

**ФГБОУ ВПО КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ
УНИВЕРСИТЕТ**

кафедра компьютерных технологий и систем

Методические указания

к лабораторным работам по дисциплине
«Базы данных»

для студентов специальностей:
«Бизнес-информатика»
«Прикладная информатика»

Краснодар, 2013

Составители: **Курносков Сергей Андреевич**, к.э.н., профессор
Ткаченко Василий Владимирович, к.э.н., доцент

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
Лабораторная работа 1	5
Лабораторная работа 2	12
Лабораторная работа 3	17
Лабораторная работа 4	22
Лабораторная работа 5	40
Лабораторная работа 6	46
Лабораторная работа 7	58
Лабораторная работа 8	63
Лабораторная работа 9	66
Список контрольных вопросов	81

ВВЕДЕНИЕ

Сегодня информацию рассматривают как один из основных ресурсов развития общества, а информационные системы и технологии как средство повышения производительности и эффективности работы людей. Ядром любой информационной системы является база данных (БД).

База данных – это поименованная совокупность структурированных данных, относящихся к определенной предметной области и находящихся под управлением СУБД. **Система управления базами данных (СУБД)** – это комплекс программных и языковых средств, предназначенных для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации.

Цель предлагаемого пособия - практическое освоение возможностей реляционных СУБД на примере СУБД *Microsoft Access 2000*. В пособии излагаются справочные сведения по среде разработки БД, предложены задания для самостоятельной работы и даны рекомендации по их выполнению. В конце пособия приведен список контрольных вопросов, по которым будет проводиться собеседование.

В курсе лабораторных работ студенты должны создать БД по проектным материалам расчетно-графического задания. По каждой лабораторной работе необходимо представить отчет, соответствующий этапу расчетно-графического задания.

Лабораторная работа 1

СОЗДАНИЕ ТАБЛИЦЫ. ФОРМАТИРОВАНИЕ ДАННЫХ ТАБЛИЦЫ

Цель работы - изучение возможностей СУБД *Microsoft Access 2000* по созданию таблиц и представлению в них информации.

Краткие сведения по среде разработки

База данных в *Access* представляет собой единый объект, который включает в себя такие составляющие, как таблицы, отчеты, запросы, формы и т.д., и позволяет хранить их в едином дисковом файле с расширением *.mdb*.

Основным структурным компонентом БД является *таблица*. Таблица состоит из набора строк (или *записей*). Структура всех записей в наборе одинакова, а количество записей в таблице – переменное. Элементы данных, из которых состоит каждая запись, называются *полями*.

Каждому полю таблицы присваивается уникальное имя, которое может содержать до 64-х символов, включая пробелы. Нельзя использовать в имени поля точку, восклицательный знак, надстрочный символ (') и квадратные скобки. Имя поля не должно начинаться с пробела или управляющего символа (коды ASCII 00-31). Желательно избегать имен полей, совпадающих с именами встроенных функций или свойств *Microsoft Access*.

Данные в столбце таблицы должны быть одного типа. СУБД *Microsoft Access 2000* поддерживает восемь типов данных (таблица 1.1).

Microsoft Access предоставляет три способа создания таблиц: *в режиме таблицы*, в режиме *Конструктора*, с помощью *Мастера таблиц*.

Таблица 1.1

Типы данных СУБД *Microsoft Access 2000*

Тип данных	Использование	Размер
Текстовый	Алфавитно-цифровые данные	До 255 байтов
Мемо	Алфавитно-цифровые данные - предложения, абзацы, тексты	До 64 Кбайт
Числовой	Числовые данные	1,2,4,8 и 16 байтов
Дата/Время	Даты и время	8 байтов
Денежный	Данные о денежных суммах, хранящиеся с 4 знаками после запятой	8 байтов
Счетчик	Уникальное длинное целое, генерируемое <i>Access</i> при создании каждой новой записи	4 байта
Логический	Логические данные	1 бит
Объект OLE	Диаграммы, рисунки и другие объекты из приложений <i>Windows</i>	До 1 Гигабайта

Задание

1. Разработайте структуры таблиц БД в соответствии с темой расчетно-графического задания.
2. Создайте в среде *Microsoft Access* в **режиме Конструктора** одну таблицу.
3. Заполните таблицу произвольными данными в соответствии с темой расчетно-графического задания (не менее 15 записей).

Последовательность выполнения работы

1. Запустите *Microsoft Access*. Для этого выберите пиктограмму *Microsoft Access*, которая находится в панели *Microsoft Office*, либо последовательно выберите кнопку “Пуск”, пункты меню “Программы” и “*Microsoft Access*”, либо в окне “Мой компьютер” найдите папку *Microsoft Office* и в ней обратитесь к пиктограмме *Microsoft Access*.

2. В диалоговом окне **Создание базы данных** (рис.1.1) поставьте переключатель в положение *Новая база данных* и нажмите кнопку *OK*.

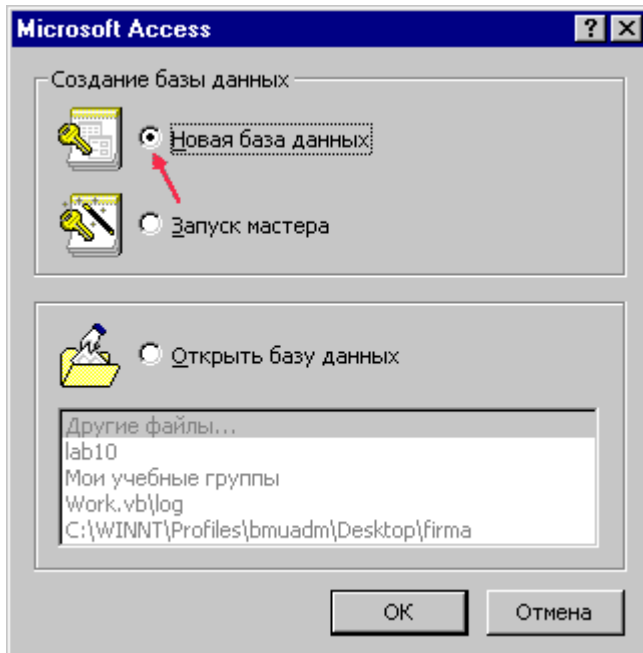


Рис.1.1 Окно Создание базы данных

3. В появившемся окне **Файл новой базы данных** укажите *диск и каталог* (папку), куда в последствии будет помещен файл базы данных, затем укажите *имя файла* и нажмите кнопку *OK*.

4. В диалоговом окне **База данных** (рис.1.2) выберите вкладку *Таблица* и нажмите кнопку *Создать*.

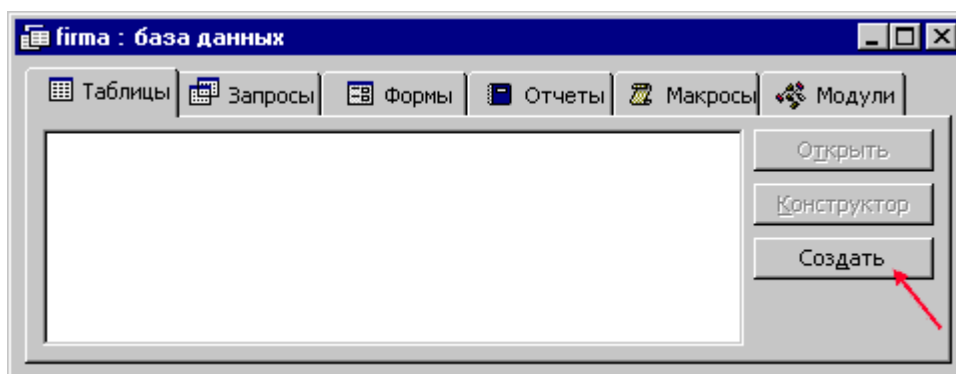


Рис. 1.2. Окно Базы данных

5. В диалоговом окне **Новая таблица** (рис. 1.3) выберите режим создания структуры новой таблицы - *Конструктор*.

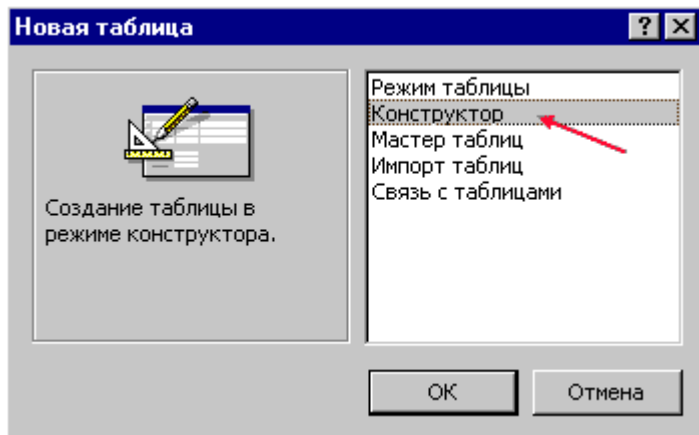


Рис. 1.3 Окно Новая таблица

6. В окне **Конструктора** таблицы (рис. 1.4) укажите для каждого поля его *имя* и *тип* (рекомендуется в колонке *Описание* указать назначение поля). Для задания типа поля щелкните в соответствующей ячейке в столбце *Тип данных*. Затем откройте символ списка, который появится справа в выбранной ячейке, и выберите нужный тип.

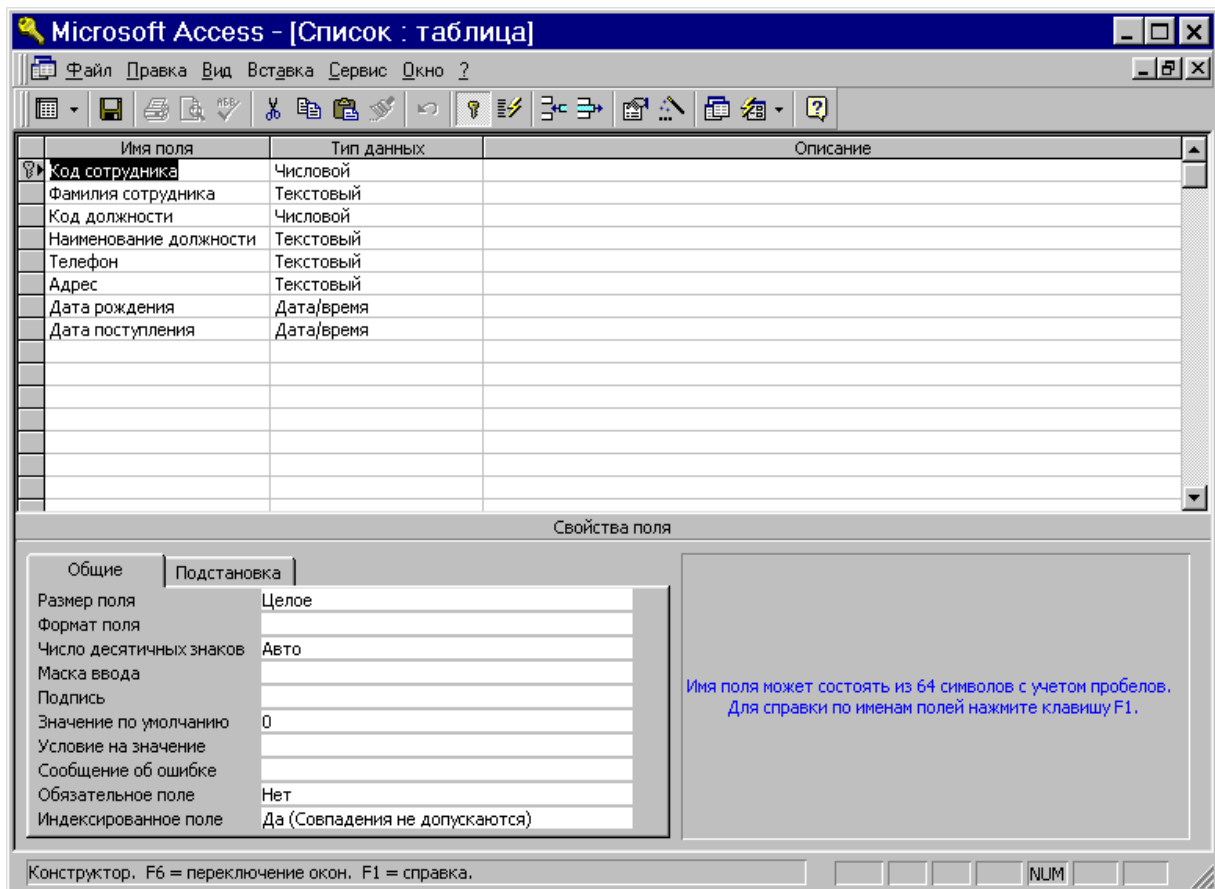


Рис. 1.4. Полная спецификация полей записи - режим *Конструктор*

Для каждого поля задайте *свойства* (перечень свойств зависит от выбранного типа данных):

- **Размер поля** задает максимальный размер данных, для хранения которых предназначено данное поле. Для *числовых* данных размер поля может быть следующим:

- байт – целые числа от 0 до 255, занимает при хранении 1 байт;
- целое – целые числа от -32 768 до 32 767, занимает 2 байта;
- длинное целое – целые числа от -2 147 483 648 до 2 147 483 647, занимает 4 байта;

- с плавающей точкой (4 байта) – числа с точностью до 6 знаков от $-3,4 \cdot 10^{38}$ до $3,4 \cdot 10^{38}$;

- с плавающей точкой (8 байта) – числа с точностью до 10 знаков от $-1,797 \cdot 10^{308}$ до $1,797 \cdot 10^{308}$;

- код репликации – глобально уникальный идентификатор (GUID) длиной 16 байт.

- **Формат поля** задает способ отображения содержимого поля.

- **Подпись** задает заголовок столбца, отличный от имени соответствующего поля. Это полезно в ситуациях, когда удобнее иметь заголовок столбца более длинный или более понятный, чем используемое имя поля. В основном, применимо в программе на Visual Basic для приложений (VBA).

- **Маска ввода** задает формат, позволяющий автоматизировать проверку вводимых символов. *Маска ввода* состоит из постоянных символов (таких как скобки, точки или дефисы) и специальных символов, указывающих, в какие позиции, в каком количестве и какого типа данные могут быть введены. *Маска ввода* обеспечивает соответствие данных определенному формату, а также заданному типу значений, вводимых в каждую позицию. Символы, используемые для задания маски ввода приведены в таблице 1.2.

- **Значение по умолчанию** задает значение, присваиваемое автоматически в качестве значения поля.

- **Условие на значение** определяет условие (ограничение), накладываемое на вводимые в поле данные. При несоответствии вводимых данных указанному условию выдается сообщение об ошибке.

- **Сообщение об ошибке** определяет то сообщение, которое будет выводиться, если при вводе данных не соблюдается условие, указанное в свойстве *Условие на значение*.

- **Обязательное поле** - свойство, указывающее требует ли поле обязательного ввода значения. Если это свойство имеет значение "Да", то при вводе новой записи необходимо ввести значение в это поле. Пустые (Null) значения в этом поле не допускаются.

- **Индексированное поле** определяет, является ли данное поле индексированным и если является, то в каком режиме. Если поле индексированное, то средствами СУБД в системе будет создан упорядоченный список содержимого данного поля, что обеспечивает более быстрый доступ к данным. Свойство *Индексированное поле* может иметь следующие значения: *Нет* (значение по умолчанию) - индекс не создается; *Да* (Допускаются совпадения) - в индексе допускаются повторяющиеся значения; *Да* (Совпадения не допускаются) - повторяющиеся значения в индексе не допускаются.

Таблица 1.2

Символы, используемые для задания маски ввода

Символ маски	Назначение
0	Цифра от 0 до 9 (ввод обязателен; символы плюс [+] и минус [-] не допускаются)
9	Цифра или пробел (ввод не обязателен; символы плюс и минус не допускаются)
#	Цифра или пробел, знаки плюс или минус (ввод не обязателен)

Продолжение таблицы 1.2

L	Буква (от A до Z или от А до Я, ввод обязателен)
?	Буква (от A до Z или от А до Я, ввод не обязателен)
A	Буква или цифра (ввод обязателен)
a	Буква или цифра (ввод необязателен)
&	Любой символ или пробел (ввод обязателен)
C	Любой символ или пробел (ввод необязателен)
. , : ; - /	Десятичный разделитель и разделители тысяч, значений дат и времени
<	Указывает перевод всех следующих символов на нижний регистр
>	Указывает перевод всех следующих символов на верхний регистр
!	Указывает заполнение маски ввода справа налево, а не слева направо
\	Указывает ввод любого следующего символа в качестве текстовой константы, даже если он является специальным символом маски. Например, \A будет выводить в маске букву А
Пароль	Создание поля для ввода пароля. Любой символ, введенный в поле, сохраняется как символ, но отображается как звездочка (*)

Специальные *форматы* для *текстовых полей* и *полей МЕМО* создаются с помощью следующих символов:

пробел — символ “пробел” вводится в формат нажатием клавиши ПРОБЕЛ и отображается в выводящемся значении;

"ABC" — выводится набор символов, заключенных в кавычки;

* — позволяет заполнить доступное пустое пространство символом, следующим за символом звездочки. Например, строка форматирования "ABCD"*х заполняет поле размером в восемь символов следующей строкой: "ABCDxxxx"

[цвет] — значение выводится в цвете, указанном в квадратных скобках. Допустимые имена цветов: *Черный (Black)*, *Синий (Blue)*, *Зеленый (Green)*, *Голубой (Cyan)*, *Красный (Red)*, *Розовый (Magenta)*, *Желтый (Yellow)*,

Белый (White). Используется для любых типов данных.

(Русские названия цветов используются при задании формата в окне свойств, английские - в конструкциях Visual Basic)

Формат для текстовых полей и полей МЕМО может состоять из двух разделов, разделяемых точкой с запятой. Первая часть является собственно форматом, а вторая – определяет значение поля, если данные в него не были введены.

7. Укажите в таблице *ключевое поле*. Выделите поле и выберите в *Панели Инструментов* пиктограмму *Ключа*. Для того, чтобы создать составной ключ, выделите необходимые поля, удерживая нажатой клавишу "Ctrl", и затем в *Панели Инструментов* выберите пиктограмму *Ключа*.

Данные ключевого поля *автоматически* сортируются в порядке возрастания. В области "Свойства поля" в поле "Индексированное поле" будет значение *"Да (Совпадения не допускаются)"*, в поле "Обязательное поле" будет значение *"Да "*.

8. Сохраните созданную структуру таблицы: выполните команду *Файл/Сохранить как*; в диалоговом окне **Сохранение** введите имя таблицы.

9. Введите данные в таблицу. Для этого перейдите из *режима проектирования* таблицы в *режим заполнения*: нажмите в *Панели Инструментов* кнопки режима заполнения *Вид таблицы* (первая слева в строке пиктограмм) либо выполните команду *Вид/Таблица* в строке Главного меню.

Лабораторная работа 2

СОЗДАНИЕ СХЕМЫ ДАННЫХ

Цель работы - создать схему базы данных, связывающую таблицы между собой.

Краткие сведения по среде разработки

База данных, как правило, состоит из нескольких таблиц. Каждая таблица содержит информацию об объектах одного класса. Между объектами разных классов различают четыре типа связей “один-к-одному”, “один-ко-многим”, “многo-к-одному”, “многo-ко-многим”. Отношение между объектами определяет отношение между таблицами.

Если связь между классами объектов “один-к-одному”, то для отображения обоих объектов и связи между ними можно использовать одну таблицу. В реальной ситуации может понадобиться разбить таблицу на более мелкие, например, некоторую информацию из этой таблицы вы используете не очень часто или нельзя давать доступ к некоторым данным всем пользователям. Например, в БД “Фирма” требуется ограничить доступ к информации об окладах сотрудников. В результате в БД включаются две таблицы “Список” и “Оклады”, связанные отношением “один-к-одному”. В этом случае каждая запись в одной таблице соответствует только одной записи в другой таблице (рис. 2.1).

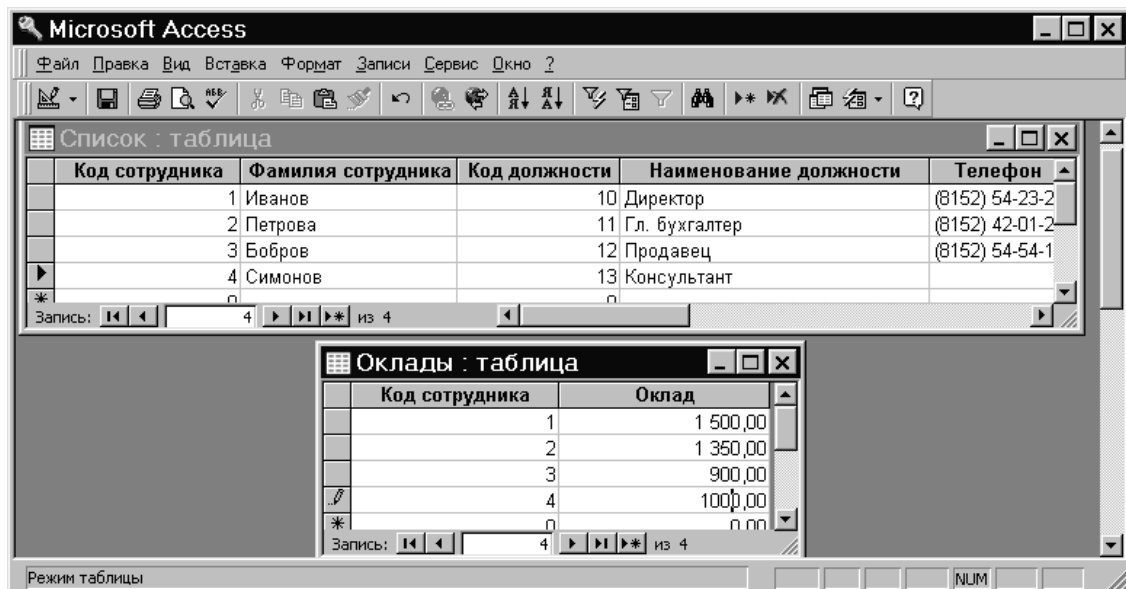


Рис. 2.1. Таблицы “Список” и “Оклады”, связанные отношением “один-к-одному”

Отношение “*один-ко-многим*” означает, что каждая запись в одной таблице соответствует многим записям другой таблицы, но в тоже время любая запись второй таблицы связана только с одной записью первой таблицы.

Отношение “*много-ко-многим*” означает, что одна запись из первой таблицы может быть связана более чем с одной записью из второй таблицы, и в то же время одна запись из второй таблицы может быть связана более чем с одной записью из первой таблицы. Отношение “*много-ко-многим*” порождает дополнительную таблицу пересечения, с помощью которой исходная связь будет сведена к двум связям типа “*один-ко-многим*”.

Задание

1. Выполните анализ предметной области с целью выявления связанных объектов, установите тип связи.
2. Выполните анализ ранее спроектированных таблиц на наличие в их структуре полей, по которым организуется связь.
3. Выполните проектирование таблиц таким образом, чтобы между ними остались отношения “*один-к-одному*” и “*один-ко-многим*”.
4. В соответствии с предложенной структурой создайте таблицы и обеспечьте целостность базы, назначением связей между всеми таблицы проектируемой базы данных.

Последовательность выполнения работы

1. Создайте таблицы, используя любой из возможных способов.
2. Установите связь между таблицами. Для этого
 - выполните команду *Сервис/Схема данных*;
 - щелкните правой кнопкой мыши на свободном пространстве схемы данных, из контекстного меню выберите команду *Добавить таблицу*;

- в диалоговом окне **Добавление таблицы** (рис 2.2) раскройте вкладку *Таблицы* и выберите из списка таблиц одну из связанных таблиц, нажмите кнопку *Добавить*; проделайте это же с таблицей, с которой будет установлена связь; (если нужно добавить к схеме несколько таблиц, повторите эту операцию для каждой таблицы);
- нажмите кнопку *Заккрыть*;

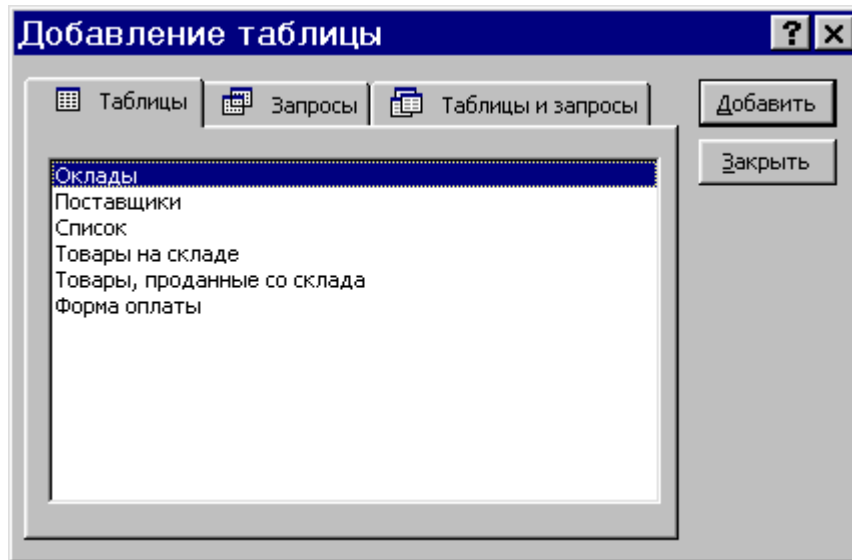


Рис. 2.2. Окно **Добавление таблицы**

- в окне **Схема данных** перенесите поле, которое следует использовать для установки связи, из списка одной таблицы к соответствующему полю другой таблицы. На экране появится диалоговое окно **Связи** (рис. 2.3); в общем случае между двумя таблицами может быть установлено несколько связей;

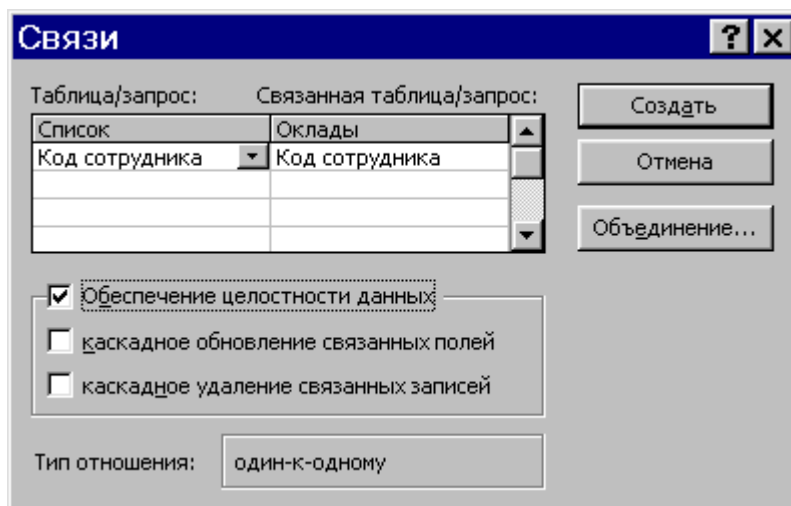


Рис.2.3. Окно **Связи**

- щелкните по флажку *Обеспечение целостности данных*, при этом для Вас станут доступными две дополнительные опции:

каскадное удаление связанных записей – это автоматическое удаление дочерних строк (*связанные строки в таблицах со стороны "многие" при связи "один-ко-многим"*) при удалении родительской строки (*связанная строка в таблице со стороны "один" при связи "один-ко-многим"*);

каскадное обновление связанных записей – это автоматическое обновление значений внешних ключей в дочерних таблицах, когда вы измените значение первичных ключей в родительской таблице.

- щелкните по кнопке **Создать** в диалоговом окне **Связи**; в окне **Схема данных** связь будет показана графически (рис. 2.4).

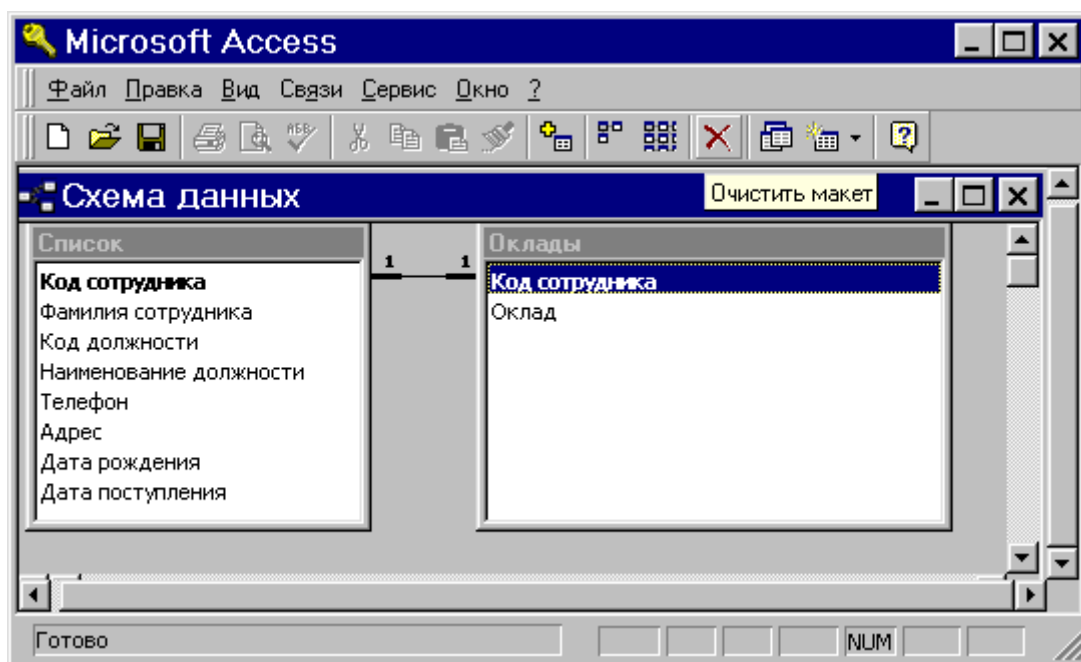


Рис. 2.4. Схема данных с установленной связью

Лабораторная работа 3

СОРТИРОВКА, ПОИСК И ФИЛЬТРАЦИЯ ДАННЫХ

Цель работы – освоить технологии поиска данных в базе данных.

Работа выполняется студентами во время, отведенное для самостоятельных занятий.

Краткие сведения по среде разработки

Сортировка, поиск и фильтрация используются для нахождения нужной информации в таблицах БД.

По умолчанию, когда таблица открывается в *режиме Таблицы*, она упорядочивается по значению ключевого поля. Если ключевое поле для таблицы не определено, записи выводятся в порядке их ввода в таблицу.

Правила сортировки:

- при сортировке в возрастающем порядке записи, содержащие пустые поля, указываются в списке первыми;
- числа, находящиеся в текстовых полях, сортируются как строки символов; если нужно отсортировать их в числовом порядке, все текстовые строки должны содержать одинаковое количество символов; если строка содержит меньшее количество символов, то в начале нужно вставить незначащие нули;
- для выполнения сортировки требуется установить курсор на любую запись в поле, по которому выполняется сортировка, и выполнить команды *Записи/Сортировка/{Сортировка по возрастанию, Сортировка по убыванию}* или использовать пиктограммы;
- для выполнения сортировки по значению нескольких полей необходимо переместить сортируемые столбцы так, чтобы они оказались рядом и были размещены с учетом приоритетов (первыми будут сортироваться значения в крайнем левом столбце); выделить все сортируемые столбцы и

выполнить команды *Записи/Сортировка/{Сортировка по возрастанию, Сортировка по убыванию}* или использовать пиктограммы.

При сохранении таблицы сохраняется и порядок ее сортировки. Для того, чтобы вернуться к первоначальному порядку сортировки по значению ключевого поля, нужно выполнить команду меню *Запись/Удалить фильтр*.

Access 2000 включает в себя возможности поиска и замены данных. Для выполнения этих операций курсор поместите в поле, содержащее данные, поиск которых необходимо произвести. Поиск начинается с активного места таблицы (активной записи, активного поля). Для просмотра всей таблицы необходимо перейти к первой записи, а затем начать поиск. Для выполнения поиска выполните команду *Правка/Найти*. Для выполнения замены – *Правка/Заменить*. В диалоговом окне **Поиск в поле (Замена в поле)** введите в текстовое поле *Образец* значение, которое ищется. Поиск можно выполнять с учетом и без учета регистра и формата поля; в текстовом поле *Образец* можно использовать шаблоны, частично или полностью заменяющие значение, поиск которого производится. Поиск производится быстрее, если поле поиска проиндексировано.

В *Access* существует четыре вида фильтров: *фильтр по выделенному фрагменту*, *обычный фильтр*, *поле Фильтр*, *расширенный фильтр*. В результате применения фильтра на экран выводятся из таблицы только те данные, которые отвечают заданным условиям отбора. Набор записей, которые были отобраны в процессе фильтрации, называется *результатирующим набором*.

Фильтр по выделенному фрагменту:

1. Открыть таблицу в режиме *Таблицы*.
2. В поле таблицы найти значение, которое должны содержать записи результирующего набора.

3. Выделить это значение и нажать кнопку *Фильтр по выделенному фрагменту* на панели инструментов *Режим таблицы* или выполнить команды меню *Записи/Фильтр/ Фильтр по выделенному фрагменту*.

Когда не удастся быстро найти образец отбора в столбце таблицы применяется **поле Фильтр**:

1. Открыть таблицу в *Режиме Таблицы*.
2. Щелкнуть правой кнопкой мыши на поле, для которого нужно указать условие отбора, а затем ввести значение условия отбора в поле *Фильтр для* в контекстном меню (в поле можно указывать не только конкретные значения, но и выражения, требующие вычислений).

3. Нажмите клавишу [Enter], чтобы применить фильтр и закрыть контекстное меню.

Чтобы применить **обычный фильтр**, необходимо:

1. Открыть таблицу в *Режиме Таблицы*.
2. Нажать кнопку *Изменить фильтр* или выполнить команду меню *Записи/Фильтр/Изменить фильтр*. Появляется окно для изменения фильтра (рис. 3.1), которое содержит линейку полей таблицы.

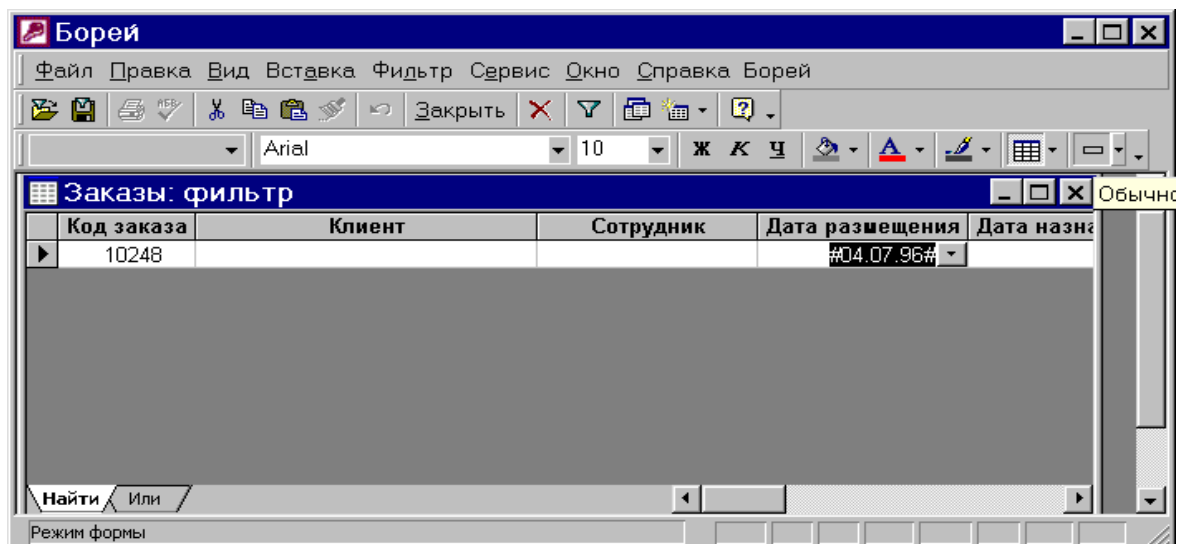


Рис. 3.1. Окно обычного фильтра

3. В любое из этих полей можно ввести или выбрать из списка значение, которое будет являться условием отбора. Условия, введенные в нескольких полях, объединяются с помощью логического оператора “И”. Для того, чтобы объединить условия оператором “ИЛИ”, нужно раскрыть другую вкладку формы, щелкнув по ярлычку *ИЛИ* в нижней части формы. При вводе условий отбора можно использовать любые выражения, которые возможны в Access.

4. Нажать кнопку *Применение фильтра* или выбрать команды меню *Фильтр/Применить фильтр*.

Применение **расширенного фильтра** выполняется в следующем порядке:

1. Открыть таблицу в *Режиме Таблицы*.

2. Выбрать пункты меню *Записи/Фильтр/Расширенный фильтр*.

появляется окно расширенного фильтра (рис. 3.2), содержащий список полей таблицы и в нижней части пустой бланк с несколькими строками.

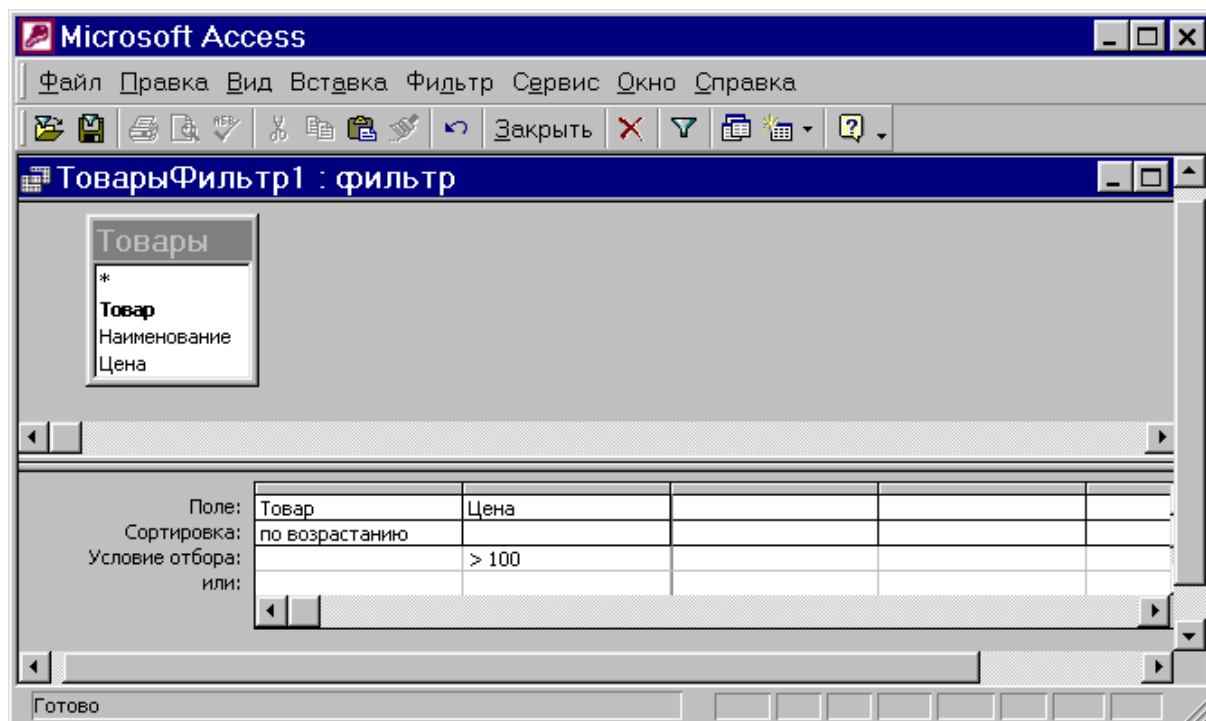


Рис. 3.2. Окно расширенного фильтра

3. Перетащить поля, по значениям которых будет выполняться отбор записей, в бланк фильтра в строку *Поле*.

4. По умолчанию в результирующем наборе записей сортировка отсутствует. При необходимости можно указать в строке *Сортировка* бланка фильтра способ вывода значений столбца. При указании порядка сортировки в нескольких полях сначала сортируются записи по крайнему левому столбцу, а затем по каждому следующему столбцу слева направо.

5. Указать в бланке фильтра в строке *Условие отбора* и в строке *Или (Or)* условия выборки из таблицы. Над условиями отбора, расположенными в одной строке, выполняется логическая операция *AND*.

6. Нажать кнопку *Применение фильтра* или выбрать команды меню *Фильтр/Применить фильтр*.

Последний фильтр, применяемый к таблице, запоминается в системе. При повторном открытии таблицы можно нажать на кнопку *Применение фильтра* на панели инструментов или воспользоваться командами меню для применения сохраненного фильтра.

Одновременно с одной таблицей может быть связан только один фильтр какого-либо одного вида. Любой из фильтров можно отменить, выбрав из меню *Записи* опцию *Удалить фильтр* или выбрать эту же опцию из контекстного меню, или повторно щелкнуть по кнопке *Применение фильтра*.

Задание

1. Самостоятельно освоить технологии сортировки, поиска, замены и фильтрации данных.

2. Защитить работу у преподавателя, продемонстрировав навыки работы с различными технологиями.

Лабораторная работа 4

СОЗДАНИЕ ЗАПРОСОВ С ПОМОЩЬЮ МАСТЕРА ЗАПРОСОВ И КОНСТРУКТОРА ЗАПРОСОВ

Цель работы - получить практические навыки разработки запросов различных типов.

Краткие сведения по среде разработки

Запрос – требование на поиск, изменение или обработку данных в БД. Запросы можно использовать в качестве источников записей для форм, отчетов и страниц доступа к данным. Запрос в *Access* является объектом, который сохраняется в файле БД и может многократно исполняться.

В *Access* существует несколько типов запросов:

- запросы на выборку;
- запросы с параметрами;
- перекрестные запросы;
- запросы на изменение (запросы на создание таблицы, удаление, обновление, добавление записей);
- запросы SQL (запросы на объединение, запросы к серверу, управляющие запросы, подчиненные запросы).

Запросы на выборку используются для выбора данных из таблиц по заданному пользователем условию. Результатом запроса является динамическая таблица, называемая также выборкой. Выборка формируется каждый раз заново на основе реальных таблиц базы данных.

Простой запрос – запрос на выборку, для формирования которого используется, как правило, одна базовая таблица. Простые запросы используются для элементарной обработки данных.

Сложный запрос – запрос, который позволяет соединять воедино данные из разных таблиц и запросов, а также из разных баз данных. Таб-

лицы (запросы) в таком запросе могут связываться с использованием внутренних, внешних, рекурсивных соединений и соединений по отношению. С помощью таких запросов можно создавать, обновлять, удалять таблицы, получать ответы на самые разнообразные вопросы.

Внутренние соединения чаще всего применяются для создания запросов на выборку. Если таблицы связаны соотношением “*один-ко-многим*”, объединения основываются на значении поля первичного ключа одной таблицы и значениях поля внешнего ключа другой таблицы. В результирующее множество запроса попадают все записи из главной таблицы (таблица “один”), для которых имеются соответствующие записи в подчиненной таблице (таблице “многие”). Внутренние соединения отображают в динамическом наборе соответствующие записи в объединенных полях таблиц как одну запись. Подобного рода соединения между таблицами создаются автоматически, если в таблицах имеются поля с одинаковыми именами и согласованными типами, причем одно из полей является ключевым либо это соединение явно задано в окне **Схема данных**. “Согласованные типы” в большинстве случаев означают одинаковые типы. Исключением является поле типа *Счетчик*, которое может иметь размер *длинного целого* или *кода репликации* и может связываться с числовыми полями соответствующего размера. Для обозначения внутренних соединений в *Access* используются зарезервированные слова *INNER JOIN*.

Для создания запроса, объединяющего все записи из одной таблицы и только те записи из второй таблицы, в которых связанные поля совпадают, используют *внешнее соединение*. В *Access* применяются *правые (RIGHT JOIN) и левые (LEFT JOIN) внешние соединения*. Запрос, основанный на таблицах с левым внешним соединением, отображает все записи главной таблицы (таблица “один”) с уникальным ключевым полем, независимо от того, имеются ли в связанных полях подчиненной таблицы (таблице “многие”) совпадающие значения. Запрос, основанный на таблицах с

правым внешним соединением, отображает все записи таблицы *многие*, независимо от того, существуют ли в связанных с ними полях главной таблицы совпадающие значения. Записи в таблице со стороны *многие*, не имеющие связанных с ними записей в таблице *один*, обычно, но не обязательно, относятся к зависшим записям. Поэтому внешние соединения используются в системе поддержки БД для удаления из таблиц “зависших” записей и дублирования данных в результате создания новых таблиц, содержащих записи с уникальными значениями.

Рекурсивные соединения (самообъединения) связывают данные в одной и той же таблице. В *Access* самообъединения создаются в результате включения в запрос дубликата таблицы (*Access* назначает дубликату псевдоним), с последующим объединением между полями копий.

Соединения по отношению (тэга-соединения) связывают данные с помощью операторов сравнения, отличных от оператора равенства (=).

Запрос на изменение – запрос, с помощью которого создают новые таблицы базы данных из таблиц запросов или вносят в таблицы значительные изменения. Такие запросы позволяют добавлять в таблицу новые записи или удалять их, или вносить изменения в записи с помощью выражений, встроенных в конструкцию запроса.

Перекрестный запрос – запрос, по требованию которого выполняются итоговые вычисления из одной или нескольких таблиц, и результаты которого представляются в виде сводной кросс-таблицы. Строки и столбцы кросс-таблицы именованные. Примеры перекрестных запросов: определить среднюю почасовую оплату, сгруппированную по виду работ и названию области; определить итоговые суммы продаж по месяцам для каждого типа изделий.

Запросы с параметрами. Для того, чтобы не формировать множество аналогичных по смыслу и структуре запросов, но с различными значениями в условиях отбора, создаются запросы с параметрами. Парамет-

рический запрос позволяет создать один-единственный запрос и ввести условие отбора посредством диалогового окна, который *Access* отображает при каждом запуске этого запроса. Запросы с параметрами фактически не являются отдельным типом запросов, поскольку функции запросов с параметрами можно добавить к запросам других типов.

Задание

1. В рамках индивидуального задания по созданию БД сформулируйте запрос простой выборки данных из разных таблиц; запрос с выборкой и группировкой с подведением итогов для разных групп, перекрестный запросы, запросы с вычислениями, параметрический, запрос на изменение.

2. Создайте перечисленные запросы с помощью **Мастера запросов** и **Конструктора запросов**. Каждый запрос сохраните.

3. Оформите отчет по работе, который должен содержать запросы, сформулированные на естественном языке, соответствующие им схемы бланка запроса и SQL-инструкции.

Последовательность выполнения работы

Создание простого запроса с помощью Мастера запросов

Мастер запросов представляет собой серию шагов, снабженных необходимыми инструкциями по созданию запроса. Перемещение между окнами *Мастера* выполняется с помощью кнопок *Назад* и *Далее*. **Мастер запросов** менее мощное средство, чем **Конструктор запросов**. Запросы, созданные в **Мастере**, могут быть затем модифицированы в **Конструкторе**. Для создания простого запроса:

1. В окне БД щелкните на закладке *Запросы*.
2. Нажмите на кнопку *Создать*.

3. В окне **Новый запрос** выберите *Простой запрос* и нажмите кнопку *OK*.

4. В окне **Создание простых запросов** (рис. 4.1.) в поле со списком *Таблицы и запросы* выберите таблицу или запрос, которые будут служить источником данных для создаваемого запроса. При формировании запроса допускается выбор нескольких таблиц или запросов.

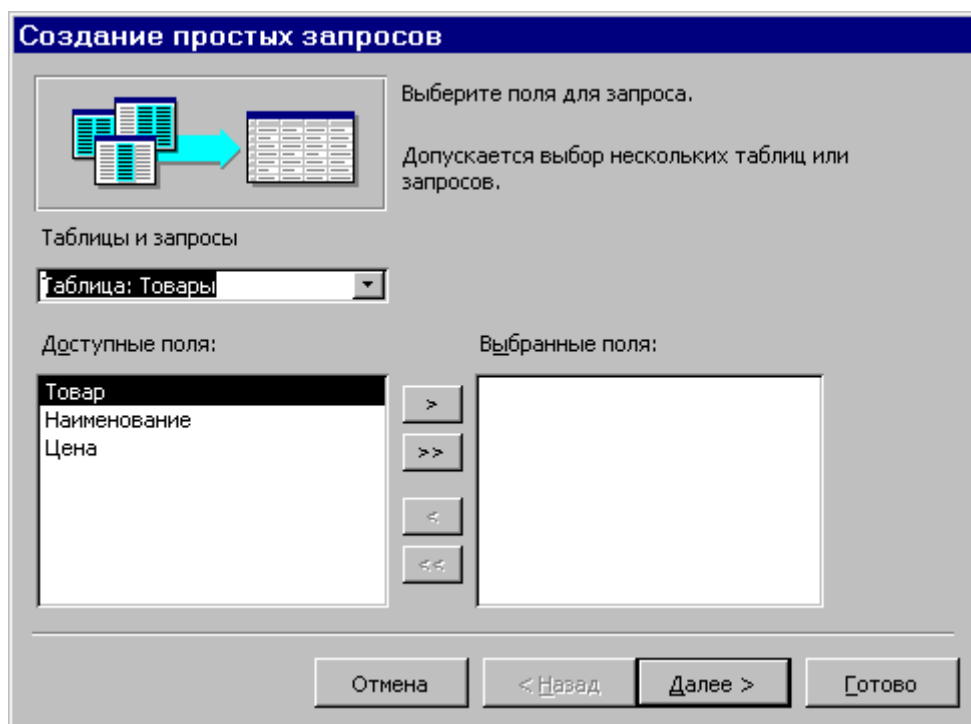


Рис. 4.1. Окно **Создание простых запросов** Мастера запросов

5. С помощью стрелок переместите из списка *Доступные поля* в список *Выбранные поля* те поля, которые необходимы в конструируемом запросе. Порядок полей в запросе будет соответствовать порядку полей в списке *Выбранные поля*. Нажмите кнопку *Далее*.

6. В следующем окне (рис.4.2) выберите *подробный* или *итоговый отчет*. В случае выбора *итогового отчета* нажмите кнопку *Итоги* для определения вида функции вычисления итоговых значений: *Sum*, *Avg*, *Min*, *Max*, *количество записей* (описание функций – Приложение 1).

7. В следующем окне **Мастера** в поле *Задайте имя запроса* введите имя создаваемого запроса.

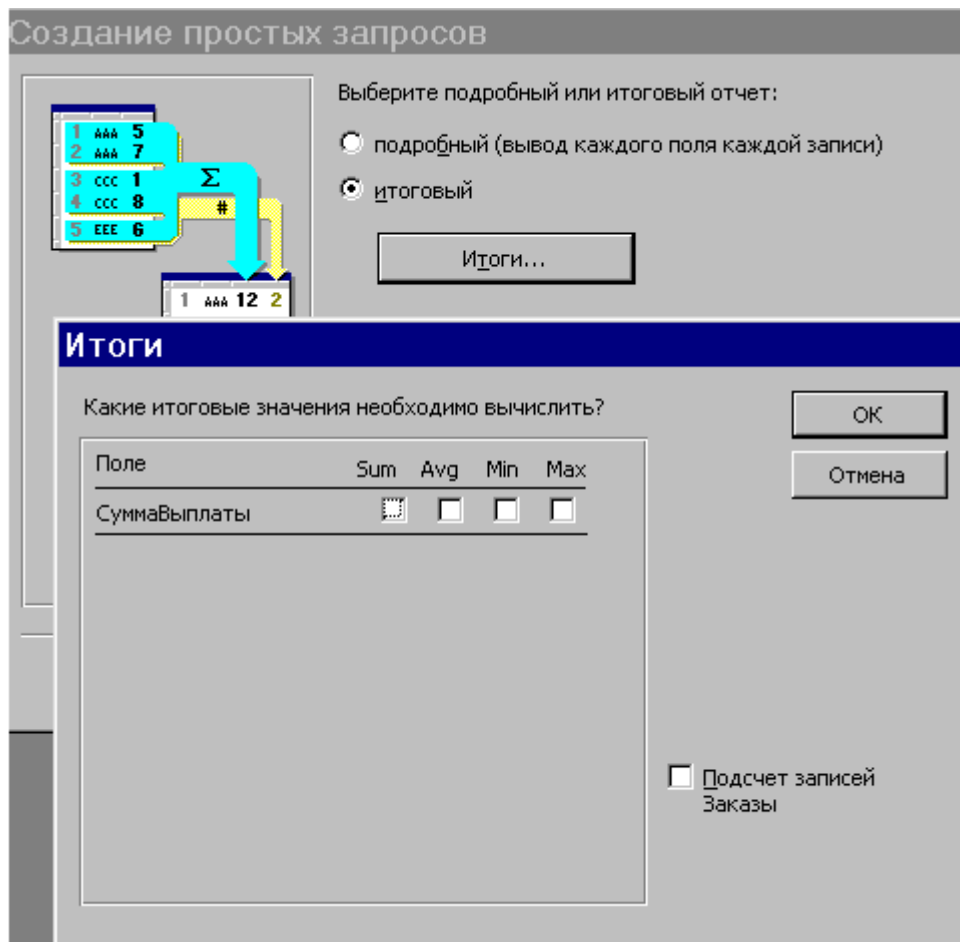


Рис. 4.2. Выбор типа запроса

8. Выберите способ открытия запроса: *Открыть запрос для просмотра данных*. По окончании работы с **Мастером** откроется окно запроса в режиме просмотра.

9. Нажмите кнопку *Готово*.

Создание запроса с помощью Конструктора запросов

Конструктор запросов позволяет создавать новые запросы и модифицировать ранее созданные запросы. По умолчанию в окне запроса в режиме конструктора создается запрос на выборку. Окно **Конструктора запросов** (рис. 4.3) состоит из двух частей: в верхней части окна расположены графические образы таблиц и запросов, данные из которых используются в текущем запросе; в нижней части окна – *бланк запроса*, называв-

шийся в ранних версиях бланком запроса по образцу (Query By Example – QBE). В *бланке запроса* пользователь указывает поля запроса, формулирует критерии отбора данных и способы представления результатов.

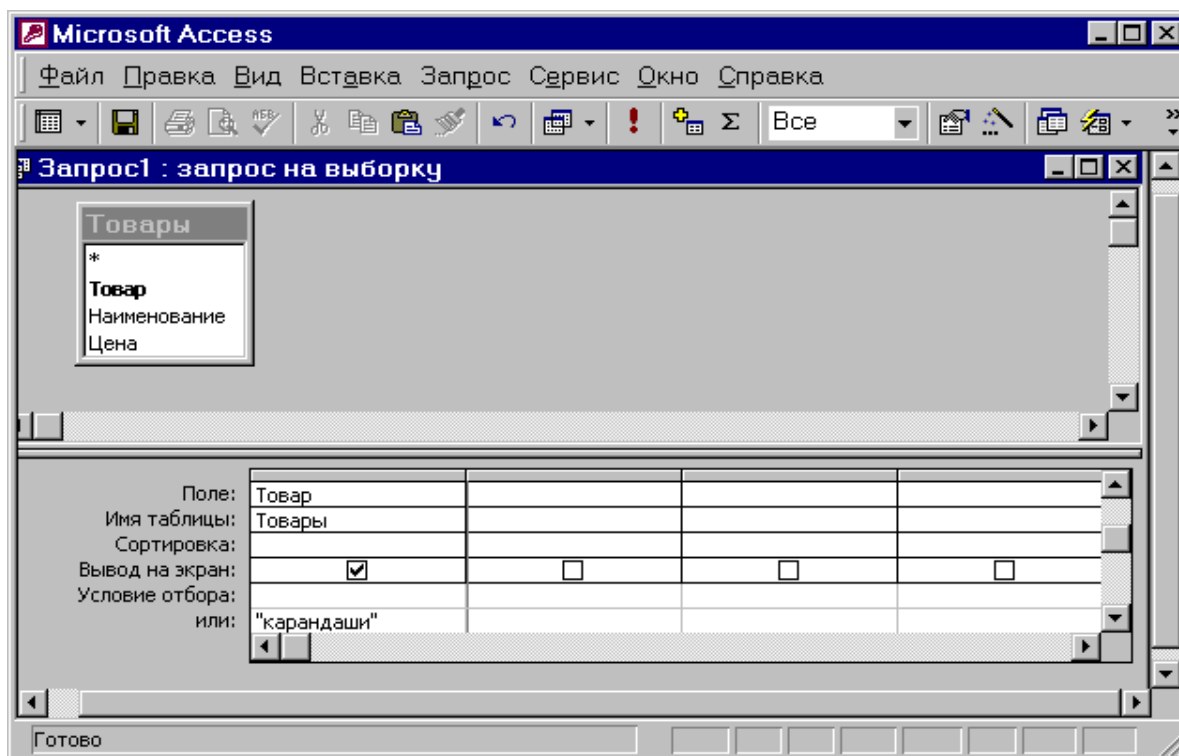


Рис. 4.3. Окно **Конструктора запросов**

Создание запроса на выборку по одной таблице

1. В **окне БД** щелкните на закладке *Запросы*.
2. Щелкните на кнопку *Создать*.
3. В **окне Новый запрос** выберите **Конструктор** и нажмите кнопку *ОК*.
4. В **окне Добавление таблицы** выберите из окна списка таблицу (или запрос), по которой будет строиться запрос и нажмите кнопку *Добавить*. Закройте текущее окно. Далее на экране появиться окно **Конструктора запросов** (рис. 4.3).
5. Добавьте поле в запрос. Для этого в *таблице-источнике*, расположенной в верхней части окна **Запрос на выборку**, выберите нужное по-

ле. Дважды щелкните левой кнопкой мыши на выделенном поле. При этом в нижней части окна **Запросов** в бланке запроса появиться столбец, соответствующий выбранному полю.

Добавить поле в запрос можно также следующим образом: нажать левую кнопку мыши на выделенном поле и, не отпуская ее, перетащить поле в нужное место бланка запроса; щелкнуть на поле ячейки в бланке запроса для отображения раскрывающегося списка полей и выбрать из него требуемое поле; щелкнуть на поле ячейки в бланке запроса и ввести имя поля.

Для добавления в запрос всех полей таблицы необходимо выделить поле, обозначенное звездочкой (*). Звездочка облегчает работу по конструированию запроса, но приводит к усложнению сортировки и ввода условия для отбора полей.

Обычно имя поля запроса совпадает с именем поля таблицы источника, но поля запроса можно переименовывать. В этом случае имя поля запроса в бланке запроса через двоеточие указывается перед именем поля таблицы-источника.

6. По умолчанию во всех полях запроса сортировка отсутствует. При необходимости можно указать в строке *Сортировка* бланка запроса способ вывода значений столбца: по возрастанию или по убыванию. При указании порядка сортировки в нескольких полях сначала сортируются записи по крайнему левому столбцу, а затем по каждому следующему столбцу слева направо.

7. В строке *Вывод на экран* проставьте флажки в полях, которые вы хотите увидеть в наборе записей после выполнения запроса. По умолчанию выводятся все поля, включенные в бланке запроса.

8. В бланке запроса в строке *Условие отбора* и в строке *Или* укажите условия выборки из базы данных. Условия представляют собой логические выражения.

Над условиями отбора, расположенными в одной строке, выполняется логическая операция *AND*. Несколько условий отбора по одному полю можно задать одним из двух способов: ввести все условия в одну ячейку строки *Условие отбора*, соединив их логическим оператором *OR*, либо ввести каждое условие в отдельную ячейку строки *Или*. Переход на следующую строку области ячеек *ИЛИ* по клавише $\langle \downarrow \rangle$. Выражения в ячейку бланка запроса вводятся с клавиатуры или для их создания используется **Построитель выражений**, который запускается командой *Построить* из контекстного меню, связанного со строкой *Условие отбора* (или можно



нажать кнопку *Построить* на панели инструментов *Access*). Основные сведения по **Построителю выражений** приводятся в *Приложении 2*.

9. Посмотрите результаты запроса, нажав кнопку *Запуск* (кнопка с изображением восклицательного знака) или кнопку *Вид* на панели инструментов.

В ситуациях, когда необходимо изменить свойство запроса, дважды щелкните на пустой области в верхней части окна **Конструктора запроса** – откроется окно **Свойства запроса**.

Запросы, созданные в режимах *Конструктора* и *Мастера*, генерируют команды на языке *SQL*. Можно просмотреть эти команды, выбрав из меню *Вид* опцию *режим SQL*.

Включение в бланк запроса групповой операции

Групповая операция используется, когда необходимо подвести итоги по группе записей, например, найти общую сумму продаж для всех клиентов или средний объем продаж по каждому месяцу и т.п. Для вычисления в запросе итоговых значений:

1. Щелкните по кнопке *Групповые операции*, знак Σ на панели инструментов *Конструктора запросов*. В бланке запроса над строкой *Сортировка* появится строка *Групповая операция*.

2. Замените в строке *Групповая операция* установку *Группировка* на требуемую *групповую операцию*, выбрав ее из поля со списком: *Sum*, *Avg*, *Count*, *Min*, *Max*, *StDev*, *Var*, *First*, *Last* (описание функций – Приложение 1), *Expression* (выражение), *Where* (условие), *Group by* (группировка).

Выражение позволяет ввести вместо названия поля в ячейку какое-либо выражение, например, для вычисления размаха значений по некоторому столбцу X : $Max([X]) - Min([X])$.

Условие показывает, что это поле будет использоваться для задания условия отбора записей. Само условие выбора записывается в ячейку строки *Условие отбора*.

3. Выполните запрос.

Включение в бланк запроса вычисляемых полей

1. Создайте запрос на выборку в режиме *Конструктора запросов*.

2. Щелкните в строке *Поле* по пустому столбцу бланка запроса. Введите имя столбца, затем двоеточие, за ним – выражение. Для ввода выражения щелкните по кнопке *Построить*, чтобы воспользоваться **Построителем выражений** (Приложение 2).

3. В верхней части **Построителя выражений** в области ввода создайте выражение.

4. В столбце вычисляемого поля переместите курсор в строку *Групповая операция* и выберите в раскрывающем списке *Выражение* (если не выбрать опцию *Выражение*, при выполнении запрос открывает диалоговое окно *Введите значение параметра* или возвращает сообщение об ошибке).

5. Запустите запрос.

Создание многотабличных запросов

Внутреннее соединение по одному полю

1. Выберите вкладку *Запросы* окна БД, щелкните кнопку *Создать*. Откроется диалоговое окно **Новый запрос**. Выберите опцию **Конструктор**.

2. В окне **Добавление таблицы** выберите последовательно две связанные таблицы, по которым будет строиться запрос. В верхней части **Конструктора запросов** отображаются связанные таблицы.

3. Заполните бланк запроса.

4. Щелкните на кнопке *Запуск* или *Вид*, чтобы отразить результаты запроса.

Аналогично создаются запросы, в которых участвуют несколько связанных таблиц, образующих цепочку. В результат запроса можно включить поля из всех таблиц, участвующих в запросе, а можно включить поля только из таблиц, находящихся на концах такой цепочки.

Внутренние соединения по нескольким полям

Между таблицами может быть задано несколько соединений и не обязательно между ключевыми полями. Связанные поля должны содержать однотипные данные.

Для создания такого соединения выполните пункты 1,2 предыдущего раздела. Затем в окне **Конструктора запроса** выберите в списке полей одной из таблиц поле, по которому образуется связь, и перетащите его на связываемое поле другой таблицы. Между связываемыми таблицами должна образоваться линия связи. Точки на концах на концах линии указывают на то, что соединение создано между полями, связь которых в схеме данных не задана. Далее выполните пункты 3,4 предыдущего раздела.

Внешние соединения

1. Выполните пункты 1- 3 раздела ***Внутреннее соединение по одному полю.***

2. Установите связь между таблицами, если ранее она не была прописана в **Схеме БД**. Для этого щелкните в первой таблице по полю, по которому будет осуществляться связь, и перетащите его в связываемое с ним поле второй таблицы

3. Щелкните на линии объединения полей, чтобы выделить ее. В результате центральная часть линии станет толще.

4. Выберите из меню команды *Вид* опцию *Параметры объединения*. Отобразится диалоговое окно *Параметры объединения* (рис.4.4).

5. Выберите тип объединения, щелкните по кнопке *ОК*. После этой операции на линии объединения появится стрелка, указывающая тип соединения (на рис.4.4 выбран второй тип соединения, на схеме ему будет соответствовать направленная вправо стрелка, что указывает на создание между таблицами левого внешнего соединения).

6. Запустите запрос на исполнение.

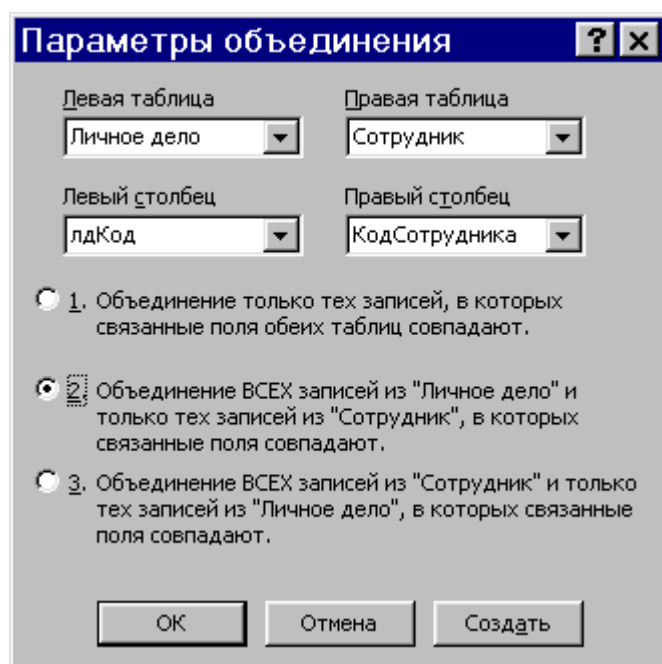


Рис.4.4. Диалоговое окно **Параметры объединения**

Создание перекрестных запросов

1. Создайте с помощью **Конструктора** новый запрос и включите в него необходимые таблицы.
2. Выберите поля, значения которых будут использованы для создания строк и столбцов динамической таблицы, и перетащите их в бланк запроса.
3. Выберите из меню команду *Запрос \Rightarrow Перекрестный*. Заголовок **Конструктора** изменится с *Запрос 1: на выборку* на *Запрос 1: перекрестный запрос*. В бланке запроса отобразится строка **Перекрестная таблица**.
4. Для полей, которые будут строками перекрестной таблицы, откройте раскрывающийся список строки *Перекрестная таблица* выберите опцию *Заголовки строк*. Каждое поле, являющееся заголовком строки должно иметь в строке *Групповая операция* установку *Группировка*.
5. Для полей, которые будут столбцами перекрестной таблицы, в зависимости от запроса, возможны следующие действия:
 в строке *Перекрестная таблица* выбрать опцию *Заголовки столбцов* и в строке *Групповая операция* установку *Группировка*, или
 в строке *Групповая операция – Условие*, в строках *Условие отбора* и *ИЛИ* записать логическое выражение.
6. Задать поле, на основе которого будет создаваться итоговое значение. В строке *Перекрестная таблица* выбрать *Значение*. В строке *Групповая операция* выбрать итоговую функцию или *Выражение*. В последнем случае в ячейке *Поле* записать выражение.
7. Запустите запрос на исполнение.

Создание параметрического запроса

Рекомендуется создать запрос обычным способом, выполнить его тестирование, а потом перейти к формированию запроса с параметрами. Это

обусловлено тем, что, если в запросе не использован ни один параметр, можно быстрее переключаться между режимами *Конструктора* и *запуска запроса*. Запрос с параметрами может содержать несколько параметров, которые связываются друг с другом логическими операциями *AND* и *OR* в зависимости от того, в каких строках бланка запроса будет указан параметр. Выполните следующие действия:

1. Создайте запрос на выборку в режиме **Конструктора запросов**.
2. Введите в одну из ячеек строки *Условие отбора* параметр, а не значение. Например, если выбор записей выполняется по фамилии, то в *параметрическом запросе* по полю *Фамилия* в условиях отбора в квадратных скобках может быть записана фраза [введите фамилию].
3. Выберите из меню *Запрос* опцию *Параметры*. В диалоговом окне *Параметры запроса* в левом столбце введите в квадратных скобках параметр, совпадающий с параметром в *бланке запроса Конструктора запросов* (параметр можно скопировать из бланка запроса). В правом столбце укажите тип данных: нажмите клавишу <F4>, из списка выберите тип данных.
4. Запустите запрос на исполнение. *Access* отобразит окно **Введите значение параметра**, в которое введите требуемое Вам значение параметра и нажмите кнопку *ОК*.

Создание запросов на изменение

Перед модификацией таблицы с помощью *запроса на изменение* всегда рекомендуется создавать резервную копию таблицы, поскольку эти запросы необратимо изменяют данные таблицы. Некорректные изменения, вносимые в таблицу *запросами на изменение*, часто трудно обнаружить.

Запрос на создание таблицы

Для того, чтобы создать таблицу из результатов запроса, использовавшегося для отбора записей в связанных таблицах, выполните следующие действия:

1. Откройте ранее созданный запрос в режиме **Конструктора**, выделив его имя в списке запросов в окне БД и нажав кнопку *Конструктор*.

2. Выберите в меню команду *Запрос ⇒ Создание таблицы*. Откроется диалоговое окно **Создание таблицы**. Введите имя новой таблицы, переключатель оставьте в положении *В текущей базе данных*.

3. Нажмите кнопку *ОК*. Запрос на выборку преобразуется в запрос на создание таблицы.

4. Закройте запрос. В окне БД ему соответствует значок, который сопровождается восклицательным знаком.

5. Запустите запрос двойным щелчком мыши по имени запроса в окне БД. Перед выполнением запроса появляется сообщение, которое предупреждает, что в новую таблицу будут внесены изменения (несмотря на то, что она еще не создана).

6. Нажмите кнопку *Да*. Появится второе предупреждающее сообщение о числе записей, которые будут помещены в новую таблицу в результате выполнения запроса.

7. Нажмите кнопку *Да*.

8. Раскройте вкладку Таблицы в окне БД; в списке должна появиться новая таблица.

9. Щелкните на кнопке *Схема данных* панели инструментов или выберите в меню команду *Сервис ⇒ Схема данных*. Появится окно **Схема данных** для установки связей между таблицами. Щелкните на кнопке *Добавить таблицу* панели инструментов и дважды щелкните по имени новой таблицы, чтобы добавить ее в окно **Схемы данных**. Щелкните по кнопке *Заккрыть*.

10. Завершите процесс включения в БД новой таблицы заданием стандартных отношений и целостности данных (см. лаб. работу 2).

Запрос на обновление записей

1. Создайте новый запрос и включите в него таблицу, в которой предполагается обновление записей.

2. Выберите в меню команду *Запрос \Rightarrow Обновление*. В бланке запроса строки *Сортировка* и *Вывод на экран* будут заменены строкой *Обновление*.

3. В строке *Обновление* задайте выражение, которое представляет собой новое значение для текущего поля.

4. Если информация обновляется не во всех записях таблицы, а только в ее части, введите условие отбора записей.

5. Выполните запрос.

Запрос на удаление записей

Рекомендуется сначала запускать запрос на выборку, чтобы отобразить удаляемые записи, а затем конвертировать его в запрос на удаление – это позволит избежать многих проблем.

1. Создайте новый запрос и включите в него таблицу, в которой предполагается удаление записей.

2. Перетащите символ звездочки *(все поля) из списка полей в строку *Поле* первого столбца бланка запроса.

3. В бланк запроса перетащите поля, по значениям которых будут отбираться поля. В строке *Условия отбора* укажите критерии отбора.

4. Запустите запрос на выборку, чтобы отобразить подлежащие удалению записи.

5. Откройте **окно базы данных**. Выберите вкладку *Таблицы*.

6. Создайте копию таблицы, с которой Вы работаете. Для этого щелкните на ее имени и нажмите комбинацию клавиш <Ctrl+C>, чтобы скопировать таблицу в буфер обмена. Затем нажмите клавишу <Ctrl+V>. Откроется диалоговое **окно Вставка таблицы**. Введите имя копии таблицы и нажмите клавишу <Enter>.

7. Активизируйте созданный запрос на выборку и перейдите в режим *Конструктора запроса*.

8. Выберите в меню команду *Запрос* \Rightarrow *Удаление*. В бланке запроса строки *Сортировка* и *Вывод на экран* будут заменены строкой *Удаление*.

9. Щелкните на кнопке *Запуск* панели инструментов. Появится окно сообщений, запрашивающее подтверждение удаления записей.

Удаление записей в таблице “*один*”, для которых в таблице “*многие*” существуют связанные записи, нарушает правило целостности данных. Поэтому опция целостности данных, установленная для связи между таблицами, может препятствовать удалению записей. Следует применить каскадное удаление.

Запрос на добавление записей

Запрос на добавление записей используется часто для того, чтобы скопировать данные из одной таблицы в другую.

1. Выберите вкладку *Запросы* окна БД, щелкните кнопку *Создать*. Откроется диалоговое окно **Новый запрос**. Выберите опцию ***Конструктор***.

2. В окне **Добавление таблицы** выберите таблицу, из которой будут браться данные.

3. Перенесите в строку *Поле* бланка запроса поля, значения которых будут включаться в другую таблицу.

4. Выберите команды меню *Запрос/Добавление*. Появится окно **Добавление** (рис. 4.5), где необходимо указать имя таблицы, в которую предполагается добавлять данные, и какой БД она принадлежит. Если БД не текущая, а какая-либо другая, то придется указать полное имя файла, содержащего БД.

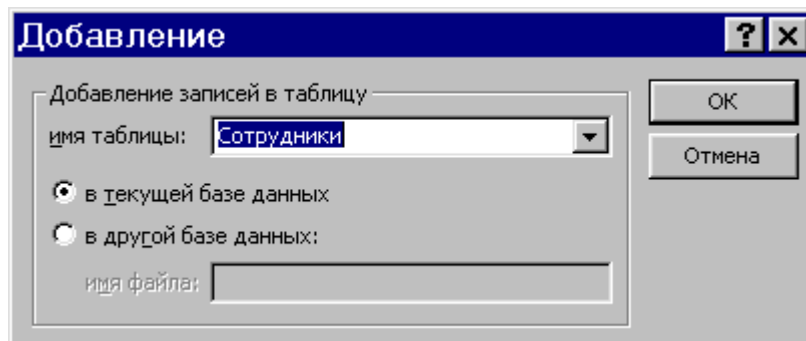


Рис.4.5. Окно выбора таблицы, в которую добавляются записи.

В бланке запроса после строки *Сортировка* будет включена строка *Добавление*, в которой требуется указать те поля таблицы, куда добавляются данные. Если имена полей совпадают Access автоматически вставит в строку *Добавление* имена полей.

5. Запустите запрос на исполнение. Посмотрите как изменилась таблица, в которую внесены новые строки.

Если в запросе на *Добавление* есть поля типа *Счетчик*, то работать с такими полями можно двумя способами:

- не включать поля типа *Счетчик* в поля, которые должны быть добавлены; тогда в таблице, куда добавляются данные, продолжится нумерация последовательности данных, т.е. если имеется таблица служащих с номерами от 1 до 100 и добавить к ней новые записи, то их нумерация начнется с 101 номера;
- включать поля типа *Счетчик* в список запроса в качестве добавляемых полей, то значения, которые они имели в первоначальной таблице будут сохранены в полях добавляемых записей; если та-

кие значения уже существуют в полях таблицы, к которой добавляются записи, это может привести к ошибке. Сообщение об ошибке отображается на экране до выполнения запроса. При наличии ошибки выбрать кнопку **Отменить** и исправить ошибку.

Лабораторная работа 5

СОЗДАНИЕ SQL-ЗАПРОСОВ

Цель работы - получить практические навыки по созданию запросов на SQL.

Краткие сведения по среде разработки

SQL (Structured Query Language) – это стандартный язык управления базами данных. Версия языка SQL, реализованная в *Access*, несколько отличается от стандартной версии. Отличаются, хотя и незначительно, версии SQL, используемые в Access 2.0, Access 95, Access 97, Access 2000. Язык Access SQL имеет теперь новое название Microsoft Jet Database Engine SQL, сокращенно Jet SQL.

Любой запрос, созданный с помощью **Мастера запросов** либо с помощью **Конструктора запросов** хранится в БД в формате SQL. Есть несколько типов запросов, которые невозможно создать с помощью **Конструктора запросов**. Перечисленные ниже типы запросов предполагают использование конструкций SQL:

- запрос на объединение в одном результирующем наборе результаты нескольких запросов и таблиц;
- управляющие запросы, позволяющие создавать, изменять и удалять таблицы, создавать индексы;

- подчиненные запросы (можно создать в режиме Конструктора запросов, но необходимо в *бланке запроса* в строке *Условия отбора* использовать инструкции SQL);

- запрос к серверу.

Структура запроса на выборку:

```
SELECT[ALL|DISTINCT|DISTINCTROW][TOP n PERCENT]] <список выбора>
FROM <список таблиц (запросов) >
[WHERE <условие отбора строк>]
[GROUP BY <список полей, по которым будут группироваться записи>]
[HAVING <условие отбора групп записей>]
[ORDER BY <определение сортировки записей>];
```

SELECT – команда, определяющая запрос на выборку. Список выбора (список полей) содержит те поля, которые должны быть включены в результирующую таблицу запроса и их имена в этой новой таблице. Имена полей отделяются друг от друга запятыми. Имя поля формируется из имени таблицы и имени столбца, разделенных точкой.

Необязательные предикаты ALL, DISTINCT, DISTINCTROW определяют способ отбора строк. При использовании ALL в результирующую таблицу включаются все строки, удовлетворяющие указанным далее условиям. Предикат DISTINCT исключает повторяющиеся строки, основываясь на данных результирующего набора записей. Предикат DISTINCTROW исключает повторяющиеся строки, основываясь на данных полных строк исходной таблицы, независимо от того, включены ли в запрос те поля таблицы, данные которых различаются.

Необязательные предикат TOP n [PERCENT] возвращает первые *n* записей или *n* процентов набора записей, удовлетворяющих запросу.

Предложение FROM определяет таблицы или запросы, которые служат источником данных. Синтаксис предложения FROM :

```
FROM <имя таблицы (имена таблиц)>
    [{INNER/ LEFT/ RIGHT }JOIN < связанная таблица>
    ON <условие объединения >]
```

Если источником служит не текущая база данных, необходимо использовать предложение IN, чтобы задать путь к базе данных:

```
IN <'имя БД-источника' >.
```

Предложение FROM позволяет определить псевдоним таблицы, который можно использовать вместо полного имени таблицы при задании имен столбцов в списке выбора: FROM <имя таблицы > [AS <псевдоним>].

INNER JOIN – внутреннее объединение таблиц; LEFT JOIN – левое внешнее объединение; RIGHT JOIN – правое внешнее объединение.

Если имя таблицы (поля) совпадает с зарезервированным словом языка SQL (например, "ORDER") или содержит пробелы, то его необходимо заключить в квадратные скобки.

Предложение WHERE задает условие отбора строк. Для написания условий отбора можно использовать операторы сравнения, логические операторы, операторы LIKE, BETWEEN, IN, IS (*Приложение 2*).

Если необходимо результат выполнения одного запроса использовать в качестве данных для другого запроса, то первый будет являться подчиненным по отношению ко второму. Структура подчиненного запроса аналогична структуре внешнего запроса. Подчиненный запрос помещается в команду WHERE и заключается в круглые скобки.

Предложение ORDER BY задает порядок сортировки записей созданного запросом набора строк: по возрастанию (ASC) или по убыванию (DESC). По умолчанию принимается значение ASC:

```
ORDER BY <имя столбца> [{ASC, DESC}].
```

ПРИМЕРЫ:

1. Найти тех клиентов, адрес которых не совпадает с адресом получателя.

```
SELECT Klient.Name, Klient.Address, Customer.AdressRecipient
FROM Klient INNER JOIN Customer ON Klient.CodeKlient = Customer.CodeKlient
WHERE Customer.AddressRecipient<> Klient.Address
```

2. Найти всех поставщиков, которые поставляют, по крайней мере, один вид комплектующих.

```
SELECT DISTINCTROW Supplier.Name
FROM Supplier
WHERE EXISTS
  (SELECT *
   FROM Component
   WHERE Component.SupplierID = Supplier.SupplierID);
```

EXISTS – оператор, генерирующий значение “истина” или “ложь”, его можно использовать отдельно в предикате или комбинировать с другими булевыми выражениями с помощью операторов AND, OR, NOT. Используя подзапрос в качестве аргумента, этот оператор оценивает его как истинный, если он (подзапрос) возвращает, по крайней мере, одну строку, а в противном случае как ложный.

Операторы SOME, ANY, ALL, также как и EXISTS, используются только с подзапросами.

При использовании ALL предикат принимает значение “истина”, если все результаты сравнения со значениями, возвращаемыми при выполнении подчиненного запроса истинны. Если некоторое значение, возвращаемое подчиненным запросом, равно NULL, результат не определен.

При использовании SOME или ANY предикат принимает значение “истина”, если результат сравнения хотя бы с одним из возвращаемых подчиненным запросом значений истинен.

ПРИМЕР: Определить товары, цена которых больше, чем стоимость любого из товаров с кодом 007.

```
SELECT Good.Description, Good.Price
FROM Good
WHERE (Good.Price) > ALL
      (SELECT Price
       FROM Good
       WHERE Good.Code="007");
```

Для подведения итогов по некоторому набору записей используются агрегатные функции (*Приложение 1*). Агрегатные (статистические) функции возвращают одно результирующее значение, вычисленное на основании набора данных. Аргументом функций является либо имя столбца, содержащего числовые данные, либо арифметическое выражение. Внутри выражения нельзя использовать другие агрегатные функции.

Предложение GROUP BY определяет столбец (столбцы) по значениям которого записи объединяются в группы.

Предложение HAVING определяет условие отбора групп записей в столбцах, указанных в предложении GROUP BY.

ПРИМЕР: Вывести список регионов, суммарная стоимость доставок по которым за период 1997-1999 г.г. превышает 100000 рублей.

```
SELECT Region, SUM(CostDelivery) AS [Total Cost Delivery]
FROM Order
WHERE CountryDelivery = "Россия"
      AND Date BETWEEN #1/01/97# AND #31/12/99#
GROUP BY Region
HAVING SUM(CostDelivery) > 100000
ORDER BY SUM(CostDelivery) DESC;
```

Для определения перекрестных запросов используется следующий синтаксис:

TRANSFORM <выражение с итоговой функцией> [AS подпись]

<инструкция SELECT>

PIVOT <выражение>

Ключевое слово TRANSFORM определяет перекрестный запрос; <инструкция SELECT> соответствует синтаксису обычного запроса и обязательно содержит предложение GROUP BY; PIVOT определяет столбец или выражение, значения которого используются в качестве заголовков столбцов перекрестной таблицы. В предложении PIVOT можно задать формат вывода заголовков столбцов.

ПРИМЕР: Определить итоги продаж товаров за каждый месяц 1999 года по каждому товару.

```

TRANSFORM Sum (MounthlySales.TotalInvoiceAmount) AS
TotalAmount
SELECT Catalog.Descripti
FROM Catalog INNER JOIN Mounthly.Sales
ON Catalog.Code= MounthlySales.Code
WHERE MounthlySales.Date LIKE “*. *.99”
GROUP BY Catalog.Description
ORDER BY Catalog.Description
PIVOT Format ([Date]
IN ("январь", "февраль", "март", "апрель", "май", "июнь", "июль",
"август",
"сентябрь", "октябрь", "ноябрь", "декабрь");
  
```

Задание

1. В рамках индивидуального задания по созданию базы данных сформулируйте

- запрос на выборку, который предполагает сортировку, группировку данных и получение итоговых значений;
- запрос, содержащий в себе подзапрос;

- перекрестный запрос.
- 2. Запишите вышеперечисленные запросы с помощью операторов SQL.
- 3. Выполните генерацию запросов в среде *Access*.
- 4. Сохраните запросы.
- 5. Оформите отчет по лабораторной работе, включив в него тексты запросов на естественном языке и соответствующие им SQL-запросы.

Последовательность выполнения работы

1. Создайте новый запрос с использованием **Конструктора запросов**.
2. Закройте диалоговое окно **Добавление таблицы**, не добавляя таблиц.
3. Выберите команду *Вид, Режим SQL* для вывода окна SQL.
4. Удалите весь текст, который может быть выведен в окне SQL.
5. Введите в окне инструкцию SQL, используя для ввода новой строки комбинацию клавиш <Ctrl>+<Enter>.
6. Щелкните по кнопке *Вид* или *Запуск*.

Лабораторная работа 6

СОЗДАНИЕ ФОРМ ДЛЯ ВВОДА, РЕДАКТИРОВАНИЯ И ПРОСМОТРА ДАННЫХ

Цель работы - получить практические навыки разработки форм для ввода, редактирования и просмотра данных.

Краткие сведения по среде разработки

Форма — это структурированный документ с незаполненными областями, в которые вводятся данные.

В Access формы служат для организации пользовательского интерфейса с таблицами данных и запросами. Содержимое и внешний вид формы зависит от ее назначений в конкретном приложении БД. Существует три типа приложений БД:

Приложения для обработки транзакций. Они предназначены для добавления в таблицы новых записей и редактирования существующих.

Приложения для поддержки принятия решений. В таких приложениях информация представлена в виде графиков, таблиц или отдельных элементов данных. Возможность ввода и редактирования данных отсутствует.

Приложения поддержки БД. Эти приложения предназначены для создания БД и таблиц БД, контроля доступа к БД, защиты данных с помощью пароля, периодического уплотнения и копирования БД.

Для создания форм в Access используются:

1. *Средства автоматического создания форм* по таблице или запросу. Разработчику предлагается три вида автоформ:

- форма, организованная в “столбец”: поля формы расположены друг под другом;
- ленточная форма: поля расположены по всему окну в форме таблицы; форма позволяет просматривать одновременно несколько записей и имеет свою линейку прокрутки;
- табличная форма, по виду сходная с таблицей.

2. *Мастер форм*. Мастер форм разбивает процесс создания формы на несколько этапов. На каждом этапе требуется установить определенные параметры в одном из диалоговых окон Мастера. С помощью Мастера можно создавать запросы как на основе одной таблицы (запроса), так и на основе нескольких таблиц и запросов.

3. *Конструктор форм.* Конструктор используется для разработки структуры формы, изменения ее внешнего вида, добавления и удаления элементов управления.

Форма в окне Конструктора состоит из пяти частей (рис.5.1): заголовок формы, область данных, верхний и нижний колонтитулы и примечание формы. Заголовок формы определяет верхнюю часть формы, при печати многостраничной формы отображается только на первой странице. В заголовке формы можно поместить текст, графику и другие элементы управления. Область данных – основная часть формы, содержащая данные, полученные из источника. Примечание определяет нижнюю часть формы, при печати многостраничной формы отображается только на последней странице. Обязательным в форме является только раздел области данных. Для добавления или удаления разделов формы используйте пункты меню Вид.

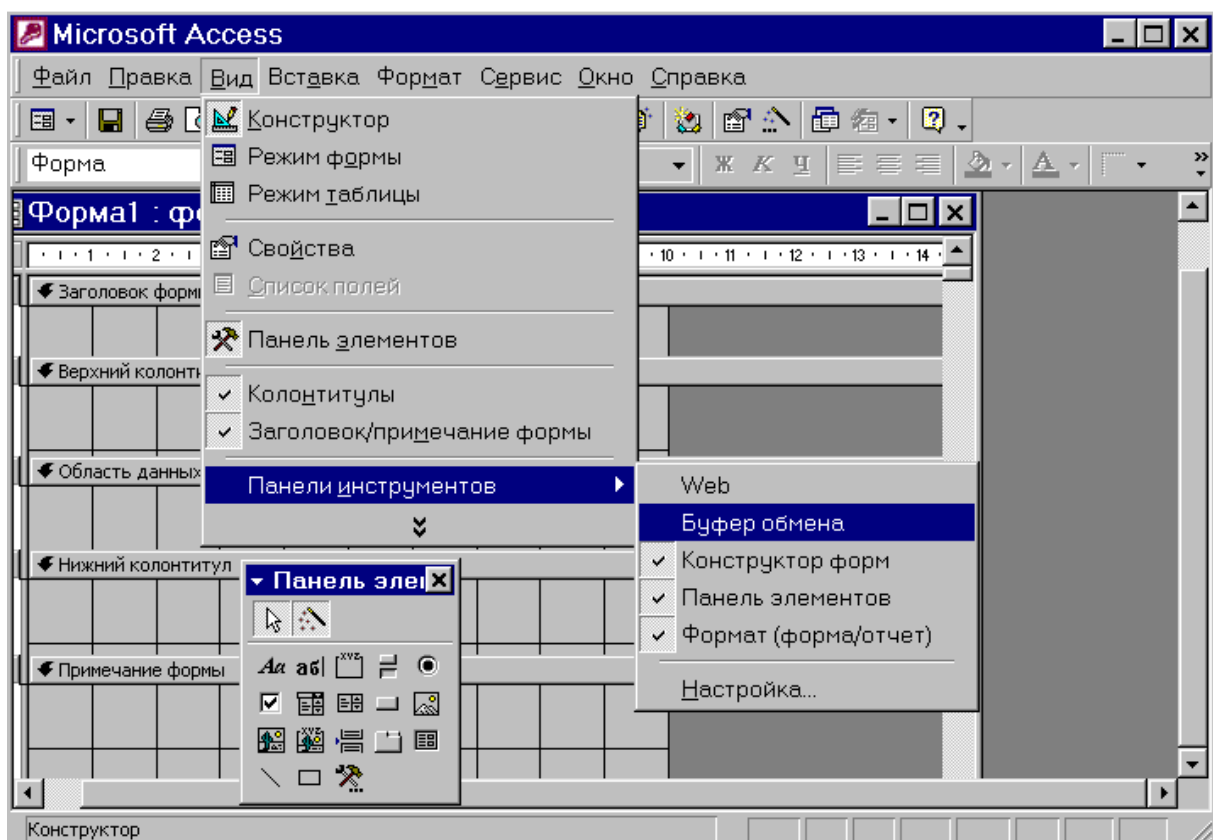


Рис. 5.1. Окно проектирования формы

Формы содержат элементы управления, с помощью которых осуществляется доступ к данным. Элементами управления являются текстовые поля, кнопки, флажки, переключатели, выключатели, списки, надписи, рисунки, рамки объектов для отображения графики и объектов OLE (*Приложение 3*). Элементы управления размещены на панели элементов, которую можно активизировать используя команды меню (рис.5.1) или кнопку



Все элементы формы в *режиме Конструктора* представлены в качестве объектов. После того, как объект выбран, он окружается рамкой с черными квадратиками. Квадратики служат для изменения размеров объекта. Можно перемещать объект, задавать его свойства, программировать событие

Рис. 5.1. Окно Конструктора форм

нию.

Цвет фона формы и ее элементов можно изменять. В форме, созданной с помощью *Мастера форм*, фон ее разделов зависит от стиля оформления, выбранного при его создании. Чтобы изменить цвет в режиме *Конструктора*: выделите раздел или элемент формы; щелкните левой кнопкой мыши по пустой области внутри выделенного раздела (заголовок, область данных или примечания), либо по элементу формы; на панели инструментов *Формат(форма/отчет)* нажмите стрелку справа от кнопки *Цвет заливки фона*; выберите в палитре нужный цвет.

В качестве фона формы можно использовать растровые рисунки, хранящиеся в графических файлах формата .WMF или .BMP. Чтобы добавить фоновый рисунок: в режиме *Конструктора* щелкните левой кнопкой мыши по квадратику в верхнем левом углу конструктора формы (должен появиться черный квадратик); на панели инструментов *Конструктор*

форм нажмите кнопку *Свойства*; раскройте вкладку *Макет*, найдите на ней свойство *Рисунок*, в поле ввода свойства введите путь и имя файла.

Формы могут быть построены на основе одной или нескольких таблиц. К многотабличным формам относят формы, построенные на основе запроса, объединяющего несколько таблиц, а также формы с внедренными в них другими формами. Форма, встроенная в другую форму, называется *подчиненной формой*. При этом форма, которая включает подчиненную форму, называется *главной*. Обычно такие формы применяются для отображения данных из связанных таблиц.

В *Access* возможно создание многостраничных форм.

Задание

1. Откройте учебную БД “Борей”. В окне БД перейдите на вкладку *Формы*. Откройте некоторые формы, чтобы иметь представление о структуре и дизайне формы.
2. Изучите самостоятельно средства автоматического создания форм по таблице или запросу и Мастер форм.
3. Для одной из таблиц БД создайте простую форму для ввода и редактирования данных с помощью *Мастера форм*.
4. Доработайте форму в режиме *Конструктора*.
5. Введите данные через окно формы.
6. Выполните редактирование записи через окно формы.
7. Создайте форму (можно на основе ранее созданной формы), содержащую элементы управления: флажки, переключатели или выключатели, поле со списком.
8. Включите в форму рисунок.
9. Создайте многотабличную форму для таблиц, между которыми существует связь “*один-ко-многим*”.
10. Создайте многотабличную форму для несвязанных таблиц.

11. Создайте форму, содержащую вычисляемые поля для отображения итоговых значений.

12. Включите в одну из форм элемент управления **Кнопка**. С кнопками должны быть связаны переходы по записям, обработка записей или работа с формой.

Последовательность выполнения работы

Создание простой формы

1. В окне **Базы данных** перейдите на вкладку *Формы*, нажмите кнопку *Создать*, выберите *Мастер форм*, укажите имя таблицы.

2. Последовательно выполняйте шаги, предлагаемые *Мастером форм*:

- выберите поля для отображения в форме;
- выберите внешний вид формы, например, в один столбец;
- выберите из предлагаемого списка стиль формы;
- задайте имя формы;
- выберите переключатель *Изменить макет формы*; проект формы будет открыт в режиме *Конструктора*.

3. Измените размер окна формы так, чтобы с ним было удобно работать (поместить указатель на границу формы и перетащить границу) либо разверните форму до максимальных размеров.

4. Переместите все поля влево так, чтобы их левая граница была прямо на отметке трех сантиметров. Выделите все поля, выберите из меню *Формат* опцию *Выровнять* и из подменю опцию *По левому краю*. Можно разместить поля в соответствии с вашими представлениями.

5. Определите свойства полей формы:

- выделите поле;
- раскройте пункты меню *Вид/Свойство*;

- для поля, которое в таблице является счетчиком, установите свойство *Доступ* в значение *Нет* (так оно не может быть изменено пользователем), свойство *Блокировка* – в значение *Да* (чтобы показать, что это поле не обновляемое). Свойства *Значение по умолчанию*, *Формат*, *Число десятичных знаков* или *Маска ввода*, *Условие на значение* определяются свойствами полей таблицы, для работы с которой строится данная форма. Для *Условия на значение* можно задать более жесткие значения.

6. Выполните редактирование надписей полей, если это необходимо. Для изменения содержания надписи, установите курсор на надпись, измените ее и нажмите [Enter] или выделите надпись и используйте команды меню *Вид/Свойства*. Изменить размер шрифта и его стиль – выделить надпись, затем используйте возможности *Панели инструментов Формат*. Настроить размер надписи по размеру, содержащегося текста – щелкните по подписи и выполните команду *Формат/Размер/По размеру данных*.

7. Добавьте в форме *Заголовок*. Для этого

- щелкните по разделу окна **Заголовок формы** (если он отсутствует, используйте команду пункта меню *Вид*);
- измените размер раздела: установите курсор на границу раздела, указатель мыши примет вид горизонтальной или вертикальной двунаправленной стрелки, щелкните мышью и перетащите границу;
- разверните *Панель элементов*, щелкните по кнопке *Надпись*; затем щелкните на левой границе раздела *Заголовок формы*, чтобы поместить туда курсор;
- введите название формы и нажмите [Enter];
- активизируйте *Панель инструментов*, выберите *Формат (форма/отчет)*; измените размер и тип шрифта;
- выберите опцию *Размер* из меню *Формат* и из подменю опцию *По размеру данных*;

- щелкните по надписи заголовка, установите курсор на верхнем левом квадрате рамки и переместите надпись так, чтобы разместить ее наилучшим образом;
- при необходимости измените размер раздела *Заголовок формы*, чтобы подогнать его точно под размер надписи.

8. Дважды щелкните в окне формы вне ее разделов. Появится окно **Свойств формы**. На закладке *Данные* в строках *Разрешить добавление*, *Разрешить изменение*, *Разрешить удаление* введите *ДА*.

9. Выведите форму в режиме *Формы* (меню *Вид/Режим формы* или кнопка *Вид*). Если не требуется вносить изменений - то сохраните форму, в противном случае вернитесь в режим *Конструктора*.

10. Введите данные через окно формы:

- откройте форму, перейдите в *режим формы*. В текстовых полях формы появятся значения полей первой записи или запроса, являющиеся источником данных формы;
- нажмите кнопку *Новая запись* (►*) на панели инструментов *Режим формы* либо кнопку *Новая запись*, расположенную в нижней части окна формы (чтобы отменить создание новой записи после нажатия кнопки *Новая запись* – нажмите кнопку перехода к предыдущей записи). Все поля новой записи будут пустыми, за исключением тех, значения которых определено по умолчанию. Значение по умолчанию можно изменять, если свойство *Блокировка* этого поля имеет значение *Нет*;
- введите данные. Введенные данные сохраняются в исходной таблице автоматически при переходе к другой записи или при закрытии формы. Если есть необходимость сохранить промежуточное состояние записи (до перемещения на другую запись), то нажмите комбинацию клавиш [Shift]+[Enter] или выберите команду *Запись/Сохранить запись*.

Последнее сохранение записи можно отменить, если сразу же нажать кнопку *Отменить* на панели инструментов *Режимы формы* или выбрать команду *Правка/Восстановить запись*.

11. Выполните редактирование записи через окно формы:

- найдите запись, которую предполагается редактировать. Для этого используйте кнопки навигации, расположенные в нижней части окна формы, или в текстовое поле, находящееся между кнопками навигации, введите номер записи и нажмите [Enter], или используйте команды меню *Правка/Найти*;
- выполните редактирование данных;
- сохраните изменение – перейдите к другой записи или выполните команду *Запись/Сохранить запись*.

Для того, чтобы запретить редактирование записи таблицы с помощью формы, в режиме *Конструктора формы* в окне свойств формы на вкладке *Данные* против свойства *Разрешить изменение* введите *Нет*. Чтобы запретить редактирование конкретного поля записи, в режиме *Конструктора формы* в окне свойств поля на вкладке *Данные* против свойства *Блокировка* введите *Да*.

12. Удалите запись через окно формы. Для удаления текущей записи выполните команду *Правка/Выделить запись* и нажмите клавишу [Delete], или щелкните по кнопке выделения записи, находящейся слева от области данных формы и нажмите клавишу [Delete]. В появившемся окне подтвердите удаление записи.

13. В режиме *Конструктора* сохраните последний вариант формы.

Создание элементов управления с помощью панели элементов

1. Откройте форму в режиме *Конструктора*.

2. Измените размеры разделов формы с учетом возможности размещения элементов управления.

3. Активизируйте *Панель управления* и нажмите на ней кнопку *Мастера*. Теперь при переносе любого элемента управления с *панели элементов* в форму будет автоматически открываться соответствующий *Мастер*.

Ниже приведен порядок создания одно из элементов управления. Для создания других элементов управления пользуйтесь **справочной системой**.

Создание элемента управления Список (Поле со списком)

1. На панели элементов нажмите кнопку *Список*.

2. Переведите указатель мыши в область нужного раздела формы. Указатель мыши примет вид крестика со значком элемента управления *Список*. Центр крестика определяет позицию верхнего левого угла элемента управления *Список*.

3. Нажмите левую кнопку мыши, и удерживая ее, перетащите указатель мыши в нижний правый угол списка. Число строк и количество символов текущего типа шрифта, которые может отобразить список, выводятся в строке состояния. При достижении элементом управления *Список* нужных размеров отпустите кнопку мыши. Появится первое окно *Мастера*.

4. Определите какие значения будет содержать список: значения из таблицы или запроса, фиксированный набор значений. Дальнейший диалог зависит от сделанного вами выбора.

5. Выполните все действия, предлагаемые *Мастером*.

Создание многотабличных форм для связанных таблиц

1. Выберите таблицы, связанные отношением “один-ко-многим”.

2. В окне **Базы данных** перейдите на вкладку *Формы*, нажмите кнопку *Создать*.

3. В окне **Новая форма** выберите *Мастер форм*, в поле со списком укажите таблицу для главной формы.

4. В первом окне диалога *Мастера форм* в списке *Доступные поля* выберите те поля, которые вы хотите включить в главную форму. В этом же окне в поле со списком *Таблицы/запросы* выберите связанную таблицу или запрос, содержащий данные из связанной таблицы. В списке *Доступные поля* выберите поля, которые должны быть включены в подчиненную форму.

5. Выполните все шаги, предлагаемые *Мастером*.

Подчиненная форма является элементом управления и может быть добавлена в главную форму с помощью *Мастера подчиненных форм*:

1. Откройте *главную форму* в режиме *Конструктора*, выведите на экран *Панель элементов*.

2. Измените размер *Области данных* так, чтобы можно было разместить подчиненную форму. Перетащите мышью в область данных формы элемент управления *Подчиненная форма/отчет*.

3. Выполните все шаги, предлагаемые *Мастером*.

Если главная и подчиненная формы созданы ранее, то разместите рядом окно БД с раскрытой вкладкой *Формы* и главную форму, открытую в режиме *Конструктора*, перетащите мышью нужную форму в главную форму, связь между формами устанавливается автоматически.

Создание многотабличных форм для несвязанных таблиц

Связь форм осуществляется с помощью свойств *Основные поля* и *Подчиненные поля*. *Мастера форм* автоматически заполняют эти свойства, основываясь на связях между таблицами. Если связи не заданы, то связу-

ющими полями считаются поля в таблицах, которые имеют одинаковые имена и совместимые типы.

1. Открыть главную форму в режиме *Конструктора*.
2. Выбрать элемент управления *Подчиненная форма* и открыть окно *Свойства* для этого элемента.
3. В строку свойства *Подчиненные поля* ввести имя связующего поля из таблицы или запроса, на котором базируется подчиненная форма.
4. В строку свойства *Основные поля* ввести имя связующего поля из таблицы или запроса, на котором базируется главная форма.

Если вы не помните имена полей в базовых таблицах или запросах, воспользуйтесь *Построителем* для связывания полей. Нажмите кнопку справа от любого из описываемых свойств, на экране появится диалоговое окно. Выберите связующие поля. Связь может задаваться с помощью нескольких полей. *Построитель* может задать максимум два поля, недостающие поля можно задать непосредственно в строках свойств, разделив их точкой с запятой. Связующие поля не обязательно должны присутствовать в формах, они обязаны присутствовать в базовых таблицах и запросах.

Создание вычисляемых полей

Первый способ создания вычисляемого поля основан на том, что форма строится на базе запроса, в который включают вычисляемые столбцы. В форме создается текстовое поле, у которого в качестве источника данных указывается вычисляемый столбец запроса. Поле блокируется, чтобы пользователь не мог изменить в нем данные. Данные вычисляются в процессе выполнения запроса. Для блокировки поля в режиме *Конструктора* в окне *Свойства* вычисляемого поля на вкладке *Данные* в строке *Блокировка* введите *ДА*.

Во втором случае форма строится на базе таблицы. В ней создается текстовое поле, в котором в качестве источника данных указывается вы-

ражение, например, = [Цена за единицу]*Количество. При ссылке на поля в выражения обязательно использование квадратных скобок, если название поля состоит более чем из одного слова. Ввод выражения допускается прямо в элемент управления (поле), если оно короткое и длина поля это допускает. В противном случае выражение вводится в ячейку свойства *Данные* окна *Свойства*. При этом можно воспользоваться построителем выражений (*Приложение 2*), если выражение не помещается в ячейке свойства, нужно нажать клавиши [Shift]+[F2] и открыть окно **Область ввода** (*Zoom*). При необходимости следует задать формат поля при выводе на экран.

Лабораторная работа 7

СОЗДАНИЕ ОТЧЕТОВ

Цель работы – получить практические навыки разработки различных типов отчетов.

Краткие сведения по среде разработки

Отчеты позволяют выбрать из БД требуемую информацию и оформить ее в виде документов, которые можно просмотреть и напечатать. Источником данных служат таблицы и запросы.

Все отчеты подразделяются на три категории: простая распечатка содержимого БД из режимов таблицы или формы; детальные отчеты; специальные отчеты, которые позволяют подготавливать почтовые наклейки и формы писем.

В Access отчеты можно построить с помощью *средств автоматического создания отчетов* (отчеты типов “в столбец”, “ленточный”), *Мастера отчетов* и *Конструктора отчетов*. Основная работа с отчетом прохо-

дит в режиме *Конструктора*. *Конструктор отчетов* предоставляет такие же средства разработки, что и *Конструктор форм*: панель инструментов, бланки свойств, список полей и панель элементов. Структуры отчетов и форм похожи. Отчеты, как правило, включают группировку записей и вывод промежуточных итогов, поэтому в них могут присутствовать дополнительные разделы *Заголовок группы* и *Примечание группы*.

В *Access* можно строить отчеты сложной структуры: отчеты, включающие подчиненные отчеты; отчеты, построенные на базе перекрестных запросов; отчеты, выводимые в виде нескольких колонок.

Подчиненные отчеты необходимо или предпочтительно использовать, когда:

главный отчет представляет собой оболочку, в которую включаются один или более подчиненных отчетов из несвязанных таблиц; в этом случае главный отчет вообще может быть не связанным с таблицей или запросом и содержать только надписи и вычисляемые элементы;

итоговые данные требуется представить в документе в виде отдельной таблицы наряду с детальными данными;

необходимо создать группы, которые содержат две и более подгруппы данных;

требуется создать отчет с двумя уровнями вложенности отчетов или (и) включить в главный отчет подчиненную форму.

Задание

1. Изучите самостоятельно *средства автоматического создания отчетов* и *Мастер отчетов*.

2. В режиме *Конструктора отчетов* постройте отчет в виде таблицы. В отчет должна входить информация по одной таблице или запросу. Предусмотрите группировку данных и подведение итогов по группе.

3. Создайте составной отчет, содержащий подчиненный отчет.

Последовательность выполнения работы

Создание простого отчета в режиме Конструктора

1. В окне **Базы данных** перейдите на вкладку *Отчеты*, нажмите кнопку *Создать*, выберите *Конструктор отчетов* и укажите имя таблицы (запроса). Щелкните кнопку *ОК*. В результате откроется окно отчета, содержащего три раздела: верхний колонтитул, нижний колонтитул и область данных.

2. Добавьте в отчет раздел *Заголовок отчета* (и при необходимости добавьте раздел *Примечаний*): выполните команду *Вид/Заголовок/Примечание отчета*.

3. По аналогии с формой спроектируйте заголовок отчета.

4. Разместите в разделе *Верхний колонтитул* заголовки колонок отчета: активизируйте панель элементов, выберите элемент *Надпись* и перенесите его в раздел.

5. Оформите раздел вывода данных по группам. Для этого выберите из меню команду *Вид/Сортировка и группировка*, откроется одноименное окно;

в столбец *Поле/выражение* введите из раскрывающегося списка поле (поля), по которым будет осуществляться группировка и сортировка данных;

в правом столбце диалогового окна укажите порядок сортировки;

в разделе *Свойства группы* строках *Заголовок группы* и *Примечание группы*, введите значение *Да*, а в строке *Не разрывать* – *Нет* (запрет разрывать данные, входящие в одну группу, по разным страницам);

в строку *Группировка* введите одно из возможных значений, выбор которого делается из раскрывающегося списка и зависит от типа данных в поле;

в строку *Интервал* следует ввести значение, которое определяется типом данных и значением в строке *Группировка*;

закройте **окно Сортировка и группировка**.

в верхний левый угол области заголовка группы поместите элемент управления, соответствующий полю таблицы, по которому осуществляется группировка записей;

6. Выберите из списка полей поля для отчета, перетащите их в *Область данных* (если список полей отсутствует, то воспользуйтесь командами меню *Вид/Список полей*).

7. Уменьшите размер *Области данных* до высоты размещенных в ней полей, передвигая мышкой ее нижнюю границу.

8. В области *Примечания группы* разместите поле (поля), в котором будет вычисляться итоговое значение; выделите его; откройте окно *Свойств поля*; перейдите на вкладку *Данные*; введите итоговую функцию.

9. Пронумеруйте страницы отчета. Выберите команду *Вставка/Номера страниц*. В появившемся окне **Номера страниц** установите нужные параметры и нажмите кнопку *ОК*.

10. Перейдите в режим *Предварительного просмотра* – *Вид/Предварительный просмотр*. Если структура отчета Вас не устраивает, вернитесь в режим *Конструктора* и внесите изменения в отчет.

11. Сохраните отчет.

Создание составного отчета

1. Создайте запрос, который будет базовым для главного отчета. На рис. 7.1 приведен в качестве примера базовый запрос для отчета “Квар-

тальные обороты” – по запросу будут отбираться все клиенты, которые имеют заказы в указанном году.

Квартальные обороты : запрос на выборку

Клиенты

- * КодКлиента
- Название
- ОбращатьсяК

Заказы

- * КодЗаказа
- КодКлиента
- КодСотрудника

1 — ∞

Поле:	Название	Город	Страна	Год заказа: Year([ДатаРазмещения])
Имя таблицы:	Клиенты	Клиенты	Клиенты	
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				[Отчетный год]
или:				

Рис. 7.1. Базовый запрос для главного отчета

2. Создайте главный отчет на основе только что созданного запроса. Можно использовать *Мастер отчетов*. В данном примере для отчета можно выбрать поля *Название*, *Город*, *Страна*.

3. Подчиненный отчет сформируйте также на базе запроса (в учебной базе “Борей” для рассматриваемого примера используется перекрестный запрос “Квартальные обороты по товарам” (рис. 7.2), для которого заданы условия выборки и введен параметр [Отчетный год:]).

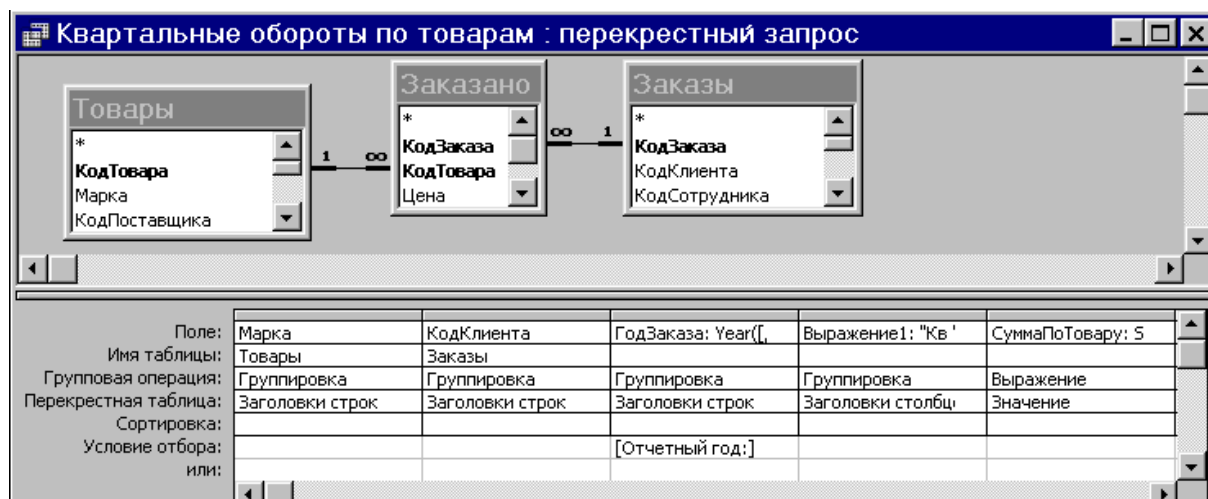


Рис. 7.2. Базовый запрос для подчиненного отчета

4. Выберите элемент управления *Подчиненная форма/отчет* на панели элементов и разместите его в разделе *Область данных* главного отчета ниже полей этого отчета (Кнопка Мастера элементов должна быть нажата).

5. Далее выполните все шаги, предлагаемые *Мастером*.

Лабораторная работа 8

РАЗРАБОТКА КОМАНДНОГО ИНТЕРФЕЙСА ПРИЛОЖЕНИЯ

Цель работы – получить практические навыки разработки пользовательского интерфейса.

Краткие сведения по среде разработки

Обеспечение удобного пользовательского интерфейса является обязательным требованием любого заказа на разработку приложений. Для того, чтобы приложение было законченным, его оснащают такими дополнительными элементами пользовательского интерфейса, как меню команд, панели инструментов, контекстные меню, клавиши быстрого вызова и т.п.

Обычно для работы с приложением создают специальную форму, которая носит название “*Главная кнопочная форма*” и содержит кнопки, позволяющие выбирать основные функции или подсистемы приложения (рис. 8.1.). Такую форму можно создать с помощью *Конструктора форм* или специального *Мастера*, который называется *Диспетчер кнопочных форм*.

Задание

В соответствии с собственным представлением об эстетике пользовательского интерфейса разработайте главную кнопочную форму приложения.

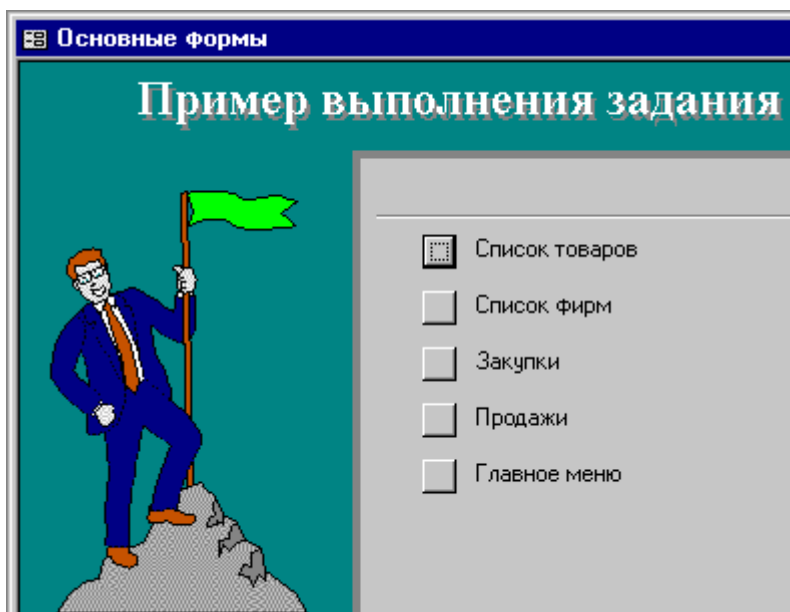


Рис.8.1. Пример главной кнопочной формы

Последовательность выполнения работы

1. Выберите команду меню *Сервис/Настройки/Диспетчер кнопочных форм*.
2. В диалоговом окне, где запрашивается, нужно ли создавать кнопочную форму нажмите *ДА*.

3. На экране появится окно **Диспетчер кнопочных форм**. В этом окне приведен список страниц кнопочной формы. В начале работы в окне одна страница - **Главная кнопочная форма**. В окне нажмите кнопку *Изменить*.

4. Открывается диалоговое окно **Изменение страницы кнопочной формы** (рис.8.2). Если считаете нужным, то в поле *Название кнопочной формы* измените название формы.

5. Нажмите кнопку *Создать*. Появляется окно **Изменение элемента кнопочной формы**.

6. В поле *Текст* введите наименование создаваемой кнопки. Имя кнопки должно ассоциироваться у пользователя с формой, отчетом, макросом и т.п., с которыми он предполагает работать. Второе поле *Команда* позволяет определить действие, которое будет выполняться после нажатия этой кнопки. Выберите из раскрывающегося списка необходимую для вашего приложения команду. В следующем поле введите имя открываемой формы. Нажмите кнопку *ОК*.

Название последнего поля в окне **Изменение элемента кнопочной формы** изменяется в зависимости от выбранной команды во втором поле, либо третье поле может вообще отсутствовать.

Изменение страницы кнопочной формы

Название кнопочной формы:

Главная кнопочная форма

Заккрыть

Элементы данной кнопочной формы:

Создать...

Изменение элемента кнопочной формы

Текст:

Форма Товары

ОК

Команда:

Открыть форму для изменения

Отмена

Форма:

Товары

Рис.8.2. Определение элементов кнопочной формы

7. Аналогично добавьте кнопки для других объектов приложения.
8. Добавьте кнопку, по которой будет осуществляться выход из приложения.
9. Закройте диалоговое окно **Изменение страницы кнопочной формы**, нажав кнопку **Заккрыть**.

Количество кнопок на одной странице формы **Диспетчером** ограничено числом 8. Дополнительные кнопки можно создать в режиме *Конструктора* или разместить их на другой странице. В этом случае на второй странице создайте кнопку, которая позволит вернуться на главную кнопочную форму. Для этого необходимо дать имя кнопки возврата, например, *Назад*, выбрать команду *Переход к кнопочной форме*, в последнем поле ввести **Главная кнопочная форма** (или имя страницы). Аналогично на странице **Главная кнопочная форма** создайте кнопку, вызывающую следующую страницу.

После создания всех необходимых кнопок закройте окно **Диспетчера кнопочной формы**. В окне базы данных перейдите на закладку **Формы** и откройте кнопочную форму.

Перейдите в режим **Конструктора** и оформите внешний вид формы, пользуясь средствами **Конструктора**.

Лабораторная работа 9

СОЗДАНИЕ МАКРОСОВ

Цель работы – познакомиться с основами создания объекта базы данных – МАКРОСА; получить практические навыки создания макросов.

Краткие сведения по среде разработки

Макрос - это последовательность макрокоманд, реализующих практически те же действия, которые можно осуществлять нажатием на клавиши или на кнопки мыши. С помощью макроса можно автоматизировать выполнение однообразных, часто повторяющихся действий. Краткий список макрокоманд, доступных для использования в макросах, приведен в таблице 9.1. Каждая макрокоманда имеет определенное имя и, возможно, один или несколько аргументов, которые задаются пользователем. Для ввода аргументов макрокоманды чаще всего требуется выбирать значения из списков или вводить выражения. Для ввода выражений рекомендуется использовать **Построитель выражений** (Приложение 2).

Таблица 9.1

Макрокоманды, доступные для использования в макросах

Макрокоманда	Назначение
1	2
ОткрытьФорму	Может открыть форму в режиме предварительного просмотра. Возможно применить фильтр и задать условия отбора
ОткрытьЗапрос	Может открыть запрос в режиме предварительного просмотра
ОткрытьТаблицу	Может открыть таблицу в режиме предварительного просмотра.
ВывестиВформате	Выводит заданную таблицу, запрос, форму, отчет или модуль в файл в формате XLS, RTF или TXT, а также запускает соответствующее приложение для редактирования файла.
Печать	Выполняет печать активной таблицы, формы или отчета. Можно задать диапазон печатаемых страниц, число копий и необходимость раскладки копий
КомандаМеню	Выполняет команду главного меню Access
Выход	Закрывает все окна и осуществляет выход из Access
ЗапускМакроса	Запускает другой макрос
ОстановитьВсе-	Останавливает работу всех макросов, включая макрос,

Макросы	который запустил этот макрос
Добавить меню	Добавляет в строке пользовательского меню раскрывающееся меню для формы или отчета. В аргументах данной макрокоманды указывается имя строки меню и имя другого макроса, который содержит все используемые в меню команды и все действия, соответствующие этим командам
КомандаМеню	Выполняет команду одного из главных меню Access. Используя эту макрокоманду внутри строки пользовательского меню, можно выполнить в пользовательском меню имеющиеся в нем команды.
КомандыКлавиатуры	Запоминает последовательность нажимаемых клавиш
ЗадатьЗначение	Изменяет значение любого элемента управления или свойства, которое можно обновить
Сигнал	Выдает звуковой сигнал
Сообщение	Выводит предупреждающее или информационное сообщение

Продолжение таблицы 9.1

Установить Сообщения	Когда она включена, клавиша Enter назначается как клавиша ответной реакции на все системные предупреждающие или информационные сообщения при работе макроса. Чтобы отменить вывод сообщений, надо использовать эту команду вместе с макрокомандой ВыводНаЭкран, установленной в состояние "Нет" (Off)
Вывести на экран	Выводит на экран информацию о промежуточных действиях, выполняемых при работе макроса
Развернуть	Увеличивает размер активного окна до полного размера
Свернуть	Сворачивает активное окно в значок
СдвигРазмер	Изменяет размер активного окна
Восстановить	Восстанавливает окно в прежних размерах

Для работы с макросами в панели инструментов предназначены следующие кнопки (рис. 9.1.):

- кнопка *Имена макросов* (с изображением «XYZ») отображает или убирает столбец «Имя макроса», где можно присваивать имена макросам, входящим в группу;

- кнопка *Условия* (с изображением блок-схемы) отображает или убирает столбец “*Условие*” в окне макроса, где можно задать условие, с которым связано какое-то действие;
- кнопка *Выполнить* (с изображением «!») запускает макрос;
- кнопка *По шагам* (с изображением стрелки и списка) выполняет макрос по шагам;

Для обслуживания некоторого объекта, входящего в базу данных, создается подчас многочисленная совокупность макросов, каждый из которых автоматизирует некоторую вполне конкретную процедуру. Рекомендуется все макросы, связанные с событиями в форме или отчету, объединить в отдельный объект (группу), соответствующий данной форме или отчету. Каждый макрос должен иметь уникальное имя в пределах группы. Имя объекта *Макрос* будет являться именем группы. Вызов макрокоманды из макрогруппы осуществляется точным указанием имени макрогруппы и макрокоманды в группе: *имяГруппы.имяМакроса*.

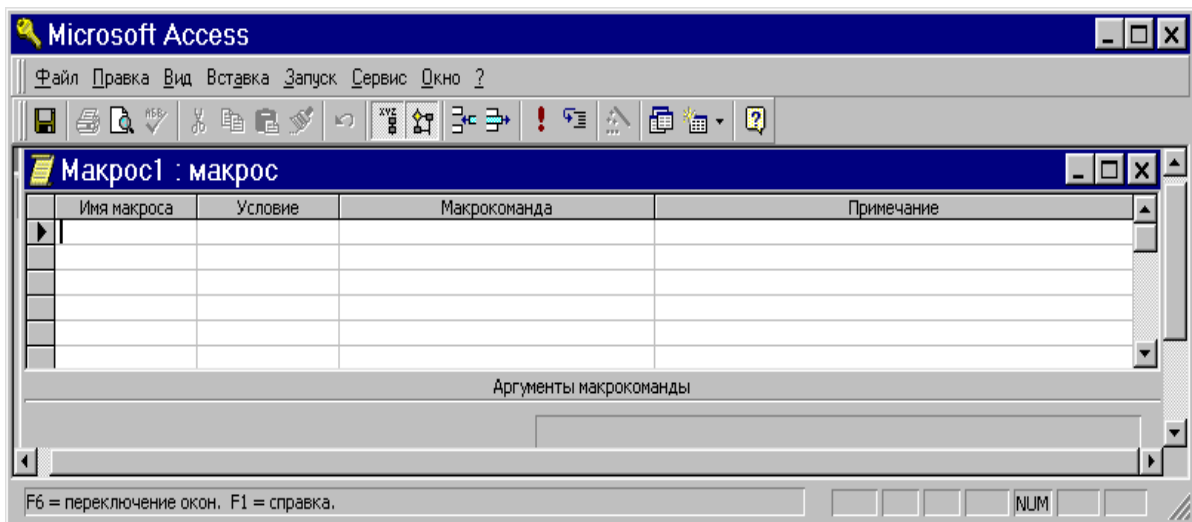


Рис. 9.1. Окно проектирования макроса

Порядок выполнения макрокоманд в макросе может быть изменен. Для этого в макросе вводятся условия выполнения или пропуска макрокоманд. Условия задаются в виде выражения в столбце *Условие*.

В макросах часто происходят ссылки на те или иные объекты БД. Ссылки указываются в столбце *Условие*. Ссылаться можно на объекты, на свойство объекта, на элемент управления в объекте.

Ссылка на форму или отчет состоит из двух частей: *имяСемейства!имяОбъекта*. Все открытые формы входят в семейство *Формы*, открытые отчеты – в семейство *Отчеты*. Например, *Формы![Заказы клиента]* или *Формы!Клиент*; ссылка на отчет – *Отчеты![Отчет о продажах]*. Если имя объекта содержит пробелы или специальные символы, оно обязательно должно быть заключено в квадратные скобки. Ссылка на свойство формы или отчета состоит из трех частей: *имяСемейства!имяОбъекта.имяСвойства*. Например: *Формы!Клиент.Вывод на экран*.

Ссылки на элементы управления форм, отчетов и их свойства:
имяСемейства!имяОбъекта!имяЭлемента (например, *Формы![Заказы клиента]! [Номер клиента]*);
имяСемейства!имяОбъекта!имяЭлемента.имяСвойства (например, *Формы![Заказы клиента]! [Доступ]*).

Задание

1. Создайте макрос, содержащий макрокоманду *ОткрытьФорму*. Форма должна открываться с помощью кнопки, расположенной в другой форме.

2. Создайте группу, включающую наряду с новыми макрокомандами предыдущую макрокоманду. Выбор макрокоманд сделайте по своему усмотрению.

3. Создайте макрос с условиями.

Порядок выполнения работы

Создание макроса

1. В окне **Базы данных** щелкните по вкладке *Макросы* и нажмите на кнопку *Создать*. Откроется окно проектирования макроса аналогичное окну, приведенному на рис. 9.1. (будут отсутствовать столбцы *Имя макроса* и *Условие*). Каждая строка в этом окне может содержать одну макрокоманду и комментарий к ней.

2. В строку, помеченную указателем, введите *Макрокоманду*. Команду можно вводить с клавиатуры или воспользоваться списком, который доступен в ячейках колонки *Макрокоманда*.

3. Желательно ввести пояснения в столбец *Примечание*. Комментарии облегчают понимание и сопровождение макросов.

4. Когда поле *Макрокоманда* заполнено, в нижней части *Конструктора макроса* появится *панель аргументов*. Список полей этой панели зависит от выбранной макрокоманды и может отсутствовать, если макрокоманда не имеет аргументов. Укажите всю требуемую системой информацию в области аргументов макрокоманды. В общем случае рекомендуется задавать аргументы макрокоманды в том порядке, в котором они перечислены, поскольку выбор одного аргумента может определять возможные значения следующего аргумента.

5. Сохраните макрос. Для этого можно воспользоваться кнопкой *Сохранить*, либо выполните команду *ФАЙЛ/Сохранить*. Присвойте имя макросу (имя должно нести смысловую нагрузку).

6. Запустите макрос на исполнение.

Вызвать и выполнить макрос можно одним из нескольких способов:

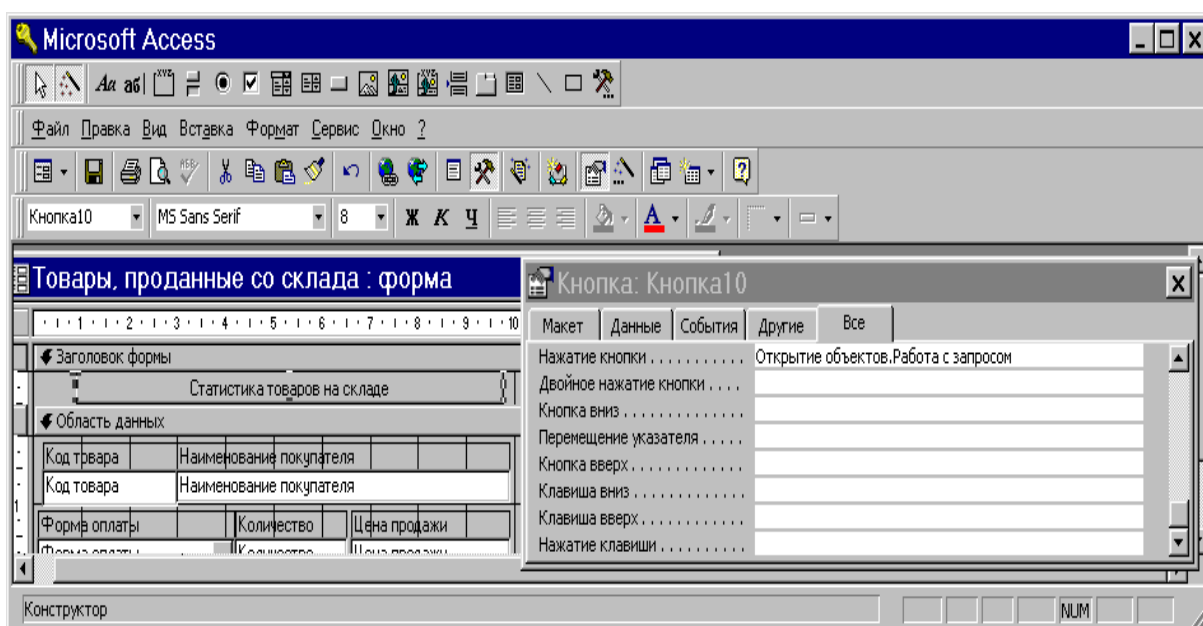
- в режиме проектирования макроса (см. рис. 9.1.) щелкнуть по кнопке *Выполнить*;
- выполнить команду *СЕРВИС/Запуск макроса* и выбрать из списка имя макроса для выполнения;
- в окне **Базы данных** на вкладке *Макросы* выполнить двойной щелчок на имени подлежащего выполнению макроса.

7. Откройте в *режиме Конструктора* какую-либо форму, отличную от формы, для которой разрабатывается макрос.

8. Создайте кнопку, разместив его в области заголовка открытой формы. Откройте окно свойств кнопки (рис. 9.2): выделите кнопку, нажмите правую кнопку мыши, в появившемся меню выберите *пункт Свойства* (открыть окно свойств можно, дважды щелкнув по кнопке). В окне свойств укажите, с каким событием связывается макрос. По умолчанию это событие *Нажатие кнопки*.

Создание групп макросов

Объедините макросы в макрогруппу (можно сделать новый объект или преобразовать ранее созданный макрос в группу макросов).



1. Откройте *Конструктор макросов*.

2. Нажмите кнопку *Имена макросов* на панели инструментов. В окне *Конструктора макросов* появится еще один столбец *Имя макроса*.

3. Введите в этот столбец имя первого макроса. В этой же строке в столбце *Макрокоманда* введите имя макрокоманды, далее все соответствующие ей аргументы. Если предполагается, что макрос включает несколько макрокоманд, то введите их в последующих строках столбца *Макрокоманда*.

4. Пропустите строчку.

5. Введите имя следующего макроса и соответствующие ему макрокоманды. На рис. 9.3, в качестве примера, представлен макрос “Поставщики” БД “Борей”.

6. Проверьте работу макроса. Выполните команду *СЕРВИС/Запуск макроса* (будет выполнен только первый макрос из макрогруппы). Вызов других макросов из группы возможен, если в диалоговом окне ввести имя макрогруппы и через точку имя макроса.

7. Сохраните макрос.

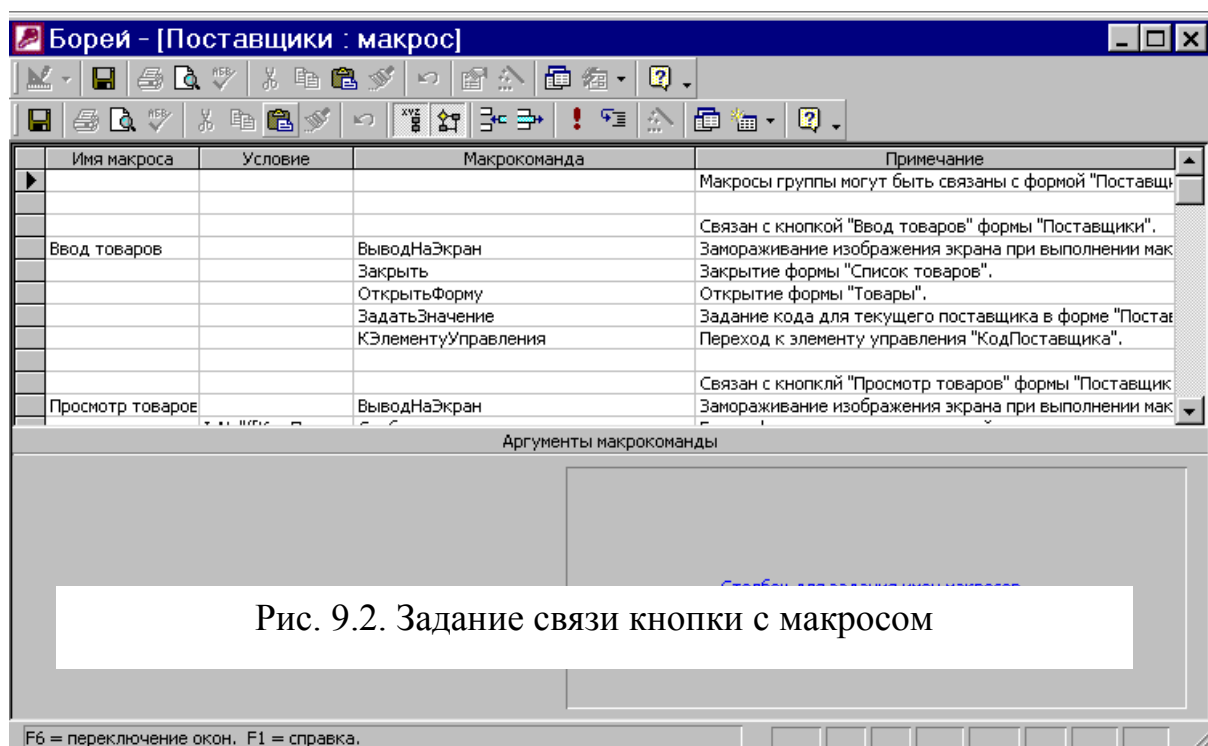


Рис. 9.2. Задание связи кнопки с макросом

Рис. №4. Структура макрогруппы
Рис. 9.3. Создание группы макросов

Создание макроса с условием

Для примера на рис. 9.4 представлено окно проектирования макроса, который будет выводить на экран таблицу о товарах, находящихся на складе фирмы. Таблица *“Товары на складе”* будет открываться лишь тогда, когда в поле *Цена* формы *“Товары, проданные со склада”* будет внесено значение, меньше 1000. Макрос связан с элементом управления *Цена* в форме *“Товары, проданные со склада”*. Подключение этого макроса к элементу управления *Цена* происходит также, как и подключение макроса кнопки. Макрос будет связываться либо с событием *Выход*, либо с событием *Вход*.

1. Откройте *Конструктор макросов*.
2. Нажмите кнопку *Условие*. Запишите в столбце *Условие* выражение, при истинности которого будет выполняться создаваемый макрос. Выражения для условия удобнее задавать в окне **Построитель выражений**.
3. Задайте макрокоманду.
4. Сохраните макрос.
5. Откройте в *режиме Конструктора* форму, в которой предполагается один из элементов управления связать с разрабатываемым макросом. Свяжите элемент управления и макрос.
6. Проверьте работу макроса.

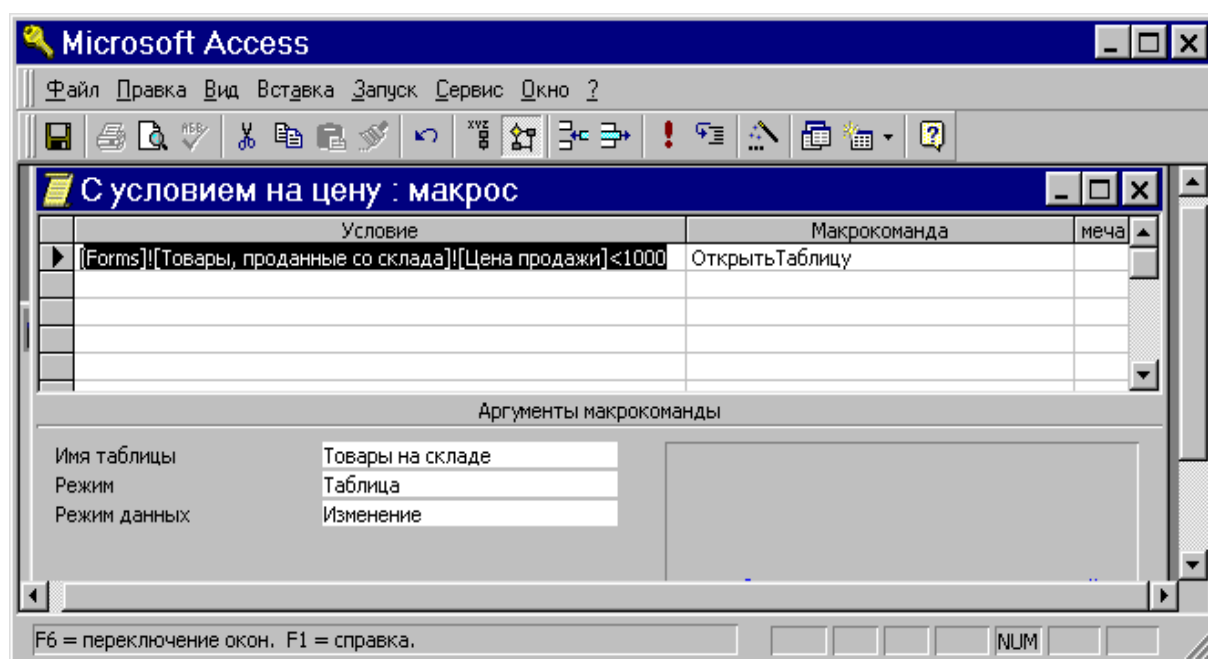


Рис. 9.4 Пример макроса с условием

Приложение 1

Статистические функции

Функция	Описание
Sum	Вычисляет сумму всех значений заданного поля в каждой группе. Используется только для числовых и денежных поле.
Avg	Вычисляет среднее арифметическое всех значений заданного поля в каждой группе. Используется только для числовых и денежных поле. Значения Null не включаются в вычисления.
Min (Max)	Возвращает наименьшее (наибольшее) значение, найденное в заданном числовом поле внутри каждой группы. Для текстовых полей - наименьшее из символьных значений независимо от регистра. Значения Null игнорируется.
Count	Возвращает число записей, в которых значения данного поля отличны от Null. Для учета значений Null используется Count(*)
StDev*	Вычисляет смещенное статистическое стандартное отклонение для всех значений данного поля в каждой группе. Применяется к числовым и денежным полям. Если в группе меньше двух строк, возвращает значение Null.
StDevP*	Вычисляет несмещенное статистическое стандартное отклонение
Var*	Вычисляет смещенную статистическую дисперсию для всех значений данного поля в каждой группе. Применяется к числовым и денежным полям. Если в группе меньше двух строк, возвращает значение Null.
VarP*	Вычисляет несмещенную статистическую дисперсию
First*	Возвращает первое значение в заданном поле.
Last*	Возвращает последнее значение в заданном поле.

Примечание: функции, отмеченные *, в *Мастере запросов* не применяются.

Приложение 2

Построитель выражений

Построитель выражений – средство построения сложных выражений. Структура окна **Построителя выражений** представлена на рис. П.2.1.

Текстовое поле для ввода выражения

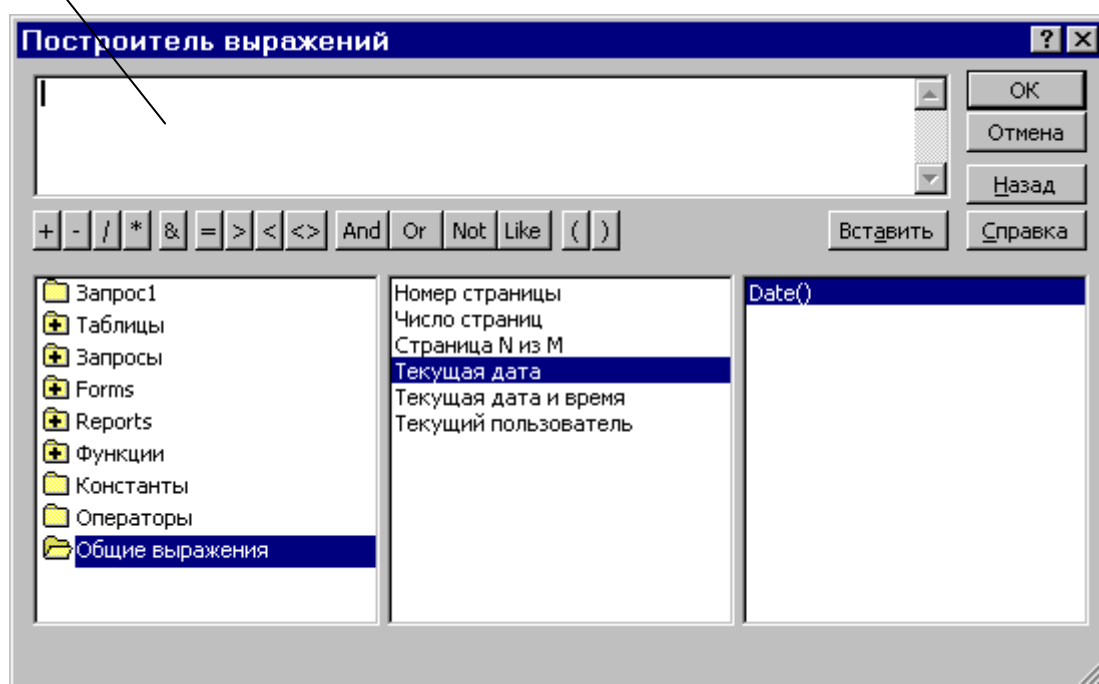


Рис. П. 2.1. Окно **Построителя выражений**

Выражение – совокупность операторов и операндов (константы, идентификаторы и функций).

Идентификаторы – имена объектов (например, полей таблиц или запросов), которые при вычислении выражений заменяются их текущими значениями. Если имя объекта состоит более чем из одного слова, то его обязательно заключают в квадратные скобки. Однако обычно скобки ставят даже, если имя объекта состоит из одного слова, чтобы имена объектов сразу бросались в глаза при работе с выражениями.

Идентификаторами являются имена констант и переменных, применяемые в программах на языке Visual Basic для приложений, а также именованные константы True, False, Yes, No, Null.

Функции – возвращают в выражение значение вместо имени функции. Большинство функций требует заключать в скобки свои аргументы. Пример функции с пустым списком аргументов: *Date ()* – возвращает текущую дату.

Для создания выражений существует шесть категорий операторов, которые представлены в таблице П. 2.1

Таблица П. 2.1

Операторы для создания выражений

Категория операторов	Оператор	Пример	Описание
1	2	3	4
Арифметические операторы	+	[Итог]+[Надбавка]	Складывает два операнда
	-	Date() – 7	Считает разность двух операндов
	*	[Коробок]*[Цена]	Перемножает два операнда
	/	[Количество] / 5.33	Делит один операнд на другой
	\	[Количество] \ 2	Делит один целый операнд на другой нацело. При этом операнды с десятичными дробями округляются до целого
	Mod	[Коробок] Mod 3	Возвращает остаток от деления нацело
Операторы сравнения ¹	^	[A] ^ [b]	Возводит операнд A в степень b
	<	1 < 100	Меньше
	<=	[a] <= 2	Меньше либо равно
	=	[b] = 3	Равно
	>	[a] > 2	Больше
	>=	[b] >= 3	Больше либо равно
Логические операторы ²	<>	1 <> 100	Неравно
	And	[A] And [B]	Конъюнкция (логическое И)
	Or	[A] Or [B]	Дизъюнкция (логическое Или)
	Not	Not [A]	Логическое отрицание
	Xor	[A] Xor [B]	Исключающее ИЛИ
	Eqv	[A] Eqv [B]	Логическая эквивалентность
Операторы конкатенации	Imp	[A] Imp [B]	Логическая импликация
	+	“Visual “+”Basic”	Объединение двух текстовых значений в единую строку. Оба операнда эквивалентны, но & - предпочтительнее
	&	“Visual “ & ”Basic”	
Операторы идентификации	!	КлассОбъекта! ИмяОбъекта	Разделитель в ссылках на объекты
	.	ИмяОбъекта.Метод	Разделитель в ссылках на методы и свойства объектов
Операторы сравнения с образцом	Between	Between(-100) And (100)	Определяет, находится ли числовое значение в определенном диапазоне

Продолжение таблицы П.2.1.

1	2	3	4
	Is	[Имя] Is Null	Используется только для сравнения со значением. В примере выражение принимает истинное значение, когда в поле Имя нет никаких данных.
	In	In ("Москва", "Киев", "Мурманск")	Определяет, является ли строковое значение элементом списка значений
	Like	Like "Ив*" Like "db??"	Определяет, начинается ли строковое значение с указанных символов (символ "*" замещает любое число знаков, "?" – замещает только один знак)

Примечания:

1. Результат применения операторов сравнения – True или False. Если один из операндов имеет значение Null (пустое значение), то любое сравнение возвращает значение Null.
2. Результат применения логических операторов – True или False. Операнды имеют значение из множества (True, False, Null). Для всех логических операторов, кроме Imp, значение одного из операндов Null приводит к значению результата Null.

Элементы выражения вводятся в текстовом поле **Построителя выражений**. Наиболее употребительные операторы представлены кнопками в средней части окна. Элементы выражений можно выбирать из списков объектов, функций, констант, операторов и общих выражений.















Чтобы скопировать элементы выражения из списков, сначала щелкните на одной из папок в левой панели. Некоторые папки имеют дочерние папки, которые также раскрываются двойным щелчком. Список элементов, соответствующий открытой папке, отображается в правой панели.

В некоторых случаях центральная панель также включает элементы, которые можно добавить в выражения (например, идентификаторы полей таблицы).







Чтобы добавить элемент из списка в выражение текстового поля, можно сделать двойной щелчок на нем или выбрать его и щелкнуть по кнопке **Вставить**.

Приложение 3

Кнопки панели элементов ACCESS

Кнопка	Название кнопки	Функция
1	2	3
	Выбор объекта	Превращает указатель мыши в инструмент выборки объектов
	Мастера	Включает и отключает мастеров создания элементов управления
	Надпись	Текст заголовков, подписей или пояснений, этот текст остается неизменным
	Поле	Создает поле для отображения и редактирования текстовых данных
	Группа	Создает рамку для размещения выключателей, переключателей, флажков. Одновременно может быть выбран только один из объектов, помещенных в рамку. Каждому элементу рамки соответствует уникальное число
  	Выключатель Переключатель Флажок	Выключатель, переключатель, флажок, не включенные в группу, определяют значение логического типа Да (<i>Yes</i>) или Нет (<i>No</i>), допускается использовать Истина (<i>True</i>) или любое положительное число, Ложь (<i>False</i>) или ноль
	Поле со списком	Создает поле со списком, содержащее поле для ввода значения, и список, в котором можно выбрать значение из набора значений
	Список	Создает раскрывающийся список для выборки значений
	Кнопка	Создает кнопку, после щелчка на которой запускается процедура обработки событий Access VBA
	Рисунок	Добавляет в форму или отчет точечный рисунок. Рисунок не относится к числу объектов OLE, после размещения в форме его изменение невозможно
	Свободная рамка	Добавляет в форму или отчет объект OLE
	Присоединенная рамка	Отображает содержимое поля OLE записи, если в поле имеется графический объект, иначе отображается значок объекта

Продолжение прил. 3

1	2	3
	Разрыв страницы	При печати формы или отчета заставляет принтер начинать новую страницу в месте разрыва страницы. В режиме просмотра формы или отчета разрыв не отображается
	Набор вкладок	Вставляет набор вкладок для создания формы, разделенной вкладками.
	Подчиненные форма/отчет	Добавляет подчиненные форму/отчет в главную форму/отчет
 	Линия Прямоугольник	Применяются для разделения формы на логические группы для имитации рамок, границ. Цвет, ширину линий, заливку прямоугольника устанавливают с помощью панели инструментов форматирования
	Дополнительные элементы	

Приложение 4

Список контрольных вопросов

1. Каков порядок определения структуры БД в СУБД ACCESS?
2. В каком режиме описывается структура таблицы БД?
3. Дайте определения терминам: “объект”, “свойство”, “метод”, “событие”.
4. Какие типы объектов можно увидеть в окне БД ACCESS?
5. Какое расширение принято для файлов БД?
6. Можно ли изменить структуру таблицы после ее создания?
7. Какие типы полей БД предусмотрены в ACCESS?
8. Каковы правила именования полей БД?
9. Какая информация может храниться в полях типа "объект OLE"?
10. Какие свойства поля должны быть определены обязательно?
11. Поясните назначение первичного ключа в реляционной БД. Какие ограничения накладываются на значения первичного ключа?
12. Поясните назначение индексов.
13. Можно ли создать индекс по части поля? Если “да”, то как это сделать?
14. Какие виды контроля данных могут автоматически выполняться в СУБД ACCESS?
15. Для чего необходимо описывать связи между таблицами базы данных?

16. В каком режиме следует описывать связи между таблицами?
17. Какие типы связей поддерживает СУБД ACCESS?
18. Какое влияние оказывает установка связей на ввод и изменение информации в БД?
19. В чем состоят задачи ввода и коррекции данных в БД?
20. Какие типичные операции включает задача коррекции данных?
21. Какие приемы контроля данных имеются в ACCESS?
22. Когда производится проверка данных?
23. Как добавить в таблицу новую запись?
24. Как удалить запись из таблицы?
25. Можно ли восстановить удаленную из БД информацию?
26. Как найти в таблице нужное значение?
27. Как заменить во всех записях БД название фирмы ТОО "Заря" на АО "Рассвет"?
28. Что такое шаблон поиска, какие символы и для чего применяют в шаблоне?
29. С какой целью и как производится сортировка записей БД?
30. Как ведут себя данные в других полях таблицы при сортировке по полю "Фирма" ?
31. Как отменить сортировку данных?
32. Что такое запрос и какие типы запросов поддерживает в ACCESS?
33. Как задать параметры при создании запроса-выборки с параметрами?
34. Что такое агрегированные (групповые) операции и как они применяются в запросах?
35. Как определить из каких таблиц взяты поля в шаблон вывода при конструировании запроса?
36. Что такое перекрестный запрос и как он конструируется?
37. К какому классу языков относится SQL? Как язык SQL используется при формировании запросов?
38. Какова структура запроса на языке SQL, содержащего подзапросы?
39. Можно ли передать параметры в запрос из некоторой формы, если можно, то как это сделать?
40. Можно ли вводить данные в таблицы через запросы, если можно, то в каких случаях, каковы ограничения, накладываемые на взаимосвязанные таблицы, на ввод данных в через запросы в таблицы?
41. Что такое поля группировки, можно использовать одновременно несколько полей группировки в одном запросе и каков смысл получаемого результата?
42. Что представляет из себя запрос на создание таблицы, как он работает, какую таблицу можно создать?
43. Как конструируются запросы на изменение в таблицах БД? Можно ли в одном запросе включать изменения в нескольких таблицах?

44. Что такое запросы на удаление, как они конструируются? Для чего могут быть использованы запросы на удаление? Могут ли быть запросы данного типа с параметрами.?
45. Что такое вычисляемые поля в запросах, для чего они нужны и как они конструируются?
46. Могут ли вычисляемые поля участвовать в формировании условия отбора в запросе?
47. Как сохранить результат выполнения запроса?
48. Что такое видимые или выводимые и невидимые или невыводимые поля запроса? Зачем нужны невидимые поля запроса?
49. Как в построителе выражений можно увидеть поля конкретной таблицы, поля данного запроса? Почему при конструировании запроса не все поля видны в построителе выражений?
50. Сколько баз данных может быть открыто одновременно?
51. Какие свойства имеет объект “таблица”?
52. Назначение объекта “форма”. Какие свойства имеет объект “форма”?
53. Позволяет ли ACCESS создавать многоэкранные формы?
54. Можно ли пользоваться одной и той же экранной формой для ввода информации в разные файлы БД?
55. Как внедрить в форму подчиненную форму?
56. Назначение объекта “отчет”. Какие свойства имеет объект “отчет”?
57. Можно ли получить отчет, содержащий только итоговые строки? Если “Да”, то как?
58. Можно ли изменять длину и формат поля в выходном документе по сравнению с тем, который был задан в файле БД?
59. Что такое макрос? Назначение макросов.
60. Какие режимы допускает окно БД для таблиц, запросов и форм? Чем они отличаются от допустимых режимов для макросов?
61. Как выявить ошибки в работе макроса?
62. Можно ли в ACCESS использовать данные, которые хранятся в других форматах?
63. Как создать таблицу путем импорта данных из рабочего листа EXCEL?