

Министерство образования Республики Беларусь

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТРАНСПОРТА**

**Кафедра микропроцессорной техники
и информационно-управляющих систем**

Н. В. РЯЗАНЦЕВА, С. Н. ХАРЛАП

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

**Практикум по выполнению лабораторных работ
по дисциплине «Системы управления базами данных»**

Гомель 2001

Министерство образования Республики Беларусь

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТРАНСПОРТА

Кафедра микропроцессорной техники
и информационно-управляющих систем

Н. В. РЯЗАНЦЕВА, С. Н. ХАРЛАП

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Практикум по выполнению лабораторных работ
по дисциплине «Системы управления базами данных»

*Одобен методической комиссией электротехнического факультета
Белорусского государственного университета транспорта*

Гомель 2001

УДК 681.325.5 (076.5)

Р 993

Рязанцева Н. В., Харлап С. Н.

Р 993 Системы управления базами данных: Практикум по выполнению лабораторных работ по дисциплине «Системы управления базами данных». – Гомель: БелГУТ, 2001. – 40 с.

Раскрыты основные вопросы работы с базами данных в *Microsoft Access*. Приведены примеры создания таблиц, запросов, форм и отчетов.

Предназначен для студентов специальности «Автоматика, телемеханика и связь на транспорте».

Р е ц е н з е н т – канд. техн. наук доцент **М. Л. Шишаков** (БелГУТ).

© Н. В. Рязанцева, С. Н. Харлап, 2001.

ВВЕДЕНИЕ

База данных – это организованная структура, предназначенная для хранения информации. Почти все современные системы управления базами данных (СУБД) основаны на реляционной модели управления данными. Название «реляционная» связано с тем, что каждая запись в такой базе содержит информацию, относящуюся (related) только к одному объекту. Кроме того, с данными двух типов можно работать как с единым целым, основанным на значениях связанных между собой данных. В реляционной системе управления базами данных все обрабатываемые данные представляются в виде таблиц. В ранних версиях СУБД термин «база данных» обычно относился только к файлам, в которых хранятся данные. В Microsoft Access база данных включает в себя все объекты, связанные с данными, в том числе и те, которые предназначены для автоматизации работы с ними.

Microsoft Access называет объектами все, что может иметь имя, такими объектами являются таблицы, запросы, формы, макросы и модули. Рассмотрим список объектов Microsoft Access.

Таблица – объект, который используется для хранения информации. Таблица содержит *поля* (столбцы), в которых хранятся различного рода данные, например, фамилия и адрес, и *записи* (строки). В записи собрана информация о конкретном предмете, например, о человеке, образце продукции и т. д.

Запрос – объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц. Для создания запроса можно использовать QBE (запрос по образцу) или инструкции SQL. Можно создавать запросы на выборку, удаление или добавление данных, можно создавать новые таблицы, используя данные одной или нескольких таблиц, которые уже существуют.

Форма – объект, предназначенный в основном для ввода данных, отображения их на экране или управления работой приложения.

Отчет – объект, предназначенный для создания документа, который впоследствии может быть распечатан или включен в документ другого приложения.

Макрос – объект, представляющий собой структурированное описание одного или нескольких действий, которые, по мнению разработчика, должен выполнить Access в ответ на определенное событие.

Модуль – объект, содержащий программы на Microsoft Access Basic, которые позволяют разбить процесс на более мелкие действия и обнаружить ошибки, которые невозможно обнаружить с помощью макросов. Модули могут быть независимыми объектами, содержащими функции, которые вызываются из любого места приложения, также модули могут быть «привязаны» к отдельным формам или отчетам для реакции на происходящие в них изменения.

Страницы – специальные объекты баз данных, реализованные в последней версии СУБД Microsoft Access (Access 2000). Более корректно их называть страницами доступа к данным. Физически это особый объект, выполненный в коде HTML, размещаемый на Web-странице и передаваемый клиенту вместе с ней. Сам по себе этот объект не является базой данных, но содержит компоненты, через которые осуществляется связь переданной Web-страницы с базой данных, остающейся на сервере. Пользуясь этими компонентами, посетитель Web-узла может просматривать записи базы в полях страницы доступа. Таким образом, страницы доступа к данным осуществляют интерфейс между клиентом, сервером и базой данных, размещенной на сервере.

Лабораторная работа № 1

СОЗДАНИЕ БАЗОВЫХ ТАБЛИЦ

Цель работы: изучение методов создания базы данных при помощи Microsoft Access, условий, необходимых для создания взаимосвязанных таблиц, и приемы их создания, ознакомление с основными типами связей, образующихся между таблицами, порядок редактирования параметров связи.

1 Краткие сведения из теории

Каждое поле таблицы базы данных СУБД Microsoft Access характеризуется свойствами. Рассмотрим основные из них.

Имя поля. Это свойство (FieldName) определяет имя поля в таблице. Оно должно удовлетворять соглашениям об именах объектов Microsoft Access. Это имя не должно совпадать с именем другого поля в этой таблице. Необходимо избегать употребления имен полей, совпадающих с именами встроенных функций или свойств Microsoft Access. Например, имя «Name» совпадает с именем встроенного свойства Name. Значение данного свойства можно задать только в верхней половине окна режима конструктора таблицы или в программе Visual Basic. Microsoft Access различает поля по их именам. После того как имя поля задано в режиме конструктора таблицы, становится возможным его использование в выражениях, в процедурах Visual Basic и в инструкциях SQL.

Тип поля. Базы данных Microsoft Access работают со следующими типами данных полей.

Текстовый – тип данных, используемый для хранения обычного неформатированного текста ограниченного размера (до 255 символов).

Поле Мемо – специальный тип данных для хранения больших объемов текста (до 65 535 символов). Физически текст не хранится в поле, а в другом месте базы данных, в поле хранится указатель на него, но для пользователя такое разделение заметно не всегда.

Числовой – тип данных для хранения действительных чисел.

Дата/время – тип данных для хранения календарных дат и текущего времени.

Денежный – тип данных для хранения денежных сумм.

Счетчик – специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием. Естественное использование – для порядковой нумерации записей.

Логический – тип для хранения логических данных (могут принимать только два значения, например, «Да» или «Нет»).

Поле объекта OLE – специальный тип данных, предназначенный для хранения объектов OLE, например мультимедийных. Реально, конечно, такие объекты в таблице не хранятся. Как и в случае полей MEMO, они хранятся в другом месте внутренней структуры файла базы данных, а в таблице хранятся только указатели на них (иначе работа с таблицами была бы чрезвычайно замедленной).

Гиперссылка – специальное поле для хранения адресов URL Web-объектов Интернета. При щелчке на ссылке автоматически происходит запуск браузера и воспроизведение объекта в его окне.

Мастер подстановок – это не специальный тип данных. Это объект, настройкой которого можно автоматизировать ввод данных в поле так, чтобы не вводить их вручную, а выбирать из раскрывающегося списка.

Размер поля. Свойство (FieldSize) определяет максимальный размер данных, которые могут сохраняться в полях с типом данных Текстовый, Числовой или Счетчик. Если свойство «Тип данных» (DataType) имеет значение «Текстовый», значением данного свойства должно быть целое число в диапазоне от 0 до 255. По умолчанию задается размер 50.

Если свойство «Тип данных» имеет значение «Счетчик», то допустимыми являются значения свойства «Размер поля», «Длинное целое» или «Код репликации». Если это свойство имеет значение «Числовой», то допустимыми являются значения свойства «Размер поля», приведенные в таблице 1.

Таблица 1 – Значения свойства «Размер поля»

Значение	Описание	Дробная часть	Размер
Байт	Числа от 0 до 255 (без дробной части).	Отсутствует	1 байт
Целое	Числа от -32 768 до 32 767 (без дробной части).	Отсутствует	2 байта
Длинное целое (значение по умолчанию)	Числа от -2 147 483 648 до 2 147 483 647 (без дробной части).	Отсутствует	4 байта
С плавающей точкой (4 байт)	Числа от -3,402823Е38 до 1,401298Е-45 для отрицательных значений и от 1,401298Е-45 до 3,402823Е38 для положительных	7	4 байта
С плавающей точкой (8 байт)	Числа от 1,79769313486232Е308 до 4,94065645841247Е для отрицательных значений и от 1,79769313486231Е308 до 4,94065645841247Е-324 для положительных	15	8 байт
Код репликации	Уникальный глобальный идентификатор (GUID).	Не определено	16 байт

Значение данного свойства можно задать в окне свойств таблицы. Для того чтобы в программе Visual Basic прочитать или задать максимальный размер текстового поля, следует использовать свойство объектов доступа к данным Size. Для полей других типов значение свойства Size автоматически определяется значением свойства Type. Пользователь имеет возможность указать стандартные размеры текстовых и числовых полей в группе «Размер поля» по умолчанию на вкладке «Таблицы/запросы» в диалоговом окне «Параметры», которое открывается командой «Параметры» в меню «Сервис». Рекомендуется задавать минимально допустимое значение свой-

ства «Размер поля» (FieldSize), поскольку обработка данных меньшего размера выполняется быстрее и требует меньше памяти.

Преобразование большего значения свойства «Размер поля» к меньшему в таблице, которая уже содержит данные, может привести к потере данных. Например, при уменьшении размера текстового поля с 255 до 50 все значения, длина которых превышает 50 символов, будут усечены.

Данные в числовом поле, которые выходят за пределы диапазона, соответствующего новому размеру поля, округляются или заменяются пустыми (Null) значениями. Например, при замене значения «С плавающей точкой (4 байта)» на «Целое» дробные числа будут округлены до ближайшего целого числа, а значения вне диапазона от -32 768 до 32 767 будут преобразованы в пустые значения. Невозможно отменить изменения данных, произошедшие из-за модификации свойства «Размер поля», после его сохранения в режиме конструктора таблицы. Рекомендуется использовать денежный тип данных для полей, в которых планируется хранить числовые значения с одним-четырьмя знаками в дробной части. При обработке числовых значений из полей типа «С плавающей точкой (4 байта)» (Single) и «С плавающей точкой (8 байт)» (Double) применяются вычисления с плавающей точкой. При обработке числовых значений из денежных полей используются более быстрые вычисления с фиксированной точкой.

Формат поля. Свойство (Format) позволяет указать форматы вывода текста, чисел, дат и значений времени на экран и на печать. Например, для поля «Цена» разумно указать в свойстве «Формат поля» формат «Денежный» и установить для его свойства Число десятичных знаков (DecimalPlaces) значение 2 или «Авто». В этом случае введенное в поле значение 4321,678 будет отображаться как 4 321,68 руб. Допустимо как использование встроенных, так и специальных форматов, созданных при помощи символов форматирования. В свойстве «Формат поля» задаются разные настройки для различных типов данных. Для элемента управления значение данного свойства задается в окне свойств. Для поля в таблице или запросе значение данного свойства задается в режиме конструктора таблицы (в разделе свойств поля) или в окне запроса (в окне свойств поля). Кроме того, значение данного свойства можно задать в макросе или в программе Visual Basic. В программе Visual Basic следует ввести строковое выражение, соответствующее одному из стандартных форматов или задающее специальный формат. Свойство «Формат поля» определяет только способ отображения данных. Оно не влияет на способ их сохранения. В Microsoft Access определены стандартные форматы для полей с типами данных «Числовой», «Дата/время», «Логический», «Текстовый» и «Поле MEMO». В качестве стандартных используются национальные форматы, выбираемые в окне «Язык и

стандарты», панели управления Windows. Набор форматов определяется настройками для конкретной страны. Например, если на вкладке «Язык и стандарты» указать Английский (США), то число 1234.56 в денежном формате будет выглядеть как \$1.234,56. Но если указать на этой вкладке «Русский», то это число будет выглядеть так: 1234,56 руб. Настройка «Формат поля» (Format), заданная в режиме конструктора таблицы, используется для отображения данных в режиме таблицы. Эта же настройка применяется при создании связанных с этим полем новых элементов управления в форме или отчете. В таблице 2 перечислены символы, используемые при определении специальных форматов для любого типа данных.

Таблица 2 – Символы форматов

Символ	Значение
(Пробел)	Выводит пробел как символьную константу
"ABC"	Все символы внутри кавычек считаются символьными константами
!	Выравнивает символы по левому, а не по правому краю
*	Заполняет доступное пустое пространство следующим символом
[Цвет]	Задаёт цвет, название которого указано в скобках. Допустимые имена цветов: Черный (Black), Синий (Blue), Зеленый (Green), Бирюзовый (Cyan), Красный (Red), Лиловый (Magenta), Желтый (Yellow), Белый (White)
\	Выводит следующий символ как символьную константу. Для этой же цели можно использовать кавычки

Не разрешается смешивать в одном формате специальные символы, предназначенные для определения числовых форматов, форматов даты/времени и текстовых форматов. Если для поля определена маска ввода, а в свойстве «Формат поля» задается другое форматирование тех же данных, то приоритет имеют настройки, задаваемые в свойстве «Формат поля», и маска ввода игнорируется. Например, если в режиме таблицы определяется маска ввода пароля одновременно с указанием значения свойства «Формат поля» либо для таблицы, либо для элемента управления в форме, то маска ввода пароля будет игнорироваться, а данные отображаться в соответствии со значением свойства «Формат поля».

Маска ввода. Свойство (InputMask) задает маску ввода, облегчающую ввод данных в элемент управления – поле. Значение данного свойства определяется автоматически, при использовании мастера по созданию масок ввода. Значение свойства «Маска ввода» может содержать до трех разделов, разделяемых точкой с запятой (таблица 3).

При создании маски ввода пользователь имеет возможность указать, что часть данных следует вводить обязательно (например, региональный код для телефонных номеров), а другие данные являются необязательными (например, добавочный номер телефона). Эти символы определяют тип данных, например номер символа, который необходимо ввести для каждого символа маски ввода. Символы, которые следует вводить в маску ввода, определяются специальными символами, приведенными в таблице 4. Значение «Пароль» свойства «Маска ввода» определяет элемент управления, предназначенный для ввода пароля. Любые символы, вводящиеся в этот элемент управления, будут отображаться на экране звездочками (*). Подобная маска ввода используется для предотвращения вывода на экран набираемых на клавиатуре символов.

Таблица 3 – Содержание разделов маски ввода

Раздел	Описание
Первый	Представляет саму маску ввода (например, !(999) 000-0000). Перечень символов, используемых для определения масок ввода, приводится ниже в таблице 4
Второй	Определяет режим занесения в таблицу строковых констант, добавляемых к символам, вводящимся пользователем. Введенный в данный компонент символ 0 указывает, что постоянные символы (например, скобки и дефисы в маске ввода телефонных номеров) сохраняются вместе с введенными пользователем символами; значение 1 или пустое значение данного компонента указывает, что сохраняются только символы, введенные пользователем
Третий	Определяет символ, используемый для изображения пустых позиций в маске ввода, в которые помещаются вводящиеся пользователем символы. В этом компоненте можно указать любой символ ANSI; пробел необходимо заключить в кавычки (" ")

Для элемента управления значение данного свойства задается в окне свойств. Для поля в таблице или запросе значение данного свойства задается в режиме конструктора таблицы (в окне свойств поля) или в режиме конструктора окна запроса (в окне свойств поля). Кроме того, значение свойства «Маска ввода» можно задать в макросе или в программе Visual Basic.

Маска ввода используется только при вводе символов в поле или в поле со списком с клавиатуры и игнорируется при всех остальных операциях, например, при импорте данных, при выполнении запроса на изменение, а также при вводе символов в элемент управления с помощью конструкции Visual Basic, в которой задается значение свойства Text элемента управления, или при выполнении макрокоманды «Задать значение» (SetValue) в

макросе. Если пользователь определяет маску ввода и одновременно задает для того же поля значение свойства «Формат поля», то при выводе данных приоритет имеет свойство «Формат поля». Это означает, что при форматировании данных сохраненная маска ввода игнорируется. Данные в базовой таблице при этом не изменяются; свойство «Формат поля» определяет только режим отображения данных.

Подпись. Свойство (Caption) определяет текст, который выводится в подписях объектов в различных режимах. Подпись поля указывает текст, который выводится в подписях, присоединенных к элементам управления, создаваемых путем переноса с помощью мыши из списка полей. Этот текст также выводится для таблицы или запроса в заголовке столбца в режиме таблицы.

Таблица 4 – Специальные символы маски ввода

Символ	Описание
0	Цифра (обязательный символ; знаки (+) и (-) не разрешены)
9	Цифра или пробел (необязательный символ; знаки (+) и (-) не разрешены)
#	Цифра или пробел (необязательный символ; незаполненные позиции выводятся как пробелы в режиме редактирования, но удаляются при сохранении данных; знаки (+) и (-) не разрешены)
L	Буква (обязательный символ)
?	Буква (необязательный символ)
A	Буква или цифра (обязательный символ)
a	Буква или цифра (необязательный символ)
&	Любой символ или пробел (обязательный символ)
C	Любой символ или пробел (необязательный символ)
<	Преобразует все символы к нижнему регистру
>	Преобразует все символы к верхнему регистру
!	Указывает, что маска ввода заполняется справа налево; этот символ следует использовать, если в левой части маски находятся позиции, заполнять которые не обязательно. Маски ввода обычно заполняются слева направо. Символ восклицательного знака можно помещать в произвольную позицию в маске ввода
\	Указывает, что следующий символ следует воспринимать как постоянный (а не специальный) символ (например, \A представляет символ «A»)
. , : ; - /	Десятичный разделитель, разделители групп разрядов, времени или даты. (Используемые символы разделителей определяются настройками, выбранными в окне Язык и стандарты панели управления Windows)

Значением свойства «Подпись» является строковое выражение, длиной до 2048 символов. Подписи форм или отчетов, которые не умецаются в строке заголовка, обрезаются. Для элементов управления значение данного свойства задается в окне свойств. Для полей значение данного свойства задается в окне свойств в режиме конструктора таблицы или в окне запроса (в окне свойств списка полей). Кроме того, можно задать значения данного свойства в макросе или в программе Visual Basic. Если пользователь не указывает текст подписи поля таблицы, то в качестве текста подписи элемента управления и заголовка столбца в режиме таблицы используется значение свойства «Имя поля». Свойство «Подпись» используется при определении назначенной клавиши для элемента управления подпись или кнопка. Для этого в подпись поля следует включить амперсанд (&) непосредственно перед символом, который указывает на назначенную клавишу и выводится в подписи с подчеркиванием. Для передачи фокуса данному элементу управления следует одновременно нажать назначенную клавишу и клавишу ALT. Если символ амперсанда должен являться частью текста подписи, введите этот символ дважды (&&). Например, чтобы создать подпись «Save & exit», следует ввести в ячейку свойства «Подпись» текст Save && exit.

Значение по умолчанию. Свойство «Значение по умолчанию» (DefaultValue) позволяет указать значение, автоматически вводимое в поле при создании новой записи. Например, в таблице «Адреса» может оказаться удобным указать автоматический ввод значения «Гомель» в поле «Город». При заполнении таблицы пользователи смогут оставить в этом поле стандартное значение или, при необходимости, указать другой город. Свойство «Значение по умолчанию» не определено для следующих элементов управления: флажки, переключатели и выключатели, входящие в группу параметров. Оно определено только для самой группы. Свойство «Значение по умолчанию» определено для всех типов полей в таблицах за исключением полей с типом данных «Счетчик» или «Поле объекта OLE». В свойстве «Значение по умолчанию» задается текст или выражение, значение которого автоматически вводится в поле при создании новой записи. Например, если в свойстве поля «Значение по умолчанию» задается выражение =Now(), то в поле автоматически выводятся текущие значения даты и времени. Максимальная длина текста или выражения, задающего значение этого свойства, составляет 255 символов. Значение данного свойства для элемента управления задается в окне его стандартных свойств. Для поля значение данного свойства можно установить в режиме конструктора таблицы (в области свойств поля), в макросе или в программе Visual Basic.

Условие на значение. Сообщение об ошибке. Свойство «Условие на значение» (ValidationRule) определяет требования к данным, вводимым в

запись, поле или элемент управления. Свойство «Сообщение об ошибке» позволяет указать текст сообщения, выводящегося на экран, если введенные данные нарушают условие, определенное в свойстве «Условие на значение». Свойства «Условие на значение» и «Сообщение об ошибке» (ValidationText) не применимы к таким элементам управления, как отдельные флажки, переключатели или выключатели, входящие в группу. Они применяются только для самой группы. Значение свойства «Условие на значение» определяется с помощью выражения, а значение свойства «Сообщение об ошибке» – с помощью строкового значения. Максимальная длина значения свойства «Условие на значение» составляет 2048 символов. Максимальная длина значения свойства «Сообщение об ошибке» составляет 255 символов.

Значения свойств «Условие на значение» и «Сообщение об ошибке» задаются:

- в разделе свойств поля в режиме конструктора таблицы (условие на значение поля);
- окне свойств таблицы, вызываемом командой «Свойства» меню «Вид» в режиме конструктора таблицы (условие на значение записи);
- окне свойств элемента управления формы;
- макросе или в программе Visual Basic. В программе Visual Basic значения данных свойств задаются при помощи строковых выражений.

Microsoft Access автоматически накладывает условия на значения, определяемые типом данных поля; например, не допускается ввод текста в числовые поля. Свойство «Условие на значение» позволяет указать дополнительные условия. Если пользователь задает значение свойства «Условие на значение», но не определяет свойство «Сообщение об ошибке», то при нарушении условия на значение Microsoft Access выводит стандартное сообщение об ошибке. Если значение свойства «Сообщение об ошибке» задано, то в сообщении об ошибке выводится текст, указанный в качестве значения этого свойства.

Проверка условий на значение, определенных для элементов управления, полей и записей, выполняется в следующих случаях:

- условия на значение поля и элемента управления проверяются после изменения данных при переводе фокуса из этого поля или элемента управления;
- условия на значение записи проверяются при переходе на другую запись;
- если условия определены и для поля, и для присоединенного к этому полю элемента управления, то проверка обоих условий выполняется при

потере фокуса этим полем или элементом управления, если его значение было изменено.

В таблице 5 приводятся примеры выражений, определяющих свойства «Условие на значение» и «Сообщение об ошибке».

По умолчанию поле, для которого определено условие на значение, не может содержать пустые (Null) значения. Для того, чтобы сделать пустые значения допустимыми, следует в выражение, определяющее условие на значение, добавить оператор Is Null, например, $\langle \rangle 8 \text{ Or Is Null}$, и убедиться, что для свойства «Обязательное поле» (Required) задано значение «Нет».

Невозможно определить условия на значение для полей или записей таблиц, созданных вне Microsoft Access (например, для таблиц dBASE, Paradox или SQL Server). Таблицы этих типов допускают определение условий на значение только для элементов управления.

Таблица 5 – Примеры выражений, определяющих свойства «Условие на значение»

Условие на значение	Сообщение об ошибке
$\langle \rangle 0$	Требуется ненулевое значение
$> 1000 \text{ Or Is Null}$	Допускаются пустые значения или значения, превышающие 1000
Like "A????"	Значение должно содержать 5 символов и начинаться с «А»
$\geq \#1/1/96\# \text{ And } \langle \#1/1/97\#$	Требуется дата, относящаяся к 1996 г.
DLookup("КодКлиента", "Клиенты", "КодКлиента = Forms!Клиенты!КодКлиента") Is Null	Требуется уникальное значение в поле «КодКлиента» (статистические функции допустимы только для условий на значение на уровне формы)

Обязательное поле. Свойство «Обязательное поле» (Required) указывает, что в этом поле обязательно требуется ввести данные.

Пустые строки. Свойство «Пустые строки» (AllowZeroLength) определяет, допускается ли ввод в данное поле пустых строк (" "). Свойство определено только для полей таблиц с типом данных «Текстовый», «Поле МЕМО» или «Гиперссылка». Допустимые значения свойства «Пустые строки» представлены в таблице 6.

Таблица 6 – Значения свойства «Пустые строки»

Значение	Описание	Visual Basic
Да	Пустые строки являются допустимыми значениями	True (-1)

Нет	(Значение по умолчанию). Пустые строки не являются допустимыми значениями	False (0)
-----	---------------------------------------------------------------------------	-----------

Значение данного свойства задается в окне свойств таблицы или в программе Visual Basic. При задании значения «Да» для свойств «Пустые строки» и «Обязательное поле» Microsoft Access различает несуществующие данные (сохраняемые в виде пустых строк) и данные, которые существуют, но не известны (сохраняются в виде пустых (Null) значений). В таблице 7 показано, к каким результатам приводит комбинирование значений свойств «Пустые строки» и «Обязательное поле».

Для различения пустых строк от значений Null используется свойство «Формат поля». При этом вместо пустых строк можно выводить строку «Пусто». Свойства «Пустые строки» и «Обязательное поле» используются независимо друг от друга. Свойство «Обязательное поле» определяет, являются ли допустимыми пустые (Null) значения поля. Если для свойства «Пустые строки» задано значение «Да», то пустые строки становятся допустимыми значениями данного поля вне зависимости от значения свойства «Обязательное поле».

Таблица 7 – Результаты использования значений свойства «Пустые строки»

Пустые строки	Обязательное поле	Действия пользователя	Значение поля
Нет	Нет	Нажимает клавишу ENTER Нажимает клавишу ПРОБЕЛ Вводит пустую строку	Null Null (не допускается)
Да	Нет	Нажимает клавишу ENTER Нажимает клавишу ПРОБЕЛ Вводит пустую строку	Null Null Пустая строка
Нет	Да	Нажимает клавишу ENTER Нажимает клавишу ПРОБЕЛ Вводит пустую строку	(не допускается) (не допускается) (не допускается)
Да	Да	Нажимает клавишу ENTER Нажимает клавишу ПРОБЕЛ. Вводит пустую строку	(не допускается) Пустая строка Пустая строка

Индексированное поле. Свойство «Индексированное поле» (Indexed) определяет индекс, создаваемый по одному полю. Индекс ускоряет выполнение запросов, в которых используются индексированные поля, и операции сортировки и группировки. Например, если часто выполняется поиск по полю "Фамилия" в таблице "Сотрудники", следует создать индекс для этого поля. Значение данного свойства можно задать только в окне свойств в ре-

жиме конструктора таблицы. Индекс по одному полю может быть определен путем установки свойства «Индексированное поле». Кроме того, можно выбрать команду «Индексы» в меню «Вид» или нажать кнопку «Индексы» на панели инструментов. Будет открыто окно индексов. После определения индекса по одному полю в окне индексов свойство «Индексированное поле» автоматически примет значение «Да». Для создания индекса по одному полю в программе Visual Basic следует вызвать метод CreateIndex объектов доступа к данным. Свойство «Indexed» используется для ускорения выполнения поиска и сортировки записей по одному полю таблицы. Индексированное поле может содержать как уникальные, так и повторяющиеся значения. Например, в таблице «Сотрудники» можно создать индекс по полю «Код Сотрудника», которое содержит уникальные значения кода или по полю «Фамилия», которое может содержать повторяющиеся значения. Не допускается создание индексов для полей MEMO, гиперссылок и объектов OLE. Допускается создание произвольного количества индексов. Индексы создаются при сохранении макета таблицы и автоматически обновляются при вводе и изменении записей. Пользователь может в любое время добавить новые или удалить ненужные индексы в режиме конструктора таблицы. Если ключ таблицы состоит из одного поля, то Microsoft Access автоматически устанавливает для свойства «Индексированное поле» этого поля значение «Да (Совпадения не допускаются)». Индексы, содержащие несколько полей, следует определять в окне индексов.

2 Индивидуальное задание

I Построить базу данных, основанную на двух таблицах. Пусть одна таблица содержит данные, которые могут отображаться для клиентов при согласовании спецификации изделия, в ней указаны розничные цены на компоненты. Вторая таблица предназначена для анализа результатов деятельности предприятия, в ней содержатся оптовые цены на компоненты и краткая информация о поставщиках (клиенты предприятия не имеют доступа к данным этой таблицы). Необходимо выполнить следующие действия.

1 Запустить программу Microsoft Access 2000 (Пуск > Программы > Microsoft Access).

2 В окне Microsoft Access включить переключатель «Новая база данных» и щелкнуть на кнопке ОК.

3 В окне «Файл» новой базы данных выбрать папку «Мои документы» и дать файлу имя: Комплектующие. Убедитесь, что в качестве типа файла выбраны Базы данных Microsoft Access, и щелкните на кнопке «Создать». Откроется окно новой базы – Комплектующие: база данных.

4 Открыть панель «Таблицы».

5 Дважды щелкнуть на значке «Создание таблицы» в режиме конструктора – откроется бланк создания структуры таблицы, приведенный на рисунке 1.

6 Для первой таблицы ввести следующие поля:

Имя поля	Тип поля
Компонент	Текстовый
Модель	Текстовый
Основной параметр	Числовой
Цена	Числовой

7 Щелкнуть на поле «Цена». В нижней части бланка задать свойство «Число десятичных знаков», равное 2.

8 Для связи с будущей таблицей поставщиков надо задать ключевое поле. Поскольку здесь ни одно поле явно не претендует на «уникальность», используем комбинацию полей «Компонент» и «Модель». Для этого следует выделить оба поля в верхней части бланка (при нажатой клавише SHIFT). Щелчком правой кнопки мыши откройте контекстное меню и выберите в нем пункт «Ключевое поле».

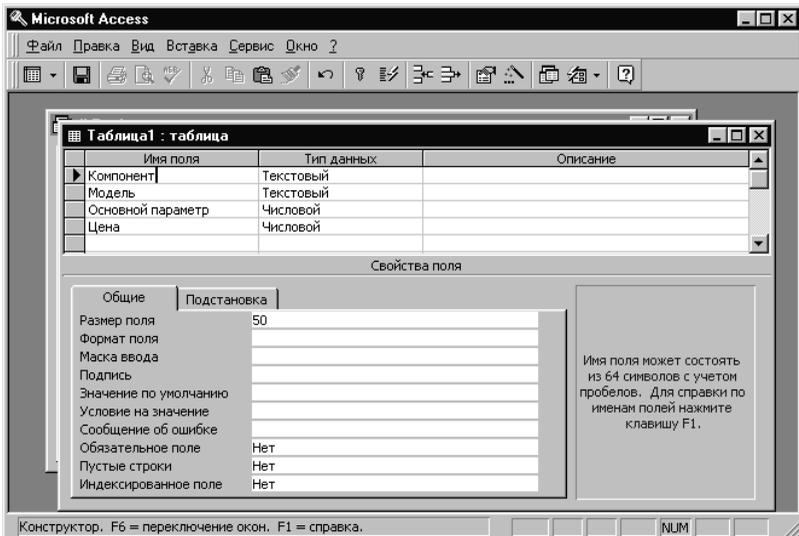


Рисунок 1 – Таблица в режиме Конструктора

9 Закройте окно Конструктора. При закрытии окна дайте таблице имя «Комплекующие».

10 Повторив действия пунктов 5–9, создайте таблицу «Поставщики», в которую входят следующие поля:

Имя поля	Тип поля
Компонент	Текстовый
Модель	Текстовый
Цена оптовая	Числовой
Поставщик	Текстовый
Телефон	Текстовый
Адрес	Текстовый
Примечание	Поле МЕМО

Здесь поле номера телефона является текстовым, несмотря на то, что обычно номера телефонов записывают цифрами. Это связано с тем, что они не имеют числового содержания. Номера телефонов не сравнивают по величине, не вычитают из одного номера другой и т. д. Это типичное текстовое поле. Ключевое поле можно не задавать, для текущей задачи оно не требуется.

11 В окне «Комплекующие»: в базе данных открыть по очереди созданные таблицы и наполнить их содержанием (3-4 записи).

Далее рассмотрим порядок создания межтабличных связей.

1 В окне Microsoft Access включить переключатель. Открыть базу данных, выбрать ранее созданную базу «Комплекующие» и щелкнуть на кнопке ОК.

2 В окне «Комплекующие»: в базе данных открыть панель «Таблицы». Убедитесь, что на ней присутствуют значки ранее созданных таблиц «Комплекующие» и «Поставщики».

3 Разыщите на панели инструментов кнопку «Схема данных». Если есть сложности, найдите команду строки меню: Сервис > Схема данных. Воспользуйтесь любым из этих средств, чтобы открыть окно «Схема данных». Одновременно с открытием этого окна открывается диалоговое окно «Добавление таблицы», на вкладке «Таблицы» которого можно выбрать таблицы, между которыми создаются связи.

4 Щелчком на кнопке «Добавить» выберите таблицы «Комплекующие» и «Поставщики», в окне «Схема данных» откроются списки полей этих таблиц.

5 При нажатой клавише SHIFT выделите в таблице «Комплекующие» два поля – «Компонент» и «Модель».

6 Перетащите эти поля на список полей таблицы «Поставщики». При отпускании кнопки мыши автоматически откроется диалоговое окно «Изменение связей».

7 На правой панели окна «Изменение связей» выберите поля «Компонент» и «Модель» таблицы «Поставщики», включаемые в связь. Не устанавливайте флажок «Обеспечение целостности данных», в данном случае это не требуется, но может препятствовать работе с таблицами.

8 Закройте диалоговое окно «Изменение связей» и в окне «Схема данных» рассмотрите образовавшуюся связь. Убедитесь в том, что линию связи можно выделить щелчком левой кнопки мыши, а щелчком правой кнопки мыши открывается контекстное меню, позволяющее разорвать связь или отредактировать ее.

9 Закройте окно «Схема данных». Закройте программу Microsoft Access.

Модифицировать структуру созданных таблиц, добавив поля всех остальных типов, например – «дата регистрации» (поле типа Дата), «налоговые льготы» (Логический тип), «электронная почта» (типа Гиперссылка) и т.д.

По заданию преподавателя в окне конструктора заполнить строки свойств поля «Формат», «Маска ввода», «Значение по умолчанию», «Подпись» и т. д.

3 Порядок выполнения работы

1 Изучить краткие сведения из теории.

2 Ознакомиться с индивидуальным заданием.

3 Запустить Microsoft Access и создать свою базу, состоящую из нескольких таблиц, согласно индивидуальному заданию. Имя таблицы должно содержать группу и фамилию студента.

4 Заполнить базу данных отладочной информацией (не менее десяти записей в каждой таблице).

5 Оформить отчет по лабораторной работе.

Содержание отчета

Наименование и цель работы; основные положения кратких сведений из теории; индивидуальное задание; распечатки экранных форм; выводы по работе.

Лабораторная работа № 2

ЗАПРОС НА ВЫБОРКУ

Цель работы: изучение методов создания запроса на выборку, основанного на связанных таблицах, формирования структуры запроса путем выбора базовых таблиц и полей, а также формулирования условий отбора.

1 Краткие сведения из теории

Объекты «Таблица» и «Запрос» в Microsoft Access являются основополагающими, т. к. часто запросы могут использоваться вместо таблиц. Например, форма или отчет могут основываться как на таблице, так и на запросе. Таблицы и запросы содержат множество полей и записей. Оба эти объекта могут предоставлять исходную информацию, необходимую для формы или отчета.

Типы запросов, создаваемых в Microsoft Access.

1 Запросы на выборку. Наиболее часто используемым запросом является запрос на выборку. Запрос на выборку возвращает данные из одной или нескольких таблиц, а также результаты, которые при желании пользователь может изменить (с некоторыми ограничениями). Также можно использовать запрос на выборку, чтобы сгруппировать записи для вычисления сумм, средних значений, пересчета и других действий

2 Запросы с параметрами. Это запросы, при выполнении которых в диалоговом окне пользователю выдается приглашение ввести данные, например условие для возвращения записей или значение, которое должно содержаться в поле. Можно создать запрос, в результате которого выводится приглашение на ввод нескольких данных, например, двух дат. В результате будут возвращены все записи, находящиеся между указанными двумя датами. Также запросы с параметрами удобно использовать в качестве основы для форм и отчетов. Например, на основе запроса с параметрами можно создать месячный отчет о доходах. При выводе данного отчета, на экране появится приглашение ввести месяц, доходы которого интересуют пользователя. После ввода месяца на экране будет представлен требуемый отчет. Для получения дополнительных сведений о запросах с параметрами нажмите кнопку. Можно создать специальную форму или диалоговое окно, которое вместо диалогового окна запроса с параметрами будет выводить приглашение на ввод параметров запроса.

3 Перекрестные запросы. В перекрестном запросе отображаются результаты статистических расчетов (такие как суммы, количество записей и

средние значения), выполненных по данным из одного поля таблицы. Эти результаты группируются по двум наборам данных, один из которых расположен в левом столбце таблицы, а второй – в верхней строке. Существует возможность вывести данные в перекрестной таблице без создания в базе данных отдельного запроса. Для этого следует использовать мастер сводных таблиц. В сводной таблице пользователь имеет возможность изменять заголовки строк или столбцов, что позволяет анализировать данные различными способами.

4 Запросы на изменение. В этом типе запросов за одну операцию вносятся изменения в несколько записей. Существует четыре типа запросов на изменение: на удаление, обновление и добавление записей, а также на создание таблицы.

- *Запрос на удаление.* Удаляет группу записей из одной или нескольких таблиц. Например, запрос на удаление позволяет удалить записи о товарах, поставки которых прекращены или на которые нет заказов. С помощью запроса на удаление можно удалять только всю запись, а не отдельные поля внутри нее.

- *Запрос на обновление записей.* Вносит общие изменения в группу записей одной или нескольких таблиц. Например, на 10 % поднимаются цены на все молочные продукты или на 5 % увеличивается зарплата сотрудников определенной категории. Запрос на обновление записей позволяет изменять данные в существующих таблицах.

- *Запрос на добавление.* Добавляет группу записей из одной или нескольких таблиц в конец одной или нескольких таблиц. Например, появилось несколько новых клиентов, а также база данных, содержащая сведения о них. Чтобы не вводить все данные вручную, их можно добавить в таблицу «Клиенты». Запрос на добавление также полезен при добавлении:

- а) полей на основе условий отбора. Например, необходимо добавить имена и адреса клиентов с очень крупными заказами;

- б) записей, если некоторые поля из одной таблицы не существуют в другой. Например, таблица «Клиенты» содержит 11 полей. Пусть требуется добавить записи из другой таблицы с полями, соответствующими 9 из 11 полям таблицы «Клиенты». После выполнения запроса на добавление данные будут добавлены в совпадающие поля, а остальные поля останутся без изменения.

- *Запрос на создание таблиц.* Создает новую таблицу на основе всех или части данных из одной или нескольких таблиц. Запрос на создание таблицы полезен для выполнения следующих действий:

- а) создания таблицы для экспорта в другую базу данных Microsoft Access. Например, требуется создать таблицу, содержащую несколько полей из таб-

лицы «Сотрудники», а затем экспортировать эту таблицу в базу данных, используемую отделом кадров;

б) создания отчетов, содержащих данные, соответствующие определенному моменту времени. Например, 15 мая 2001 года необходимо напечатать отчет об объеме продаж, сделанных в первом квартале, основанный на данных, содержащихся в базовой таблице на 9:00 А.М. 1 апреля 2001. Отчет, основанный на запросе или инструкции SQL, выбирает из таблиц самые последние данные (данные на 15 мая 2001), а не записи на указанный момент времени. Чтобы получить данные на 9:00 А.М. 1 апреля 2001, необходимо разработать запрос на создание таблицы, в котором требуемые записи отбираются в зависимости от указанного момента времени и помещаются в новую таблицу. Затем в качестве основы для отчета следует использовать эту таблицу, а не запрос;

в) создания резервной копии таблицы;

г) создания архивной таблицы, содержащей старые записи. Например, можно создать таблицу, сохраняющую все старые заказы, прежде чем удалить их из текущей таблицы «Заказы»;

д) повышения быстродействия форм и отчетов, базирующихся на многотабличных запросах или инструкциях SQL. Например, требуется вывести на печать несколько отчетов, базирующихся на запросе, включающем пять таблиц, в котором рассчитываются общие итоги. Чтобы ускорить процесс, необходимо разработать запрос на создание таблицы, извлекающий нужные записи и сохраняющий их в одной таблице. Затем на базе этой таблицы создать отчет или указать ее в инструкции SQL как источник записей для формы или отчета. Это позволит обойтись без повторных запусков запроса для каждого отчета. Однако следует помнить, что после выполнения запроса на создание таблицы, данные в этой таблице не изменяются.

5 Запросы SQL создаются при помощи инструкций SQL (структурированный язык запросов). Примерами запросов SQL могут служить запросы на объединение, запросы к серверу, перекрестные и подчиненные запросы.

В режиме Конструктора запрос позволяет ввести условия отбора выводимых на экран полей и записей таблицы, и установить порядок их отображения.

Перед тем, как изучить процесс создания и работы с запросом, познакомимся с диалоговым окном «Запрос на выборку» в режиме конструктора, внешний вид которого представлен на рисунке 1.

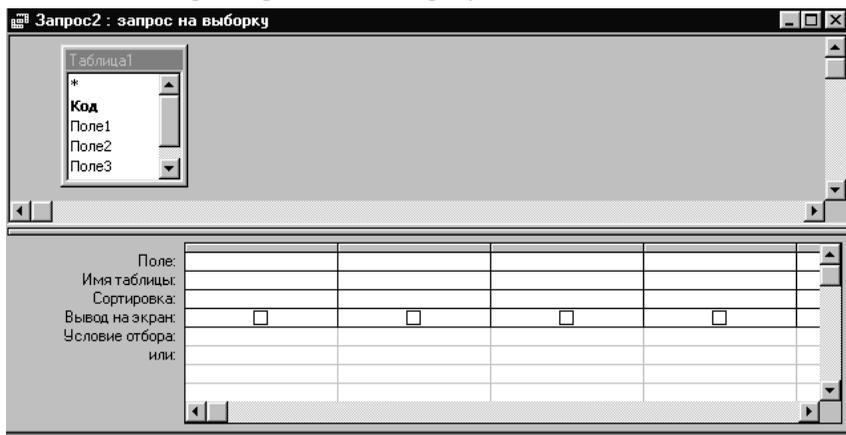


Рисунок 1 – Окно запроса в режиме конструктора

Панель, расположенная в верхней части окна, содержит список всех полей из таблицы. Нижняя половина окна запроса включает в себя область, называемую бланк запроса.

В первой строке определяется набор полей, включаемый в динамический набор.

Создадим запрос на выборку жестких дисков, имеющих емкость не менее 8 Гбайт при цене менее 150 у. е. Результирующая таблица должна содержать также адрес поставщика и номер его телефона.

1 Запустите программу Microsoft Access 2000 (Пуск ^ Программы ^ Microsoft Access).

2 В окне Microsoft Access включите переключатель «Открыть базу данных», выберите ранее созданную базу «Комплектующие» и щелкните на кнопке ОК.

3 В окне «Комплектующие» (база данных) откройте панель «Запросы». Дважды щелкните на значке «Создание запроса» в режиме Конструктора, откроется бланк запроса по образцу. Одновременно с ним откроется диалоговое окно «Добавление таблицы».

4 В окне «Добавление таблицы» выберите таблицу «Поставщики» и щелкните на кнопке «Добавить». Закройте окно «Добавление таблицы».

5 В списке полей таблицы «Поставщики» выберите поля, включаемые в результирующую таблицу: «Компонент», «Модель», «Цена оптовая», «По-

ставщик», «Телефон». Выбор производите двойными щелчками на именах полей.

6 Задайте условие отбора для поля «Компонент». В соответствующую строку введите: «Жесткий диск» Из таблицы будут выбираться не все изделия, а только жесткие диски.

7 Задайте условие отбора для поля «Цена оптовая». В соответствующую строку введите: < 150 . Из таблицы будут выбираться только изделия, имеющие цену менее 150 у. е.

8 Надо задать условие отбора по основному потребительскому параметру емкости жесткого диска. Однако в таблице «Поставщики» такого поля нет. С другой стороны, в ней есть поле «Модель», которое однозначно определяет параметры изделия. Благодаря тому, что по полю «Модель» у нас установлена связь с таблицей «Комплектующие», мы получаем возможность ввести в запрос поле «Основной параметр», взяв его из другой таблицы.

Добавьте список полей таблицы «Комплектующие» в верхнюю часть бланка запроса по образцу. Для этого щелкните правой кнопкой мыши в верхней области бланка, в открывшемся контекстном меню выберите пункт «Добавить таблицу» – откроется уже знакомое нам окно Добавление таблицы. Выберите в нем таблицу «Комплектующие».

9 Двойным щелчком на поле «Основной параметр» в списке полей таблицы «Комплектующие» введите это поле в бланк запроса по образцу.

10 В строке «Условие отбора» столбца «Основной параметр» введите условие >8 (емкость диска более восьми гигабайт). Внешний вид диалогового окна «Запрос по образцу» представлен на рисунке 2.

11 Закройте бланк запроса по образцу. При закрытии запроса введите его имя – Выбор комплекующих.

12. В окне «Комплектующие»: база данных откройте только что созданный запрос и рассмотрите результирующую таблицу. Ее содержательность зависит от того, что было введено в таблицы «Комплектующие» и «Поставщики» при их наполнении в лабораторной работе № 1. Если ни одно изделие не соответствует условию отбора, и получившаяся результирующая таблица не имеет данных, откройте базовые таблицы и наполните их модельными данными, позволяющими проверить работу запроса.

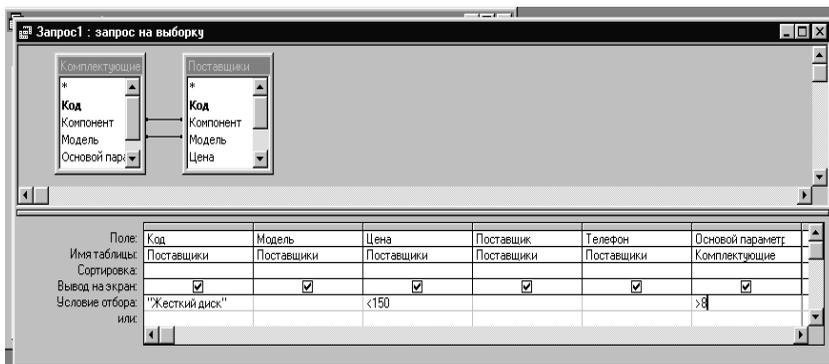


Рисунок 2 – Пример создания бланка запроса по образцу

2 Индивидуальное задание

Создать запрос на выборку, основанный на таблицах, созданных на предыдущей лабораторной работе. В строке «Условие отбора» ввести условие по заданию преподавателя

3 Порядок выполнения работы

- 1 Изучить краткие сведения из теории.
- 2 Ознакомиться с индивидуальным заданием.
- 3 Запустить Microsoft Access, и создать запрос согласно индивидуальному заданию. Имя запроса должно содержать фамилию студента.
- 4 Оформить отчет по лабораторной работе.

Содержание отчета

Наименование и цель работы; основные положения кратких сведений из теории; индивидуальное задание; распечатки экранных форм, выводы по работе.

Лабораторная работа № 3

СОЗДАНИЕ ЗАПРОСОВ «С ПАРАМЕТРОМ»

Цель работы: приобретение навыков создания запроса «с параметром».

1 Краткие сведения из теории

Специальный тип запросов, называемый запросами «с параметром», позволяет пользователю самому ввести критерий отбора данных на этапе запуска запроса. Этим приемом обеспечивается гибкость работы с базой. Пользователь базы данных работает с запросами, которые ему подготовил разработчик. Если, например, разработчик предусмотрел запрос, отбирающий изделия, имеющие цену менее 150 у. е., то пользователь базы уже не в состоянии отобрать изделия, цена которых менее 140 у. е., поскольку у него нет соответствующего запроса.

Создадим простой запрос, позволяющий отбирать процессоры, предельную цену которых пользователь может задать сам при запуске запроса.

1 Запустите программу Microsoft Access 2000 (Пуск > Программы > Microsoft Access).

2 В окне Microsoft Access включите переключатель «Открыть базу данных», выберите ранее созданную базу «Комплектующие» и щелкните на кнопке ОК.

3 В окне «Комплектующие»: база данных откройте панель «Запросы». Дважды щелкните на значке «Создание запроса» в режиме Конструктора – откроется бланк запроса по образцу.

4 Согласно предыдущей лабораторной работе, создайте запрос на выборку, основанный на таблице «Поставщики», в который войдут следующие поля:

- компонент;
- модель;
- цена оптовая;
- поставщик;
- телефон.

5 В строке «Условие отбора» поля «Компонент» введите: Процессор.

6 Строку «Условие отбора» для поля «Цена оптовая» надо заполнить таким образом, чтобы при запуске запроса пользователь получал предложение ввести нужное значение (рисунок 1).

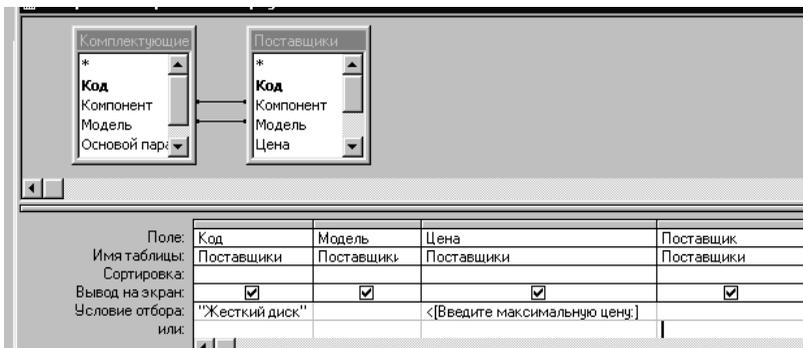


Рисунок 1 – Ввод условия отбора

Текст, обращенный к пользователю, должен быть заключен в квадратные скобки. Если бы мы хотели отобрать процессоры, цена которых больше 100 у. е., мы бы написали: >100. Если бы нам были нужны процессоры дешевле 80 у. е., мы бы написали <80. Но если мы хотим дать пользователю возможность выбора, мы должны написать: <[Введите максимальную цену:].

7 Закройте запрос. При закрытии сохраните его под именем «Выбор комплектующих».

8 В окне «Комплектующие»: база данных, откройте панель «Запросы» и запустите запрос «Выбор комплектующих», на экране появится диалоговое окно «Введите значение параметра», изображенное на рисунке 2.

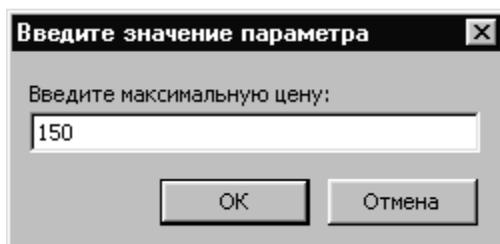


Рисунок 2 – Диалоговое окно «Введите значение параметра» для условия отбора

9 Введите какое-либо число и щелкните на кнопке ОК. В зависимости от того, что реально содержится в таблице «Поставщики», по результатам запроса будет сформирована результирующая таблица.

10 Закройте все объекты базы данных. Закройте программу Microsoft Access.

2 Индивидуальное задание

Создать запрос «с параметром», основанный на таблицах, созданных на предыдущих лабораторных работах. По заданию преподавателя создайте диалоговое окно для ввода значения параметра.

3 Порядок выполнения работы

- 1 Изучить краткие сведения из теории.
- 2 Ознакомиться с индивидуальным заданием.
- 3 Запустить Microsoft Access, и создать запрос согласно индивидуальному заданию.
- 4 Оформить отчет по лабораторной работе.

Содержание отчета

Наименование и цель работы; основные положения кратких сведений из теории; индивидуальное задание; распечатки экранных форм, выводы по работе.

Лабораторная работа № 4

СОЗДАНИЕ ИТОГОВОГО ЗАПРОСА

Цель работы: приобретение навыков создания итогового запроса.

1 Краткие сведения из теории

Запросы, выполняющие вычисления по всем записям для какого-либо числового поля, называются итоговыми запросами. В итоговом запросе может рассчитываться сумма значений или величина среднего значения по всем ячейкам поля, может выбираться максимальное или минимальное значение данных в поле, может также исполняться иная итоговая функция. Ито-

говые запросы, как и запросы на выборку, готовятся с помощью бланка запроса по образцу.

Предположим, что малое предприятие собирает компьютеры трех классов: «Элитный», «Деловой» и «Экономичный». Несмотря на то, что архитектура у всех компьютеров близка, их компоненты заметно отличаются по цене и техническим параметрам. Соответственно имеются значительные отличия в цене этих трех моделей, что важно для захвата разных секторов рынка. Наша задача – подготовить итоговый отчет, с помощью которого можно определять цену каждой из модели компьютеров и динамично ее отслеживать при изменении входящих компонентов или их поставщиков.

1 Запустите программу Microsoft Access 2000 (Пуск > Программы > Microsoft Access).

2 В окне Microsoft Access включите переключатель «Открыть базу данных», выберите ранее созданную базу «Комплекующие» и щелкните на кнопке ОК.

3 В окне «Комплекующие»: база данных откройте панель «Таблицы». Выберите таблицу «Комплекующие».

4 Щелчком на значке «Конструктор» откройте таблицу в режиме проектирования – нам это необходимо для создания дополнительного поля «Класс», в котором будут храниться данные о том, для какого класса изделий предназначены компоненты.

5 В начало структуры таблицы вставьте новое поле. Для этого выделите первое поле (Компонент) и нажмите клавишу INSERT.

6 Введите имя нового поля – Класс и определите его тип – Текстовый.

7 Закройте окно «Конструктора». При закрытии подтвердите необходимость изменить структуру таблицы.

8 Откройте таблицу «Комплекующие» и наполните ее содержанием, введя для каждого класса данные по следующим изделиям:

- материнская плата;
- процессор;
- оперативная память;
- жесткий диск;
- корпус;
- дисковод CD-ROM;
- дисковод гибких дисков;
- видеоадаптер;
- звуковая карта;
- клавиатура;
- мышь.

Цены на эти изделия для каждого класса проставьте произвольно. Прочие поля таблицы можно не заполнять, в формировании итогового запроса они участвовать не будут.

9 Закройте таблицу «Комплектующие».

10 Откройте панель «Запросы» щелчком на одноименной кнопке окна «Комплектующие».

11 Выполните двойной щелчок на значке «Создание запроса» в режиме конструктора. В открывшемся диалоговом окне «Добавление таблицы» выберите таблицу «Комплектующие», на основе которой будет разрабатываться итоговый запрос. Закройте окно «Добавление таблицы».

12 В бланк запроса по образцу введите следующие поля таблицы «Комплектующие»: «Класс», «Компонент», «Цена».

13 Для поля «Класс» включите сортировку по возрастанию. Включите также сортировку по полю «Цена», но на этот раз – по убыванию.

14 На панели инструментов Microsoft Access щелкните на кнопке «Групповые операции» или воспользуйтесь строкой меню (Вид > Групповые операции). Эта команда необходима для создания в нижней части бланка строки «Групповые операции». Именно на ее базе и создаются итоговые вычисления. Все поля, отобранные для запроса, получают в этой строке значение «Группировка».

15 Для поля, по которому производится группировка записей (в нашем случае – Класс), оставьте в строке «Групповые операции» значение «Группировка». Для остальных полей щелкните в этой строке – появится кнопка раскрывающегося списка, из которого можно выбрать итоговую функцию для расчета значений в данном поле.

16 Для поля «Цена» выберите итоговую функцию Sum для определения стоимости изделия как суммы стоимостей комплектующих.

17 Для поля «Компонент» выберите итоговую функцию Count, определяющую общее количество записей, вошедших в группу. В нашем случае это количество узлов, из которых собран компьютер.

18 Закройте бланк запроса по образцу и дайте ему имя: Расчет стоимости изделия. Запустите запрос и убедитесь, что он правильно работает.

19 Закройте все объекты базы данных. Завершите работу с программой Microsoft Access.

2 Индивидуальное задание

Создать итоговый запрос, основанный на таблицах, созданных на предыдущих лабораторных работах. По заданию преподавателя в строке «Групповые операции» выберите из списка одну из функций и создайте запрос.

3 Порядок выполнения работы

- 1 Изучить краткие сведения из теории.
- 2 Ознакомиться с индивидуальным заданием.
- 3 Запустить Microsoft Access и создать запрос согласно индивидуальному заданию.
- 4 Оформить отчет по лабораторной работе.

Содержание отчета

Наименование и цель работы; основные положения кратких сведений из теории; индивидуальное задание; распечатки экранных форм, выводы по работе.

Лабораторная работа № 5

ПОЛЬЗОВАТЕЛЬСКИЕ ФОРМЫ

Цель работы: изучение методов создания пользовательских форм.

1 Краткие сведения из теории

Если запросы – это специальные средства для отбора и анализа данных, то формы – это средства для ввода данных. С помощью форм данные можно не только вводить, но и отображать. Запросы тоже отображают данные, но делают это в виде результирующей таблицы, не имеющей почти никаких средств оформления. При выводе данных с помощью форм можно применять специальные средства оформления.

Пользователь имеет возможность создать форму самостоятельно или воспользоваться мастером. "Мастер форм" ускоряет процесс создания формы, так как автоматически выполняет всю основную работу. При использовании мастера Microsoft Access выдает приглашение на ввод данных, на основе которых создается форма. В созданных формах мастер можно использовать для быстрого создания элементов управления в форме. Для настройки формы следует переключиться в режим конструктора.

Кроме того, форму можно создать с помощью кнопки «Новый объект», без помощи мастера, на основе нескольких таблиц, из записей, отфильтрованных в таблице, запросе или другой форме.

Рассмотрим создание формы с помощью мастера.

1 В окне базы данных выберите вкладку «Формы».

2 Нажмите кнопку «Создать».

3 В диалоговом окне «Новая форма» выберите нужного мастера. Описание мастера появляется в левой части диалогового окна.

4 Выберите имя таблицы или запроса, содержащих данные, на основе которых будет создана форма. При использовании мастера форм источник данных для формы следует указывать в диалоговом окне мастера.

5 Нажмите кнопку ОК.

6 Если на шаге 3 были выбраны «Мастер форм», «Диаграмма» или «Сводная таблица», то при создании формы следуйте инструкциям, выводимым в диалоговых окнах соответствующего мастера.

При выборе элементов «Автоформа»: в столбец, «Автоформа»: ленточная или «Автоформа»: табличная форма создается автоматически. Для просмотра записей с небольшим количеством полей удобно использовать ленточную форму. В отличие от простой формы, которая показывает только одну запись, в ленточной форме отформатированные записи выводятся на экран одна за другой, как в таблице. Изменить созданную форму можно в режиме конструктора.

Структура формы. Форма имеет три основных раздела: область заголовка, область данных и область примечания. Линии, разделяющие разделы, перетаскиваются по вертикали с помощью мыши, это позволяет изменять размеры разделов так, как требуется (рисунок 1).



Рисунок 1 – Структура формы, созданной мастером

Разделы заголовка и примечания имеют чисто оформительское назначение, их содержимое напрямую не связано с таблицей или запросом, на кото-

ром основана форма. Раздел данных имеет содержательное значение, в нем представлены элементы управления, с помощью которых выполняется отображение данных или их ввод. Разработчик формы может разместить здесь дополнительные элементы управления для автоматизации ввода данных: переключатели, флажки, списки и другие, типичные для приложений Windows элементы управления (рисунок 2).

Все сведения в форме или отчете содержатся в элементах управления. Элементы управления – это объекты формы или отчета, которые служат для вывода данных на экран, выполнения макрокоманд или оформления формы или отчета. Например, поле можно использовать для вывода данных на экран в форме или отчете, кнопку – для открытия другой формы или отчета, а линию или прямоугольник – для разделения и группировки элементов управления с тем, чтобы они лучше воспринимались пользователем.

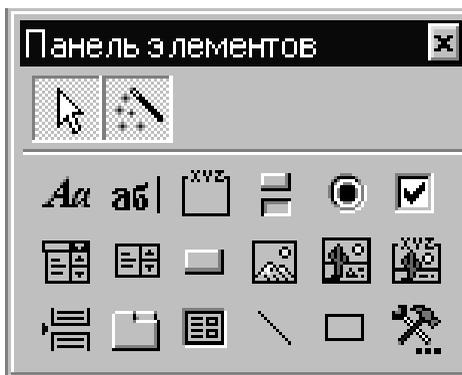


Рисунок 2 – Элементы управления формы

В Microsoft Access существуют следующие типы элементов управления, которые содержатся на панели элементов в режиме конструктора формы или режиме конструктора запроса: поле, надпись, группа, переключатель, флажок, выключатель, поле со списком, список, кнопка, рисунок, присоединенная рамка объекта, свободная рамка объекта, набор вкладок, подчиненная форма/отчет, разрыв страницы, линия, прямоугольник и дополнительные элементы ActiveX. Элементы управления могут быть связанными, свободными или вычисляемыми. *Связанный* элемент управления присоединен к полю базовой таблицы или запроса. Такие элементы управления используются для отображения, ввода или обновления значений из полей базы данных. Для *вычисляемого* элемента управления в качестве источника данных используется выражение. В выражении могут быть использованы данные из поля базовой таблицы или запроса для формы или отчета, а также данные другого элемента управления формы или отчета. Для свободного элемента управления источника данных не существует. *Свободные* элементы управле-

ния используются для вывода на экран данных, линий, прямоугольников и рисунков.

Надписи предназначены для отображения в форме или отчете описательных текстов, таких как заголовки, подписи или краткие инструкции. В надписях не выводятся значения полей или выражений; они всегда являются свободными и не меняются при переходе от записи к записи. Надпись может быть присоединена к другому элементу управления (такую надпись называют подписью). Например, поле создается с присоединенной надписью, которая содержит подпись этого поля. Эта надпись появляется как заголовок столбца в форме в режиме таблицы.

Надпись, созданная с помощью инструмента «Надпись», размещается отдельно и не присоединяется ни к какому элементу управления. Такие надписи используются для отображения разных сведений, например, заголовков формы или отчета, а также для вывода поясняющего текста. Надписи, не присоединенные к элементам управления, не отображаются в режиме таблицы.

Поля используются в форме или отчете для отображения данных из таблицы, запроса или инструкции SQL. Поле такого типа называют присоединенным, потому что оно связано с данными в поле в источнике данных. Кроме того, существуют свободные поля. Например, можно создать свободное поле для отображения результатов вычислений или для приема данных, вводимых пользователем. Содержимое свободного поля нигде не сохраняется.

Группа используется в форме или отчете для вывода ограниченного набора параметров. Группа делает выбор параметра простым и наглядным. В каждый момент времени в группе может быть выбран только один параметр. Группа состоит из рамки группы и набора флажков, переключателей или выключателей. При присоединении группы к полю к нему присоединяется только рамка группы, а не находящиеся в ней флажки, выключатели или переключатели. Пользователь не должен определять свойство «Данные» (ControlSource) для каждого элемента управления в группе. Вместо этого следует задать в свойстве «Значение параметра» (OptionValue) каждого флажка, выключателя или переключателя число, являющееся допустимым для поля, к которому присоединена рамка группы. При выборе параметра в группе Microsoft Access вводит в поле значение, равное значению свойства "Значение параметра" (OptionValue) выбранного элемента. В свойстве «Значение параметра» (OptionValue) требуется задавать число, так как значением группы может быть только числовое, а не текстовое значение. Microsoft Access сохраняет это число в базовой таблице. Группа может быть также связана с выражением или быть свободной. Свободные группы применяются

в специальных диалоговых окнах для принятия данных, вводимых пользователем, и для выполнения действий, основанных на этих данных.

Выключатели. В форме или отчете выключатель может быть использован как отдельный элемент управления, в котором отображаются значения логического поля из базовой таблицы, запроса или инструкции SQL. Если кнопка выключателя нажата, поле в таблице имеет значение «Да»; если кнопка выключателя не нажата, поле имеет значение «Нет».

Когда пользователь нажимает кнопку выключателя, присоединенного к логическому полю, Microsoft Access отображает значение в базовой таблице в формате, который определяется значением свойства поля «Формат поля» (Format) («Да»/«Нет», «Истина»/«Ложь» или «Вкл»/«Выкл»). Выключатели особенно удобны при использовании в группах. В такой группе легко видеть, какой из выключателей нажат. Вместо подписи на выключатель можно поместить рисунок. Свободные выключатели используются также в специальных диалоговых окнах для приема данных, вводимых пользователем.

Переключатели. Флажки. Свойства аналогичны выключателям. Кроме того, флажки включаются в группу для отображения набора выбираемых значений.

Поля со списком. Во многих случаях удобнее выбрать значение из списка, чем вводить конкретное значение с клавиатуры по памяти. Поле со списком позволяет выбрать любой из этих способов ввода значения, не требуя при этом значительного места в форме. Поле со списком является комбинацией двух элементов: поля и раскрывающегося списка. Значение, выбранное или введенное в присоединенное поле со списком, вставляется в поле, к которому присоединено поле со списком. В поле со списком список состоит из строк с данными. Строки содержат один или несколько столбцов, с заголовками или без заголовков. Если поле со списком, содержащим несколько столбцов, является присоединенным, то сохраняется значение одного из столбцов. Свободное поле со списком позволяет сохранять значение, используемое в другом элементе управления. Например, с помощью свободного поля со списком можно ограничить значения, отбираемые в другом поле со списком или в специальном диалоговом окне. Свободное поле применяется также для поиска записи с помощью значения, выбранного или введенного в поле со списком. Поля со списком имеют свойство "Ограничиться списком" (LimitToList), которое определяет, допускается ли ввод в поле любых значений или только совпадающих с одним из значений списка. Если в форме достаточно свободного места и требуется, чтобы список постоянно находился на экране, а также, если требуется ограничить вводимые данные имеющимся списком, вместо поля со списком можно использовать список.

Списки. Во многих случаях удобнее выбрать значение из списка, чем вводить конкретное значение по памяти. Кроме того, выбор из списка позволяет быть уверенным, что введенное значение является допустимым. Список состоит из строк с данными. Строки содержат один или несколько столбцов, которые могут быть снабжены заголовками. Если список из нескольких столбцов является присоединенным, то сохраняется значение одного из столбцов. Свободный список позволяет хранить значение, используемое в другом элементе управления. Например, с помощью свободного списка можно ограничить значения, отбираемые в другом списке или в специальном диалоговом окне. «Свободный список» применяется также для поиска записи с помощью значения, выбранного в списке. В тех случаях, когда в форме недостаточно места для отображения списка, или если наряду с выбором из списка требуется вводить новые значения с клавиатуры, вместо списка следует использовать поле со списком.

Кнопки используются в формах для выполнения определенного действия или ряда действий. Например, можно создать в форме кнопку, открывающую другую форму. Чтобы кнопка выполняла какое-либо действие, следует создать макрос или процедуру обработки события и связать их со свойством кнопки "Нажатие кнопки" (OnClick). Мастер кнопок позволяет создавать кнопки более 30 разных типов. При создании кнопки с помощью мастера для нее определяется процедура обработки события.

Текст надписи на кнопке задается в качестве значения свойства "Подпись" (Caption). Чтобы поместить на кнопку рисунок, следует указать его в свойстве кнопки "Рисунок" (Picture).

Подчиненная форма – это форма, находящаяся внутри другой формы. Первичная форма называется главной формой, а форма внутри формы называется подчиненной формой. Комбинацию «форма/подчиненная форма» часто называют также иерархической формой или комбинацией «родительской» и «дочерней» форм. Подчиненная форма удобна для вывода данных из таблиц или запросов, связанных с отношением «один-ко-многим». Например, можно создать форму с подчиненной формой для вывода данных из таблицы «Типы» и из таблицы «Товары». Данные в таблице «Типы» находятся на стороне «один» отношения. Данные в таблице «Товары» находятся на стороне «многие» отношения, каждый тип может иметь несколько товаров. Главная форма и подчиненная форма в этом типе форм связаны таким образом, что в подчиненной форме выводятся только те записи, которые связаны с текущей записью в главной форме. При использовании формы с подчиненной формой для ввода новых записей текущая запись в главной форме сохраняется при входе в подчиненную форму. Это гарантирует, что записи из таблицы на стороне «многие» будут иметь связанную запись в

таблице на стороне «один». Это также автоматически сохраняет каждую запись, добавляемую в подчиненную форму. Подчиненная форма может быть выведена в режиме таблицы или она может быть выведена как простая или ленточная форма. Главная форма может быть выведена только как простая форма.

Главная форма может содержать любое число подчиненных форм, если каждая подчиненная форма помещается в главную форму. Имеется также возможность создавать подчиненные формы двух уровней вложенности. Это означает, что можно иметь подчиненную форму внутри главной формы, а другую подчиненную форму внутри этой подчиненной формы. Например, можно иметь главную форму, в которой выводятся данные о клиентах, подчиненную форму, выводящую данные о заказах и другую подчиненную форму, которая отображает то, что заказано.

Элементы управления, которыми может пользоваться разработчик, представлены на Панели элементов. Ее открывают щелчком на соответствующей кнопке панели инструментов Microsoft Access или командой Вид > Панель элементов. Выбор элемента управления выполняется одним щелчком на его значке в Панели элементов, после чего следующим щелчком в поле формы отмечается место, куда он должен быть поставлен. Вместе с элементом в поле формы вставляется его присоединенная надпись. По умолчанию эта надпись стандартная, например, для переключателей это Переключатель 1, Переключатель 2 и т. д. Редактированием свойства элемента управления (доступ к свойствам открывается через контекстное меню) можно дать элементу управления более содержательную подпись.

Основными элементами оформления формы являются текстовые надписи и рисунки. Для создания в форме текстовых надписей служат два элемента управления – «Надпись» и «Поле». В качестве надписи можно задать произвольный текст. Элемент «Поле» отличается тем, что в нем отображается содержимое одного из полей таблицы, на которой основана форма, то есть при переходе от записи к записи текст может меняться.

Для создания графических элементов оформления служат элементы управления "Рисунок", "Свободная рамка объекта" и "Присоединенная рамка объекта". Рисунок выбирается из графического файла и вставляется в форму. Элемент "Свободная рамка объекта" отличается тем, что это не обязательно рисунок – это может быть любой другой объект OLE, например, мультимедийный. Элемент "Присоединенная рамка объекта" тоже в какой-то степени может служить для оформления формы, но его содержимое берется не из назначенного файла, а непосредственно из таблицы базы данных (если она имеет поле объекта OLE). Естественно, что при переходе между записями содержимое этого элемента будет меняться.

Дизайн формы. В то время как таблицы базы данных скрыты от посторонних глаз, формы базы данных – это средства, с помощью которых с ней общаются пользователи. Поэтому к формам предъявляются повышенные требования по дизайну. В первую очередь, все элементы управления форм должны быть аккуратно выровнены. Это обеспечивается командой **Формат > Выровнить**. Если нужно равномерно распределить элементы управления по полю формы, используют средства меню **Формат > Интервал по горизонтали** или **Формат ^ Интервал по вертикали**.

Ручное изменение размеров и положения элементов управления тоже возможно, но редко приводит к качественным результатам. При работе вручную используют перетаскивание маркеров, которые видны вокруг элемента управления в тот момент, когда он выделен. Особый статус имеет маркер левого верхнего угла. Обычно элементы управления перетаскиваются вместе с присоединенными к ним надписями. Перетаскивание с помощью этого маркера позволяет оторвать присоединенную надпись от элемента.

Существенную помощь при разработке дизайна формы оказывает вспомогательная сетка. Ее отображение включают командой **Вид ^ Сетка**. Автоматическую привязку элементов к узлам сетки включают командой **Формат > Привязать к сетке**.

Управление последовательностью перехода. Пользователь, для которого, собственно, и разрабатывается форма, ожидает, что ввод данных в нее должен происходить по элементам управления слева направо и сверху вниз. Однако при проектировании сложных форм, когда в процессе дизайна элементы управления многократно перемещаются с места на место, очень легко перепутать их последовательность и создать неудобный порядок ввода данных.

Физически последовательность перехода – это порядок перехода к следующему полю по окончании работы с предыдущим, что легко проверяется с помощью клавиши **ТАВ**. Если при последовательных нажатиях этой клавиши фокус ввода «мечется» по всей форме, значит, последовательность перехода нерациональна, и ее надо править.

Для управления последовательностью перехода служит диалоговое окно **Последовательность перехода**. В нем представлен список элементов управления формы. Порядок элементов в списке соответствует текущему порядку перехода. Изменение порядка перехода выполняется в два приема:

- щелчком на кнопке маркера слева от названия выделяется элемент управления (кнопка мыши отпускается);
- после повторного щелчка с перетаскиванием элемент перемещается на новое место.

Закончив разработку макета формы, ее следует закрыть и сохранить под заданным именем. После открытия формы в окне "База данных", с ней можно работать: просматривать или редактировать данные из базовой таблицы. Проверку последовательности перехода выполняют клавишей TAB.

2 Индивидуальное задание

Создать главную и подчиненную форму для просмотра, ввода и редактирования данных в таблицах, созданных на предыдущих лабораторных работах. По заданию преподавателя поместить на форму элементы управления и установить последовательность перехода.

3 Порядок выполнения работы

- 1 Изучить краткие сведения из теории.
- 2 Ознакомиться с индивидуальным заданием.
- 3 Запустить Microsoft Access и создать форму согласно индивидуальному заданию.
- 4 Оформить отчет по лабораторной работе.

Содержание отчета

Наименование и цель работы; основные положения кратких сведений из теории; индивидуальное задание; распечатки экранных форм, выводы по работе.

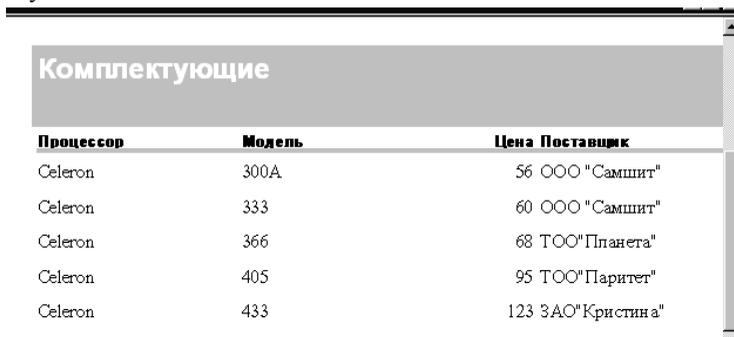
Лабораторная работа № 6

СОЗДАНИЕ ОТЧЕТОВ

Цель работы: изучение методов создания отчетов.

1 Краткие сведения из теории

По своим свойствам и структуре отчеты во многом похожи на формы, но предназначены только для вывода данных, причем для вывода не на экран, а на печатающее устройство (принтер). Пример простейшего отчета приведен на рисунке 1.



Процессор	Модель	Цена	Поставщик
Celeron	300A	56 000	"Самшит"
Celeron	333	60 000	"Самшит"
Celeron	366	68 000	"Планета"
Celeron	405	95 000	"Паритет"
Celeron	433	123 300	"Кристина"

Рисунок 1 – Пример простейшего отчета

Большая часть того, что было сказано о формах, относится и к отчетам. Здесь также существуют средства автоматического, автоматизированного и ручного проектирования. Средства автоматического проектирования реализованы автоотчетами. Кроме автоотчетов «в столбец», существуют «ленточные» автоотчеты. Средством автоматизированного создания отчетов является Мастер отчетов. Он запускается двойным щелчком на значке «Создание отчета» с помощью мастера в окне «База данных». Мастер отчетов работает в шесть этапов. При его работе выполняется выбор базовых таблиц или запросов, на которых отчет базируется, выбор полей, отображаемых в отчете, выбор полей группировки, выбор полей и методов сортировки, выбор формы печатного макета и стиля оформления.

Рассмотрим создание отчета с помощью мастера.

1 В окне базы данных выберите вкладку Отчеты.

2 Нажмите кнопку "Создать".

3 В диалоговом окне Новый отчет выберите нужного мастера. Описание действий, выполняемых мастером, выводится в левой половине диалогового окна.

4 Выберите имя таблицы или запроса, содержащих данные, по которым строится отчет.

Microsoft Access по умолчанию использует эту таблицу или запрос как базовый источник данных для отчета. Однако мастер позволяет изменить источник данных, а также выбрать поля из других таблиц или запросов.

5 Нажмите кнопку ОК.

6 Если на шаге 3 выбран мастер отчетов, мастер диаграмм или мастер наклеек, выполняйте инструкции мастера, выводющиеся в диалоговом окне. Если выбран один из мастеров автоотчетов, отчет создается автоматически.

Если созданный мастером отчет требует внесения изменений, это можно сделать в режиме конструктора отчета.

Структура готового отчета отличается от структуры формы только увеличенным количеством разделов. Кроме разделов заголовка, примечания и данных, отчет может содержать разделы верхнего и нижнего колонтитулов. Если отчет занимает более одной страницы, эти разделы необходимы для печати служебной информации, например, номеров страниц. Чем больше страниц занимает отчет, тем важнее роль данных, выводимых на печать через эти разделы. Если для каких-то полей отчета применена группировка, количество разделов отчета увеличивается, поскольку оформление заголовков групп выполняется в отдельных разделах.

Редактирование структуры отчета выполняют в режиме Конструктора (режим запускается кнопкой «Конструктор» в окне «База данных»). Приемы редактирования те же, что и для форм. Элементы управления в данном случае выполняют функции элементов оформления, поскольку печатный отчет не интерактивный объект, в отличие от электронных форм и Web-страниц. Размещение элементов управления выполняют с помощью Панели элементов (Вид > Панель элементов), которая по составу практически не отличается от Панели элементов формы. Важной особенностью отчетов является наличие средства для вставки в область верхнего или нижнего колонтитула текущего номера страницы и полного количества страниц. Эту операцию выполняют с помощью диалогового окна «Номера страниц» (Вставка ^ Номера страниц).

Как и для форм, для отчетов возможно присоединение элементов управления. Кроме тех, о которых шла речь ранее, для отчетов можно использовать такой элемент управления, как подчиненные отчеты. Подчиненным отчетом называют отчет, вставленный в другой отчет. При комбинировании отчетов один из отчетов является главным. Главный отчет может быть как присоединенным, так и свободным, т. е. не базирующимся на таблице, запросе или инструкции SQL.

Свободный главный отчет может служить контейнером нескольких не связанных между собой отчетов, которые требуется объединить.

Главный отчет связывают с таблицей, запросом или инструкцией SQL в тех случаях, когда в него требуется вставить подчиненные отчеты, в которых выводятся данные, связанные с данными в главном отчете. Например, в главном отчете могут быть выведены все записи о продажах за год, а в подчиненном отчете итоговые данные, например, суммы продаж за каждый

квартал. В главном отчете могут также содержаться данные, являющиеся общими для двух или нескольких подчиненных отчетов. В этом случае области данных выводятся в подчиненных отчетах.

В главный отчет наряду с подчиненными отчетами включают также подчиненные формы, причем число таких подчиненных форм не ограничивается. Более того, главный отчет может содержать подчиненные формы или отчеты двух уровней вложенности. Например, в отчете может содержаться подчиненный отчет, который в свою очередь содержит подчиненную форму или подчиненный отчет.

Гиперссылки также используются в формах и таблицах для перехода к объектам той же самой или другой базы данных Microsoft Access, к документам, созданным в Microsoft Word, Microsoft Excel и Microsoft PowerPoint, а также к документам, расположенным в глобальной сети Интернет или в местной корпоративной сети. Гиперссылки можно также добавлять в отчеты. Однако при просмотре отчета в Microsoft Access гиперссылки работать не будут. Для их использования отчет необходимо вывести в формате Word, Microsoft Excel или HTML.

2 Индивидуальное задание

Создать главный и подчиненный отчеты на основе данных таблиц или запросов, созданных на предыдущих лабораторных работах. По заданию преподавателя присоединить элементы управления.

3 Порядок выполнения работы

- 1 Изучить краткие сведения из теории.
- 2 Ознакомиться с индивидуальным заданием.
- 3 Запустить Microsoft Access, и создать отчеты согласно индивидуальному заданию.
- 4 Оформить отчет по лабораторной работе.

Содержание отчета

Наименование и цель работы; основные положения кратких сведений из теории; индивидуальное задание; распечатки экранных форм, выводы по работе.

СПИСОК ЛИТЕРАТУРЫ

- 1 *Сигел Ч.* Изучи сам Access 97/ Пер. с англ. – Мн.: Попурри, 1998. – 339 с.
- 2 *Вейкас Д.* Эффективная работа с Microsoft Access. – СПб.: Питер Пресс, 1995. – 855 с.

СОДЕРЖАНИЕ

Введение.....	3
Лабораторная работа № 1 Создание базовых таблиц	4
Лабораторная работа № 2 Запрос на выборку.....	19
Лабораторная работа № 3 Создание запросов «с параметром»	25
Лабораторная работа № 4 Создание итогового запроса.....	27
Лабораторная работа № 5 Пользовательские формы	30
Лабораторная работа № 6 Создание отчетов	38
Список литературы	42

Учебное издание

Наталья Васильевна РЯЗАНЦЕВА, Сергей Николаевич ХАРЛАП

Системы управления базами данных

Практикум по выполнению лабораторных работ

Редактор О. В. Занина

Технический редактор В. Н. Кучерова

Корректор М. П. Дежко

Подписано в печать 23.11.2001 г. Формат бумаги 60x84 ¹/₁₆.

Бумага газетная. Гарнитура *Times New Roman*. Печать офсетная.

Усл. печ. л. 2,32. Уч.-изд. л. 2,48. Тираж 15 0 экз.

Зак. № . Изд. №

Редакционно-издательский отдел БелГУТа, 246653, г. Гомель, ул. Кирова, 34.

Лицензия ЛВ № 57 от 22.10.97 г.

Типография БелГУТа, 246022, г. Гомель, ул. Кирова, 34.

Лицензия ЛП № 360 от 26.07.99 г.