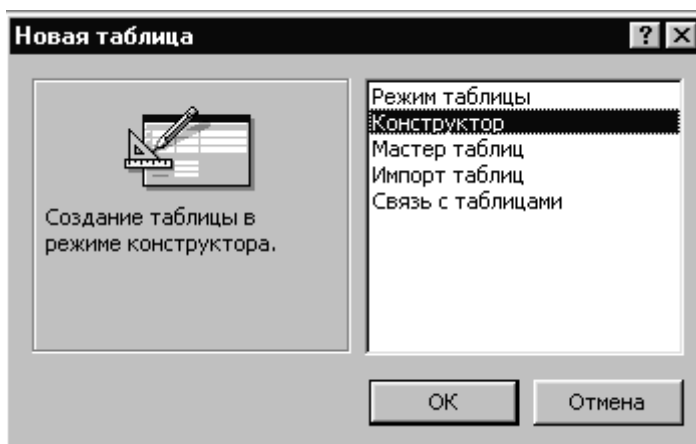


Рис. 3. Выбор способа создания таблицы

Принятая почти во всех современных СУБД реляционная модель предполагает, что каждому предмету соответствует одна запись – строка таблицы, а каждой характеристике выделено поле фиксированной длины – столбец таблицы. Таким образом, запись также имеет фиксированную длину, что значительно облегчает доступ к записям таблицы.

Начнем формирование базы данных с организации ввода информации о поставщиках. Каждому поставщику будет соответствовать запись таблицы. Для реализации замысла прежде всего требуется создать структуру соответствующей таблицы. Выделим набор наиболее важных характеристик поставщика: название фирмы, адрес, директор, контактные телефоны и другая полезная информация... Список может оказаться довольно внушительным.

Для создания таблицы надо войти в раздел «Таблицы» окна базы данных, нажать кнопку «Создать», в открывшемся диалоговом окне (рис. 3) выбрать пункт меню «Конструктор» и нажать «ОК». На экране появится окно «Конструктора таблиц» (рис. 4). В этом режиме формируется структура таблицы. Верхняя часть представляет собой таблицу, состоящую из трех столбцов: имя поля, тип данных, описание. Имя поля обозначает соответствующий параметр предмета. Тип данных выбирается из списка: Текстовый, Поле МЕМО, Числовой, Дата/время, Денежный, Счетчик, Логический, Поле объекта OLE, Гиперссылка, Мастер подстановок. Поскольку данное пособие не претендует на роль справочника, типы данных мы будем рассматривать по ходу разработки структур таблиц. В столбце «Описание» для каждого из полей можно ввести пояснительный текст. Итак, приступим к формированию структуры таблицы.

- Первому полю дадим имя «поставщик» и присвоим тип данных «Счетчик». Этот тип данных представляет собой натуральное число. В первой записи устанавливается значение поля 1. Далее при вводе каждой новой записи это значение автоматически увеличивается на единицу. Таким образом, при заполнении таблицы это поле заполняется автоматически и естественным образом обеспечивается уникальность его значения. Сделаем это поле ключевым. Для этого щелкнем клавишей мыши по значку «ключик» на соответствующей панели инструментов окна приложения. Убедимся, что слева от имени поля «поставщик» появился «ключик». Объявление поля «ключевым» обеспечивает уникальность содержимого этого поля. Поскольку поле типа «Счетчик» и так имеет уникальное значение, может возникнуть впечатление, что «ключик» при нем – просто излишество. Тем не менее в дальнейшем мы убедимся, что последнее действие не лишено смысла.

- Следующему полю (рис. 4) дадим имя «название» и присвоим тип «Текстовый». Это поле будет хранить название фирмы-поставщика. Теперь настало время обратить внимание на набор свойств поля, расположенный внизу окна конструктора. В общем случае этот набор зависит

от типа поля. Так, для типа «Текстовый» первым будет свойство «Размер поля». По умолчанию размер обычно равен 50. Допустив, что наши поставщики не имеют таких длинных названий, исправим последнее значение на 30. Сделаем это поле обязательным, установив в соответствующей строке значение «Да». Теперь система откажется вносить новые записи, в которых это поле не заполнено. Остальные поля обязательными делать не будем, дабы не брать на себя заботу повсеместно заполнять их!

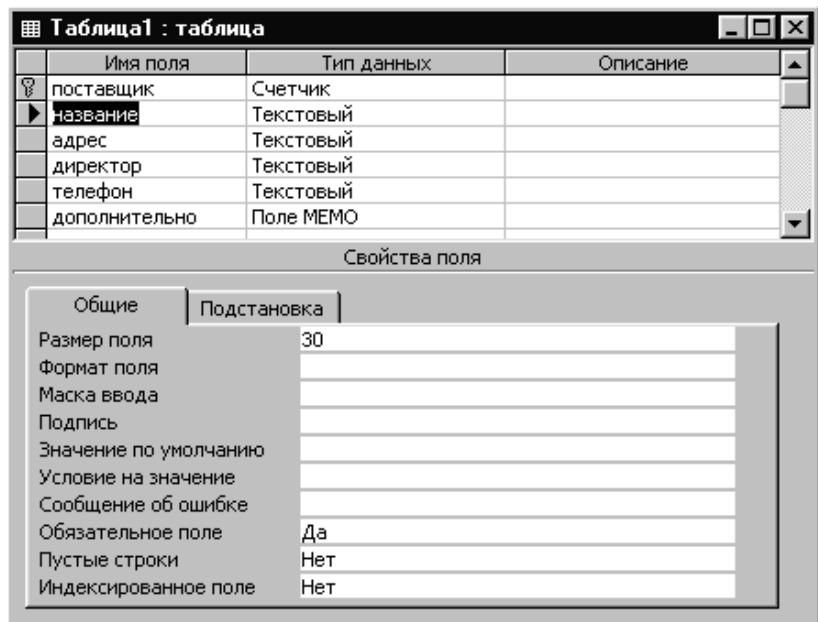


Рис. 4. Создание таблицы в режиме конструктора

- Поле «адрес» также сделаем текстовым длины 40.
- «Директор» пусть будет текстовым полем длины 25.
- «Телефон» также определим как текстовое поле. Выбор длины оставляем на усмотрение разработчика. При этом в качестве маски ввода введем соответствующей длины последовательность цифр «9». Теперь на этих местах могут стоять только цифры или пробелы. Любой другой символ при вводе будет отвергнут. Маска может не только обеспечить ввод форматированных данных, но и преобразовать, например,

все вводимые символы к верхнему регистру или вставить в нужные позиции скобки и дефисы.

- Особое место занимает поле «МЕМО». Как уже было отмечено выше, реляционная модель предполагает равенство в каждой записи длин соответствующих полей. Часто пользователь хочет оставить за собой возможность ввода дополнительной текстовой информации о предмете. Во-первых, не все можно предусмотреть на стадии разработки. Во-вторых, текстовая строка, которую Вы хотели бы разместить в данном поле, может иметь широкий диапазон длин. Тогда, зарезервировав небольшую длину, Вы лишаете себя возможности в некоторых записях хранить весь текст. Зарезервировав же максимально возможную длину, Вы существенно увеличиваете объем каждой записи, а значит, и всей таблицы, что ведет не только к неэффективному использованию памяти, но и к замедлению значительной части операций над данными таблицы. Поле «МЕМО» решает эту задачу путем компромисса: текстовое поле имеет неограниченную длину, но зато хранится вне таблицы. В таблице хранится только ссылка на текст. Таким образом, доступ к информации, хранящейся в полях «МЕМО», значительно замедляется и эти поля нецелесообразно брать за основу сортировки или использовать в операциях типа поиска. Такие поля, как правило, служат для того, чтобы пользователь имел возможность получить более подробную или не укладывающуюся в структуру таблицы информацию об отдельном предмете. Сформировав структуру таблицы, выйдем из «Конструктора». Нажмем «крестик» в правом верхнем углу окна «Конструктора» и в появившемся диалоговом окне на вопрос «Сохранить изменения?» ответим «Да». В окне «Сохранение» дадим таблице имя «поставщики». Любым объектам базы данных желательно присваивать имена, непосредственно указывающие на их назначение! Теперь в разделе «Таблицы» окна базы данных появится имя первой таблицы (рис. 5).

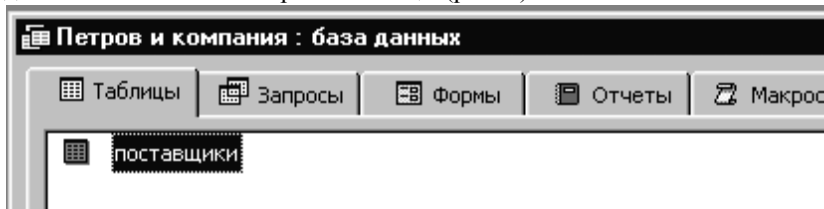


Рис. 5. Создана таблица «Поставщики»

В дальнейшем, если возникнет необходимость изменить структуру таблицы, надо выделить ее имя и нажать кнопку «Конструктор» в окне базы данных. Теперь можно приступить к вводу данных. При вводе поле «поставщик», которому мы присвоили тип «Счетчик», заполняется автоматически. Для удобства ввода можно расширить изображение любого столбца на экране, установив указатель курсора на границе между названиями соответствующих полей, нажав левую клавишу мыши и передвинув границу вправо. Но при большой длине записи нам придется постоянно пользоваться полосой прокрутки.

поставщики : таблица						
	поставщик	название	адрес	директор	телефон	допол
▶	2	Заборская макар	г. Заборск,	Епифанов	776655	
	3	Урюпинский рыбо	г. Урюпинс	Репин Сид	242325	
	4	Лабаз ООО	г. Заборск,	Кирибеич	228872	
	5	Зеленый чай ООС	г. Слоновс	Набобов Е	332117	
	6	Апельсиновый ра	г. Южногор	Хабибулин	213345	
	7	Семиструйский м	г. Семистр	Скуратом	233244	
	8	Кузякинская мака	г. Кузякин,	Бурлаков	125419	
	9	Сельхозпродукт п	п. Горохов	Пупков Се	1256	
	10	Садоводческое тс	п. Жучкино	Мичурин С	2345	
	11	Семиструйский хл	г. Семистр	Калашник	237765	
	12	Лукоморская рыб	п. Лукомор	Дубов Вит	5412	
	13	Кузякинский хлеб	г. Кузякин,	Зубов Лав	345434	
	14	Семиструйские м	г. Семистр	Пивков Ле	338821	
	15	Урюпинский мясо	г. Урюпинс	Мясоедов	217652	
	16	Дары юга ООО	г. Кошарск	Абдулов А	345412	
	17	Севливер ООО	п. Козлодо	Пузанов А	2543	
	18	Буренка АО	п. Жучкино	Мурлыкин	1243	
	19	Лукоморский мол	п. Лукомор	Черномор	3412	
	20	Урюпинская конд	г. Урюпинс	Осипов Ил	875623	
	21	Кузмолоко ООО	г. Кузякинс	Косарев И	127734	
*	(Счетчик)					

Рис. 6. Заполненная таблица «поставщики»

В любом случае в таблице неудобно заполнять поля «МЕМО», содержащие произвольной длины текст. Итак, у нас возникло два противоре-

чивых желания. С одной стороны, мы хотели бы иметь возможность просматривать все записи, ограничиваясь небольшим количеством наиболее важных полей. С другой стороны, иногда желательно видеть одновременно всю информацию о предмете. Разрешением этого противоречия мы займемся в следующем разделе.

1.3. Создание запроса. Запрос представляет собой логическую таблицу и позволяет пользователю получить нужные данные из одной или нескольких таблиц. Запросы можно создавать на выборку, обновление, удаление или добавление данных. С помощью запросов, используя информацию из одной или нескольких таблиц, можно создать новую таблицу. Более подробно с запросами мы познакомимся по ходу разработки приложения, а пока организуем просмотр списка поставщиков. Для этого перейдем в раздел «Запросы» окна базы данных и нажмем «Создать».

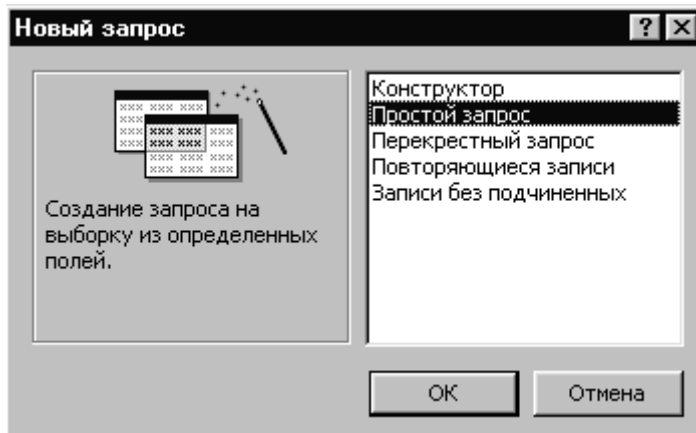


Рис. 7. Окно диалога «Новый запрос»

Выберем пункт меню «Простой запрос» и нажмем «ОК». Появится окно «Создание простых запросов» (рис. 8). В поле «Таблицы/запросы» развернем меню и выберем таблицу «поставщики» (других таблиц у нас пока и нет!). Внизу окна слева размещается список всех полей указанной таблицы, а справа список отобранных полей. Стрелки между списками означают:

- > Перенести поле в список отобранных полей
- >> Перенести все поля в список отобранных полей

- < Исключить одно поле из списка отобранных полей
- << Исключить все поля из списка отобранных полей

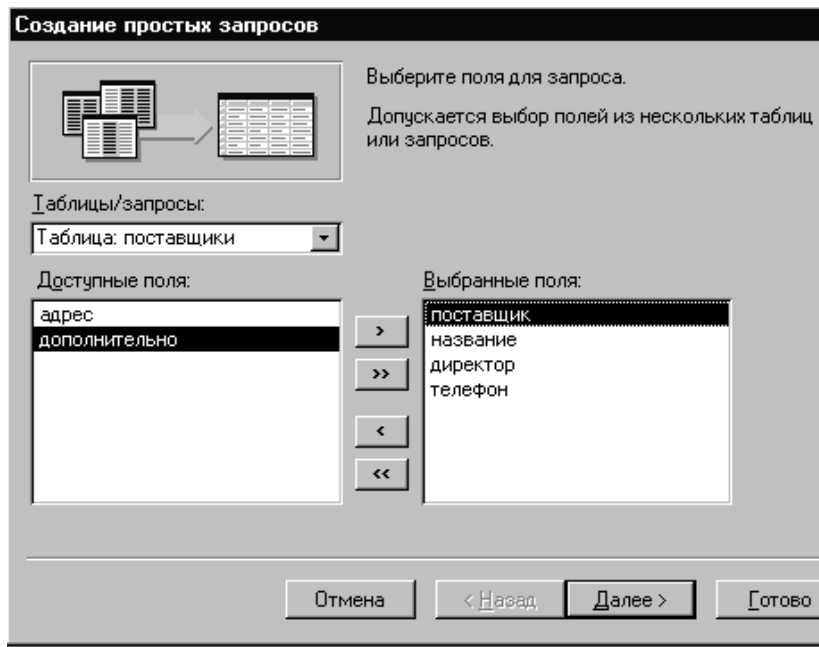


Рис. 8. Отбор полей для запроса

Мы отберем для запроса поля: поставщик, название, директор и телефон. Нажимаем «Дальше». В следующем окне предлагается дать имя запросу. Оставим предложенное системой имя «поставщики Запрос» и нажмем «Готово». На экране появится таблица (рис. 9). Эта таблица содержит выбранные нами поля таблицы «поставщики».

поставщики Запрос : запрос на выборку				
	поставщи	название	директор	телефо
▶	2	Заборская макаронная фабр	Елифанов Кирил Петрович	776655
	3	Урюпинский рыбозавод	Репин Сидор Поликарпович	242325
	4	Лабаз ООО	Кирибевч Василий Петрович	228872

Рис. 9. Результат выполнения запроса

Закроем таблицу и на вопрос «Сохранить изменения?» ответим «Да». Теперь в разделе «Запросы» окна базы данных появился первый объект – запрос «поставщики Запрос». Откроем созданный запрос в режиме конструктора. Для этого выделим запрос и нажмем «Конструктор». На экране появится окно «Конструктора запросов» (рис. 10).

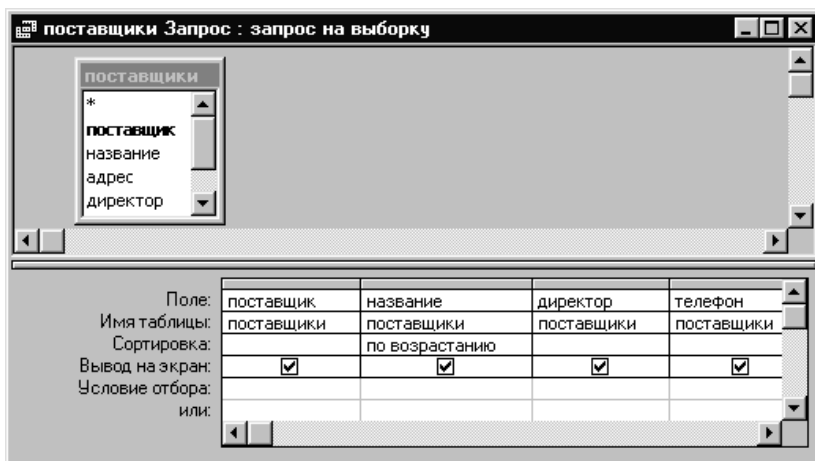


Рис. 10. Окно конструктора запросов

Следует заметить, что в режим конструктора мы могли бы перейти сразу, не закрывая запрос. Для этого достаточно нажать на изображение «зеленого угольника» в левом верхнем углу окна приложения.

Как видно из рисунка, в верхней части окна конструктора расположены таблицы (в данном случае одна таблица), которые послужили источником данных для запроса. В нижней части – таблица, столбцы которой представляют поля запроса, а в строках указываются характеристики полей.

- Поле. В этой строке задаются имена полей, которые должны попасть в запрос или используются только для задания условий отбора и сортировки данных из таблицы.
- Имя таблицы. Указывается имя таблицы, из которой выбрано поле.
- Сортировка. В этой строке для любого поля можно развернуть меню и выбрать способ сортировки (по возрастанию или по убыванию) либо отказаться от сортировки.

- Вывод на экран. В этой строке помечены «галочкой» поля, которые должны выводиться на экран. Щелчком левой клавиши мышки можно изменить соответствующую характеристику поля.
- Условие отбора. Здесь задаются условия отбора записей из таблицы. Более подробно этот вопрос мы рассмотрим в процессе разработки приложения.

Припишем полю «название» в строке «Сортировка» характеристику «по возрастанию».

При формировании структуры таблицы мы обратили внимание на то, что поля могут обладать рядом свойств (рис. 4). Поля, представленные в запросе, наследуют свойства соответствующих полей таблицы. Если возникнет необходимость изменить некоторые из них, Вы можете щелкнуть правой клавишей мыши на интересующем Вас поле и в развернувшемся меню выбрать пункт «Свойства». Появится диалоговое окно, в котором Вы можете изменить ряд свойств поля.

Итак, мы создали запрос в режиме диалога, а затем в режиме конструктора внесли в него необходимые изменения. В большинстве случаев такой подход наиболее целесообразен, хотя запрос можно строить и изначально в режиме конструктора.

Теперь выйдем из конструктора, сохранив изменения, и откроем запрос «поставщики Запрос». Вы можете убедиться, что данные представлены в алфавитном порядке (по названиям поставщиков). Таким образом, одно из наших пожеланий удалось выполнить. А именно, мы получили возможность, пользуясь полосами прокрутки, быстро найти нужного поставщика и наиболее важную информацию о нем.

2. Шаг второй. Формы и элементы управления

В предыдущем разделе мы заметили, что ввод, редактирование и просмотр данных непосредственно в таблицах и запросах не всегда приятное занятие. Кроме того, на предприятии с одной базой данных могут работать разные категории удаленных пользователей, у которых разные интересы и разные права доступа. Это также одна из причин применения форм – основного средства организации интерфейса между пользователем и базой данных. Форма – объект Microsoft Access, предназначенный для ввода и просмотра данных, а также для управления работой приложения.

2.1. Создание формы с полями, расположенными в столбец. Теперь пришло время осуществить второе желание – предоставить пользователю возможность просмотра одновременно всех полей, относящихся к конкретному поставщику. Для этого перейдем в раздел «Формы» окна базы данных и нажмем «Создать».

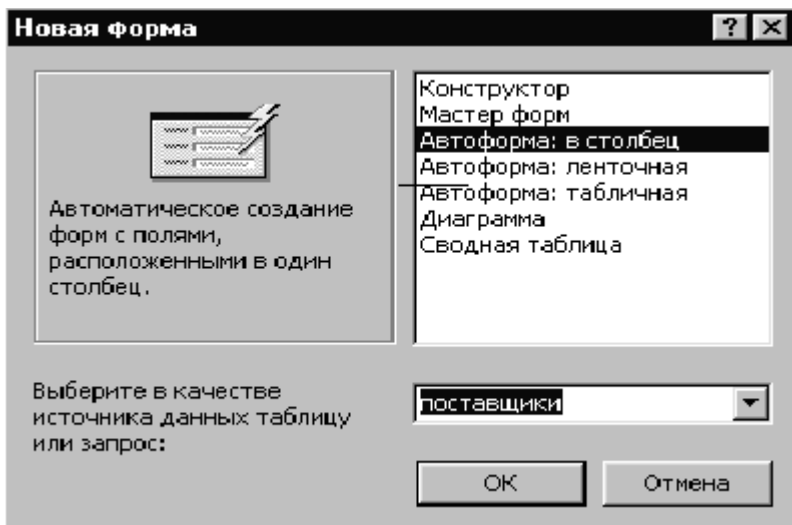


Рис. 11. Создание автоформы «в столбец»

На экране появится диалоговое окно «Новая форма» (рис. 11). Выберем «автоформу в столбец», в нижней части окна развернем меню и в качестве источника данных для формы назначим таблицу «поставщики». Теперь для создания формы достаточно нажать «ОК». На рис. 12 представлена первая созданная нами форма «в столбец». Справа от названий полей расположены сами поля ввода/вывода. В этих полях Вы можете просматривать и редактировать данные о конкретном производителе. Полю «дополнительно» мы приписали тип «МЕМО», и, таким образом, это поле может хранить текстовую строку произвольной длины. При вводе данных в этом поле справа появляется полоса прокрутки. Часть формы, в которой расположены поля ввода/вывода, называется областью данных. В правом верхнем углу находятся кнопки оконного меню, которые в комментариях не нуждаются. В левой части окна формы Вы видите помеченную стрелкой область выделения. Если по этой области

щелкнуть левой клавишей мыши (область окрасится в черный цвет) и нажать на клавиатуре компьютера клавишу «Delete», текущая запись будет удалена. Внизу формы расположено поле номера записи (рис. 13).

Рис. 12. Форма «в столбец»

Пользуясь элементами управления этого поля, Вы можете перемещаться по записям, а также вводить новые записи. Закроем форму и на вопрос «Сохранить изменения?» ответим «Да». В диалоговом окне «Сохранение» зададим имя формы «поставщики_столбец». Теперь в разделе «Формы» окна базы данных появился первый объект.



Рис. 13. Поле номера записи

Созданная в автоматическом режиме форма не всегда удовлетворяет нашим требованиям. Так, в представленной на рис. 12 форме не полностью умещается название поставщика. Для корректировки формы откроем ее в режиме конструктора. Это делается точно так же, как для таблиц и запросов. В результате на экране появится окно конструктора форм (рис. 14).

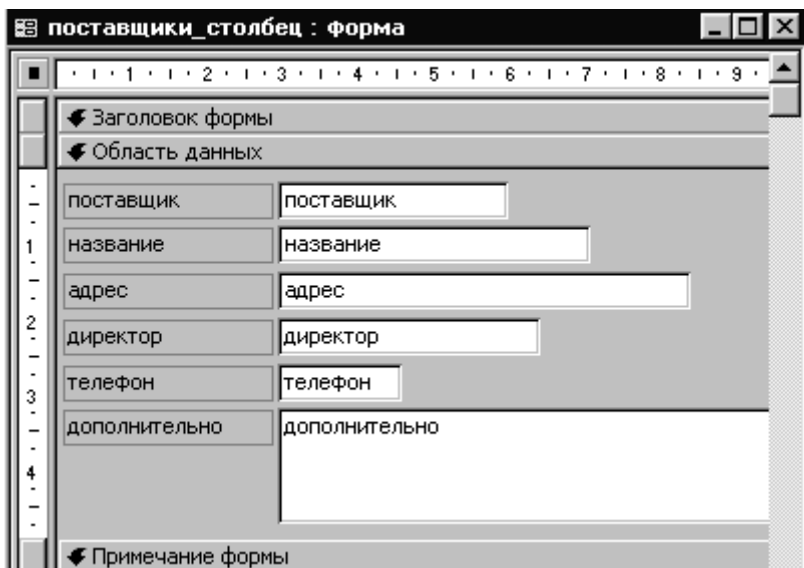


Рис. 14. Форма в режиме конструктора

Объект базы данных «Форма» сам является контейнером для объектов, которые принято называть «элементы управления». Все поля ввода/вывода и надписи, которые мы видели на рис. 12, являются элементами управления (в дальнейшем мы познакомимся с рядом других). Для простейших манипуляций с любым из представленных на рис. 14 элементов управления прежде всего надо выделить нужный элемент. Теперь, если Вы нажмете на клавиатуре компьютера клавишу «Delete», элемент управления будет удален. Если Вы установите указатель курсора в левый верхний угол элемента, курсор примет форму «указывающей» руки. Тогда, нажав (и не отпуская) левую клавишу мыши, Вы можете переместить элемент управления. Установив указатель на одном из черных прямоугольников, расположенных по углам или на сторонах

прямоугольника, ограничивающего выделенный элемент управления (курсор тогда примет форму $\leftarrow\rightarrow$), и нажав левую клавишу мыши, Вы можете соответствующим движением руки изменить размер элемента управления. Наконец, нажав левой клавишей на невыделенный элемент управления (появляется ладонь), Вы, не отпуская клавиши, можете переместить элемент управления.

Если пользователь работает с созданной Вами формой часто, с некоторого момента, возможно, его начнет раздражать необходимость каждый раз при выходе нажимать на маленький крестик. В таком случае можно предусмотреть кнопку выхода. Кнопку следует взять с «Панели элементов» (рис. 15). Но, прежде, поясним назначение представленных на панели объектов.

- Выбор объектов (по умолчанию кнопка этого инструмента нажата) – инструмент используется для выделения, изменения размеров, перемещения и редактирования элементов управления.
- Кнопка «Мастера» активизирует «Мастера по созданию элементов управления». Эту кнопку мы всегда будем держать в состоянии, изображенном на рис. 15.
- Надпись – инструмент, предназначенный для создания элемента управления, содержащего постоянный текст.
- Поле – инструмент для создания элемента управления, предназначенного для отображения, редактирования и ввода текстов, чисел, даты, времени и вычисляемых полей.
- Группа переключателей – инструмент для создания группы переключателей. Когда переключатели объединены в группу, каждому переключателю соответствует отдельное числовое значение. Одновременно всегда установлен лишь один переключатель. Численное значение, приписанное установленному переключателю, является значением всей группы.
- Выключить – инструмент, создающий элемент управления «Выключатель», принимающий одно из двух значений Вкл/Выкл, Истина/Ложь или Да/Нет. Вкл, Истина и Да представляются значением «-1». Выкл, Ложь и Нет представляются значением «0».
- Переключатель – инструмент, создающий элемент управления «Переключатель», иногда называемый также радиокнопкой и принимающий одно из двух значений Вкл/Выкл, Истина/Ложь или Да/Нет. Вкл, Истина и Да представляются значением «-1». Выкл, Ложь и Нет представляются значением «0».

- Флажок – создает одноименный элемент управления, принимающий одно из двух значений Вкл/Выкл, Истина/Ложь или Да/Нет. Вкл, Истина и Да представляются значением «-1». Выкл, Ложь и Нет представляются значением «0».

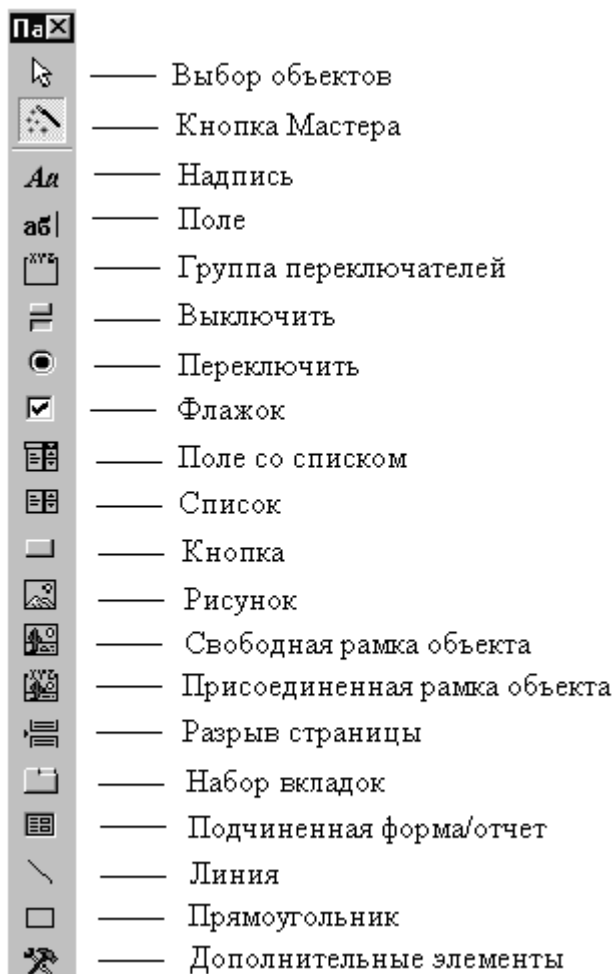


Рис. 15. Панель элементов

- Поле со списком – создает элемент управления, который задает поле и список. Значение поля выбирается из списка.
 - Список – создает элемент управления, содержащий перечень возможных значений. Список всегда раскрыт, а выделенное значение присваивается элементу управления.
 - Кнопка – создает командную кнопку, запускающую макрос или процедуру встроенного в Microsoft Access языка Visual Basic for Applications (VBA).
 - Рисунок – позволяет поместить в форму статический рисунок.
 - Свободная рамка объекта – используется для включения в форму объекта из приложения, поддерживающего OLE (Object Linking and Embedding). Технология OLE позволяет устанавливать связь с объектами другого приложения или внедрять некоторые объекты в базу данных Access.
 - Присоединенная рамка используется для включения в форму объекта OLE, хранящегося в таблице Access.
 - Разрыв страницы используют для создания многостраничной формы.
 - Набор вкладок – инструмент, позволяющий создать в форме несколько вкладок, в которых можно разместить различные элементы управления.
 - Подчиненная форма используется для внедрения в текущую форму некоторой другой формы.
 - Линия – инструмент, позволяющий включать в форму прямые линии.
 - Прямоугольник – инструмент для создания в форме прямоугольников. Например, в прямоугольник можно заключить набор близких по содержанию полей.
 - Дополнительные элементы – кнопка, открывающая список установленных в системе дополнительных элементов управления. Однако не все эти элементы могут работать в Access.
- Итак, приступим к созданию кнопки. Для этого щелкнем левой клавишей мыши по элементу «Кнопка», а затем щелкнем той же клавишей в примечании формы (ниже заголовка «Примечание формы»). Появится диалоговое окно (рис. 16).

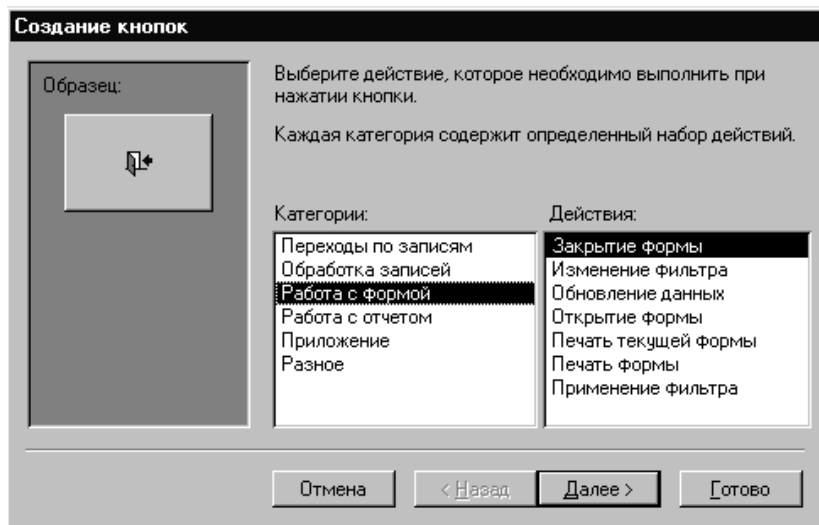


Рис. 16. Выбор назначения кнопки

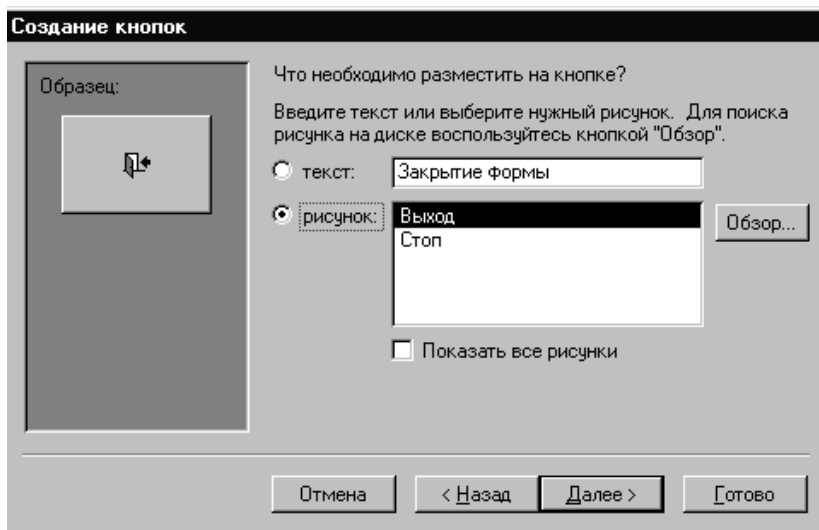


Рис. 17. Выбор оформления кнопки

В разделе «Категории» диалогового окна выберем пункт меню «Работа с формой», а в разделе «Действия» – «Закрытие формы». Теперь нажмем «Далее». Появится диалоговое окно (рис. 17). В новом диалоговом окне задается оформление кнопки. Если выбрать переключатель «текст» и в поле, расположенном справа от переключателя, ввести текст, этот текст будет расположен непосредственно на кнопке. Мы оставим значение переключателя «рисунок». В левой части окна изображен рисунок, который будет находиться на кнопке. Если Вы хотите выбрать другой рисунок, то должны установить «флажок», расположенный возле надписи «Показать все рисунки». Тогда в поле «рисунок» развернется список имеющихся в нашем распоряжении стандартных рисунков, и Вы можете выбрать тот, который Вас больше устраивает. В данном случае предложенный рисунок нас вполне устраивает и мы сразу нажмем «Далее». На экране появится последнее диалоговое окно, в котором следует задать имя элемента управления. Имя, предлагаемое системой, нам ничего не говорит. По ряду причин полезно давать объектам содержательные имена. В дальнейшем мы еще коснемся этого вопроса. Итак, дадим кнопке имя «Выход» и нажмем «Готово».

На этом работу в режиме конструктора закончим, закроем форму и, как прежде, подтвердим желание сохранить изменения. Откроем форму в рабочем режиме и убедимся, что при нажатии на кнопку «с дверью» форма действительно закрывается.

Созданная форма удобна для просмотра, ввода и редактирования информации, относящейся к конкретному поставщику. Однако при большом количестве записей найти нужного производителя в такой форме не легко. Прежде чем приступить к разрешению противоречия между эффективностью поиска и удобством просмотра, мы должны создать еще одну форму.

2.2. Создание ленточной формы. В разделе «Формы» окна базы данных нажмем кнопку «Создать». На экране появится диалоговое окно «Новая форма» (рис. 19). Выберем в меню ленточную форму. Затем в нижней части окна развернем меню, выберем источник данных для формы – запрос «поставщики Запрос» и нажмем «ОК». На экране появится форма (рис. 20). Закроем форму и подтвердим желание сохранить ее. В диалоговом окне «Сохранение» зададим имя формы «поставщики_лента».

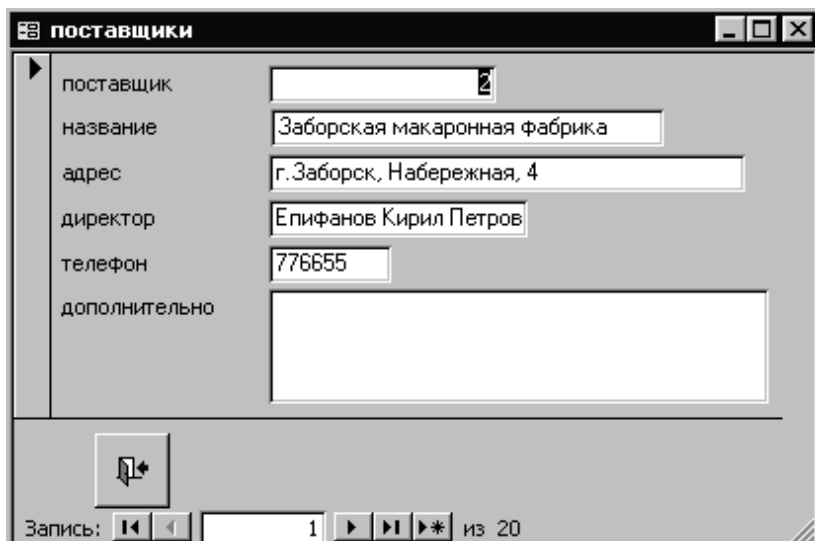


Рис. 18. Окончательный вид формы

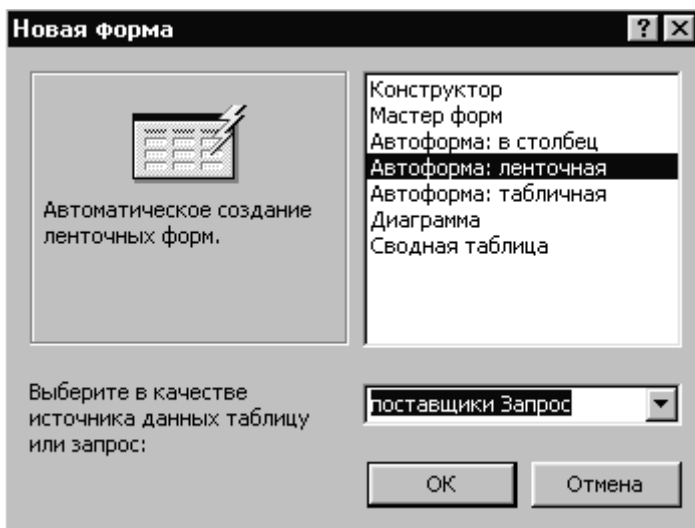


Рис. 19. Создание ленточной формы

Теперь в окне базы данных выделим объект «поставщики_лента» и нажмем кнопку «Конструктор». Ленточная форма открыта в режиме кон-

структора (рис. 21). Если есть необходимость, расширим рамку формы. Для этого установим указатель курсора на рамку (курсор примет вид $\leftarrow\rightarrow$) и, не отрывая руки от клавиши, переместим рамку. Затем, таким же образом, увеличим ширину формы.



Рис. 20. Ленточная форма



Рис. 21. Ленточная форма в режиме конструктора

Теперь рассмотрим окно конструктора формы. Окно разбито на три области: заголовок формы, область данных (рис. 21) и примечание формы. Перетащим вправо поле «телефон» и его заголовок, а затем расширим поле «директор» так, чтобы текст полностью умещался в данном поле. При необходимости увеличим размер поля «поставщик». Наконец, так же, как и в предыдущем разделе, создадим кнопку выхода. Очевидно, ее надо поместить в примечание формы. Закроем конструктор, подтвердив сохранение. Итак, мы создали две формы. Первая удобна для просмотра отдельной записи, но неудобна для поиска записи. Вторая, наоборот,

удобна для поиска записи, но не дает возможность просмотреть всю информацию о поставщике.

3. Шаг третий. Свойства форм

3.1. Установка связи между формами. Следующая наша задача – связать две ранее созданные нами формы. Для этого откроем форму «поставщики_лента» в режиме конструктора, возьмем с панели элементов кнопку и поместим ее в примечание формы. В открывшемся диалоговом окне «Создание кнопок» выберем категорию «Работа с формой» и действие «Открытие формы» (рис. 22).

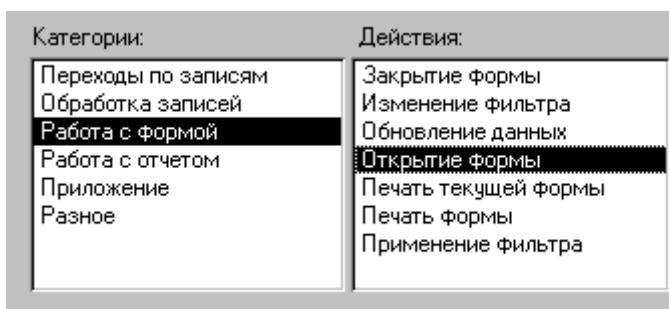


Рис. 22. Открытие формы

Нажмем «Далее», в списке выберем форму «поставщики_столбец» (рис. 23) и опять нажмем «Далее».

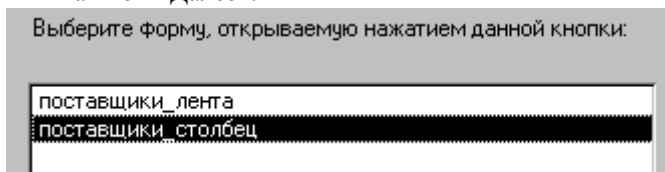


Рис. 23. Выбор формы

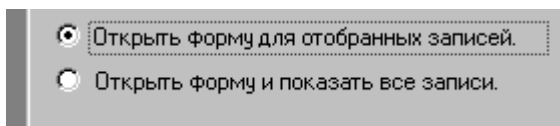


Рис. 24. Выбор способа открытия формы

Нас интересует только конкретная, выбранная в форме «поставщики_лента» запись, поэтому установим переключатель «Открыть форму для отобранных записей» (рис. 24) и нажмем «Далее».

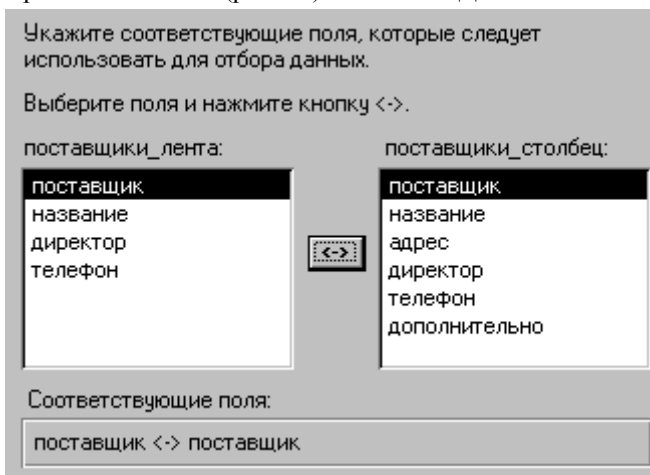


Рис. 25. Установка соответствия между полями форм

В диалоговом окне (рис. 25) Вы видите два списка полей: «поставщики_лента» (слева) и «поставщики_столбец» (справа). Выделим в обоих списках поле «поставщик». Мы знаем, что это поле содержит уникальный код поставщика, и поэтому целесообразно именно его использовать для создания связи между таблицами. Нажмем кнопку \leftrightarrow , расположенную между списками полей. Ниже заголовка «Соответствующие поля» появится текст «поставщик \leftrightarrow поставщик». Теперь, при нажатии на создаваемую нами кнопку, должна открываться форма «поставщики_столбец» с одной лишь записью, соответствующей выбранному поставщику. Нажмем «Далее», оставим переключатель в положении «Рисунок», но будем считать, что на этот раз предложенный системой рисунок нас не устраивает. Поэтому установим «флажок» в поле «Показать все рисунки». В поле «рисунок» появится список названий стандартных рисунков. При движении по списку в поле с заголовком «Образец» будет появляться сам рисунок. Выберем «Бинокль 1» (рис. 26) и нажмем кнопку «Далее». В поле под заголовком «Задайте имя кнопки» (рис. 27) зададим имя «вход». Мы сейчас не претендуем на самый удачный выбор имен. Главное, чтобы имя говорило разработчику о назначе-

нии объекта и, иногда, желательно, чтобы имя не было слишком длинным.

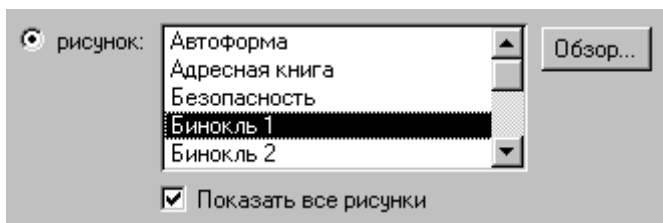


Рис. 26. Выбор рисунка

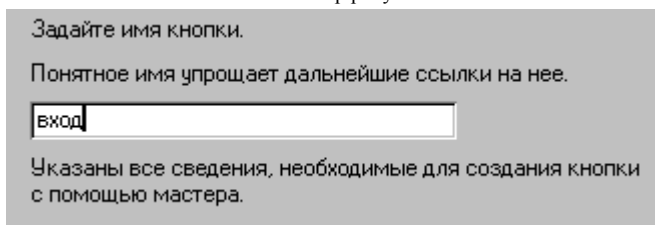


Рис. 27. Имя кнопки

Нажмем «Готово». После закрытия диалогового окна мы обнаружим в конструкторе ленточной формы новую кнопку. Переместим ее в примечании формы в место, которое нам представляется наиболее удачным, и выйдем из режима конструктора, подтвердив сохранение изменений.

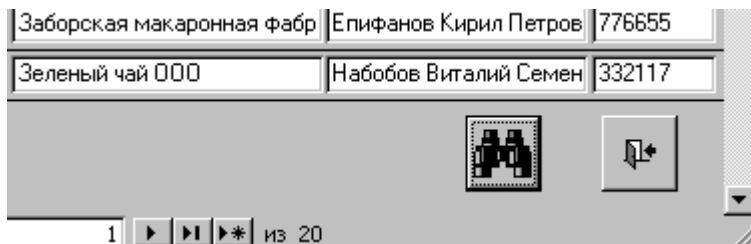


Рис. 28. Новая кнопка

Откроем форму «поставщики_лента». В нижней части формы Вы увидите кнопку с изображением бинокля. Теперь, чтобы получить подробную информацию о поставщике, Вам достаточно, пользуясь полосами прокрутки, выбрать нужного поставщика и нажать «Бинокль».

поставщик	название	директор
6	Апельсиновый рай АО	Хабибулин Фед
18	Буренка АО	Мурлыкин Иван

поставщики	
поставщик	18
название	Буренка АО
адрес	п.Жучкино, Липовая, 21
директор	Мурлыкин Иван Ильич
телефон	1243
дополнительно	

Запись: из 1 (Фильтр)

Рис. 29. Получение подробной информации о поставщике

3.2. Ввод и удаление записей. Теперь, когда мы разработали простейшую систему для поиска и просмотра записей о поставщиках, настало время определиться, в какой из форм будет производиться ввод и удаление записей. Очевидно, эти действия, дабы избежать противоречивых ситуаций, должны производиться в ленточной форме! Почему? Предлагаем читателю подумать над этим. Откроем ленточную форму. Для ввода новой записи достаточно в поле номера записи нажать значок «Новая запись» (рис. 13). Тогда в положении «текущая» окажется пустая запись (рис. 30). Этому же эффекта можно добиться, опустив вниз до конца полосу прокрутки. Для ввода записи необходимо заполнить обязательное поле «название», так как поле «поставщик» заполняется автоматически, а остальные поля – необязательные. Если записей слишком много и Вам не хочется каждый раз пользоваться полосой прокрутки, то можно создать еще одну кнопку – «Добавление записи».

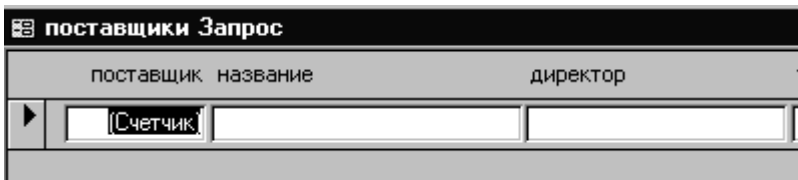


Рис. 30. Пустая запись

Для этого в режиме конструктора возьмем с панели элементов кнопку, поместим ее в примечание формы и в открывшемся диалоговом окне выберем категорию «Обработка записей» и действие «Добавление записи» (рис. 31).

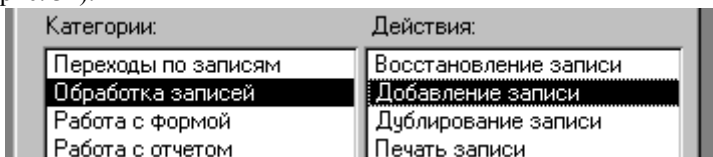


Рис. 31. Создание кнопки «Добавление записи»

Нажмем «Далее» и посмотрим предлагаемый системой рисунок. Пусть вид этого рисунка нас «устраивает» и мы еще раз нажмем «Далее», дадим кнопке имя «добавить», нажмем «Готово» и выйдем из режима конструктора. Откроем ленточную форму и для проверки введем новую запись.

Внимание! При вводе новой записи в области выделения (слева от записи) появляется изображение карандаша. Пока высвечивается изображение карандаша, запись фактически не является сохраненной в таблице и «биноклем» пользоваться мы не можем. Сохранить текущую запись можно, например, выделив другую запись. Однако лучше в режиме конструктора создать еще одну кнопку. В соответствующем диалоге выберем категорию «Обработка записей» и действие «Сохранение записи». Дадим кнопке имя «сохранить». Теперь после ввода, по крайней мере, поля «название» и нажатия кнопки «сохранить» мы можем нажать «бинокль» и внести более подробную информацию.

Примечание формы содержит уже четыре кнопки. На случай, если пользователь забудет их назначение, мы можем в режиме конструктора добавить к ним надписи. Для этого возьмем соответствующий объект с «Панели элементов» (рис. 15). На рис. 32 представлен окончательный вид ленточной формы. Поставщик с кодом 27 только что введен нами для проверки работы приложения.

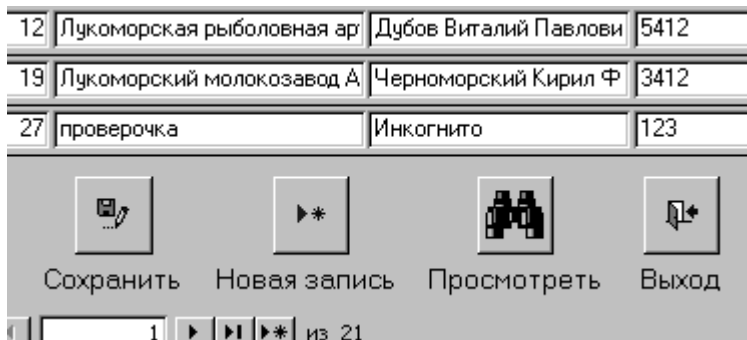


Рис. 32. Окончательный вид ленточной формы

Для удаления записи нужно щелкнуть левой клавишей по полосе выделения слева от соответствующей записи. Полоса окрасится в черный цвет. Теперь для удаления записи достаточно нажать клавишу «Delete». На рис. 33 после нажатия этой клавиши будет удалена запись, соответствующая поставщику с кодом 27 (название – «проверочка»).

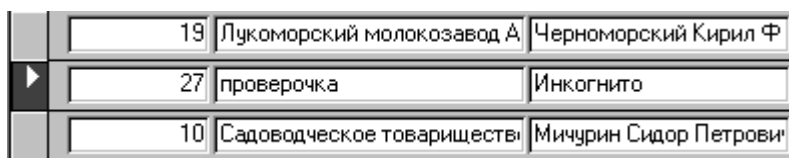


Рис. 33. Поставщик «проверочка» подготовлен для удаления

3.3. Свойства формы. Каждый объект Microsoft Access обладает некоторым набором свойств, определяющих форму, параметры этого объекта, а также его реакцию на события, которые могут с этим объектом происходить. Свойства, приписанные объекту при его создании, не всегда устраивают нас. Так, если мы в ленточной форме нажмем на «бинокль», то на фоне ленточной формы появится форма «в столбец». Теперь, допустим, мы случайно щелкнули левой клавишей мыши за пределами формы «в столбец» по ленточной форме. Тогда на первый план выйдет ленточная форма, закрыв собой форму, интересующую нас. Конечно, это не катастрофа и мы можем просто сдвинуть на экране ленточную форму. Однако есть и другие проблемы. Так, можно закрыть ленточную форму до закрытия формы «в столбец». В данном случае и это не страшно. Но, иногда, имеет принципиальное значение такой порядок, при котором первой должна закрываться только форма, открытая

последней. Установить такой порядок можно, изменив значение соответствующего свойства, сделав окно «модальным».

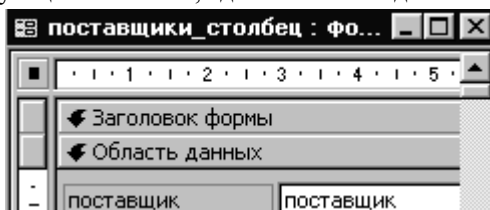


Рис. 34. Левый верхний угол окна формы



Рис. 35. Окно свойств

Для просмотра и изменения свойств формы достаточно дважды щелкнуть левой клавишей по квадратику, расположенному в левом верхнем углу окна формы слева от линейки (рис. 35). Этот квадратик устанавливается или убирается вместе с линейкой командой меню окна приложения «Вид → Линейка». Окно свойств имеет пять заставок: **Макет**, **Данные**, **События**, **Другие**, **Все**. Каждая заставка представляет собой список, в левой части которого указано свойство (то есть название свойства), а в правой – значение свойства. При щелчке левой клавиши мыши в соответствующей строке списка иногда появляется направленная вниз стрелка – предложение развернуть меню. В таком случае значение свойства выбирается из меню. Изменим значения ряда свойств. Для этого выберем заставку «Макет» и установим следующие свойства.

Свойство	<i>Установленное значение</i>	<i>Комментарий</i>
Подпись	Информация о поставщике	Эта подпись всегда расположена вверху формы
Допустимые режимы	Форма	Теперь нельзя в процессе работы перейти в режим таблицы
Полосы прокрутки	Отсутствуют	В рассматриваемой форме мы работаем только с одной записью. Кроме того, мы договорились в этой форме не удалять и не добавлять записей
Область выделения	Нет	
Поле номера записи		
Кнопка оконного меню	Нет	В рабочем режиме мы не будем менять размеры окна или сворачивать окно
Кнопки размеров окна	Отсутствуют	

Теперь перейдем к заставке «Данные». В первой строке установлен источник данных – таблица «поставщики». Установим значения «Нет» для свойств «Разрешить удаление» и «Разрешить добавление». Пропустим заставку «События» и сразу перейдем к заставке «Другие».

Рис. 36. Окно поставщика после изменения значений свойств

Свойству «Модальное окно» присвоим значение «Да», чтобы форма не перекрывалась другими окнами. Закроем окно свойств и покинем режим конструктора. Форма «поставщики_столбец» теперь будет иметь вид, представленный на рис. 36 (сравните с рис. 12). В этой форме Вы не можете перемещаться по записям, удалять или вводить новые записи!

3.4. Последний штрих. При работе с ленточной формой для получения или ввода более подробной информации о поставщике мы сначала устанавливаем запись, как текущую, щелкнув в соответствующей строке левой клавишей мышки, а уже затем нажимаем «бинокль» внизу формы. Эту операцию можно упростить. Откроем ленточную форму в режиме конструктора и дважды щелкнем левой клавишей мышки по кнопке с изображением бинокля. Откроется окно свойств элемента управления (рис. 37).

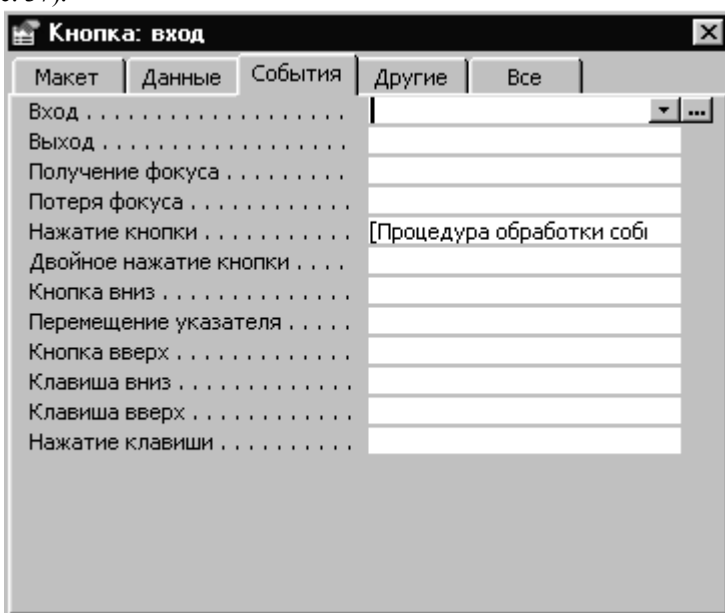


Рис. 37. Окно свойств элемента управления

Обратите внимание на присвоенную окну подпись «Кнопка: вход». Здесь указан класс, к которому принадлежит элемент управления – «Кнопка», и заданное нами в диалоговом окне при создании кнопки имя представителя класса – «вход». Выберем заставку «События». С любым элементом управления могут происходить некоторые события. Набор событий в общем случае зависит от класса, к которому принадлежит объект, и каждому событию можно предписать определенные действия. Набор событий, которые могут происходить с кнопкой, представлен на рис. 37. Подробное истолкование этих событий можно найти в любом справочнике по Microsoft Access. Нас сейчас интересует событие «Нажатие кнопки». Этому событию, как видно из рисунка, соответствует «Процедура обработки событий». Дело в том, что когда, при создании кнопки, мы в диалоговом окне задавали параметры элемента управления и, в частности, связанные с этой кнопкой действия, система сама написала за нас процедуру (то есть небольшую программу) на встроенном языке Visual Basic for Applications (VBA).

Чтобы увидеть текст этой процедуры, надо щелкнуть клавишей мышки по соответствующей строке. В конце строки появится маленькая кнопка с троеточием. После нажатия на эту кнопку откроется «Окно модуля» (рис. 38). В этом окне расположены тексты процедур на VBA, но не всех, а только обслуживающих данную форму. В таких случаях говорят о модуле формы.

Обратите внимание! В верхней части «Окна модуля» слева расположено название элемента управления (вход), а справа – название события (Click). Событие Click как раз и есть нажатие кнопки. Пользуясь полосою прокрутки, просмотрим содержимое окна.

В самом верху Вы увидите две инструкции:

- Option Compare Database – задает сравнение строк на основе порядка сортировки, определяемого национальной настройкой базы данных, в которой производится сравнение строк.
- Option Explicit – налагает на уровне модуля требование на явное описание всех переменных этого модуля.

Дальше следуют тексты четырех процедур, соответствующих четырем созданным нами кнопкам: Private Sub выход_Click(), Private Sub вход_Click(), Private Sub добавить_Click(), Private Sub сохранить_Click(). Каждая процедура начинается с одного из приведенных выше заголовков и заканчивается оператором End Sub. Все, что расположено между заголовком и End Sub, является телом процедуры, со-

державшим собственно ее код, то есть последовательность операторов языка.

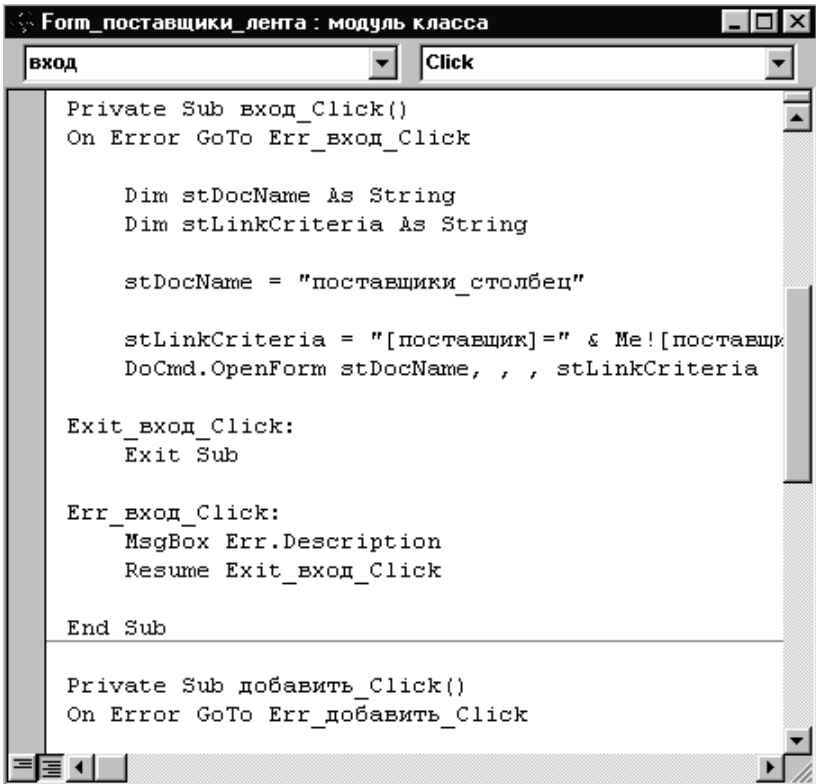


Рис. 38. Окно модуля формы

Первое ключевое слово в заголовке – Private означает, что данная процедура может быть вызвана только из «своей» формы. Далее следует имя процедуры. Имя состоит из двух частей: имени элемента управления (или другого объекта), с которым связана процедура, и события, которое обрабатывается этой процедурой (в данном случае Click). Вот почему мы дали кнопкам содержательные имена! По имени процедуры мы сразу можем сказать, нажатие какой из четырех кнопок она обслуживает.

Приняв к сведению рассмотренный выше материал, закроем окно модуля, а затем и окно свойств. Таким образом, мы вернемся в ленточную

форму, находящуюся в режиме конструктора. Двойным щелчком откроем окно свойств элемента управления «название» в области данных. Появится окно свойств, озаглавленное «Поле: название». Если Вы по ошибке щелкнете по надписи «название» в заголовке формы, то откроется окно «Надпись: название Надпись». Выберем заставку «События» и щелкнем левой клавишей на строке, соответствующей событию «Двойное нажатие кнопки». Мы хотим двойному щелчку по названию поставщика поставить в соответствие процедуру, открывающую форму с подробной информацией о данном поставщике. Щелкнем по трюетичию, как мы поступили выше, когда открывали окно модуля формы. Поскольку событию еще не поставлен в соответствие метод его обработки, откроется диалоговое окно «Построитель» (рис. 39). В этом окне мы выберем строку «Программы» и нажмем «ОК». Откроется окно модуля. В этом окне мы увидим процедуру с пустым телом:

```
Private Sub название_Db1Click(Cancel As Integer)
End Sub
```

Обратите внимание, что имя процедуры – название_Db1Click опять состоит из двух частей: имени элемента управления («название») и события (Db1Click, то есть двойной щелчок). Между заголовком и End Sub мы должны внести код процедуры, обрабатывающей данное событие. Почему мы выбрали двойное нажатие, а не одинарное? Если в качестве события выбрать одинарное нажатие, форма «поставщики_столбец» будет открываться при каждой попытке установить указатель курсора в поле «название» ленточной формы. Теперь дело только за кодом процедуры. Но этот код за нас один раз уже написала система и он находится в теле процедуры вход_Click().

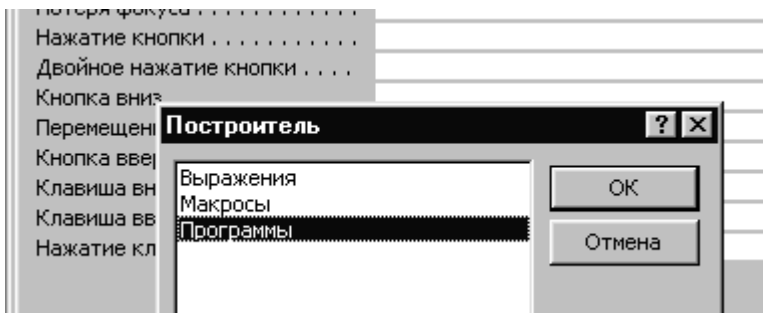


Рис. 39. Диалоговое окно «Построитель»

Было бы неразумно писать этот код еще раз. Мы просто вставим в тело процедуры название `_DbClick` вызов процедуры `вход_Click()`, то есть оператор «`Call вход_Click()`». Новая процедура примет вид:

```
Private Sub название_DbClick(Cancel As Integer)
```

```
    Call вход_Click
```

```
End Sub
```

Закроем последовательно окно модуля формы, окно свойств и окно конструктора формы.

4. Шаг четвертый, или первый самостоятельный

Теперь Вы должны самостоятельно разработать аналогичные объекты для просмотра, ввода и редактирования данных о потребителях продукции, а также по номенклатуре товаров. Возможно, повторение – не самое интересное занятие, но, во-первых, это неизбежный шаг в процессе разработки приложения, во-вторых, Вам предоставляется возможность убедиться, насколько хорошо Вы усвоили весь пройденный материал, и, в-третьих, приобретенные навыки нуждаются в закреплении.

4.1. Таблица «потребители». Прежде всего, войдем в раздел «Таблицы» и создадим в режиме конструктора новую таблицу, которой присвоим имя «потребители». Структура таблицы представлена на рис. 40.

	Имя поля	Тип данных
▼	потребитель	Счетчик
	название	Текстовый
	адрес	Текстовый
	директор	Текстовый
	телефон	Текстовый
	дополнительно	Поле МЕМО

Рис. 40. Создание таблицы «потребители» в режиме конструктора

Обратите внимание!

- Поле «потребитель» содержит код потребителя. Этому полю следует присвоить тип «Счетчик» и сделать его ключевым.

- Целесообразно поле «название» сделать обязательным, а остальные поля оставить как необязательные.

☰ потребители : таблица						
	потребит	название	адрес	директор	телефон	доп
	1	Фиалка, маг.	Садовая, 5	Сидоров Иван Ие	123456	
	2	Хлебпродукты, маг	Зеленая, 12	Ходарковский Ил	990001	
	3	Хлеб, ларек	Мясоедовский ры	Глюкин Петр Нико	346621	
	4	Фикус, гастр.	Кукушкина, 88	Левинская Роза Г	1313131	
▶	5	Мясопродукты, маг.	Школьная, 28	Дынин Алексей Г	281328	
	6	Зигзаг, гастр.	Речная, 25	Корягина Алена А	131430	
	7	Молокопродукты, м	Морская, 25	Раскорякина Али	130143	
	8	Овощи-Фрукты, маг	Огородная,8	Городницкая Вер	321111	
	9	Слон, универсам	Центральная, 33	Барыгин Никита С	666999	
*	(Счетчик)					

Рис. 41. Таблица «потребители»

Итак, склад обслуживает сеть торговых точек. Заполним форму, как подскажет фантазия (рис. 41). При вводе названий потребителей будем придерживаться того же принципа, что и при вводе названий поставщиков: первые буквы в названии должны нести максимум информации! Поэтому слова магазин, ларек, универмаг и так далее мы пишем в конце названия. По этой же причине мы не ставим названия в кавычки.

4.2. Запрос «потребители_Запрос». Перейдем в раздел «Запросы» окна базы данных и, нажав «Создать», в режиме диалога создадим запрос, которому дадим имя «потребители_Запрос». В режиме конструктора (рис. 42) упорядочим данные по возрастанию (в алфавитном порядке) названий потребителей.

4.3. Формы для потребителей. Теперь создадим еще одну ленточную форму и форму «в столбец» так же, как мы это делали раньше (Шаг второй). Источником данных для первой будет запрос «потребители_Запрос», а для второй – таблица «потребители». Когда мы создавали эти формы для поставщиков, мы неоднократно открывали каждую из них в режиме конструктора. Мы оценивали эти формы в работе, находили недостатки, а потом старались усовершенствовать объекты. Для первых шагов такой подход оправдан, так как если бы мы сразу выполнили задачу наилучшим образом, мы не смогли бы понять смысл всех проделанных манипуляций. Теперь Вы можете справиться с заданием

гораздо быстрее, открыв каждую форму в режиме конструктора только раз и внося в нее сразу все необходимые корректировки (на самом деле нормальный человек иногда делает ошибки, для исправления которых придется возвращаться в режим конструктора).

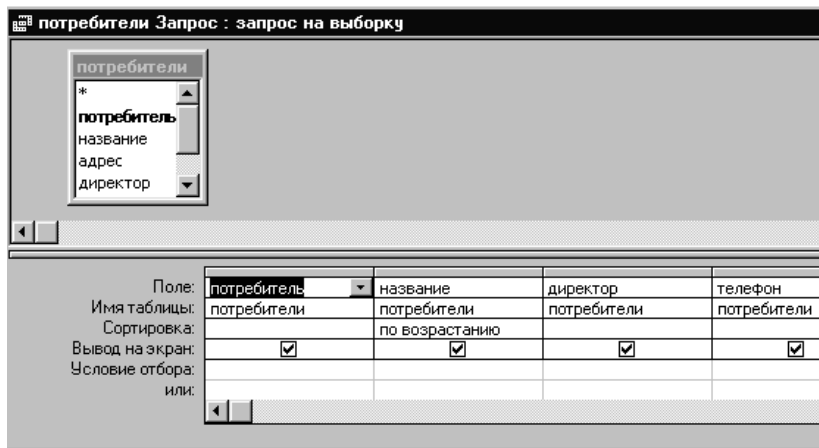


Рис. 42. Запрос в режиме конструктора

Надо заметить, что система работы с данными о поставщиках была организована не наилучшим образом. При желании Вы что-то можете сделать лучше. Но все же, пока, постараемся создать для потребителей такие же формы, какие мы создали для поставщиков. И, наконец, между формами следует установить связь (вспомним Шаг третий). Таким образом, в разделе форм в окне базы данных будут представлены четыре формы (рис. 43).

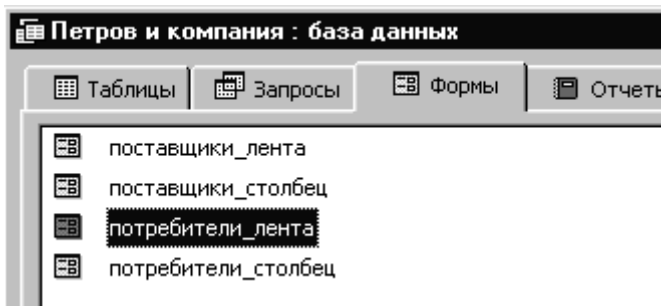


Рис. 43. Раздел формы окна базы данных

4.4. Данные о товарах. Теперь Вам придется самостоятельно создать в режиме конструктора еще одну таблицу, которой мы дадим имя «товары».

Таблица1 : таблица	
Имя поля	Тип данных
товар	Счетчик
название	Текстовый

Рис. 44. Структура таблицы «товары»

товар	название
29	рыба Осетр (кг)
30	рыба Треска (кг)
31	яблоки (кг)
32	груши (кг)
33	виноград (кг)
34	картофель (кг)
35	ветчина (кг)
36	пельмени Сибирские (0,5 кг)
37	пельмени Новосибирские

Запись: 1 из 37

Рис. 45. Форма «товары»

Полная информация о каждом товаре потребовала бы значительное количество полей: производитель, срок годности, информация о качестве

и так далее. Мы ограничимся всего двумя полями. Поле «товар» будет хранить код товара и имеет тип «Счетчик». Поле «название» – текстовая строка длиной 30 символов (рис. 44).

В заключение создадим ленточную форму «товары» для ввода и редактирования данных о товарах. Форму отредактируем и добавим кнопку выхода. Заполним таблицу «товары», используя форму «товары» (рис. 45). Отметим еще раз тот факт, что мы значительно упростили задачу. На самом деле, например, товар «колбаса Докторская» производства разных предприятий – это разные продукты. Мы же условно будем считать ее всегда одним и тем же товаром. В некоторых случаях такой подход оправдан.

Итак, на данный момент мы умеем создавать таблицы, простейшие запросы и формы.

5. Шаг пятый. Мастер подстановок

5.1. Создание таблицы «приход». На этом шаге мы создадим таблицу «приход» для хранения данных о поставках продукции. Таблица будет содержать поля: «номер», «дата поставки» (которое коротко назовем «дата»), «товар», «цена», «количество» и «поставщик». Полю «номер» присвоим тип «счетчик» и сделаем его ключевым. Полю «дата» присвоим тип «дата/время», но здесь тип нуждается в конкретизации. В нижней части окна конструктора в строке «Тип поля» развернем меню и выберем «Длинный формат даты». Поскольку без указания даты поставки большая часть отчетности теряет смысл, сделаем это поле обязательным, установив в строке «Обязательное поле» значение «Да» (рис. 46). После того, как мы внесли в таблицу поля «номер» и «дата», выйдем из конструктора, сохраним таблицу под именем «приход» и снова откроем ее в режиме конструктора. По ряду причин желательно, чтобы перед использованием мастера подстановок таблица уже имела имя. Поле «товар» сделаем обязательным. Но здесь мы впервые сталкиваемся с проблемами рационального использования памяти и целостности данных. Допустим, мы решили вводить в это поле информацию о товаре в виде текста. Подробное описание товара занимает много места. Значит, если в течение месяца этот товар поступал многократно, мы должны были многократно заполнять поле большого размера. Здесь налицо и неэффективное расходование памяти, и перерасход времени кладов-

щика или оператора, которому поручен ввод данных. Наконец, один и тот же текст разные люди могут ввести не идентично. Появилось искушение сократить длинное слово – и, с точки зрения компьютера, введено новое название. О возможных ошибках даже говорить не приходится.

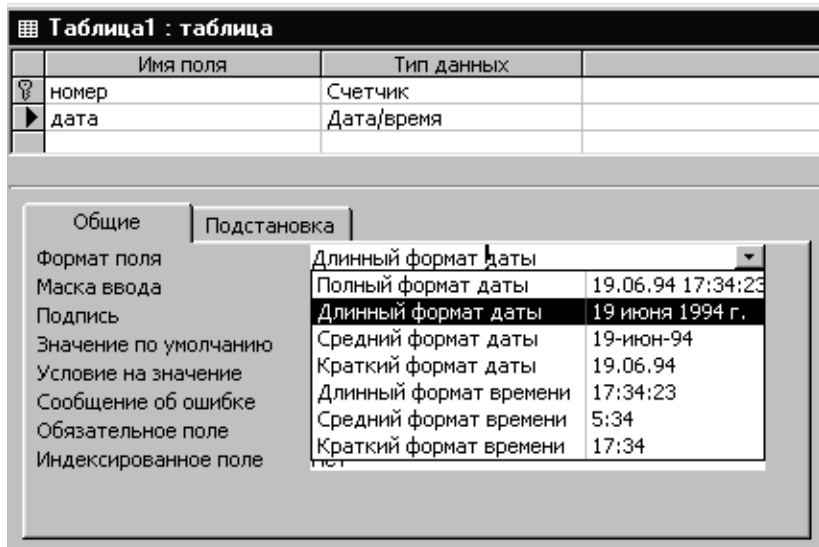


Рис. 46. Выбор формата даты

С точки зрения эффективности хранения данных здесь наиболее целесообразно хранение кодов, в частности кодов товара и поставщика. Вот почему, конструируя соответствующие таблицы, мы сделали эти коды уникальными. Однако пользователю трудно держать в голове большое количество кодов. И здесь приходит на помощь мастер подстановок (рис. 47). Мастер подстановок обеспечивает такой способ хранения данных, при котором в таблице хранятся коды объектов, а названия, расшифровывающие смысл этих кодов, хранятся в другой таблице, которая играет роль справочника. Пользователь видит только названия! В процессе создания подстановок принципиально, чтобы коды в таблице-справочнике были помечены как ключевые поля, что мы и сделали при создании структур таблиц «поставщики», «потребители» и «товары». После выбора (рис. 47) строки «Мастер подстановок» на экране появится диалоговое окно (рис. 48), в котором нам будет предложено при фор-

мировании столбца подстановки использовать значения либо из таблицы или запроса, либо ввести фиксированный набор значений.



Рис. 47. Мастер подстановок

Есть параметры объектов, которые могут иметь только фиксированный и неизменный набор свойств. Например, пол может принимать только два значения: «мужской» или «женский». В таком случае можно перевернуть переключатель в нижнее положение и заполнить список подставляемых значений.

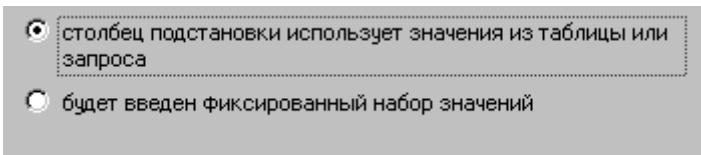


Рис. 48. Выбор способа формирования столбца подстановок

Список товаров, напротив, вещь очень непостоянная и его лучше хранить в таблице. Мы оставим переключатель в положении, представленном на рис. 48, и нажмем «Далее». Система предложит нам выбрать таблицу или запрос, которые мы собираемся использовать для создания подстановки. Мы выберем «товары» (рис. 49) и нажмем «Далее».

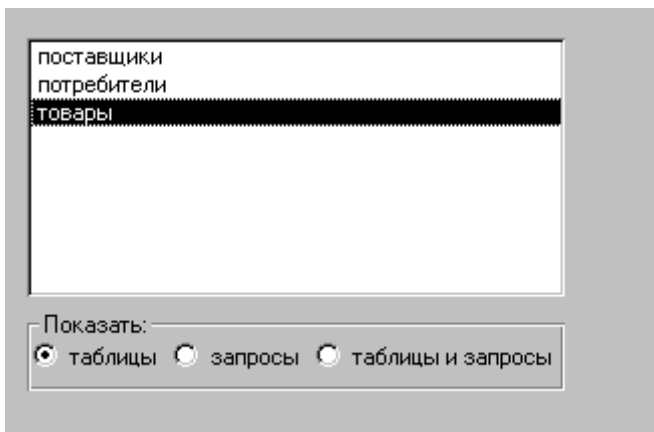


Рис. 49. Выбор таблицы

В диалоговом окне (рис. 50) перебросим поля «товар» и «название» в раздел «Выбранные поля» и нажмем «Далее». После этого система предоставит нам возможность посмотреть, как выглядит столбец подстановки (рис. 51). Ключевой столбец по умолчанию скрыт! Нажмем «Далее». В ответ на предложение: «Задайте подпись, которую содержит столбец подстановки» введите имя поля «товар» и нажмите «Готово».

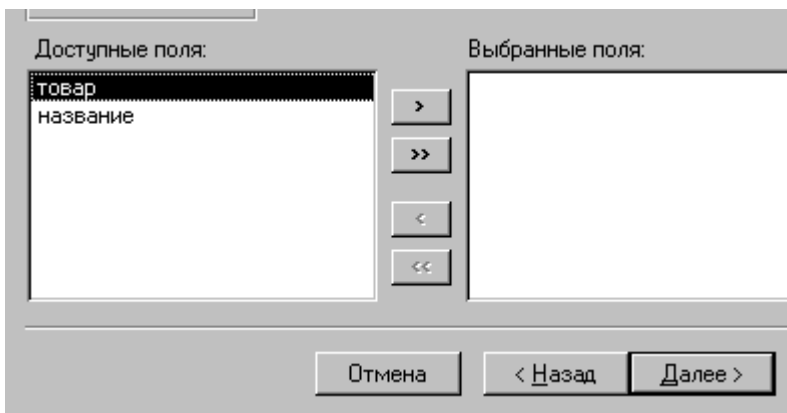


Рис. 50. Выбор полей

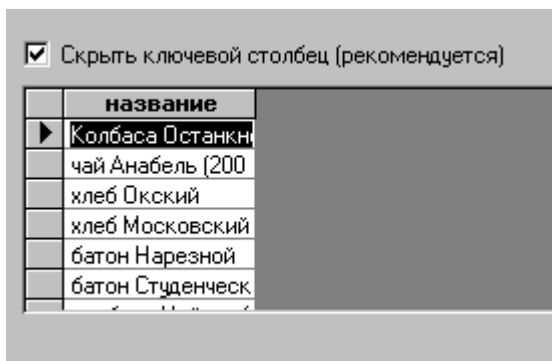


Рис. 51. Столбец подстановки

Таким образом, Вы создали связь между таблицами «приход» и «товары». Система потребует сохранить таблицу. На вопрос «Выполнить это сейчас?» выбираем ответ «Да». Обратите внимание, что тип поля «товар» оказался числовым. Точно так же создайте поле «поставщик», используя для подстановки таблицу «поставщики». Только теперь в качестве «Выбранных полей» Вы берете не все поля таблицы, а только ключевое поле «поставщик» и поле «название». Полю «цена» присвоим тип «числовой». При этом по умолчанию в наборе свойств поля в строке «Размер поля» появится значение «Длинное целое». Это свойство также нуждается в конкретизации. Поэтому в строке «Формат поля» выберем «Фиксированный». Это значит, что в записи числа будет содержаться два знака после запятой.

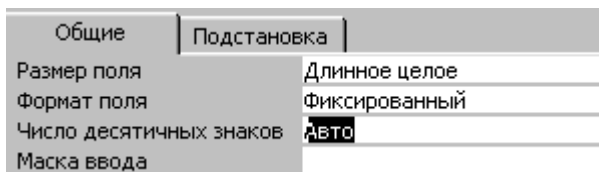


Рис. 52. Свойства поля «цена»

Поле «количество» будет хранить объем поставки. По смыслу задачи поле должно быть числовым, а размер поля, вероятно, достаточно сделать целым. Теперь закроем конструктор, сохранив изменения макета. Откроем таблицу в рабочем режиме (рис. 53) и заполним несколько записей. При этом следует обратить внимание на то, что дата вводится в числовом формате с точкой в качестве разделителя. Например, Вы вво-

дите «4.2.5», а на экране отображается «4 февраля 2005 г.». Товар или производителя теперь Вы можете вводить, раскрыв соответствующее меню. Кроме того, часто имеет смысл сначала набрать первые буквы названия. Тогда система попытается угадать все слово.

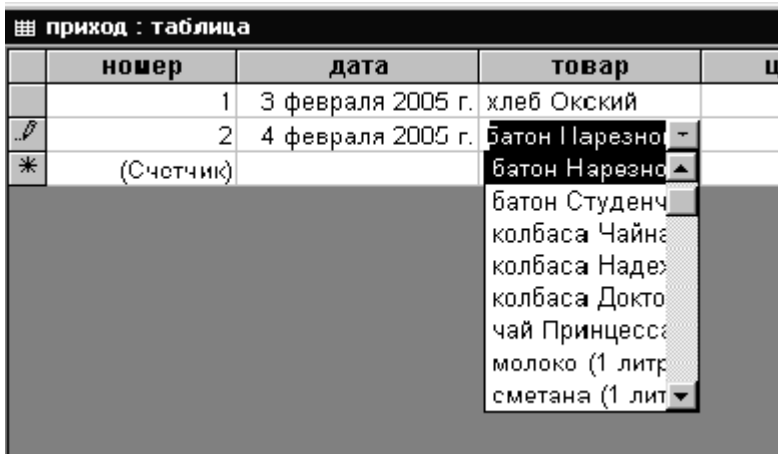


Рис. 53. Ввод значения поля из меню

Пока мы заполнили только две записи, но уже почувствовали неудобства, связанные с тем, что тексты в столбце подстановки расположены в произвольном (неалфавитном) порядке. Не поленимся еще раз войти в режим конструктора. Выделим поле «товар» и в свойствах поля внизу окна конструктора перейдем на заставку «Подстановка» (рис. 54).

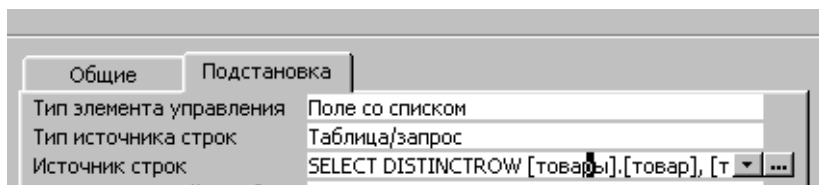


Рис. 54. Источник данных для подстановки

Установим указатель в поле с надписью «Источник строк» и нажмем троеточие в конце поля. На экране появится уже знакомое нам окно построителя запросов (рис. 55).

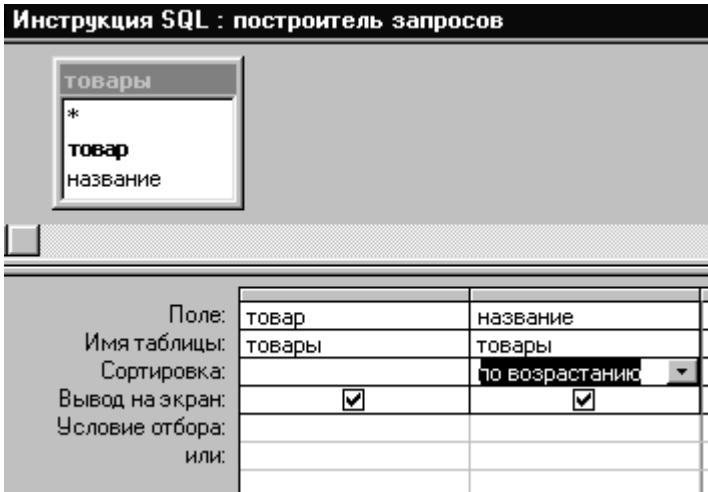


Рис. 55. Источник данных для подстановки в окне запроса

В этом окне в строке «Сортировка» для поля «название» установим значение «по возрастанию». Закроем окно запроса и подтвердим сохранение изменений. Теперь, не выходя из режима конструктора, те же действия проделаем с полем «поставщик». Покинем конструктор, сохранив изменения, и откроем таблицу в рабочем режиме. Теперь ввод товаров и поставщиков значительно упростился. Например, если Вы ищете в меню колбасу данного вида, то Вам достаточно набрать в поле «товар» текст «ко» и развернуть меню. Перед Вами предстанут все колбасы, какие фигурировали когда-либо на складе. Вот почему автор упорно настаивал на том, чтобы первые буквы текста в таблицах-справочниках несли максимум информации об объекте. Например, не рекомендуется заключать названия предприятий в кавычки или начинать их с «АО», «ООО» и прочих повторяющихся последовательностей символов. Следующие задания читателю предлагается выполнить самостоятельно.

5.2. Создание запроса «приход Запрос». Указанный объект следует создать, как простой запрос, а затем в режиме конструктора упорядочить даты поступления товаров по возрастанию. При построении запроса используем все поля источника!

5.3. Создание формы «приход». В разделе «Формы» окна базы данных нажмем кнопку «Создать», в диалоге выберем ленточную форму, в качестве источника данных возьмем «приход Запрос» и выберем для формы все поля источника! Сохраним форму, дав ей имя «приход». То, что имя формы совпадает с именем таблицы, не должно нас беспокоить. Мы могли, конечно, пойти по проторенной дорожке, создав форму «в столбец», а затем связав ее с ленточной формой, но хотя бы для разнообразия поступим иначе. Откроем форму в рабочем режиме и посмотрим, что нам в ней не нравится. Для некоторых полей система выделила слишком много места, а для других места явно не хватает. Поле «номер» (вместе с подписью к нему) можно без ущерба для содержания удалить из формы. Полезно разместить в примечании формы кнопку выхода. Перечисленные манипуляции не должны вызывать у нас каких-либо затруднений, мы их не раз проделывали. Но иногда также при построении формы в автоматическом режиме игнорируются созданные нами подстановки и форма выглядит, как на рис. 56. Как видно на рисунке, вместо названий товаров и поставщиков представлены только их коды. Чтобы устранить подобный дефект, нужно развернуть «Список полей» (рис. 57) с панели инструментов «Конструктора форм».

	дата	товар	цена	колич.	поставщик
▶	3 февраля 2005 г.	3	11,00	200	4
	4 февраля 2005 г.	5	8,00	50	11
*		0	0,00	0	0

Запись: [←] [←] [1] [→] [→] [→] [→] из 2

Рис. 56. Форма с полями без подстановок

Удаляем в форме поля «товар» и «поставщик» и переносим эти поля из «Списка полей» в форму. Для такого переноса поля надо в «Списке

полей» щелкнуть левой клавишей мыши по имени соответствующего поля и, не отрывая руки от клавиши, перенести это поле на место удаленного. При этом вместе с полем перенесется в область данных формы и относящаяся к полю надпись, что совершенно некстати. Для удаления лишнего экземпляра надписи щелкните клавишей за пределами поля, а затем по надписи и нажмите клавишу «Delete».

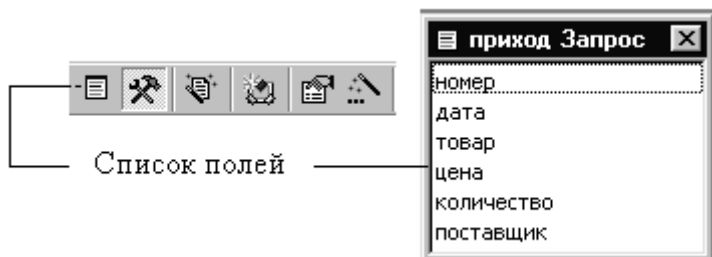


Рис. 57. Список полей

Аккуратно расположите поля в одной строке области данных, чтобы каждое поле располагалось под своей надписью. Корректировки могут нарушить последовательность перехода между полями в форме. В таком случае Вы можете установить правильную последовательность перехода, воспользовавшись пунктом меню «Вид → Последовательность перехода» (рис. 59). Закроем «Конструктор».

дата	товар	цена	колич.	поставщик
3 февраля 2005 г.	хлеб Окский	11,00	200	Лабаз 000
4 февраля 2005 г.	батон Нарезной	8,00	50	Семиструйский хле
		0,00	0	

Рис. 58. Заголовки полей и область данных после корректировки

Теперь – самое скучное! Вам предстоит, пользуясь формой или непосредственно в таблице, занести как можно больше информации о поставках. Стоит выделить на это не менее пятнадцати минут.

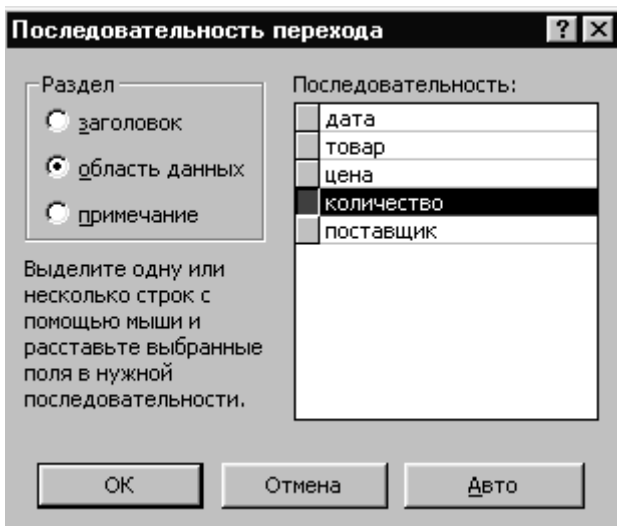


Рис. 59. Последовательность перехода при вводе данных

расход : таблица		
	Имя поля	Тип данных
?	номер	Счетчик
	дата	Дата/время
	товар	Числовой
	количество	Числовой
	потребитель	Числовой

Рис. 60. Поля таблицы «расход»

5.4. Таблица «расход». Теперь Вам предлагается продемонстрировать, насколько хорошо Вы освоили материал пунктов 5.1-5.3. Таблица «расход» будет содержать данные о расходе товаров. Ее поля представлены на рис. 60. Поля «номер», «дата», «товар» и «количество» имеют тот же смысл, что и в таблице «приход», только теперь речь идет об отпуске партий товара со склада. Поле «цена» мы не задействовали, считая, что склад обслуживает торговые точки нашей фирмы, а значит, товары отпускаются со склада по закупочной цене.

Поле «потребитель» использует подстановку, а источником столбцов для подстановки служит таблица «потребители».

5.5. Форма «расход». Создадим запрос «расход Запрос» и в режиме конструктора введем упорядочивание записей по дате (теперь под датой подразумевается дата отпуска товара). Используя построенный запрос в качестве источника данных, создадим ленточную форму «расход» точно так же, как в разделе 5.3.

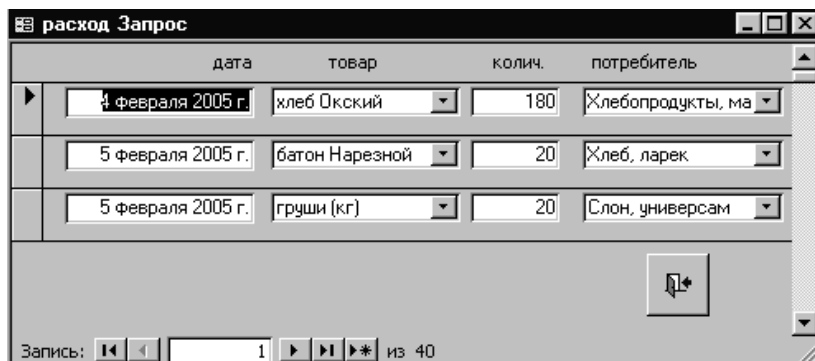


Рис. 61. Форма «расход»

Используя созданную выше форму, заполним таблицу «расход». Чем больше записей Вы введете, тем более удачные иллюстрации работы приложения можно будет в дальнейшем получать. При этом будем считать, что наш склад только открылся, и постараемся, чтобы, например, мы не поставили потребителям больше картофеля, чем получили от поставщиков.

5.6. Связи между таблицами. В рассмотренных выше примерах при вводе товаров, поставщиков и потребителей мы видим на экране их названия, хотя теперь знаем, что физически таблица хранит только коды. Такое удобство создал нам мастер подстановок. Одна из таблиц играет роль справочника по кодам, хранящимся в другой таблице. Таким образом, между таблицами можно установить связи. Виды связей в этом разделе мы разбирать не будем. Однако заметим, что связи между таблицами всегда устанавливаются по определенным полям. Для того, чтобы получить схему связей между таблицами, надо в строке горизонтального меню выбрать пункт *Сервис* → *Схема данных*. Если в появившемся окне «Схема данных» (рис. 62) интересующая Вас связь отсутствует, то можете щелкнуть правой клавишей мыши и в развернувшемся окне выбрать пункт «Отобразить все».

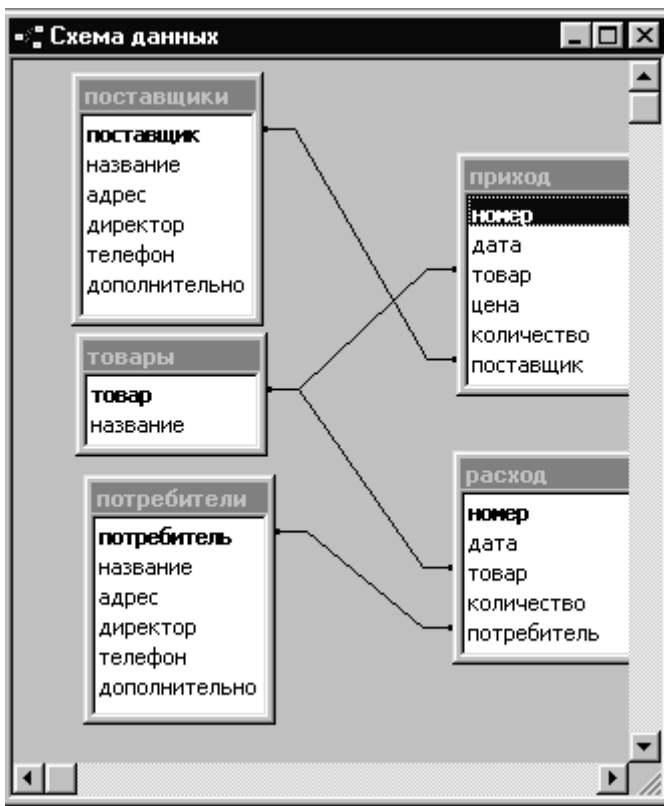


Рис. 62. Окно схемы данных

Итак, мы разработали объекты, обеспечивающие хранение и редактирование данных о товарах, поставщиках и потребителях, а также о приходе и расходе товаров. В разделе «Формы» окна базы данных мы видим имена семи построенных нами форм (рис. 63). Отражаемая в форме информация может быть распечатана так же, как во многих других приложениях, с помощью строки меню **Файл → Печать**. Однако представленная в формах информация, как правило, далеко не то, что хотел бы видеть на своем столе начальник. Руководитель обычно нуждается в обобщенной и прошедшей предварительную обработку информации, представленной в удобном для чтения виде. Этим вопросом мы и займемся далее.

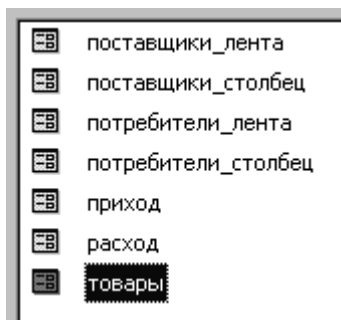


Рис. 63. Раздел «Формы» окна базы данных

6. Шаг шестой. Отчеты

Отчет – это объект базы данных, представляющий информацию в виде готового для печати документа. Информацию можно распечатать непосредственно из таблиц, запросов и форм, но отчеты обладают принципиальным преимуществом.

- Они предоставляют широкие возможности отражения в печатном документе больших наборов данных, в случае необходимости, сгруппированных по заданным признакам. При этом в документе могут отражаться промежуточные (по группам) и общие итоговые значения.
- Отчеты могут использоваться для получения соответствующим образом оформленных счетов, документов для презентации и других полезных в работе вещей, оформление которых вручную отнимает уйму времени!

6.1. Простейший отчет по приходу. Войдем в раздел «Отчеты» окна базы данных и нажмем клавишу «Создать». На экране появится диалоговое окно «Новый отчет» (рис. 64). В диалоговом окне выберем «Автоотчет: ленточный», а в качестве источника данных возьмем таблицу «приход» и нажмем кнопку «ОК». Отчет готов! Сохраним его под именем «приход_подробный». Откроем отчет в режиме конструктора и проведем ряд корректировок (отчет так же, как и форма, является контейнером для элементов управления). Во-первых, уберем поле номера с

соответствующей надписью и откорректируем размеры полей. Подпись «ПРИХОД» в заголовке отчета исправим на «Приход товаров».

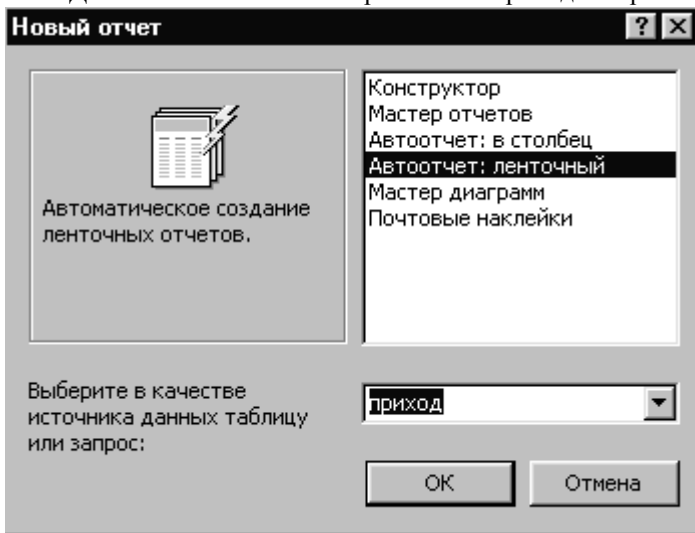


Рис. 64. Диалоговое окно создания нового отчета

Если отчет подготовлен в альбомном формате, уплотните данные, сдвиньте правую границу страницы и в меню «**Файл→Параметры страницы**» смените формат страницы на «книжный». Здесь поначалу может возникнуть ряд затруднений, но вполне преодолимых. Зато привычка самостоятельно решать свои проблемы всегда пригодится! Итак, пусть окно конструктора приняло вид, как на рис. 65. Обратите внимание на структуру окна конструктора. В верхней части находится «Заголовок отчета». Представленная в нем информация, независимо от количества листов, печатается только один раз в начале отчета. «Верхний колонтитул» содержит информацию, которая отображается вверху каждой страницы текста, обычно здесь находятся заголовки полей. Область данных – набор полей, взятый из источника данных. «Нижний колонтитул» – информация, отображаемая внизу каждой страницы (номер страницы, дата). Средством отображения текущей даты служит вычисляемое поле, содержащее функцию =Now(). Кроме всего перечисленного, на рисунке Вы видите «Примечание отчета», в котором элементы управления пока отсутствуют.

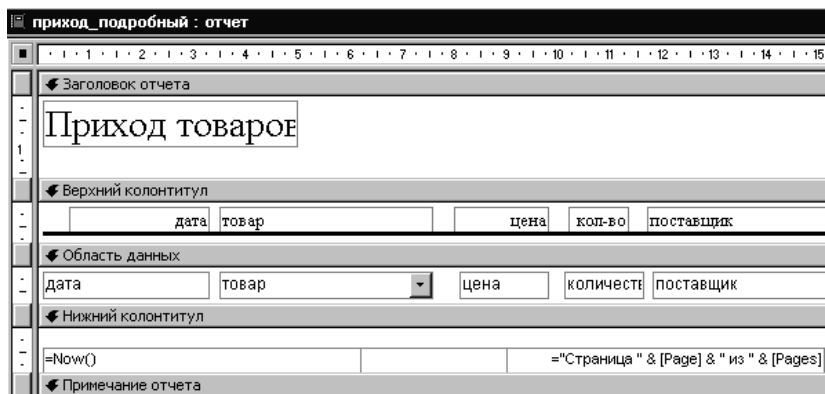


Рис. 65. Окно конструктора отчетов

Закроем «Конструктор» и посмотрим отчет (рис. 66). Здесь, как и прежде, мы не пытались найти сразу наилучшее решение. Но если в течение отчетного периода было большое количество поставок, то вряд ли такой отчет удовлетворит Ваше начальство.

Приход товаров

дата	товар	цена	кол-во	поставщик
3 февраля 2005 г.	хлеб Окский	11,00	200	Лабаз ООО
4 февраля 2005 г.	батон Нарезной	8,00	50	Семиструйск
4 февраля 2005 г.	груши (кг)	80,00	30	Садоводческ
4 февраля 2005 г.	ветчина (кг)	100,00	20	Севливер ОС
5 февраля 2005 г.	ветчина (кг)	90,00	15	Урюпинский
5 февраля 2005 г.	картофель (кг)	10,00	50	Апельсиновъ
6 февраля 2005 г.	сметана (1 литр)	20,00	100	Лукоморский
6 февраля 2005 г.	молоко (1 литр)	13,00	150	Лукоморский
7 февраля 2005 г.	груши (кг)	70,00	60	Апельсиновъ
10 февраля 2005 г.	яблоки (кг)	50,00	100	Апельсиновъ

Рис. 66. Отчет по поставкам товаров

6.2. Группировки в отчете. Допустим, директору фирмы важно видеть данные, сгруппированные по поставщикам. Это для нас не проблема. Вернемся в режим конструктора и выберем строку меню «Вид→Сортировка и группировка».

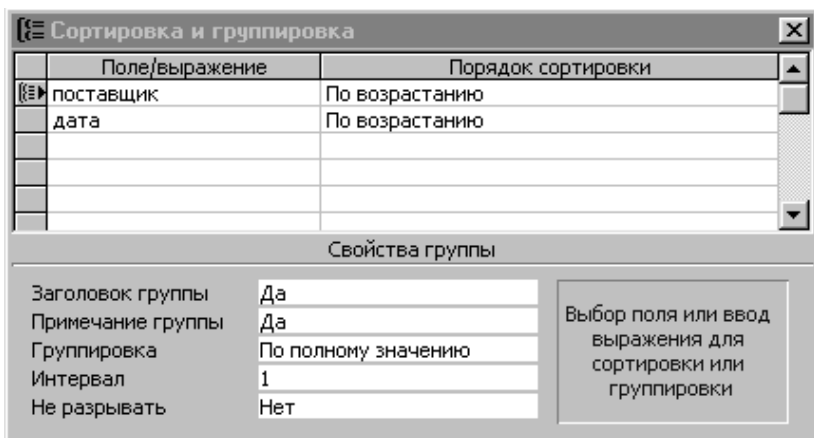


Рис. 67. Сортировка и группировка

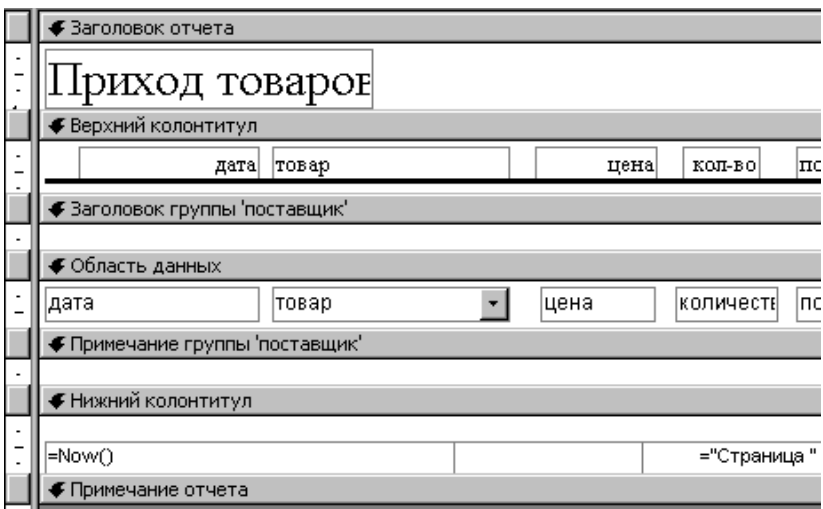


Рис. 68. Окно конструктора. Группировка по поставщикам

Откроется диалоговое окно «Сортировка и группировка» (рис. 67). Установим указатель на первой строке столбца «Поле/выражение», развернем меню и выберем поле «поставщик». Автоматически в соответствующей строке второго столбца появится текст «По возрастанию». Это значит, что записи в запросе будут сгруппированы по поставщикам. В нижней части окна в полях с подписями «Заголовок группы» и «Примечание группы» установим значения «Да». Во второй строке точно так же установим поле «дата», для того чтобы поставки были упорядочены по датам, но не будем требовать наличия «Заголовка группы» и «Примечания группы»! Закроем окно. Окно конструктора примет вид, представленный на рис. 68. Теперь перетащим в «Заголовок группы» из верхнего колонтитула подпись «поставщик», а из области данных – поле «поставщик» (рис. 69). Установим для поля «поставщик» жирный шрифт. Чтобы лишний раз не открывать конструктор, заодно решим еще одну проблему. Нам хотелось бы видеть в отчете данные о стоимости поставок. Добавим в верхний колонтитул надпись «стоимость», а в область данных – поле. Элементы управления, как и в случае конструирования форм, берутся с «Панели элементов». Подпись, которая появится слева от поля, удалим. Теперь занесем в поле вычисляемое выражение «=цена*количество». Квадратные скобки система вставит сама (рис. 70).

дата	товар	цена	кол-во
<div style="border: 1px solid black; padding: 2px;"> ☛ Заголовок группы 'поставщик' </div> <div style="border: 1px solid black; padding: 2px;"> поставщик </div> <div style="border: 1px solid black; padding: 2px;"> ☛ Область данных </div> <div style="border: 1px solid black; padding: 2px;"> дата товар цена количество </div> <div style="border: 1px solid black; padding: 2px;"> ☛ Примечание группы 'поставщик' </div> <div style="border: 1px solid black; padding: 2px;"> ☛ Нижний колонтитул </div>			

Рис. 69. Название поставщика теперь размещено в примечании группы

Чтобы иметь возможность видеть итоговые данные, возьмем с «Панели элементов» поле и разместим в примечании группы. В качестве подписи введем «Итого». А в свободное поле поместим выражение «=Sum([цена]*[количество])». Поскольку это выражение находится в примечании группы, сумма стоимостей будет вычисляться только по данной группе (рис. 71).

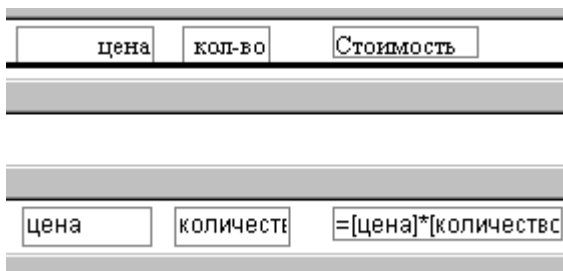


Рис. 70. Вычисляемое поле

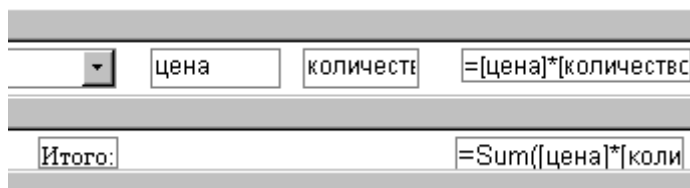


Рис. 71. Суммарная стоимость по группе

Подведем черту под строкой «Итого» (черту берем с Панели элементов). Затем скопируем содержимое «Примечания группы» в «Примечание отчета», а надпись «Итого» заменим на «Итого по всем поставщикам». Закроем конструктор, сохранив изменения, и откроем отчет для просмотра. На рис. 72 изображено начало отчета.

Приход товаров

	дата	товар	цена	кол-во	Стоимость
поставщик	Заборская макаронная фаб				
	18 февраля 2005 г.	макароны Рожки	32,00	150	4800
	18 марта 2005 г.	макароны Рожки	32,00	150	4800
	Итого				9600
поставщик	Урюпинский рыбозавод				
	6 марта 2005 г.	рыба Горбуша (кг)	70,00	50	3500
	26 марта 2005 г.	рыба Ряпушка (кг)	50,00	50	2500
	Итого				6000

Рис. 72. Отчет с группировкой по поставщикам

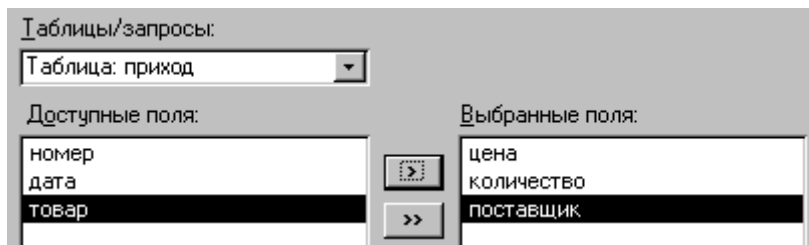


Рис. 73. Выбор полей для запроса

6.3. Отчет по итоговому запросу. Теперь директор хотел бы видеть итоговый отчет только по объему поставок по поставщикам в денежном выражении. И это не проблема! Сначала перейдем в раздел «Запросы», нажмем «Создать», в диалоговом окне выберем «Простой запрос» и нажмем «ОК». В следующем диалоговом окне в качестве источника данных возьмем таблицу «приход» и выберем в ней поля «цена», «количество» и «поставщик» (рис. 73). Нажмем «Далее», оставим переключатель в положении «подробный» и опять нажмем «Далее». Дадим запросу имя «приход по стоимости» и нажмем «Готово». Откроем получившийся запрос в режиме конструктора и добавим вычисляемое поле «стоимость: [цена]*[количество]» (рис. 74).

поставщик	стоимость: [цена]*[количество]
приход Запрос	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 74. Вычисляемое поле в запросе

Обратите внимание на формат вычисляемого поля. Сначала идет название поля, а затем после двоеточия формула для вычисления его значения. Вид получившегося отчета в режиме таблицы представлен на рис. 75. Теперь создадим еще один простой запрос, источник данных для которого – запрос «приход по стоимости». В источнике выберем поля «поставщик» и «стоимость» (рис. 76). Нажмем «Далее», установим переключатель в положение «итоговый» и нажмем кнопку «Итоги». Появится диалоговое окно (рис. 77). В этом окне мы поставим «птичку» в строке «стоимость» под заголовком «Sum». Строку «поставщик» не трогаем. Это значит, что группировка будет проводиться по полю «по-

ставщик» и нас интересует суммарное значение стоимости по каждой группе.

приход по стоимости : запрос на выборку				
	цена	количество	поставщик	стоимость
▶	11,00	200	Лабаз ООО	2200
	8,00	50	Семиструйский хлеб	400
	80,00	30	Садоводческое това	2400
	100,00	20	Севливер ООО	2000
	90,00	15	Урюпинский мясобо	1350
	10,00	50	Апельсиновый рай А	500
	20,00	100	Лукоморский молокс	2000
	13,00	150	Лукоморский молокс	1950
	70,00	60	Апельсиновый рай А	4200
	50,00	100	Апельсиновый рай А	5000
	75,00	30	Лукоморская рыболк	2250
	100,00	100	Лукоморская рыболк	10000
	110,00	200	Сельхозпродукт п/о	22000

Запись: [←] [↩] [1] [▶] [▶▶] [⌘] из 40

Рис. 75. Запрос с вычисляемым полем «стоимость»

Таблицы/запросы:
 Запрос: приход по стоимости ▾

Доступные поля:
 цена
 количество

Выбранные поля:
 поставщик
 стоимость

[↔] [>>]

Рис. 76. Выбор полей из запроса «приход по стоимости»

Нажмем «ОК», затем «Далее». Дадим запросу имя «итоги приход по поставщикам» и нажмем «Готово». Вид запроса в режиме таблицы представлен на рис. 78. Обратите внимание на тот факт, что система, не советуясь с нами, присвоила вычисляемому полю имя «Sum_стоимость». Откорректируем имя. Для этого откроем запрос в режиме конструктор и вместо «Sum_стоимость» запишем «объем_поставки» (рис. 79).

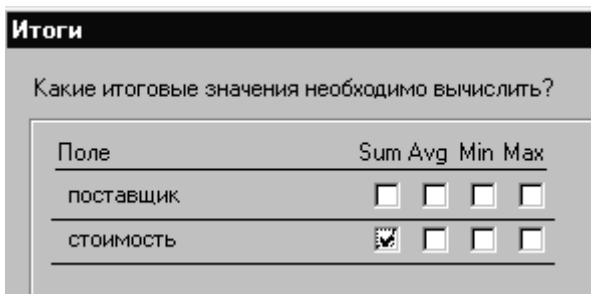


Рис. 77. Окно «Итоги»

поставщик	Sum_стоимости
Заборская макаронн	9600
Урюпинский рыбозав	6000
Лабаз ООО	4400
Алва...	10100

Рис. 78. Запрос «Итоги по поставщикам»

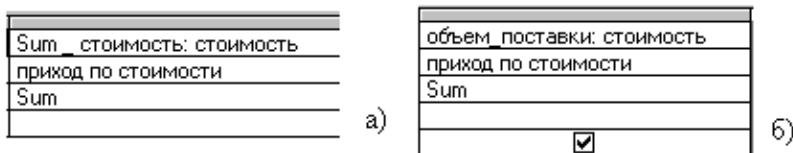


Рис. 79. Изменение имени вычисляемого поля в запросе

Двоеточие и следующее за ним название исходного поля «стоимость», по которому ведется суммирование, удалять или изменять нельзя!

Теперь осталось создать ленточный отчет, точно так же, как мы это делали в пункте 6.1. В режиме конструктора откорректируем отчет. Изменим заголовок на «Объемы поставок (в рублях)». В примечании отчета поместим вычисляемое поле, отражающее суммарный объем поставок по всем поставщикам (рис. 80). Вычисляемое поле в примечании отчета содержит формулу «=Sum([объем_поставки])». Предназначенный для печати документ представлен на рис. 81.

Таким образом, мы создали два отчета. В первом поставки сгруппированы по поставщикам. Во втором отражена обобщенная информация, то есть объемы поставок в рублях по всем поставщикам.

В качестве самостоятельной работы предлагается создать следующие объекты:

- Отчет по поставкам, сгруппированным по отдельным товарам.
- Итоговый отчет по отдельным товарам с указанием объема поставки, общей стоимости поставки и средней цены товара (в общем случае один и тот же товар за некоторый период времени может поставляться по разным ценам).

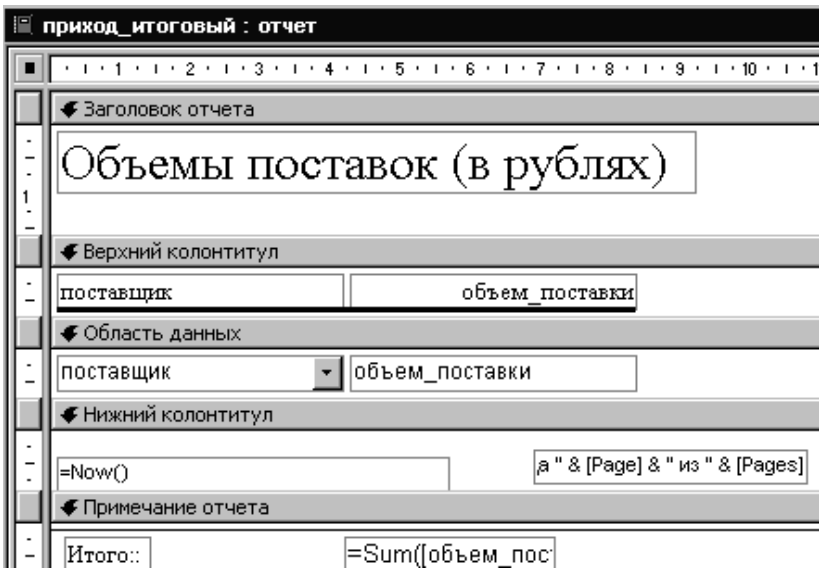


Рис. 80. Отчет «приход_итоговый» в режиме конструктора

Может ли созданный таким образом отчет представлять практический интерес? Очевидно, нет. Склад может работать годы, а отчет директор требует за конкретный период времени. Наконец, неплохо бы в заголовке отчета поместить название фирмы, а в примечании отчета фамилию с инициалами лица, представляющего отчет, – заведующего складом. Этим в дальнейшем мы и займемся.

Объемы поставок (в рублях)

<u>поставщик</u>	<u>объем_поставки</u>
Заборская макаронная фа	9600
Урюпинский рыбозавод	6000
Лабаз ООО	4400
Апельсиновый рай АО	19400
Кузякинская макаронная ф	6000
Сельхозпродукт п/о	44000
Садоводческое товарищес	4800
Семиструйский хлебозаво	800
Лукоморская рыболовная	14500
Кузякинский хлебокомбина	4400
Урюпинский мясокомбинат	2700
Дары юга ООО	6600
Севливер ООО	4000
Буренка АО	1200
Лукоморский молокозавод	7900
Кузмолоко ООО	2400
Итого:	138700

Рис. 81. Предназначенный для печати документ

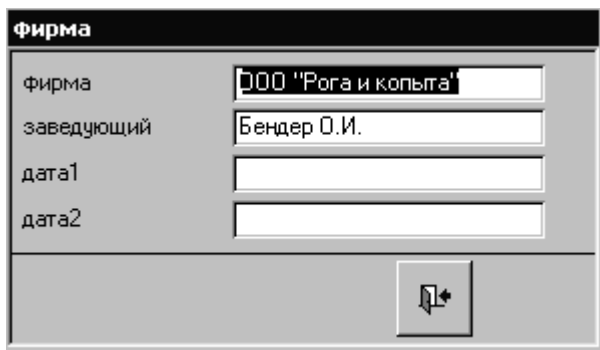
7. Шаг седьмой. Доработка отчетов

7.1. Дополнительная информация. Прежде всего создадим в режиме конструктора таблицу, которая будет содержать дополнительную информацию, отраженную в полях:

- «фирма» – название фирмы. Текстовая строка, длину которой Вы можете установить сами;
- «заведующий» – фамилия, имя, отчество заведующего складом;
- «дата1» и «дата2» – эти поля (длинный формат даты) будут хранить начало и конец временного интервала, за который составляется отчет.

Разработчик, если сочтет нужным, может пополнить структуру таблицы и другими полями, но мы, как всегда, постараемся ограничиться необходимым минимумом. При сохранении таблицы подтвердим, что не собираемся задавать ключевое поле. Дадим таблице имя «фирма» и заполним только первую запись. Например, «фирма» – «ООО «Рога и копыта», «заведующий» – «Бендер О. И.». Создадим форму «в столбец», источником данных которой будет служить таблица «фирма» (рис. 82).

Рис. 82. Дополнительная информация



The image shows a form window titled "Фирма". It contains four input fields arranged vertically. The first field, labeled "фирма", contains the text "ООО 'Рога и копыта'". The second field, labeled "заведующий", contains the text "Бендер О.И.". The third and fourth fields, labeled "дата1" and "дата2" respectively, are currently empty. At the bottom right of the form, there is a button with a right-pointing arrow and a plus sign, indicating a "Next" or "Add" action.

Дадим форме имя «фирма». Поскольку таблица «фирма» всегда будет состоять из одной-единственной записи, внесем в свойства формы изменения, которые мы уже вносили в разделе 3.3. То есть удалим полосы прокрутки, область выделения, поле номера записи, кнопки оконного меню и кнопки размеров окна, запретим удаление и добавление записей. Наконец, сделаем окно модальным, закроем окно свойств, создадим кнопку выхода и покинем режим конструктора. Форма готова!

На самом деле нам нужны две формы, отличные от той, которую мы только что создали: одна для просмотра и корректировки названия фирмы и фамилии заведующего складом (по крайней мере, заведующих складом иногда меняют), другая для задания начала и конца отчетного периода.

Создадим копию получившейся формы. Для этого воспользуемся пунктом меню «Правка → Копировать», а затем «Правка → Вставить». В открывшемся диалоговом окне дадим копии имя «отчет_приход». Затем поочередно откроем обе формы в режиме конструктора и внесем в них небольшие корректировки. В форме «фирма» удалим поля «дата1» и «дата2» (рис. 83).

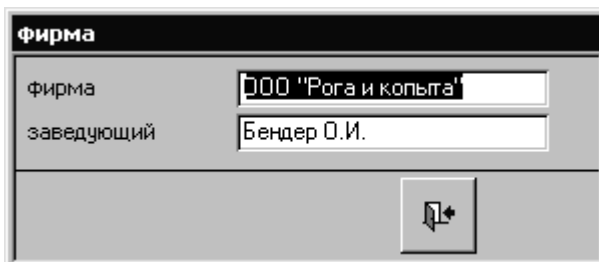


Рис. 83. Форма «фирма»

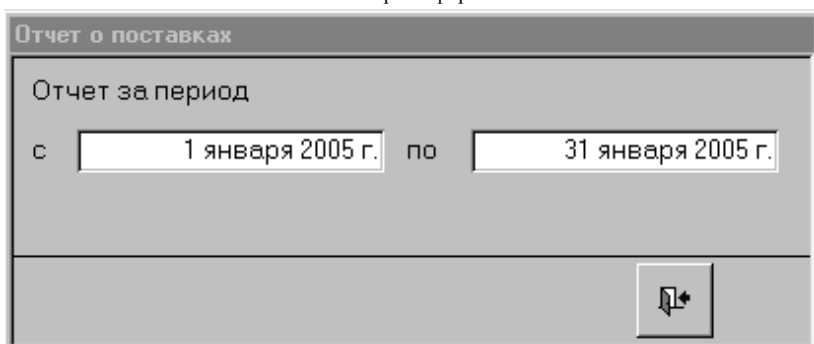


Рис. 84. Форма «отчет_приход»

В форме «отчет_приход» установим подпись: «Отчет о поставках», удалим поля «фирма» и «заведующий», а также изменим расположение полей и содержание надписей к ним (рис. 84). Еще раз обратим Ваше внимание на то, что подпись в верхней части формы и имя формы – разные

вещи. Смысл преобразований будет более понятен, когда мы организуем вызов из этого окна отчетов за заданный период времени.

7.2. Группа переключателей. Откроем форму «отчет_приход» в режиме конструктора. Поместим в форму элемент управления «Группа переключателей» (см. Шаг второй, рис. 15). Откроется диалоговое окно (рис. 85).

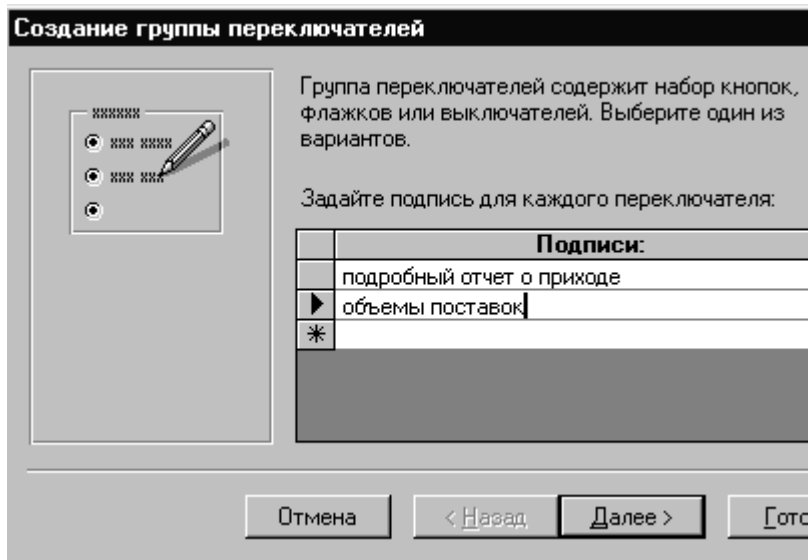


Рис. 85. Подписи к переключателям

У нас пока создано два отчета. Возможно, Вы создали больше. В любом случае каждому отчету мы поставим в соответствие переключатель. Введем подписи к двум переключателям (рис. 85) и нажмем «Далее». Теперь система потребует задать переключатель, используемый по умолчанию. Здесь Вы можете поступать по своему усмотрению. Нажмите «Далее». Вы увидите список, в котором каждой подписи соответствует натуральное число. Система по умолчанию в каждой строке устанавливает число, равное номеру строки. Мы имеем право изменить заданные системой значения, но не хотим. Поэтому нажмем «Далее». Оставим переключатель (рис. 86) в положении «Сохранить значение для дальнейшего использования» и нажмем «Далее».

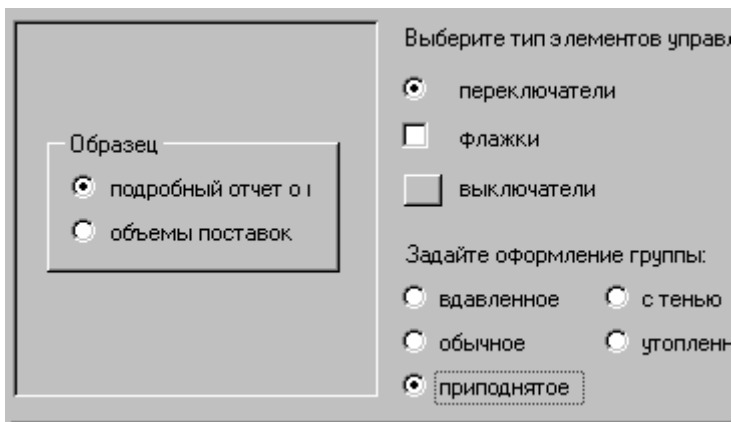


Рис. 86. Выбор варианта использования значения переключателя

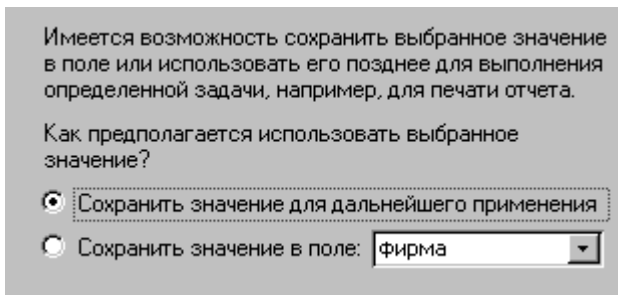


Рис. 87. Выбор типа элемента управления

В диалоговом окне (рис. 87) мы можем заменить переключатель флажком или кнопкой, но не будем этого делать. Поставим оформление группы «приподнятое» (здесь автор не настаивает) и нажмем «Далее». Зададим подпись к группе переключателей «Выбор формы отчета» и нажмем «Готово». Выйдем из режима конструктора и откроем форму в рабочем режиме (рис. 88). Если Вы создали, например, пять отчетов, то у Вас сейчас должно быть пять переключателей с соответствующими подписями. Испытаем переключатели в работе. В положении «Да» всегда находится только один переключатель. Правда, само по себе изменение положения переключателя не влечет за собой равным счетом никаких действий. Эту проблему мы решим в следующем пункте, а пока дадим группе переключателей имя.

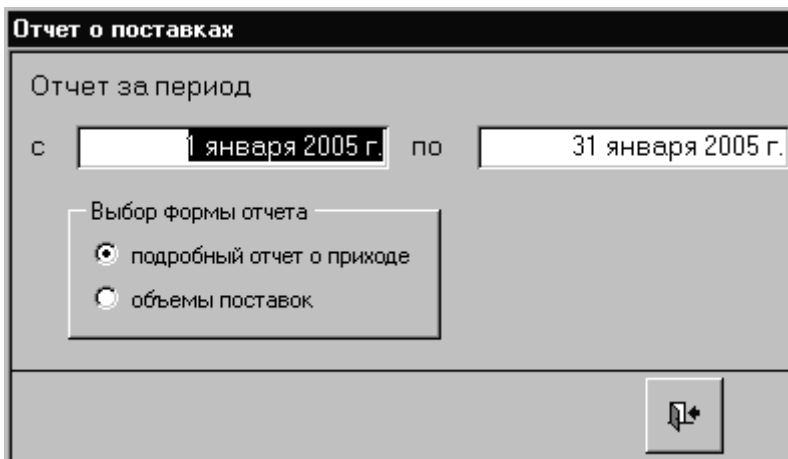


Рис. 88. Группа переключателей в форме «отчет_приход»

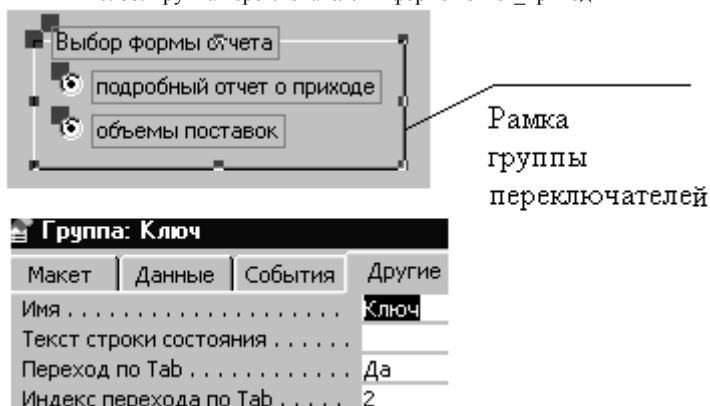


Рис. 89. Свойства группы переключателей

Конечно, система при создании любого элемента управления непременно дает ему имя, но имя типа «Кнопка7» или «Группа16» нам ни о чем не говорит. Поэтому в режиме конструктора два раза щелкнем левой клавишей мышки по рамке группы переключателей, как мы поступаем всегда, когда хотим открыть окно свойств элемента управления. Обратите внимание, что щелкнуть надо именно по рамке (рис. 89)! Если форма является контейнером для элементов управления, то группа пе-

реключателей, в свою очередь, является контейнером для переключателей. Поэтому щелкнув клавишей в другом месте, Вы можете открыть свойства надписи или, например, конкретного переключателя. Теперь в окне свойств группы переключателей установим имя «Ключ» и покинем конструктор.

7.3. Макросы. Нам предстоит использовать новый объект Microsoft Access – макрос, то есть структурированное описание одного или нескольких действий, которые должно выполнить приложение Access в ответ на определенное событие. Макрос строится из макрокоманд. Microsoft Access предоставляет в распоряжение пользователя свыше 40 макрокоманд, охватывающих большой набор стандартных действий, таких как открытие таблиц, форм, выполнение запросов, выбор пунктов меню и так далее. Войдем в раздел «Макросы» и нажмем кнопку «Создать». Откроется окно макросов в виде таблицы в две колонки: «Макрокоманда» и «Примечание». Их названия говорят сами за себя. Добавим в таблицу еще две колонки, воспользовавшись пунктами меню «**Вид → Имена макросов**» и «**Вид → Условия**». В первой строке первой колонки зададим имя макроса: «отчет_приход». Макрос, который мы хотим создать, должен запускаться из формы «отчет_приход». В зависимости от значения элемента управления, которому мы в предыдущем пункте дали имя «Ключ», макрос должен выдавать тот или иной отчет. В столбце «Условия» введем условие вывода первого отчета: «`[Forms]![отчет_приход]![Ключ]=1`». Обратите внимание, первый отчет формируется, если значение объекта с именем «Ключ» равно 1. Чтобы сослаться на этот объект, недостаточно его имени, так как в разных формах могут встречаться одноименные элементы управления. Поэтому в условии мы указываем, что объект принадлежит форме, даем название формы и уже потом имя объекта. В столбце «Макрокоманда» разворачиваем меню и выбираем макрокоманду «ОткрытьОтчет». В нижней части окна макросов слева расположена таблица аргументов макрокоманд. Здесь мы задаем имя отчета «приход_подробный» и режим «Просмотр». Аналогично оформим строку для второго отчета (рис. 90). При желании в столбце «Примечание» можно указать, какие действия вызывают заданные макрокоманды. Сохраним макрос под именем «Макрос1» и вернемся в раздел «Формы». Откроем в режиме конструктора форму «отчет_приход», на панели элементов отожмем «Кнопку мастера» (рис. 15), чтобы отключить «Мастера по созданию элементов управления».

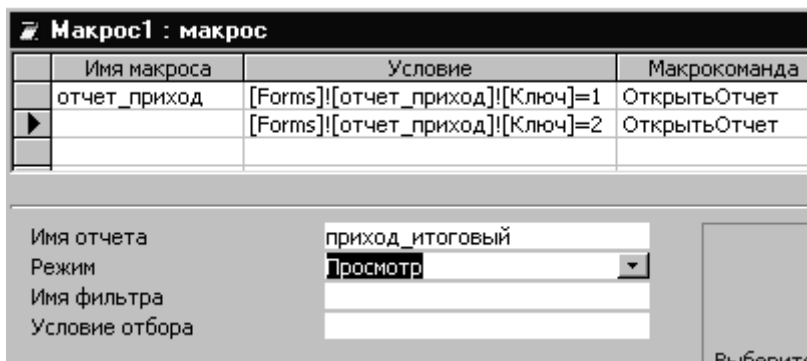


Рис. 90. Макрос «отчет_приход»

Возьмем с панели элементов кнопку и бросим в форму. Теперь – никакого диалога! Просто в форме появится кнопка, нажатие на которую не вызывает никаких действий. Откроем окно свойств кнопки, на заставке «Макет» щелкнем по строке «Рисунок», выберем рисунок (рис. 91) и нажмем «ОК».

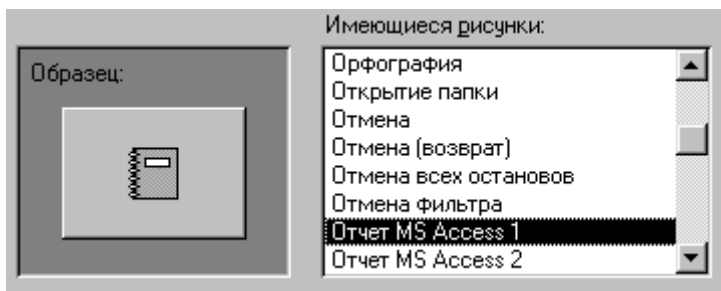


Рис. 91. Выбор рисунка для кнопки

Выберем заставку «События» и щелкнем клавишей по строке «Нажатие кнопки». Нажав в правой части строки стрелку, направленную вниз, развернем меню и выберем в качестве действия «Макрос1.отчет_приход» (рис. 92). Закроем окно свойств, выберем наиболее удачное расположение кнопки и выйдем из конструктора. Теперь в рабочем режиме форма будет иметь вид, представленный на рис. 93. Убедитесь, что при нажатии на кнопку в правой части формы на экран выводится отчет, соответствующий установленному положению переключателя.

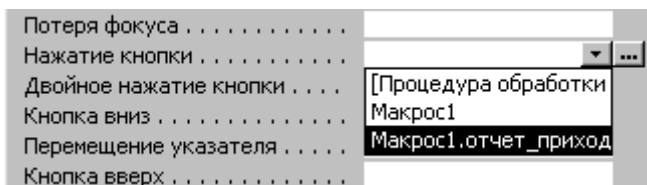


Рис. 92. Событию «Нажатие кнопки» приписано действие

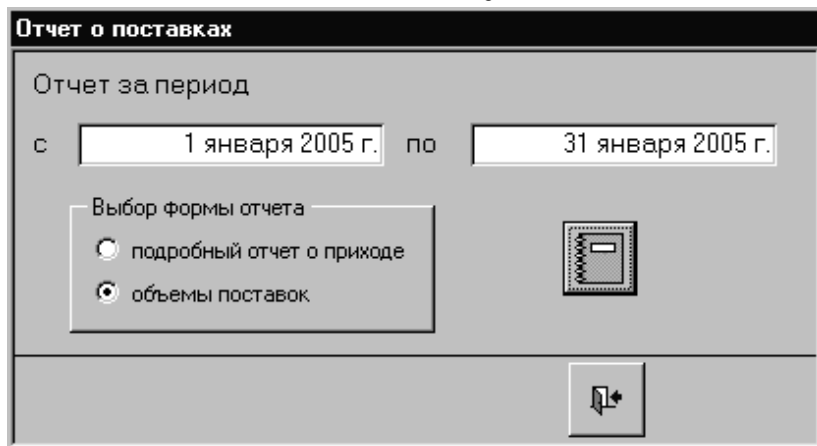


Рис. 93. Для получения выбранного отчета достаточно нажать кнопку слева

Точно так же, создав соответствующую группу переключателей и макрос, можно организовать вывод любого количества отчетов.

Теперь настало время приступить к усовершенствованию созданных отчетов.

7.4. Условия отбора данных. При создании отчета «приход_итоговый» в качестве источника данных взят запрос «итоги приход по поставщикам», для которого источником данных является в свою очередь запрос «приход по стоимости». Запрос «приход по стоимости» построен на основе запроса «приход_Запрос», а запрос «приход_Запрос» фактически содержит все поля и записи таблицы «приход», упорядоченные по возрастанию даты. Источником данных отчета «приход_подробный» является непосредственно таблица «приход». Поскольку наборы полей в таблице «приход» и в запросе «приход_Запрос» совпадают один к одному, мы можем в отчете «приход_подробный» заменить источник данных. Откроем указанный отчет в режиме конструктора, а затем окно

свойств отчета. Окно свойств отчета открывается точно так же, как и окно свойств формы. Выберем заставку «Данные» и щелкнем клавишей в строке «Источник записей». Развернем меню и выберем в качестве источника «приход Запрос» (рис. 94).

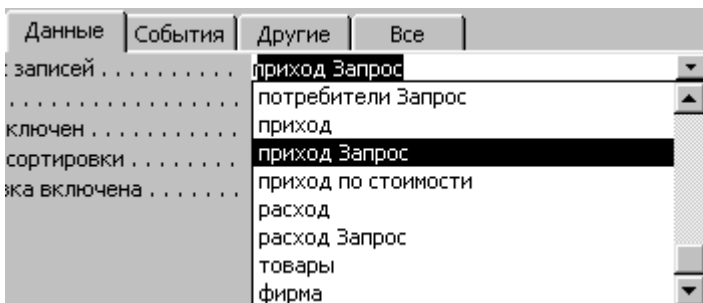


Рис. 94. Смена источника данных в отчете «приход_подробный»

Закроем окно свойств и конструктор. Теперь, открыв отчет для просмотра, Вы можете убедиться, что ровным счетом ничего не изменилось. Зато теперь оба отчета основаны на данных запроса «приход Запрос»! Перейдем в раздел «Запросы» и откроем «приход Запрос» в режиме конструктора.

Поле:	номер	дата
Имя таблицы:	приход	приход
Сортировка:		по возрастанию
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		Between [Forms]![отчет_приход]![дата1] And [Forms]![отчет_приход]![дата2]

Рис. 95. Условие отбора записей по дате

В строке «Сортировка» поля «дата» зададим условие отбора записей: **Between [Forms]![отчет_приход]![дата1] And [Forms]![отчет_приход]![дата2]**.

Оператор «Between» задает диапазон значений поля. Запрос производит отбор записей, которые в соответствующем поле имеют значение, попадающее в заданный диапазон. Например, мы можем отобразить записи о поставках в течение отчетного периода: с 1.01.05 по 31.01.05. Вместо «Between» мы могли бы использовать два неравенства, объединенных оператором «And»:

>=[Forms]![отчет_приход]![дата1] And <=[Forms]![отчет_приход]![дата2].

Открыв форму «отчет_приход», задайте интересующий Вас диапазон дат и убедитесь, что оба отчета отражают данные о поставках за заданный диапазон дат.

Заголовок отчета				
=[Forms]![отчет_приход]![фирма]				
Приход товаров				
Верхний колонтитул				
дата	товар	цена	кол-во	Стоимость
Заголовок группы 'поставщик'				
поставщик	поставщик			
Область данных				
дата	товар	цена	количество	:[цена]*[количество]
Примечание группы 'поставщик'				
Итого:				a*[количество]
Нижний колонтитул				
=Now()	="Страница " & [Page] & " из " & [Pages]			
Примечание отчета				
Итого по всем поставщикам:				a*[количество]
Заведующий складом	=[Forms]![отчет_приход]![заведующий]			
=Now()				

Рис. 96. Дополнительная информация в отчете

7.5. Дополнительная информация в отчете. Осталось внести в отчет две небольших корректировки: поместить в заголовок отчета название фирмы, а в примечание отчета – поле для подписи. В форме «отчет_приход» установим переключатель в положение «подробный отчет о приходе», откроем соответствующий отчет и переведем его в режим конструктора. С панели элементов возьмем элемент управления «поле» и поместим его в заголовок отчета. Надпись, соответствующую полю, удалим. В случае необходимости расширим заголовок и сдвинем вниз название отчета. Поместим в поле выражение `= [Forms]![отчет_приход]![фирма]`, то есть название фирмы. Теперь становится ясно, зачем мы назначили в качестве источника данных для формы «отчет_приход» таблицу «фирма»! Аналогично в примечание отчета поместим надпись «Заведующий складом», содержащую выражение `= [Forms]![отчет_приход]![заведующий]`, то есть Ф. И. О. заведующего складом, а также текущую дату, определяемую функцией

Now() (рис. 96). Аналогичным образом отредактируем и отчет по объемам поставок.

Замечание! Мы создали форму «приход», используя соответствующие запросы. Теперь, когда мы изменили запросы, нам придется заменить в этой форме источник данных. Для этого откроем форму «приход» в режиме конструктора, откроем окно свойств формы и в разделе «Данные» заменим источник данных на таблицу «приход».

8. Шаг восьмой. Несколько итоговых запросов

8.1. Создание итогового запроса в режиме конструктора. При разработке соответствующих отчетов по расходу товаров возникает проблема определения цен. Поскольку мы договорились, что товары распределяются по торговым точкам, принадлежащим нашей фирме, цены, по которым товары отпускаются, должны соответствовать ценам, по которым мы эти товары закупили. Но имеющиеся в наличии партии одного и того же товара могут быть приобретены по разным ценам. Значит, речь должна идти о средних ценах. Среднюю цену будем считать за заданное количество дней до последней поставки. Прежде всего, для каждого товара найдем дату последней поставки. Для этого в разделе «Запросы» нажмем кнопку «Создать». На этот раз создадим запрос в режиме конструктора. В меню диалогового окна выберем строку «Конструктор», а затем в окне «Добавление таблицы» выберем таблицу «приход». Нажмем «Добавить», а потом «Закрыть». В таблице «приход» в верхней части окна запроса два раза щелкнем по названиям полей «товар» и «дата». Выберем пункт меню «Вид → Групповые операции». В наборе свойств полей появится новая строка – «Групповая операция». В этой строке для поля «товар» выберем в меню значение «Группировка», а для поля «Дата» – значение Max (рис. 97). Это значит, что в запросе, как в логической таблице, каждый товар будет представлен один раз, а в поле, соответствующем полю «дата», будет стоять наибольшее значение даты поставки этого товара. Созданный запрос представлен на рис. 98. Теперь откроем в режиме конструктора таблицу «фирма» и добавим еще одно поле – «период», которое будет содержать количество дней до последней поставки, то есть период, за который считается средняя цена. Очевидно, это поле должно быть числовым.

Поле:	товар	дата
Имя таблицы:	приход	приход
Групповая операция:	Группировка	Max
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		
или:		

Рис. 97. Групповые операции в запросе

последняя_поставка : запрос на выборку		
	товар	Max_дата
▶	хлеб Окский	03.03.05
	хлеб Московский	20.03.05
	батон Нарезной	04.03.05
	колбаса Докторская	13.03.05
	молоко (1 литр)	23.03.05
	сметана (1 литр)	06.03.05
	макароны Рожки	18.03.05
	макароны Вермишель	17.03.05
	рыба Горбуша (кг)	12.03.05
	рыба Ряпушка (кг)	26.03.05
	рыба Осетр (кг)	12.02.05
	рыба Треска (кг)	12.03.05
	яблоки (кг)	10.03.05

Запись: ⏪ ⏩ 1 ▶ ▶▶ ▶* из 16

Рис. 98. Запрос «последняя_поставка»

После того, как Вы создадите форму «отчет_расход», аналогичную форме «отчет_приход», в этой форме надо поместить новое поле «период» (рис. 99). Чтобы обеспечить удобную Вам последовательность переходов, воспользуйтесь пунктом меню «Вид → Последовательность переходов». В дальнейшем из этой формы могут вызываться отчеты, использующие среднее значение цены.

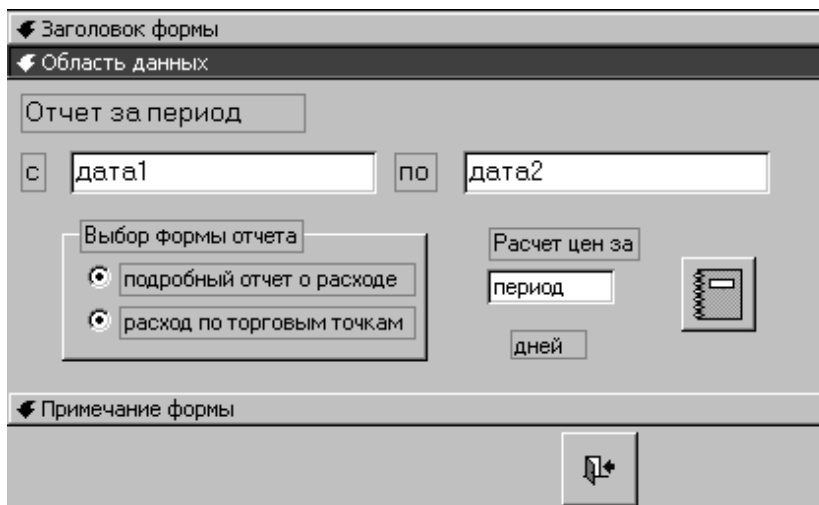


Рис. 99. Вид формы «отчет_расход» в режиме конструктора

8.2. Расчет диапазонов дат. Создадим простой запрос, источником данных для которого будет запрос «последняя_поставка». Дадим запросу имя «диапазон_дат», откроем его в режиме конструктора и добавим вычисляемое поле (рис. 100):

Min_дата: DateAdd("y";-[Forms]![отчет_расход]![период];[Max_дата]).

Max_дата	Min_дата: DateAdd("y";-[Forms]![отчет_расход]![период];[Max_дата])
последняя_поставка	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 100. Запрос «диапазон_дат» в режиме конструктора

Здесь «Min_дата» – имя вычисляемого поля, содержащего дату, предшествующую Max_дата. Функция DateAdd вычисляет дату, отличающуюся от исходной на заданное количество единиц времени. Первый аргумент «y» задает единицу времени – день. Кроме того, время может измеряться в неделях, месяцах, годах и даже в минутах, секундах и часах. Второй аргумент: -[Forms]![отчет_расход]![период]. Этот аргумент указывает, сколько единиц времени надо отступить от заданной в третьем аргументе даты. В качестве его значения берется значение поля «период» формы «отчет_расход» со знаком «минус», указывающим, что отсчет идет в обратном направлении. Тот факт, что запрос использует

значение поля из формы «отчет_расход», не означает, что мы не можем открыть этот запрос автономно. Просто при попытке открыть запрос появится диалоговое окно, в котором Вы введете это значение (рис. 101). Того же результата можно достичь, если сначала открыть форму «отчет_расход», а потом запрос. Созданный запрос представлен на рис. 102. Теперь каждому товару поставлен в соответствии интервал времени поставок, по которому следует вычислять среднюю цену (в нашем примере – 10 дней).

Рис. 101. Ввод количества дней

диапазон_дат : запрос на выборку			
	товар	Max_дата	Min_дата
▶	хлеб Окский	03.03.05	21.02.05
	хлеб Московский	20.03.05	10.03.05
	батон Нарезной	04.03.05	22.02.05
	колбаса Докторская	13.03.05	03.03.05
	молоко (1 литр)	23.03.05	13.03.05
	сметана (1 литр)	06.03.05	24.02.05
	макароны Рожки	18.03.05	08.03.05
	макароны Вермишель	17.03.05	07.03.05
	рыба Горбуша (кг)	12.03.05	02.03.05
	рыба Ряпушка (кг)	26.03.05	16.03.05
	рыба Осетр (кг)	12.02.05	02.02.05
	рыба Треска (кг)	12.03.05	02.03.05
	яблоки (кг)	10.03.05	28.02.05

Запись: 1 из 16

Рис. 102. Запрос «диапазон_дат»

9. Шаг девятый. Установка связей между таблицами

Теперь мы умеем формировать практически любые отчеты по приходу и расходу товаров. Но для принятия решений ничуть не меньший интерес представляет информация о наличии товаров на данный момент. Для этого мы должны научиться устанавливать связи между таблицами. Связь – способ объединения информации, хранящейся в разных таблицах.

9.1. Отбор данных по диапазону. Создадим простой запрос, в качестве источника данных для которого назначим таблицу «приход». Выберем поля «дата», «товар», «цена» и «количество». Сохраним запрос под именем «диапазон_стоимость». В этом запросе будут отражены поставки товаров за заданный период времени до последней поставки. Диапазон дат для каждого товара будет свой и в общем случае различный. Откроем запрос в режиме конструктора, щелкнем правой клавишей в его верхней части (в области таблиц) и выберем строку «Добавить таблицу». В открывшемся окне перейдем на заставку «Запросы», выберем запрос «диапазон_дат» и нажмем «Добавить», после чего окно «Добавление таблицы» закроем. Установим связь между таблицей «приход» и запросом «диапазон_дат» по полю «товар». Для этого нажмем левую клавишу мыши на поле «товар» в окне таблицы «приход», не отпуская клавиши, перетащим указатель на поле «товар» в окне «диапазон_дат». Теперь два одноименных поля соединены линией, что означает установление связи (рис. 103)! Если по линии связи дважды щелкнуть левой клавишей, появится окно «Параметры объединения». По умолчанию мы получили объединение только тех записей, в которых связанные поля обеих таблиц совпадают. Это нас вполне устраивает. Осталось только добавить в запрос поля «Min_дата» и «Max_дата». Для этого надо в окне «диапазон_дат» дважды щелкнуть левой клавишей по соответствующим полям. В строке «Условие отбора» для поля «дата» зададим условие: `Between [Min_дата] And [Max_дата]`. В строке «Вывод на экран» погасим галочку у полей «дата», «Min_дата» и «Max_дата», так как вывод на экран этих полей в дальнейшем не нужен. И, наконец, создадим в запросе вычисляемое поле «стоимость: [количество]*[цена]». В рабочем режиме построенный запрос будет иметь вид, представленный на рис. 104.

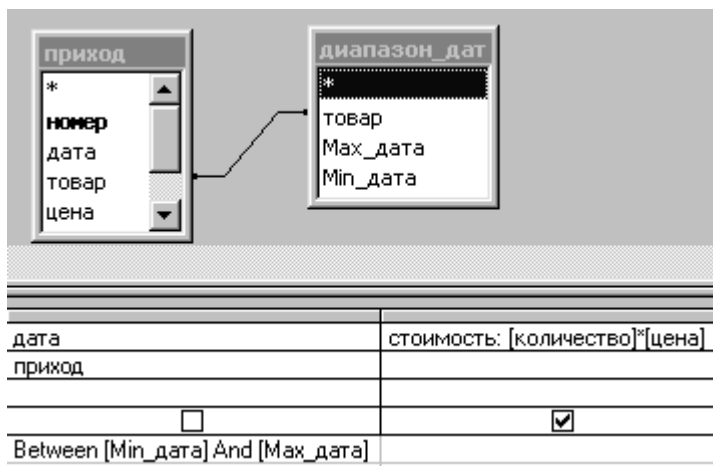


Рис. 103. Связь таблиц

диапазон_стоимость : запрос на выборку

	товар	цена	количество	стоимость
▶	рыба Горбуша (кг)	70,00	50	3500
	хлеб Окский	11,00	200	2200
	батон Нарезной	8,00	50	400
	яблоки (кг)	80,00	30	2400
	ветчина (кг)	100,00	20	2000
	ветчина (кг)	90,00	15	1350
	картофель (кг)	10,00	50	500
	сметана (1 литр)	20,00	100	2000
	яблоки (кг)	50,00	100	5000
	рыба Горбуша (кг)	75,00	30	2250
	рыба Треска (кг)	80,00	100	8000
	колбаса Докторская	110,00	200	22000
	молоко (1 литр)	12,00	100	1200

Запись: 1 из 20

Рис. 104. Запрос «диапазон_стоимость»

9.2. Расчет цен. Создадим простой запрос, в качестве источника данных для которого назначим последний созданный запрос «диапазон_ стоимость». Выберем поля «товар», «количество» и «стоимость».

Сделаем запрос итоговым. Поэтому поле «товар», по которому будет производиться группировка, должно стоять в списке первым. По полям «количество» и «стоимость» будем производить суммирование (рис. 105).

Поле	Sum	Avg	Min	Max
товар	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
количество	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
стоимость	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рис. 105. Суммирование по полям «количество» и «товар»

Сохраним запрос под именем «цена» и откроем его в режиме конструктора. Система присвоила двум вычисляемым полям имена «Sum _ количество» и «Sum _ стоимость». Исправим эти названия на «сум_количество» и «сум_стоимость». Теперь добавим в запрос вычисляемое поле

цена: $\text{If}([\text{сум_количество}]>0;[\text{сум_стоимость}]/[\text{сум_количество}];0)$.

Логическая функция If имеет три аргумента. Первый аргумент – логическое выражение. Если логическое выражение истинно, возвращается значение первого аргумента, иначе – второго. Мы использовали функцию «If» для того, чтобы заданное по ошибке нулевое значение поля «количество» не приводило к сбою в работе приложения.

На вычисляемом поле «цена» щелкнем правой клавишей мыши и откроем свойства поля. В разделе свойств «Общие» установим фиксированный формат поля. Этот формат отражает два знака после точки. В строке «Групповая операция» поля «цена» должно быть установлено значение «Выражение». Итак, мы построили запрос, который выдает средние взвешенные цены каждого товара, рассчитанные по соответствующему диапазону дат (рис. 106).

9.3. Расчет наличия товаров. Откроем запрос «расход Запрос» и добавим запрос «цена». Объединим таблицы по полю «товар» и добавим в запрос поле «цена» из запроса «цена». Теперь «расход Запрос» так же,

как и «приход Запрос», содержит цены всех товаров, правда, средние. Этот запрос может являться источником данных для множества расчетов по расходу товаров, аналогичных составленным ранее отчетам о приходе товаров.

цена : запрос на выборку				
	товар	сум_количест	сум_стоимость	цена
	хлеб Окский	200	2200	11,00
	хлеб Московский	200	2200	11,00
	батон Нарезной	50	400	8,00
	колбаса Докторская	200	22000	110,00
	молоко (1 литр)	150	1800	12,00
	сметана (1 литр)	100	2000	20,00
	макароны Рожки	150	4800	32,00
	макароны Вермишель	100	3000	30,00
	рыба Горбуша (кг)	80	5750	71,88
	рыба Ряпушка (кг)	50	2500	50,00
▶	рыба Осетр (кг)	100	10000	100,00
	рыба Треска (кг)	100	8000	80,00
	яблоки (кг)	130	7400	56,92

Запись: 11 из 16

Рис. 106. Запрос «цена»

Отчет о наличии товаров

Отчет за период


с января 2005 г. по января 2005 г.

Выбор формы отчета

цены

наличие

Расчет цен за дней






Рис. 107. Форма для вызова отчета о наличии товаров

товар	Sum_ количество: количество
приход	приход
Группировка	Sum
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис. 108. Запрос «приход_наличие»

Теперь для расчета наличия на основании таблиц «приход» и «расход» составим два простых итоговых запроса, выдающих соответственно для каждого товара сумму всего прихода и сумму всего расхода. Оба запроса содержат поля «товар» и «количество». Присвоим запросам соответственно имена «приход_наличие» и «расход_наличие». Вместо присвоенных системой имен «Sum_ количество» дадим в запросах суммарным количествам имена «приход» и «расход» соответственно (рис. 109). Тогда разность значений этих полей для соответствующего товара даст наличие товара на данный момент. Построим запрос для определения наличия товаров. Для этого создадим простой запрос, источником данных для которого будет «приход_наличие», и сохраним его под именем «наличие».

приход_наличие : запрос на выборку		расход_наличие : запрос на выборку	
товар	приход	товар	расход
хлеб Окский	600	хлеб Окский	350
хлеб Московский	600	хлеб Московский	370
батон Нарезной	250	батон Нарезной	65
колбаса Докторская	620	колбаса Докторская	347
молоко (1 литр)	900	молоко (1 литр)	473
сметана (1 литр)	300	сметана (1 литр)	160
макароны Рожки	450	макароны Рожки	260
макароны Вермишель	300	макароны Вермишель	175
рыба Горбуша (кг)	140	рыба Горбуша (кг)	85
рыба Ряпушка (кг)	50	рыба Ряпушка (кг)	45
рыба Осетр (кг)	200	рыба Осетр (кг)	67
рыба Треска (кг)	100	рыба Треска (кг)	98
яблоки (кг)	330	яблоки (кг)	197

Рис. 109. Запросы «приход_наличие» и «расход_наличие»

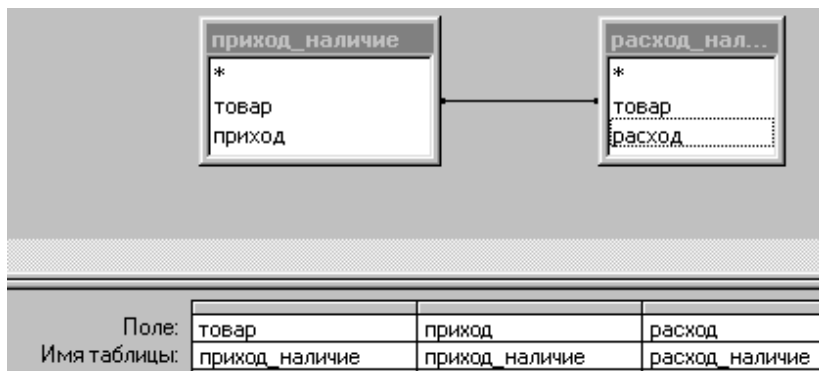


Рис. 110. Запрос «приход» в режиме конструктора

Фактически этот запрос будет просто копией своего источника и его можно было бы создать из запроса «приход_наличие», используя команды меню «Правка → Копировать» и «Правка → Вставить». Теперь откроем новый запрос в режиме конструктора, нажмем правую клавишу мышки в области таблиц запроса и выберем пункт меню «Добавить таблицу». В диалоговом окне перейдем на заставку «Запросы», добавим «расход_наличие» и закроем окно. Установим связь между запросами по полю «товар» (рис. 110). Мы уже знаем, что параметры объединения можно получить, щелкнув дважды левой клавишей по линии, связывающей таблицы или запросы. По умолчанию будут объединены только те записи, в которых связанные поля обеих таблиц совпадают. Это значит, что если некоторый товар имеется в таблице «приход», но не фигурирует нигде в таблице «расход», то в запросе «наличие» он не будет отражен. Это нас совершенно не устраивает. Ведь товар в наличии имеется! Поэтому мы должны изменить параметры объединения, установив переключатель в положение «Объединение всех записей из «приход_наличие» и только тех записей из «расход_наличие», в которых связанные поля совпадают» (рис. 111). Теперь можете убедиться, что любой товар, фигурирующий в приходе, но отсутствующий в расходе, будет отражен в запросе «наличие». В соответствующей ячейке Вы увидите «пробел». Однако фактически поле получит значение «Null», которое иначе называют «пустым значением». Это значение нельзя путать с нулем или с пустой строкой. Значение «Null» не равно никакому другому значению и поэтому не может участвовать в арифметических выражениях. Добавим в запрос «наличие» вычисляемое поле «наличие»:

наличие: [приход]-[расход].

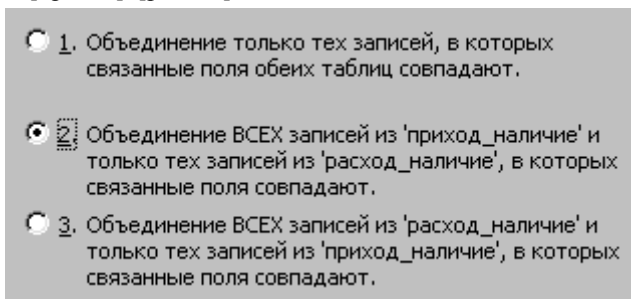


Рис. 111. Выбор параметров объединения

В нашем случае это значит, что мы, вместо имеющегося в наличии количества товаров, получим опять поле со значением Null. Чтобы избежать подобных затруднений, воспользуемся функцией Nz(). Строка «Поле» в запросе примет вид:

наличие: [приход]-Nz([расход]).

Функция Nz() возвратит значение 0, если значение поля «расход» будет Null (рис. 112). Если мы хотим вместо нуля получать какое-либо другое значение, мы введем это значение в качестве второго параметра функции.

наличие : запрос на выборку				
	товар	приход	расход	наличие
▶	хлеб Окский	600	350	250
	хлеб Московский	600	370	230
	батон Нарезной	250	65	185
	батон Студенческий	30		30
	колбаса Докторская	620	347	273
	молоко (1 литр)	900	473	427
	сметана (1 литр)	300	160	140

Рис. 112. Запрос «наличие»

Теперь осталось только создать отчет по наличию товаров. Вывод и печать отчетов по наличию товаров и средним ценам будет происходить из формы «Отчет о наличии товаров» (рис. 113).

Отчет о наличии товаров

Отчет за период

с по

Выбор формы отчета

цены

наличие

Расчет цен за

дней

Рис. 113. Форма «Отчет о наличии товаров»

9.4. Задание для самостоятельной работы. Теперь, когда создан запрос по расходу, содержащий среднее взвешенное значение цены каждого товара, Вам предлагается создать все отчеты по расходу, аналогичные соответствующим отчетам по приходу. Это большая, но крайне полезная работа.

10. Шаг десятый. Диспетчер кнопочных форм

Мы разработали набор отдельных объектов Microsoft Access. Но, пока, эти объекты существуют как бы сами по себе. Чтобы создать удобный для пользователя интерфейс с приложением, воспользуемся «Диспетчером кнопочных форм».

10.1. Создание кнопочной формы. В Access-97 диспетчер кнопочных форм вызывается командой меню «Сервис → Настройки → Диспетчер кнопочных форм». В версиях, начиная с 2000-й, – командой «Сервис → Служебные программы → Диспетчер кнопочных форм». При первом вызове «Диспетчера» на экране появится окно с сообщением: «Не удастся найти кнопочную форму в этой базе данных. Создать кнопочную форму?» Подтвердим желание создать форму. На экране появится окно диспетчера кнопочных форм с одной страницей. Нажмем кнопку «Из-

менить» и введем вместо предложенного системой новое имя страницы кнопочной формы – «Склад» (рис. 114) и приступим к созданию кнопок.

Рис. 114. Создание страницы кнопочной формы

Нажмем «Создать». Первой кнопке поставим в соответствие текст «Приход», команду «Открытие формы в режиме редактирования» и форму «приход» (рис. 115).

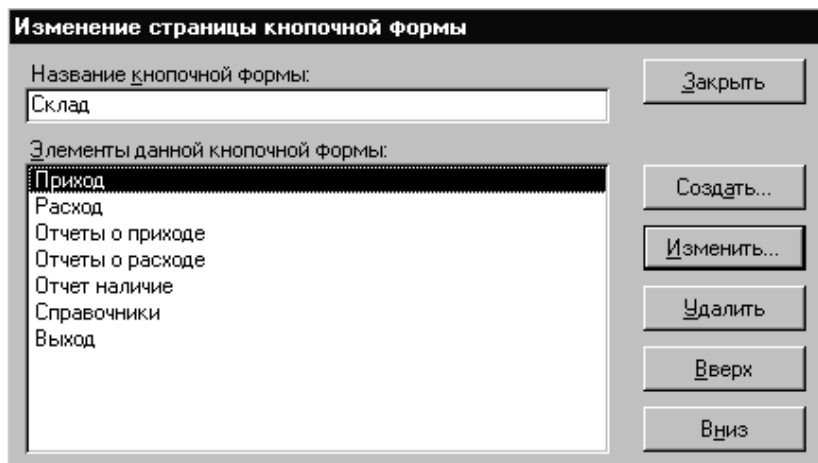
Рис. 115. Создание кнопок

Точно так же создадим кнопки для открытия форм «расход», «отчет_приход», «отчет_расход», «отчет_наличие». Закроем страницу кнопочной формы, а затем и само окно диспетчера. Проверим форму в работе. В разделе «Формы» окна базы данных появилась новая форма. Мы имеем право переименовать ее. Дадим кнопочной форме имя «Главная».

10.2. Редактирование кнопочной формы. Прежде, чем вносить корректировки в кнопочной форме, войдем в раздел «Макросы», откроем «Макрос1» в режиме конструктора и создадим под именем «Выход из приложения» макрос, состоящий из одной-единственной макрокоманды «Выход». Еще раз откроем кнопочную форму и в первом диалоговом окне нажмем кнопку «Создать» и, таким образом, создадим новую страницу под именем «Справочники». С этой страницы в дальнейшем мы будем вызывать формы для редактирования данных о поставщиках, потребителях и товарах. Нажмем кнопку «Изменить» и создадим кнопки

для открытия в режиме редактирования форм «поставщики_лента», «потребители_лента» и «товары», а затем кнопку «Главное меню» для перехода к первой странице кнопочной формы «Склад».

Рис. 116. Страница «Склад» кнопочной формы



Закроем страницу, выделим страницу «Склад» и откроем ее для изменения. На этой странице создадим кнопку с надписью «Справочники» и поставим ей в соответствие команду «Переход к кнопочной форме», ниже выберем в меню страницу кнопочной формы «Справочники». Также создадим кнопку «Выход», которая будет выполнять запуск макроса, созданного нами под именем «Макрос1.выход из приложения». При нажатии на эту кнопку будет осуществляться выход не только из базы данных, но и из Microsoft Access. Если теперь Вы войдете в раздел «Таблицы», то обнаружите новую таблицу, которая содержит всю информацию о кнопочной форме. Эту таблицу переименовывать нельзя и вообще лучше не трогать! Кнопочную форму можно редактировать в режиме конструктора, как и любую другую. В частности, Вы можете вставить рисунок, соответствующий назначению приложения. Проверьте форму «Главная» в работе (рис. 117). Теперь приложение готово к эксплуатации.

10.3. Последние штрихи. Таким образом, мы завершили разработку приложения. Многое можно было сделать лучше. И тем не менее при-

ложение справляется со своими задачами – а это главное! Напоследок – несколько советов.

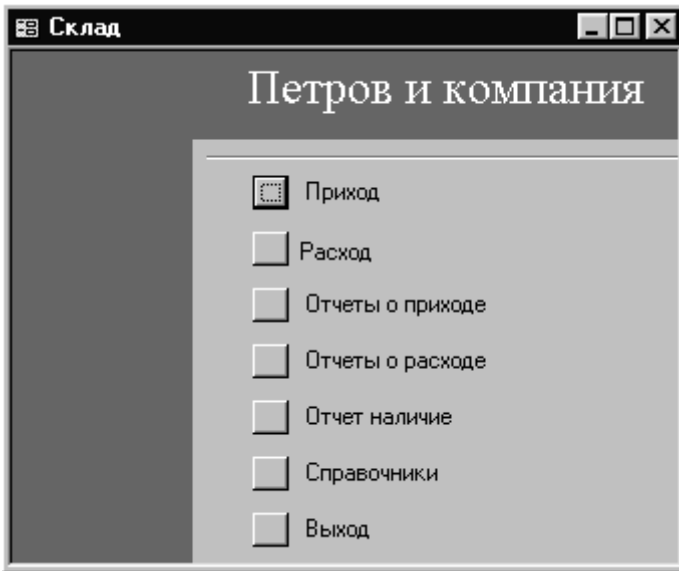


Рис. 117. Кнопочная форма

- В процессе разработки приложения мы не раз редактировали таблицы, запросы, формы и другие объекты базы. После корректировок файл с базой может содержать много ненужной информации. В частности, записи, удаленные в таблице, физически продолжают храниться в файле. Для удаления ненужной информации достаточно выбрать пункт меню «Сервис → Служебные программы → Сжать базу данных».
- Если Вы хотите, чтобы Ваше приложение работало не только на Вашем компьютере, то не следует создавать его в последней версии Access. Так, приложение, разработанное на Access-97, можно всегда преобразовать для работы с Access-2000 и выше, но обратное преобразование выполняется далеко не всегда. Для преобразования базы используют команду меню «Сервис → Служебные программы → Преобразовать базу данных». Прежде, чем перенести приложение на другой компьютер, следует войти в раздел «Модули» и откомпилировать все входящие в приложение модули.

- Окно базы данных необходимо для разработки приложения, но пользователю вовсе не обязательно его видеть. Кроме того, хотелось бы сразу после запуска приложения видеть главную кнопочную форму. Для этого выберем пункт меню «Сервис → Параметры запуска» и в открывшемся окне в поле под заголовком «Форма» установим имя «Главная». Кроме того, погасим все флажки, отвечающие за вывод на экран меню, панелей и прочих объектов. Теперь двойным щелчком по значку приложения будет запускаться нужная кнопочная (или другая заданная) форма. Если же Вы снова захотите войти в созданное приложение как разработчик, Вам достаточно войти в приложение, не отрывая левой руки от клавиши «Shift».
- Для защиты содержащейся в базе данных информации можно использовать пароль. Для этого надо открыть приложение в монопольном режиме и командой меню «Сервис → Защита → Задать пароль базы данных» вызвать соответствующее диалоговое окно. Пароль решает проблему несанкционированного доступа, но не мешает пользователю просматривать и изменять созданные Вами формы и другие объекты в режиме конструктора. Если Вы хотите лишить пользователя этого удовольствия, Вам придется в разделе «Служебные программы» создать MDE-файл.
- Если Вы разрабатываете приложение, предназначенное для эксплуатации другими пользователями, есть резон хранить все таблицы в отдельной базе данных. Для установки связей с таблицами, хранящимися в другой базе, используют команду меню «Файл → Внешние данные → Связь с таблицами». Это позволит Вам создавать новые версии приложения, не трогая таблиц с данными. Кроме того, такой подход удобен при организации доступа к данным с нескольких компьютеров. Теперь Вы имеете все необходимые начальные навыки по созданию приложений Microsoft Access. Остальное – дело опыта. Автору остается только пожелать Вам дальнейших успехов!

Список литературы

1. *Вейскас Дж.* Эффективная работа с Microsoft Access 97. СПб: «Питер», 2000.
2. *Винтер Рик.* Microsoft Access 97: Справочник. СПб: «Питер», 1998.
3. *Дейт Крис.* Введение в системы баз данных. М.: «Вильямс», 2001.
4. *Джонс Э., Джонс Дж.* Access 97: Книга ответов. СПб: «Питер», 1998.
5. *Камминг Стив.* VBA для «чайников». М.: «Вильямс», 2002.
6. *Назаров С. В., Мельников П. П.* Программирование на MS VISUAL BASIC. М.: Финансы и статистика, 2002.

Содержание

Предисловие	3
1. Шаг первый. Работа с таблицами и запросами	5
2. Шаг второй. Формы и элементы управления	14
3. Шаг третий. Свойства форм	25
4. Шаг четвертый, или первый самостоятельный	37
5. Шаг пятый. Мастер подстановок	41
6. Шаг шестой. Отчеты	53
7. Шаг седьмой. Доработка отчетов	64
8. Шаг восьмой. Несколько итоговых запросов	74
9. Шаг девятый. Установка связей между таблицами	78
10. Шаг десятый. Диспетчер кнопочных форм	85
Список литературы	90

Учебное издание

Белый Евгений Константинович

Десять шагов в Microsoft Access

Учебное пособие

Редактор С. Л. Смирнова

Компьютерный набор и верстка – Е. К. Белый

Художник обложки – А. И. Морозов

Подписано в печать 26.09.05. Формат 60x84 ¹/₁₆.
Бумага офсетная. Печать офсетная. Уч.-изд. л. 5,7.
Тираж 350 экз. Изд. № 87.

Государственное образовательное учреждение
высшего профессионального образования
**ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ**

Отпечатано в типографии Издательства ПетрГУ
185910, г. Петрозаводск, пр. Ленина, 33