

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ
И РАДИОЭЛЕКТРОНИКИ

Кафедра экономической информатики

В. Н. Комличенко, И. Г. Орешко

Лабораторный практикум
по курсу

«Базы данных. Технология организации хранения и обработки данных»

для студентов экономических специальностей БГУИР

В 2-х частях

Часть 1

Минск 2002

УДК 002.5+681.3 (075)

ББК 32.97+65.39 я 73

Л 12

Авторы:

Л 12 Лабораторный практикум по курсу «Базы данных. Технология организации хранения и обработки данных». для студентов экономических специальностей БГУИР

В 2-х ч. ч 1/. В. Н. Комличенко, И. Г. Орешко. - Мн.: БГУИР, 2002 - 42 с.: ил.

ISBN 985-444-338-8 (ч.1)

В практикуме представлен курс из четырех лабораторных работ, даны краткие теоретические сведения, примеры и варианты заданий для лабораторных работ.

Часть 1. Лабораторный практикум по курсу «Базы данных. Технология организации хранения и обработки данных» для студентов экономической специальности. В 2-х ч. Ч. 1. / В. Н. Комличенко, И. Г. Орешко. - Мн.: БГУИР, 2002, с. 48: ил.

УДК 002.5+681.3 (075)

ББК 32.97+65.39 я 73

ISBN 985-444-338-8 (ч.1)

ISBN 985-337-X (ч.2)

© Коллектив авторов, 2002

Содержание

Лабораторная работа №1

Проектирование структуры базы данных и создание таблиц

Лабораторная работа №2

Проектирование запросов для управления данными

Лабораторная работа №3.

Проектирование форм представления и управления данными

Лабораторная работа №4.

Макросы MS Access

Список литературы

Библиотека БГУИР

Лабораторная работа №1

Проектирование структуры базы данных и создание таблиц

Цели: получить навыки проектирования баз данных;

изучить основные приемы создания реляционных таблиц средствами MS Access.

Задача. Предположим, на некотором предприятии требуется автоматизировать складской учет. Для этого, прежде всего, необходимо создать базу данных, в которой будут содержаться все необходимые сведения о товарах, заказах, поставщиках, сотрудниках, а также сведения о движении товаров, выполнении заказов и др. На основании этих данных можно строить системы управления товарооборотом, контроля выполнения заказов, оптимизации транспортных потоков, автоматизированные системы складского учета и др.

Основные понятия баз данных

Под базой данных (БД) понимают совокупность сведений, логически связанных таким образом, чтобы составлять единую совокупность данных, хранимых в запоминающих устройствах вычислительной машины. Эта совокупность выступает в качестве исходных данных задач, решаемых в процессе функционирования автоматизированных систем управления, систем обработки данных, информационных и вычислительных систем.

Следует различать различные способы представления данных. **Физические данные** – это данные, хранящиеся в памяти ЭВМ.

Логическое представление данных соответствует пользовательскому представлению физических данных. Различие между физическим и соответствующим логическим представлением данных состоит в том, что последнее отражает некоторые важные взаимосвязи между физическими данными.

Обзор моделей данных

Самым естественным способом представления данных для пользователя не-программиста является двумерная таблица. А так как доказано, что любая сетевая структура, с некоторой избыточностью, может быть представлена в виде совокупности древовидных структур, а те, в свою очередь, тоже с некоторой избыточностью, могут быть описаны плоскими таблицами, то табличное представление данных стало доминирующим в современных СУБД. Таблица – это прямоугольный массив данных, который можно описать математически.

Таблицу, в терминологии реляционных БД, называют *отношением*. Каждый столбец таблицы является *атрибутом*. Значения атрибута выделяются из домена (множества допустимых значений атрибута). Число столбцов называется *степенью* (арностью) отношения, а число его строк – его *мощностью* (или кардинальным числом). Строки отношения называются *кортежами*. Один или ряд столбцов отношения называют возможным *ключом отношения*, если их значения однозначно идентифицируют строки таблицы. Если таких наборов больше одного, то один из них рассматривают в качестве *первичного ключа*.

Понятие нормализации

Процесс выявления объектов и их взаимосвязей с помощью концепций реляционной модели и табличной формы представления называется процессом **нормализации**.

Процесс нормализации таблиц позволяет привести их к виду, удовлетворяющему основным необходимым свойствам реляционных структур (отношению) :

- Все столбцы таблицы однородны.
- Каждому столбцу присвоено уникальное имя.
- Все столбцы атомарны, т. е. отношение не может иметь в качестве компонента другое отношение
- Отсутствуют одинаковые кортежи (строки), каждая строка имеет уникальный идентификатор (ключ)
- Все кортежи имеют одну и ту же структуру
- В операциях с таблицами все строки и столбцы могут просматриваться в любой последовательности безотносительно к информационному содержанию и смыслу.

В физическом представлении каждому отношению (таблице) соответствует файл БД. Реляционная БД реализуется одним или несколькими *отношениями*. Связь между кортежами разных отношений (для совместной обработки) реализуется с использованием одного и того же ключа в различных отношениях, помещением ключа одного отношения в качестве атрибута кортежа другого отношения, созданием специальных связующих отношений.

Проектирование БД

Основные понятия и термины

Предметная область - часть реального мира, которая описывается или моделируется с помощью БД и разрабатываемого программного обеспечения.

Это может быть: предприятие, его некоторая функциональная часть, процесс, система и т.д. Модель предметной области создается с использованием понятий информационных объектов и функций, выполняемых этими объектами или для них.

Информационный объект - идентифицируемый объект реального мира, некоторое понятие или процесс, относящееся к предметной области, о котором хранятся описательные данные (люди, счета, изделия, события и т.д.).

Информационный объект описывается совокупностью заданных элементов данных (характеристик объекта), которым присваиваются конкретные значения (символы, числа, коды).

Элемент данных - характеристика объекта, которая определяется именем и одним или совокупностью некоторых значений (величин).

Именами элемента данных могут быть, например, номер работающего, дата рождения, цена, квалификация и т.д. Значение элемента данных определяет эту характеристику для конкретного экземпляра информационного элемента.

Совокупность таких значений элементов данных, описывающих конкретный экземпляр объекта, представляют собой *запись*. Например, если объект ИЗДЕЛИЕ описывается элементами данных: КОД ИЗДЕЛИЯ, НАИМЕНОВАНИЕ ИЗДЕЛИЯ, МАТЕРИАЛ, СТОИМОСТЬ, КОЛИЧЕСТВО, то совокупность значений этих элементов для конкретного изделия и представляет запись.

Набор записей - это множество записей об объекте для всех экземпляров данного типа объектов, например, для всех экземпляров (производимых изделий) объекта ИЗДЕЛИЕ.

Поиск отдельных записей осуществляется по содержанию идентифицирующего поля (идентификатора).

Идентификатор - это элемент данных (или совокупность элементов), значение которого используется для определения одного или нескольких значений связанных с ним других элементов.

Идентификация может быть *уникальной* (однозначной) или *неуникальной* (многозначной). Если идентификация однозначна, то говорят, что идентифицируемый элемент (атрибут) является *функционально зависимым* от ключа (идентифицирующего элемента). Например, можно предположить, что для объекта ИЗДЕЛИЕ значение элемента КОД_ИЗДЕЛИЯ однозначно идентифицирует значения других элементов: НАИМЕНОВАНИЕ_ИЗДЕЛИЯ, МАТЕРИАЛ и пр.,

Ключ является идентификатором, уникально определяющим запись об объекте. Ключ в наборе записей может включать в себя один (простой ключ) или более (составной) элементов данных, совокупность которых однозначно идентифицирует запись об объекте.

Элементы данных, не входящие в состав ключа, называются **атрибутами**.

В записи об объекте значение атрибутов идентифицируется значением ключа.

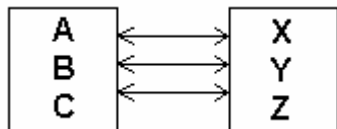
В реляционных моделях баз данных набор записей представляется в виде отношения (двумерной таблицы), в котором кортежи (строки таблицы) соответствуют экземплярам объекта, а атрибуты (колонки таблицы) определяют набор и значение его характеристик (элементов данных).

Связи данных

Традиционными средствами для представления характера взаимосвязей между парами связанных элементов данных являются *отображения* (двусторонние связи) и *ассоциации* (односторонние связи).

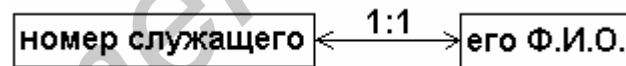
Отображения

Рассмотрим основные типы отображений и их изображение на диаграммах.

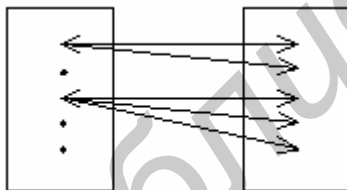


уникальна как слева направо, так и в обратном направлении.

Отображение 1:1. Один экземпляр элемента, от которого направлена связь, идентифицирует один экземпляр элемента, к которому направлена связь. Связь



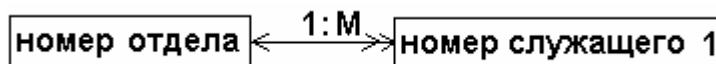
Пример.



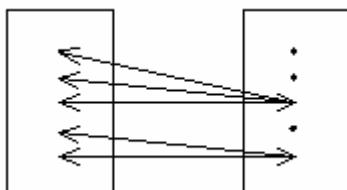
которого направлена связь, т.е. в этом направлении связь уникальна.

Отображение 1:M. Один экземпляр элемента с выходящей связью идентифицирует несколько экземпляров элемента данных, принимающих эту связь.

Подчеркнем, что в направлении слева направо связь не является уникальной. Однако в обратном направлении любой экземпляр элемента, принимающий связь идентифицирует только один экземпляр объекта, от

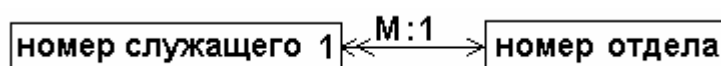


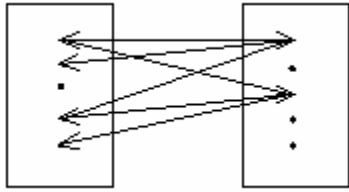
Пример:



Отображение M:1. Это отображение подобно предыдущему с той лишь разницей, что слева направо связь уникальна, а в обратном направлении не является таковой.

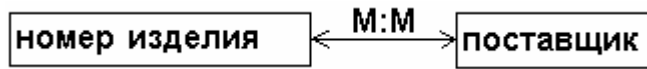
Пример:





Отображение М:М. Каждый экземпляр элемента данных с выходящей связью идентифицирует некоторое число экземпляров элемента данных, принимающих эту связь, и наоборот. Другими словами идентификация не является уникальной в обоих направлениях.

Пример:



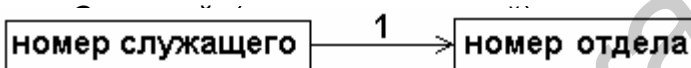
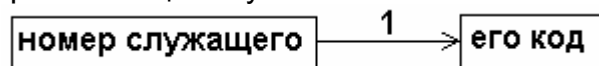
Ассоциации

Для определения связи *ключ - атрибут* чаще всего используются *ассоциации* (односторонние связи между парами элементов данных), поскольку в большинстве случаев в БД связи от атрибутов к ключам не определяются. Ассоциации, определенные в обе стороны, представляют собой *отображения*.

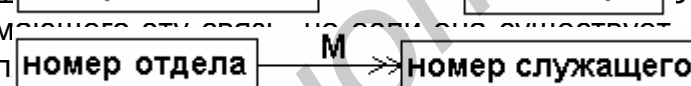
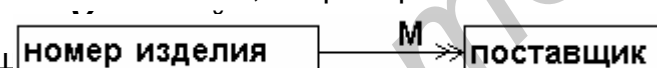
Существует три типа ассоциаций:

- простая (тип 1);
- сложная (тип М);
- условная (тип С).

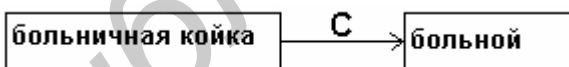
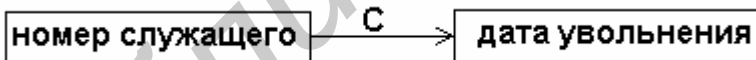
Простой называется ассоциация, в которой экземпляр элемента данных с выходящей связью идентифицирует один, и только один, экземпляр элемента данных, принимающих эту связь.



Сложная ассоциация, в которой экземпляр идентифицирует несколько (один или более) экземпляров элементов данных, к которым эта связь направлена. Идентификация не обязательно является уникальной и может представлять многозначную зависимость. На диаграмме сложную ассоциацию будем отображать двойной стрелочкой и помечать символом М, например:



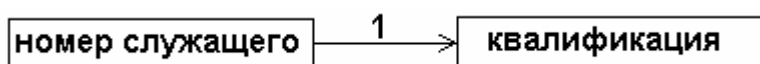
Условная ассоциация, в которой для элемента данных с выходящей связью существует элемент данных, принимающий эту связь, но сам элемент данных не существует. Это относится к единственному экземпляру, который существует, является уникальной, например.



Реляционные ключи

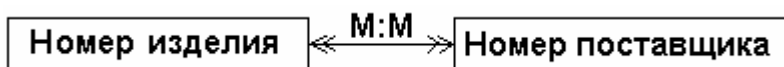
Введем определение реляционных ключей. **Простым ключом** будем называть ключ, состоящий только из одного элемента. Его значение должно быть уникальным в отношении (таблице):

В отношении "номер служащего - квалификация" ключ состоит из единственного поля "номер служащего", а каждое значение данного поля является уникальным в отношении и однозначно определяет аргумент "квалификация".



Составной ключ - ключ, который содержит два или более элемента данных, каждый из которых необходим для однозначной идентификации объекта.

Полностью составной ключ - ключ, содержащий несколько атрибутов, между которыми существует отображение M:M, а одиночная ассоциация типа M или ассоциация вообще отсутствует. Атрибуты, составляющие такой ключ, не зависят друг от друга, ни один из них не является дополнительным квалификационным признаком другого атрибута, например:



Полностью составной ключ включает оба атрибута, т.е. ключ состоит из двух полей "номер_изделия" и "номер_поставщика". Этим достигается однозначная идентификация объекта.

Полусоставной ключ - ключ, содержащий несколько атрибутов и построенный с использованием отображения M:1 (которые подразумевают функциональную зависимость). Атрибуты такого ключа можно считать упорядоченными в том смысле,

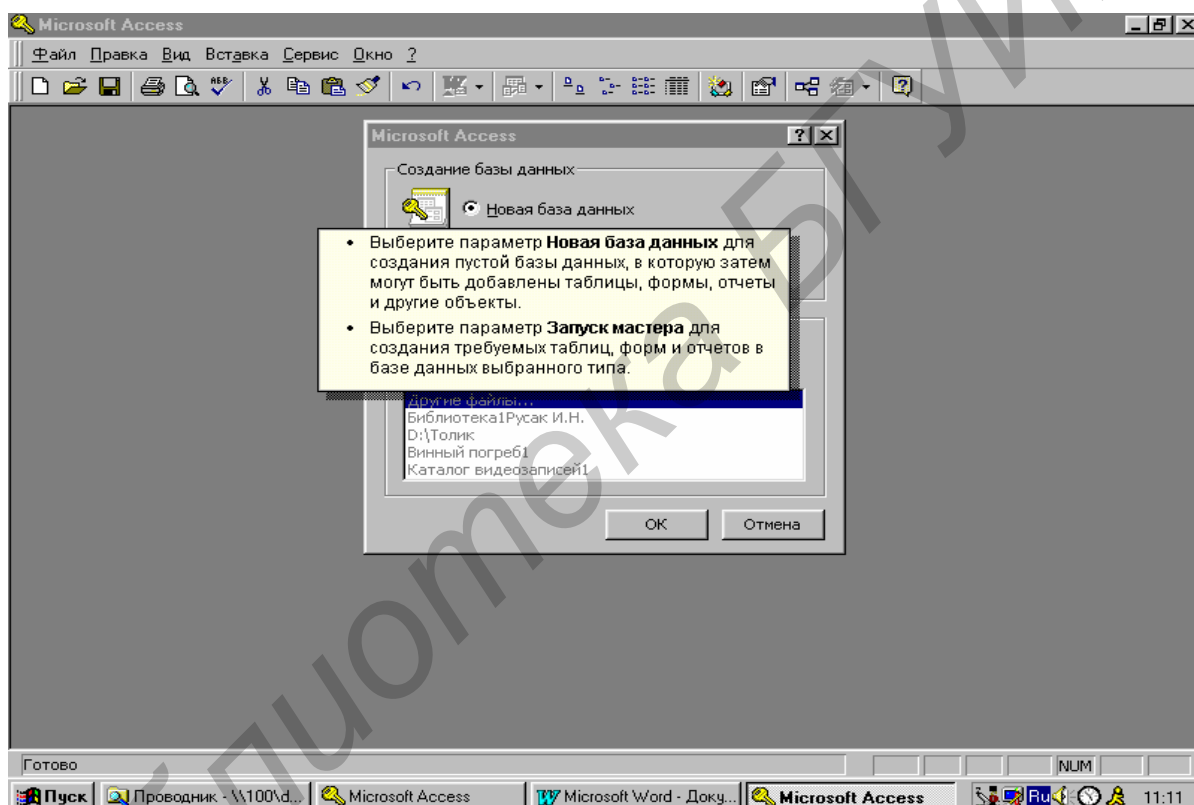


Рис. 1.1 Окно начального диалога в MS Access

что каждый следующий атрибут в ключе является дополнительным квалификационным признаком предшествующих атрибутов. Например, для следующей структуры:

полусоставной ключ будет включать оба атрибута "номер_отделения" и "номер_служащего", что позволяет добиться однозначной идентификации записи в отношении.

Общие сведения о MS Access

Работу с Access рассмотрим на примере создания базы данных для решения поставленной задачи.

Пример 1

Создать базу данных "Склад", содержащую сведения о товарах, поставщиках, заказах, а также сведения о предприятии.

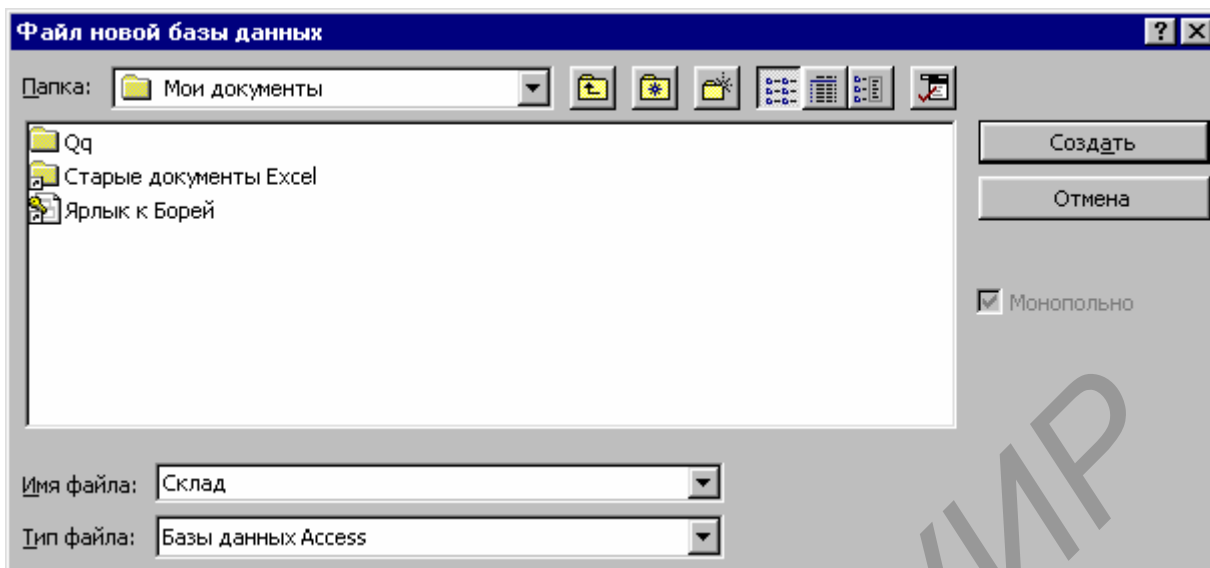


Рис. 1.2 Окно "Файл новой базы данных"

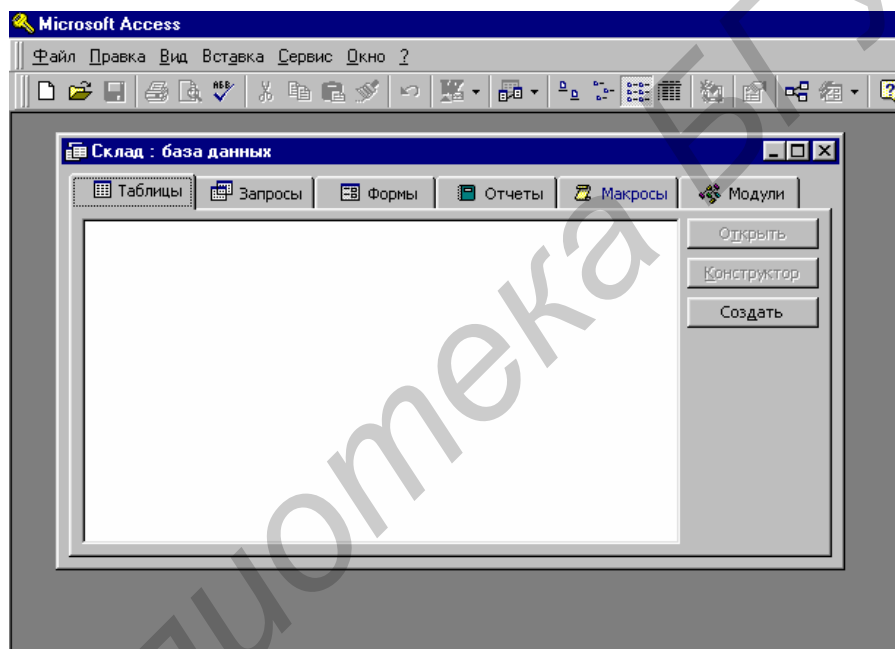


Рис. 1.3. Окно базы данных

Для того чтобы создать новую базу данных, после запуска СУБД Access в **Окне начального диалога** указать **Новая база данных** (рис. 1.1.) и подтвердить создание новой базы данных нажатием кнопки **Ок**. В результате отобразится окно **Файл новой базы данных** (рис. 1.2), в котором предлагается указать путь и имя файла новой базы данных. По умолчанию файлу присваивается имя db1, и размещается он в папке "Мои документы", также принятой по умолчанию.

Переименуем файл, указав в поле **Имя файла**, новое имя - "Склад". Затем остается подтвердить создание нового файла базы данных, нажав на кнопку **Создать**. В результате в указанной папке ("Мои документы") создан файл БД "Склад". При этом в окне Microsoft Access автоматически появляется окно новой базы данных - **Склад: база данных** (рис. 1.3).

Это окно предназначено для хранения, создания и редактирования различных объектов базы данных. В окне новой базы данных автоматически выбирается вкладка **Таблицы**. Дальнейшие наши действия будут происходить на этой вкладке окна базы данных

Пример 2

Создать таблицу, содержащую сведения о товарах.

Сначала определим состав и типы полей будущей таблицы.

Предположим, наше торговое предприятие осуществляет продажу различных продуктов питания. Поэтому нас будут интересовать следующие характеристики объекта базы данных:

Код товара	- код, однозначно определяющий конкретный товар.
Марка	- марка товара (указанная на упаковке).
Описание товара	- описание, уточняющее состав данного товара.
Код типа	- код типа товара.
Серийный номер	- серийный номер, присвоенный партии товара.
Цена	- стоимость единицы товара.
Минимальный запас	- минимальный запас товара, хранящийся на складе.
Время задержки	- время задержки доставки товара.

Для создания новой таблицы воспользуемся кнопкой **Создать** в окне базы данных. В результате выдается окно **Новая таблица** (рис. 1.4), в котором требуется указать способ создания новой таблицы. Выберем **Конструктор** для создания структуры таблицы.

В режиме конструктора (рис. 1.5) опишем элементы структуры создаваемой таблицы - присвоим соответствующие имена полей, выберем их типы и, если необходимо, укажем их размеры. Имена полей в Access могут быть длиной до 64 символов и содержать как латинские буквы, так и символы кириллицы.

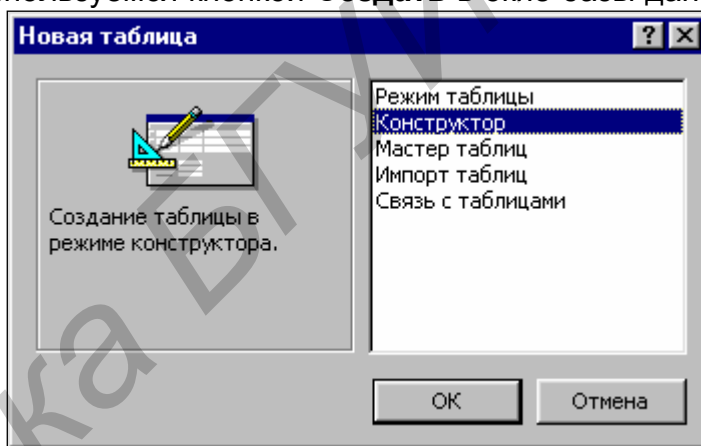


Рис. 1.4 Выбор режима создания новой таблицы

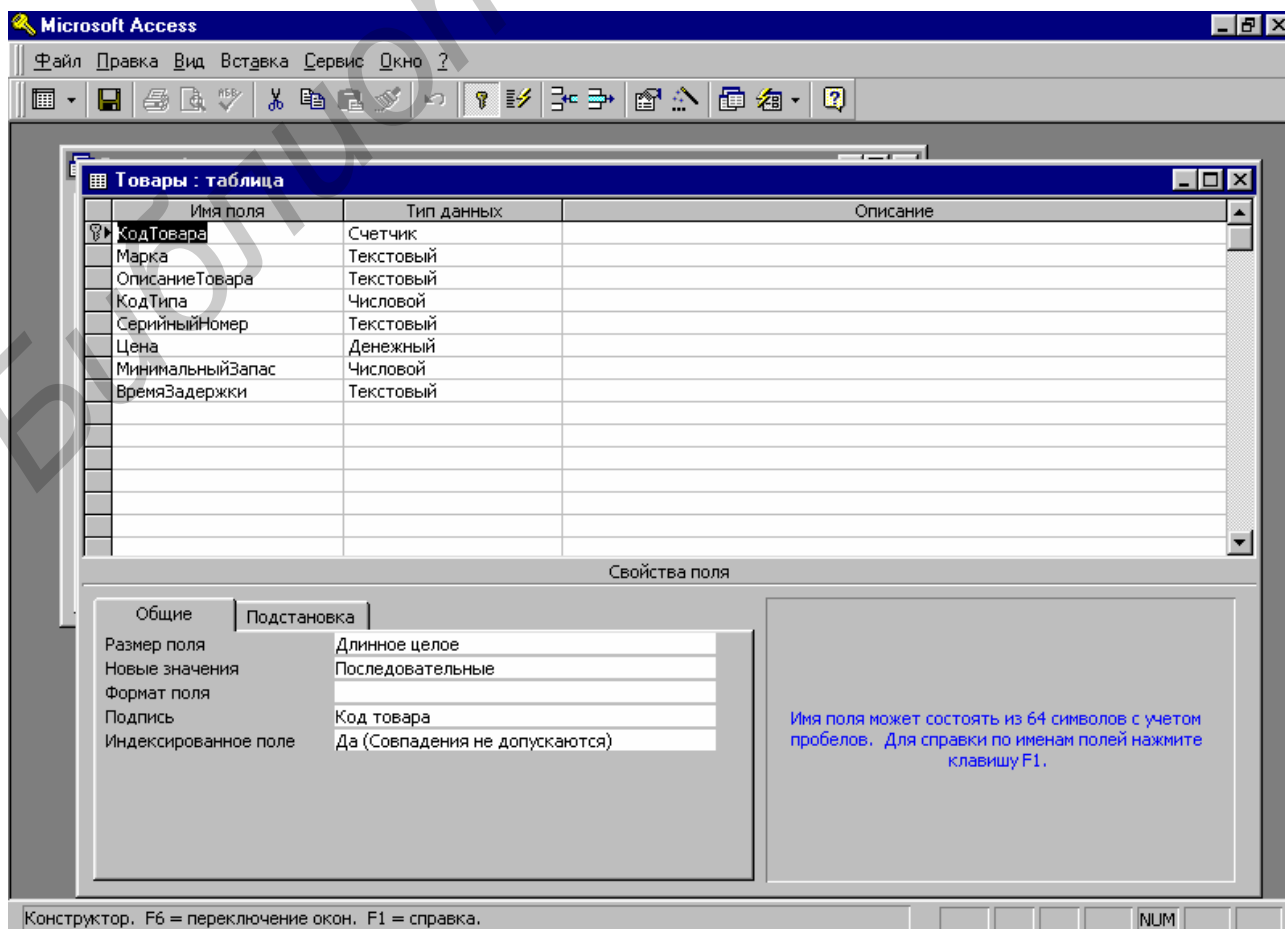


Рис. 1.5. Окно конструктора таблицы

Поэтому для имен полей можем использовать названия соответствующих характеристик товара. Тип поля будем выбирать в соответствии с информацией, которую предполагается вносить в данное поле. Длину поля выберем, исходя из максимально возможной длины данных, которые могут быть введены в это поле.

Заполнив таким образом поля структуры "Имя поля" и "Тип данных", получим следующую структуру таблицы "Товары":

<u>Свойства</u>			
Число записей:	5	Дата изменения:	10.03.98 9:19:07
Дата создания:	10.03.98 9:19:03	Допускаются обновления:	Истина
<u>Столбец</u>			
Имя	Тип	Размер	
КодТовара	Числовой (длинное целое)	4	
Марка	Текстовый	50	
ОписаниеТовара	Текстовый	255	
КодТипа	Числовой (длинное целое)	4	
СерийныйНомер	Текстовый	50	
Цена	Денежный	8	
МинимальныйЗапас	Числовой (длинное целое)	4	
ВремяЗадержки	Текстовый	30	

После завершения работы с конструктором таблиц можно сразу же перейти к вводу информации в созданную таблицу. Для этого можно

воспользоваться кнопкой **Вид** панели инструментов "Конструктор таблиц" либо меню **Вид - Режим таблицы** (Рис. 1.6). Можно также закрыть окно Конструктора, сохранив макет таблицы, а затем нажать кнопку **Открыть** в **Окне базы данных** (рис. 1.3).

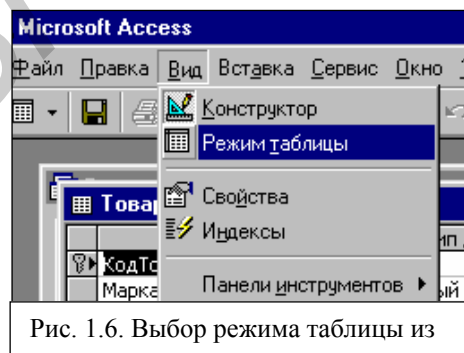
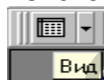


Рис. 1.6. Выбор режима таблицы из

Результатом проделанной работы является таблица "Товары", содержащая следующие сведения:

Код товара	Марка	Описание	Код типа	Серийный	Цена	Мин. запас	Срок
1	Цейлонский		кондитерские		18р.	10	10 дней
2	Тибетское		напитки		19р.	25	10 дней
3	Анисовый си-		приправы		10р.	25	10 дней
4	Индийская	Индийская	приправы		22р.	0	10 дней
5	Смесь Антона	Смесь Гумбо	приправы		21р.	0	10 дней

Выполнение работы

Создать базу данных "Склад", содержащую сведения о товарах, поставщиках, заказах, а также сведения о предприятии.

1) Создать таблицу "Поставщики" с помощью **Мастера таблиц**; таблица должна содержать следующие столбцы:

- КодПоставщика
- НазваниеПоставщика
- ДолжностьПолучателя
- Адрес
- Город
- Область
- Страна
- Факс

- УсловияОплаты
- Заметки

2) В режиме таблицы ввести сведения о товарах:

- КодТовара
- Наименование
- Цена
- Количество
- Стоимость
- СрокПоставки

3) В режиме конструктора создать таблицы "Заказы" и "Клиенты":
Структура таблицы "Заказы":

- КодЗаказа
- КодТовара
- КодПоставщика
- КодКлиента
- Наименование
- ДатаЗаказа .

Структура таблицы "Клиенты":

- КодКлиента
- Наименование
- ФИО_Руководителя
- Адрес
- Область
- Страна
- Телефон

4) Используя окно схемы данных, создать связи между таблицами.

5) Ввести данные в полученные таблицы. (не менее 10 записей)

Лабораторная работа №2

Проектирование запросов для управления данными

Цели: получить навыки конструирования запросов различного типа для выборки данных из реляционных таблиц и управления данными;
освоить технологию создания запросов QBE;

изучить основные конструкции реляционного языка запросов SQL.

Задача. Пусть требуется получить определенные сведения из одной или нескольких таблиц базы данных. Для решения этой задачи целесообразно создать запросы на выборку, содержащие требуемые критерии отбора данных. В ряде случаев требуется удалять или добавлять записи в имеющиеся таблицы базы данных. Эту задачу решают при помощи запросов на удаление или добавление соответственно, в которых задаются необходимые критерии, указывающие, какие записи должны быть удалены или, наоборот, добавлены в таблицу.

Общие сведения о запросах MS Access

В больших базах данных часто возникает проблема поиска необходимой информации (или отбора записей), удовлетворяющей определенным критериям. Задача поиска информации является одной из самых трудоемких и во многих случаях - одной из главных.

Для решения этой задачи предназначен механизм запросов. Этот механизм является стандартным и применяется почти одинаково во всех (или, во всяком случае, подавляющем большинстве) СУБД реляционного типа. Он представляет собой набор команд на языке SQL, определяющих критерий отбора записей в реляционной таблице. Таким образом, чтобы получить необходимую информацию из базы данных, следует записать соответствующие команды на языке SQL или, иначе говоря, сформировать запрос.

Многие СУБД обладают механизмом автоматизации проектирования запросов. Чаще всего запрос формируется на специальном бланке. Такой метод формирования запроса называется QBE (**Query By Example - Запрос по образцу**). В MS Access процесс создания запроса подобен процессу создания таблиц (см. Лабораторная работа №1 «Проектирование структуры базы данных и создание таблиц») Для того, чтобы начать проектирование нового запроса, необходимо перейти на вкладку **Запросы** окна базы данных и нажать кнопку **Создать**. В результате появится окно диалога **Новый Запрос** (рис. 2.1), аналогичное окну **Новая таблица**. В этом окне

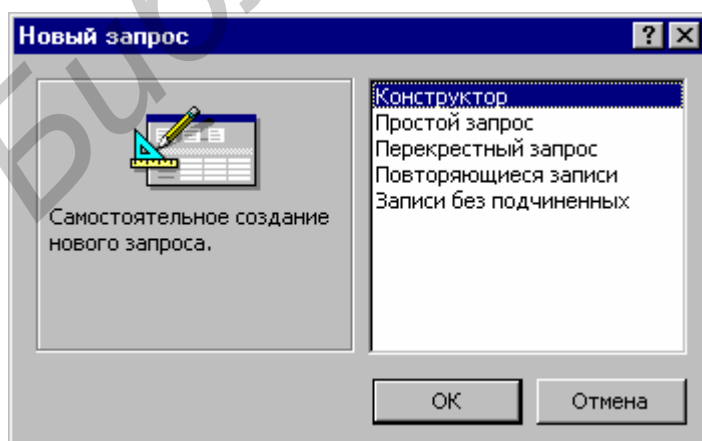


Рис. 2.1 Окно нового запроса

будет предложено выбрать один из вариантов создания запроса.

Запрос можно создать самостоятельно при помощи **Конструктора** или использовать готовый. Назначение каждого из режимов поясняется в левой части окна при указании мышью. Как показывает практика, большинство запросов создается с помощью **Конструктора**. По этой причине мы более подробно рассмотрим этот способ.

После подтверждения запуска

Конструктора открывается бланк запроса (рис. 2.2) и окно **Добавление таблицы** (рис. 2.3). В этом окне пользователю предоставляется возможность выделить одну или несколько таблиц, участвующих в запросе (выделение таблиц осуществляется аналогично выделению файлов в операционной системе Windows). Чтобы выделенные таблицы поместить в запрос, следует нажать кнопку **Добавить**. Указанные таблицы отображаются в верхней части окна **Конструктора Запроса** вместе со всеми связями, если они имеются.

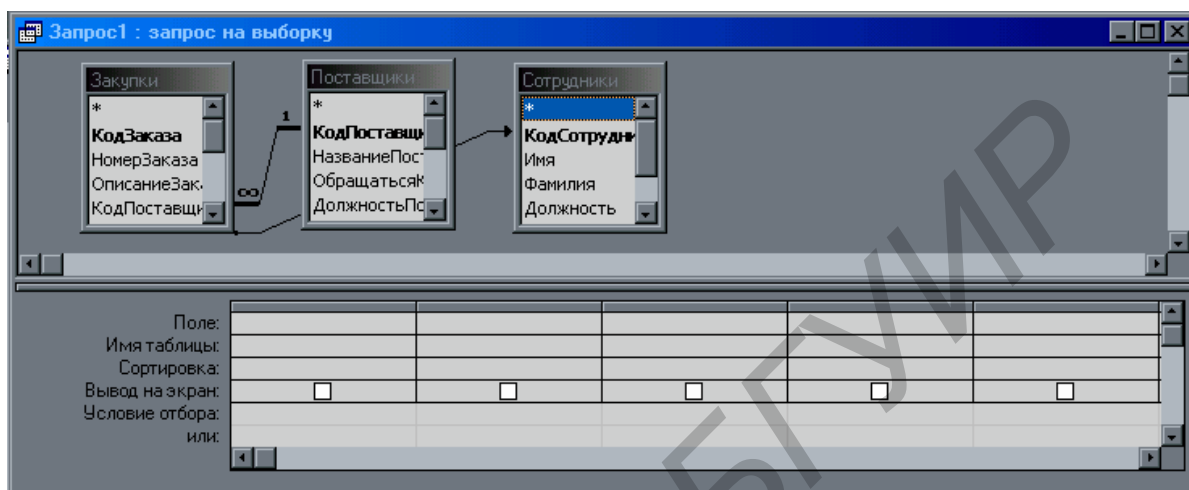


Рис. 2.2 Конструктор запроса

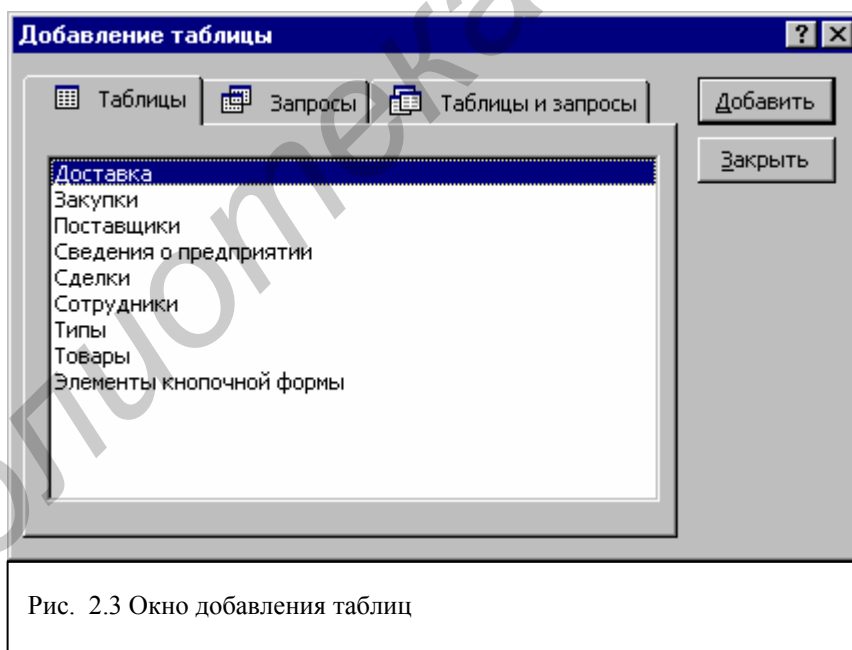


Рис. 2.3 Окно добавления таблиц

Теперь, чтобы сформировать запрос, необходимо в бланке **Конструктора** сформировать образец. Он составляется из полей соответствующих таблиц, логических условий и выражений. В простейшем случае, если в бланк помещаются только некоторые поля из таблиц, мы получаем запрос на выборку определенных полей из одной или нескольких таблиц. Указать наименование поля, значения которого должны выводиться или участвовать в запросе, можно одним из следующих способов (рис. 2.4):

- ☞ Выбрать поле из списка **Поле** в окне **Конструктора**.
- ☞ В таблице из верхней части окна **Конструктора** выделить необходимые поля и перетащить их мышью в ячейку **Поле** бланка запроса в нижней части окна - выбранные поля помещаются в бланке запроса последовательно, начиная с той ячейки, где была освобождена кнопка мыши.
- ☞ Если в образец запроса необходимо поместить все поля из таблицы, то можно одним из вышеуказанных способов поместить в бланк символ * ("звездочка").

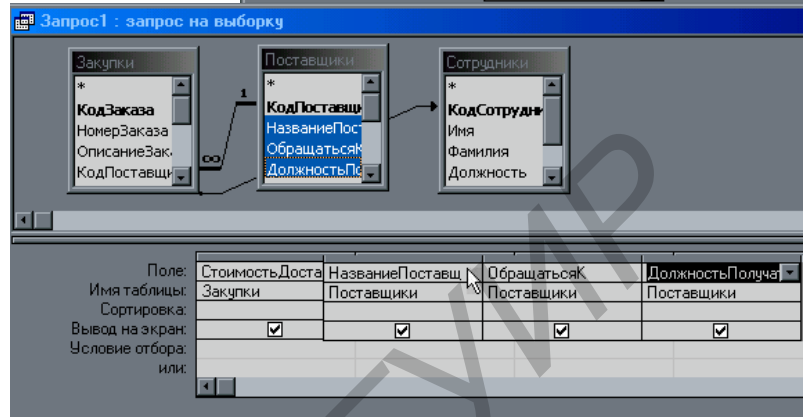
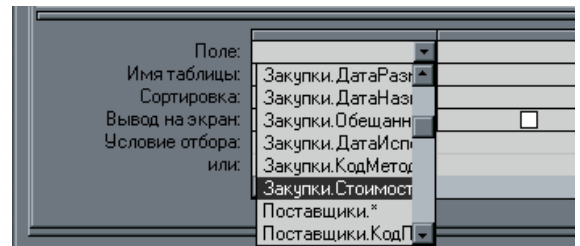


Рис. 2.4. Работа с бланком запроса в режиме конструктора

Варианты заданий

1. Создать запрос на выборку, позволяющий получить следующие сведения из базы данных:

таблица "Поставщики":

- НазваниеПоставщика
- Адрес
- Город
- Факс

таблица "Товары":

- Наименование
- Цена
- Количество
- Стоимость

таблица Клиенты":

- Наименование
- ФИО_Руководителя
- Адрес

Результирующие данные сортировать по наименованию товара.

2. Создать запрос на выборку для получения данных, удовлетворяющих определенным критериям:

- 1) вывести всех поставщиков, обслуживающих клиентов в городах, названия которых начинаются на букву "М";
- 2) получить сведения о клиентах, заказавших товары стоимостью от 200 до 400\$;
- 3) вывести адреса поставщиков, наименования которых начинаются с букв "Д ... И", обслуживающих клиентов, заказавших товары стоимостью не менее 250\$;
- 4) получить те записи из связанных таблиц (исключая коды), где содержатся сведения об определенном товаре. Записи сортировать по стоимости товара в порядке убывания;

- 5) получить сведения о зарубежных клиентах (расположенных не в Беларуси), оформивших заказ на сумму, превышающую 300\$;
- 6) вывести записи о поставщиках, расположенных в Гомеле и Могилеве;
- 7) получить данные о клиентах, которые проживают в Минске и Бресте или которые сделали заказ на поставку компьютеров;
- 8) выдать информацию о клиентах, проживающих в городах, названия которых начинаются на букву "Д", или в Могилевской области, но не заказавших телевизоры;
- 9) получить сведения о заказах, срок поставки которых истекает после 01/04/01
- 10) вывести информацию о поставщиках города Минска, которые должны выполнить заказы в период времени с 20/03/01 по 15/04/01;
- 11) получить сведения о клиентах, заказавших товары стоимостью от 200 до 400\$, получающих заказы 12/04/01 или 20/05/01;
- 12) уточнить информацию о поставщиках, наименование которых точно неизвестно, возможно, "АЛВИГ", "АДВИГ" или "АМВИГ", срок поставки которых для клиентов, проживающих в городах Добруш и Светлогорск, истек 5 дней назад.

Библиотека БГУИР

Лабораторная работа №3. Проектирование форм представления и управления данными

Цели: освоить основные приемы создания форм представления данных для отображения, редактирования и управления данными реляционных таблиц; изучить основные элементы форм.

Задача. Необходимо создать форму для представления данных, позволяющую просматривать и редактировать данные из нескольких связанных таблиц базы данных, предоставить пользователю простейшие элементы для управления данными. Для решения поставленной задачи создадим форму для отображения данных, включающую элементы управления данными.

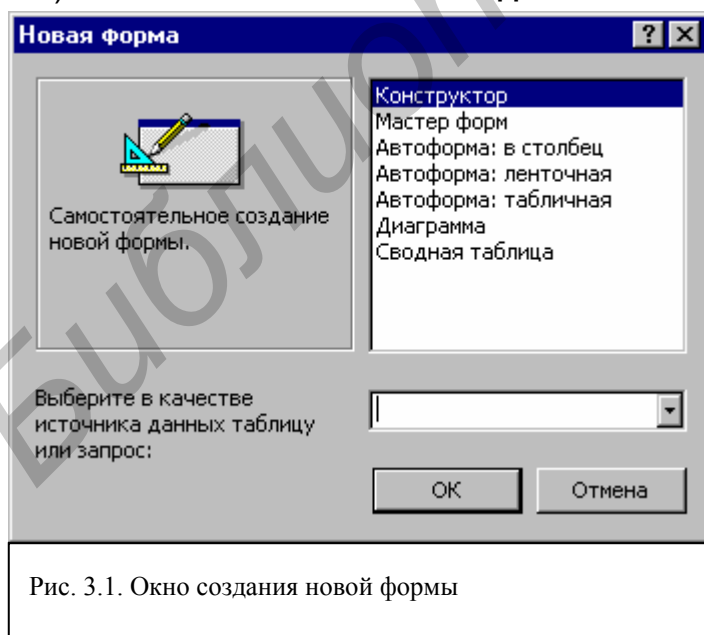
Общие сведения о формах MS Access

Табличная форма представления данных в реляционных базах данных хоть и является простой с точки зрения реализации, не всегда оказывается удобной для просмотра данных в больших таблицах. При значительном количестве полей таблицы доступ к данным может быть затруднен. Для большинства неподготовленных пользователей табличная форма является трудной для восприятия, они желали бы просматривать данные в привычном для себя виде – в виде бланка, документа определенной структуры или в каком-то другом удобном для восприятия виде.

В MS Access имеется возможность представлять данные в произвольном порядке, имитируя практически любую структуру документа. Для этой цели используются *формы*.

Форма – это средство отображения и редактирования данных с помощью различных элементов управления.

Также как таблицы и запросы, формы создаются различными способами. Для создания новой формы используется вкладка **Формы** окна Базы Данных (см. рис. 1.3). после нажатия кнопки **Создать** появляется окно диалога **Новая форма** (рис.



3.1). как и в предыдущих случаях, здесь имеется возможность выбрать один из вариантов создания формы. Для отображения простой структуры данных можно воспользоваться готовой формой (*Автоформа в столбец*, *Автоформа ленточная* и *Автоформа табличная*). Для создания более сложных форм используется *Конструктор* и *Мастер форм*. Режим *Диаграмма* предназначен для графического представления числовых данных в виде диаграмм и графиков. *Сводная таблица* используется для анализа данных в электронных таблицах MS Excel. При выборе режимов в окне диалога их назначение поясняется в левой части

ти окна.

Наиболее оптимальным представляется следующий прием. Сначала при помощи **Мастера форм** создать заготовку формы, затем в режиме **Конструктора** вне-

сти необходимые изменения и дополнения. Например, изменить порядок расположения полей, добавить вычисляемые поля, элементы управления и т. д. Впрочем, часто оказывается достаточным воспользоваться **Мастером**. Рассмотрим процесс создания формы на примере.

Пример 1

Создать форму отображения сведений о поставщиках. Сведения о поставщиках отобразить в виде бланка свободной формы.

Для решения поставленной задачи воспользуемся указанным выше приемом. С помощью **Мастера форм** создадим заготовку со стандартным расположением полей и оформлением. Затем в режиме конструктора изменим расположение полей данных в соответствии с формой бланка. После нажатия кнопки **Создать** на вкладке *Формы* в окне **Новая форма** (рис. 3.1) следует выбирать режим **Мастер форм**. В раскрывающемся списке в нижней части окна следует указать источник данных создаваемой формы. В нашем случае это таблица *Поставщики*. После подтверждения (нажатием кнопки **Ок**) запускается мастер. Рассмотрим последовательно шаги построения формы с помощью **Мастера**.

Шаг 1. Выбрать поля для отображения в форме.

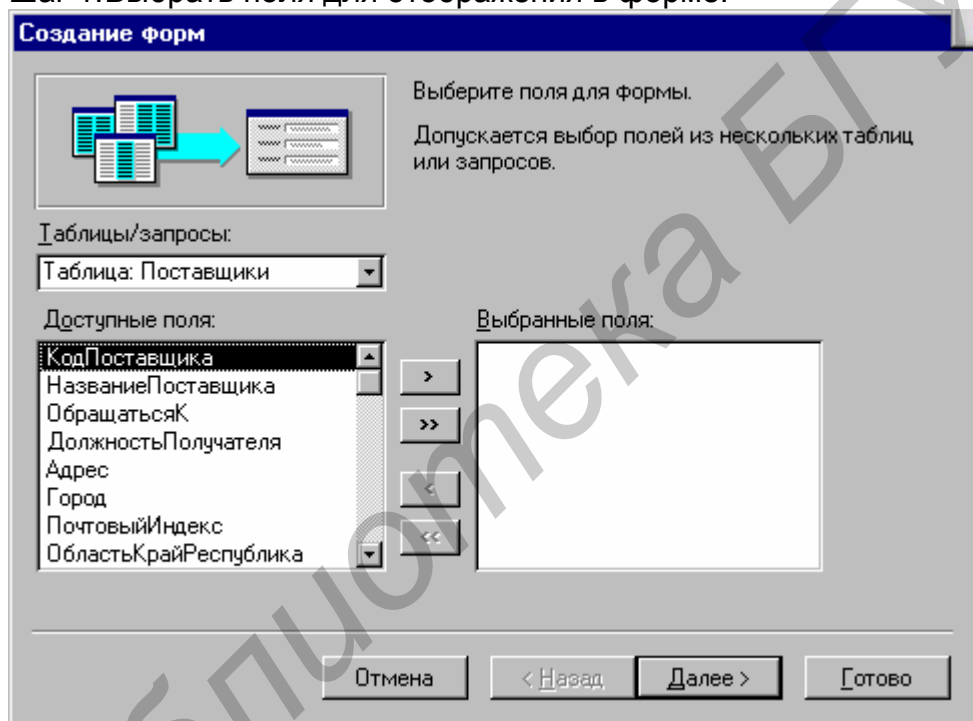


Рис. 3.1. Выбор полей в Мастере форм

Выберем следующие поля:

- КодПоставщика;
- НазваниеПоставщика;
- Город;
- ОбластьКрай Республика;
- Страна;
- НомерТелефона.

Шаг 2. Выбрать внешний вид формы.

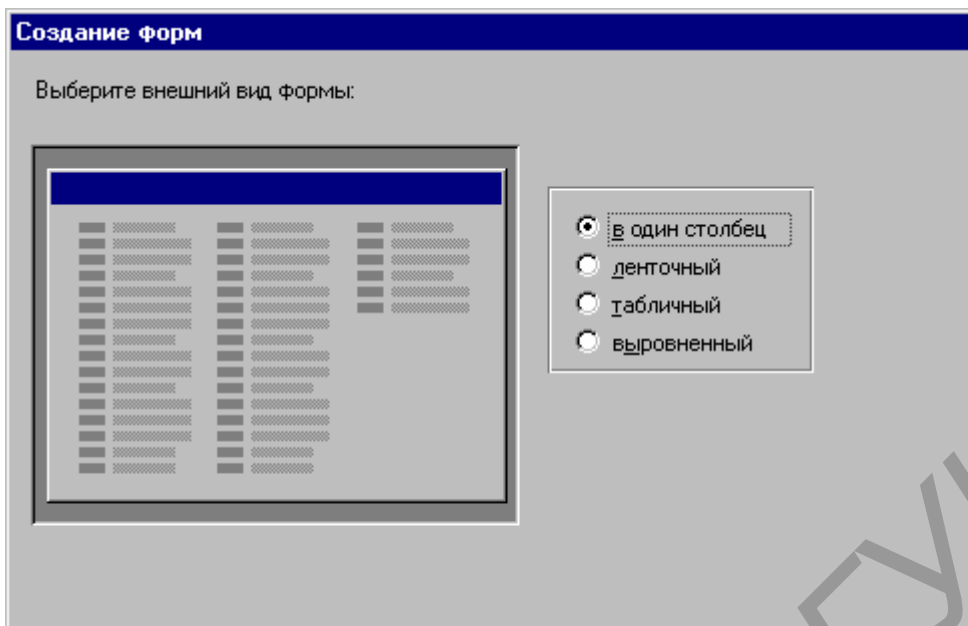


Рис. 3.3. Выбор внешнего вида формы.

В данном случае вид формы значения не имеет, поэтому оставим форму *В один столбец*.

Шаг 3. Выбрать стиль формы.

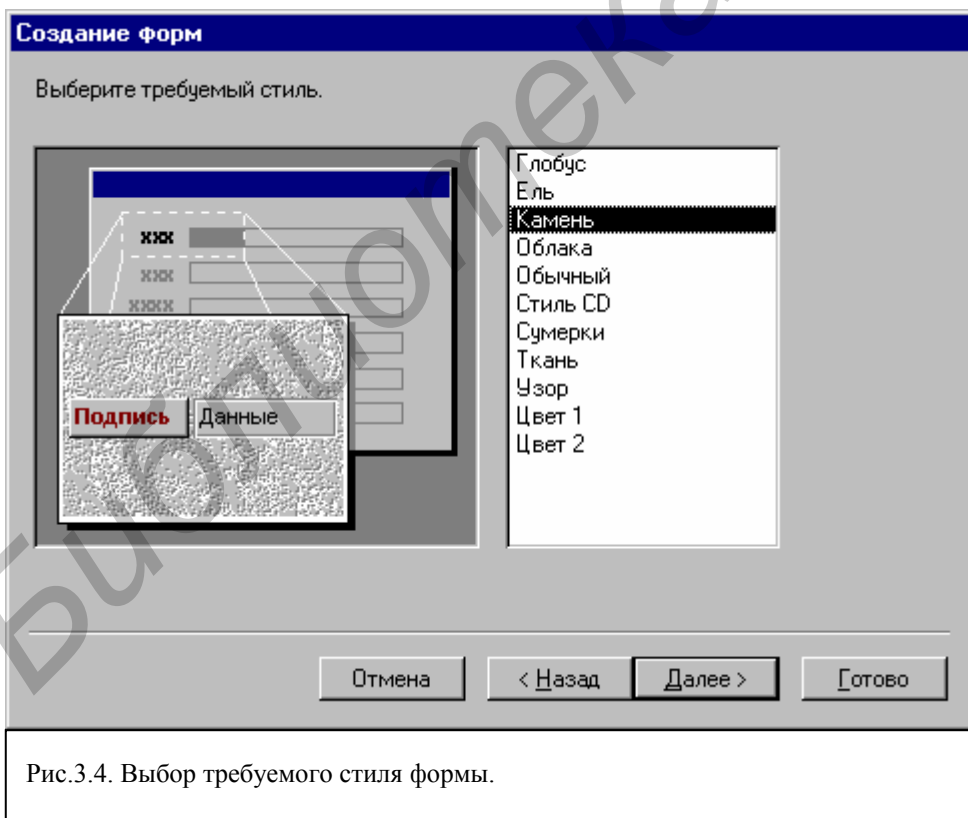


Рис.3.4. Выбор требуемого стиля формы.

В правой части окна показан список возможных стилей. При указании стиля в левой части окна отображается пример формы. Выбираем стиль «Камень».

Шаг 4. Задать имя формы.

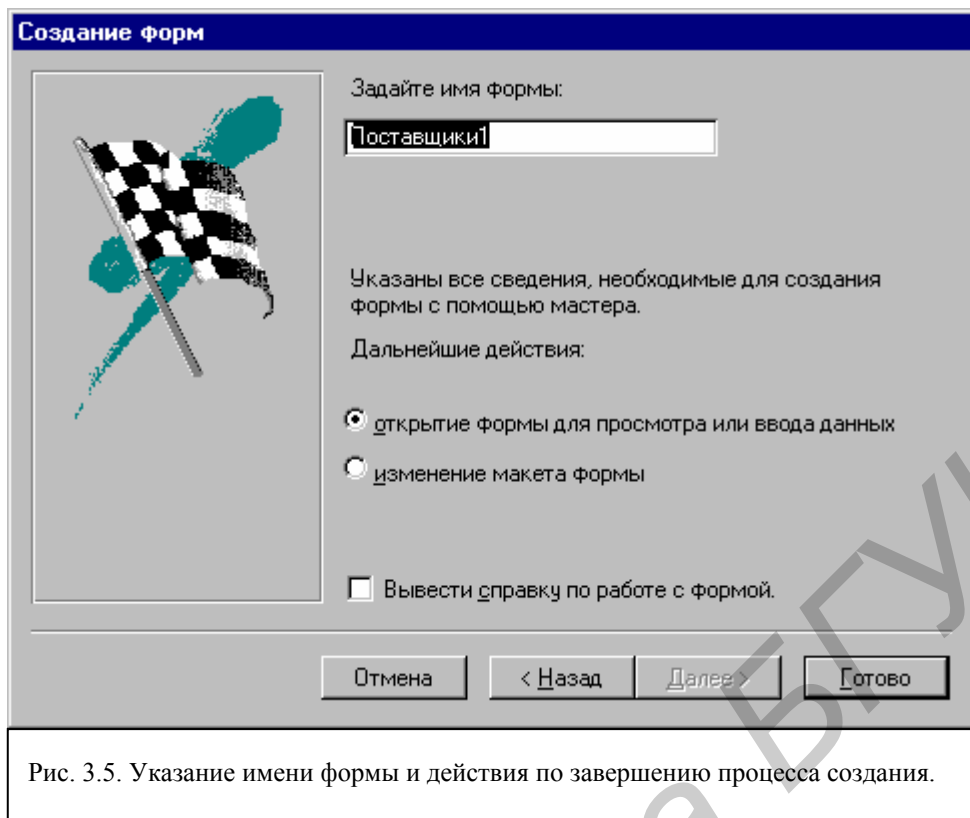


Рис. 3.5. Указание имени формы и действия по завершению процесса создания.

Это последний шаг проектирования формы с помощью мастера. Здесь задается имя, под которым форма будет сохранена в базе данных, и указываются дальнейшие действия, которые следует предпринять по завершению работы с Мастером.

После завершения работы с Мастером (нажать на кнопку **Готово**) получаем форму для просмотра и редактирования данных таблицы *Поставщики* (рис. 3.6).

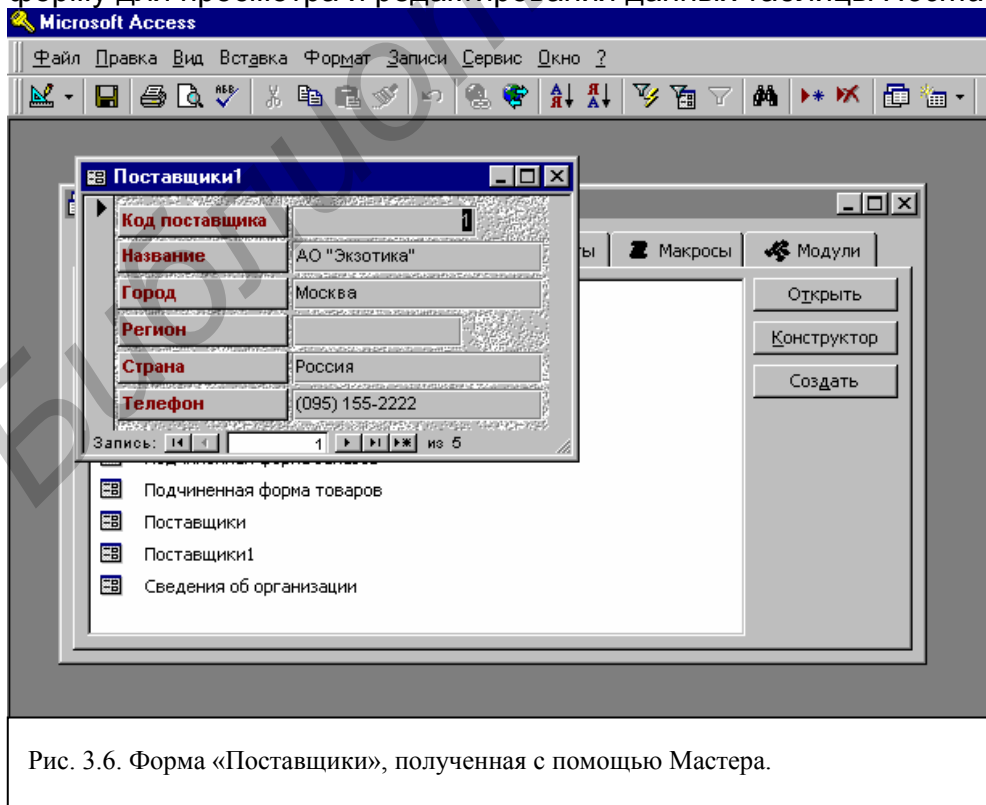



Рис. 3.6. Форма «Поставщики», полученная с помощью Мастера.

Теперь на основании полученной формы можно в режиме **Конструктора** сформировать бланк отображения данных.

Для редактирования формы в режиме **Конструктора** можно воспользоваться одним из следующих способов:

- ☞ находясь в режиме просмотра формы нажать кнопку **Вид** ;
- ☞ в окне Базы Данных выбрать форму, затем нажать на кнопку **Конструктор**;
- ☞ в окне Базы Данных щелкнуть правой кнопкой мыши и воспользоваться командой **Конструктор** контекстного меню.

В режиме конструктора переместим элементы формы соответствующим образом, чтобы получить такое расположение полей данных, как в требуемом бланке отображения (рис. 3.7).

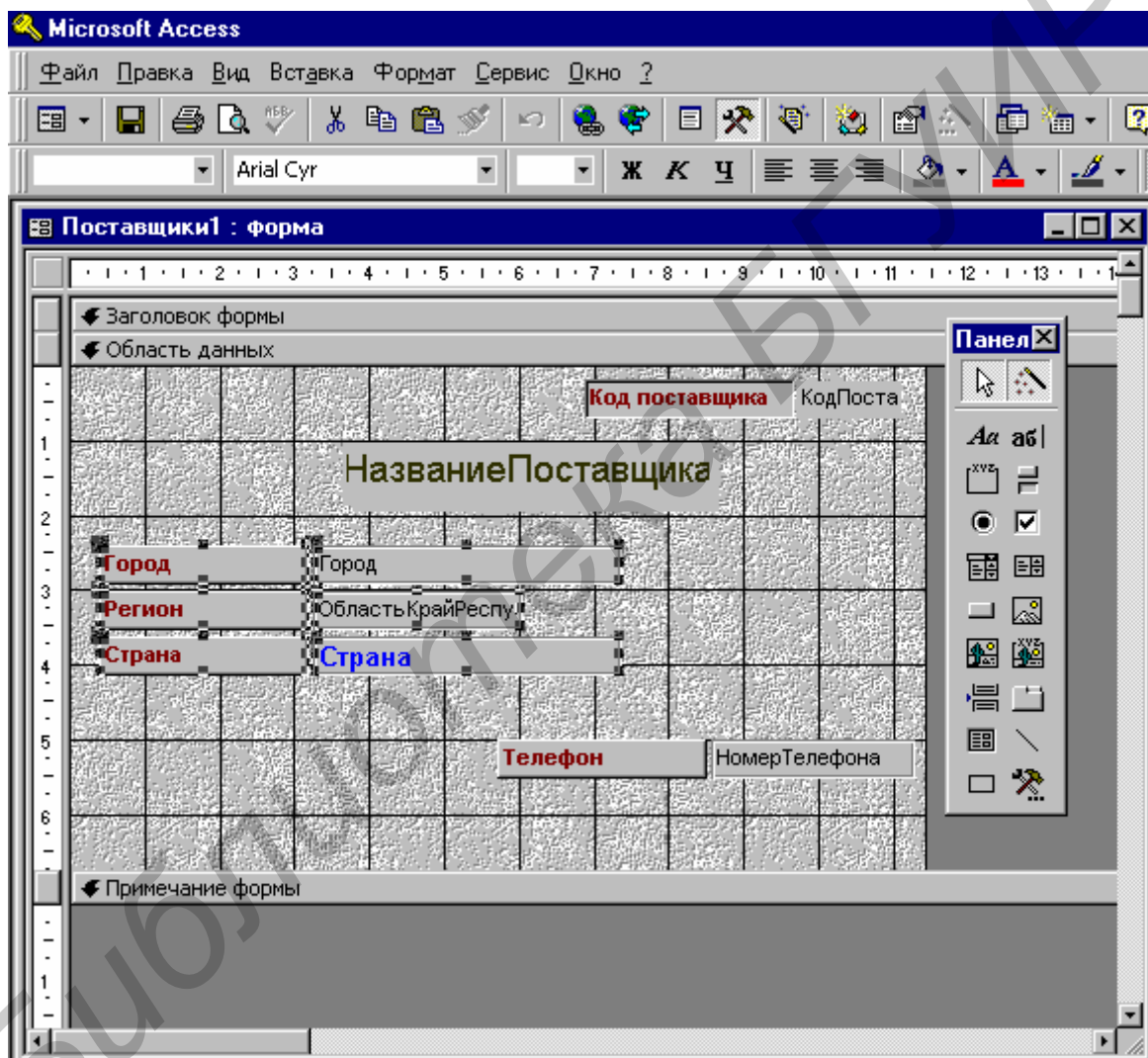


Рис.3.7. Форма в режиме конструктора.

Варианты заданий

Создать форму в соответствии с вариантом задания.

1. Форму «Клиенты», отображающую следующие сведения о клиентах:
 - Наименование
 - ФИО_Руководителя
 - Адрес
2. «Товары», содержащую следующие сведения о товарах:

- Наименование
- Цена
- Количество
- Стоимость

Рис. 3.8 Бланк сведений о товарах

сведения должны выводиться в бланке, представленном на рис. 3.8 *Стоимость* – вычисляемое поле. Значение вычисляется по формуле: $Стоимость = Цена * Количество$.

3. Отобразить сведения о поставщиках. По каждому поставщику отображать сведения о товарах (см. вариант 2) в табличном виде. Сведения о поставщиках отобразить на бланке, показанном на рис.
4. Сведения о зарубежных клиентах (расположенных не в Беларуси). По каждому клиенту показать информацию о сделанных заказах. Поля клиентов выводить в ленточной форме, поля заказов – в столбец.
5. Информация о просроченных заказах, с истекшим сроком поставки
6. Сведения о поставщиках и клиентах, сделавших заказ у этих поставщиков. Поля выводить в виде свободного бланка. На каждой странице формы отображать логотип организации-поставщика.
7. Бланк заказов. На бланке расположить поле со списком городов, где проживают клиенты, сделавшие заказ. В подчиненной форме вывести сведения о клиентах, проживающих в указанном городе.
8. Сведения о зарубежных клиентах и обслуживающих их поставщиках. Продвижение по записям таблиц осуществлять с помощью кнопок, добавленных в форму.
9. Бланк регистрации клиента. Изменение поля `КодКлиента` запретить. Удаление записи о клиенте осуществлять с помощью кнопки, добавленной в форму.
10. Список заказанных товаров. В форму добавить кнопку запуска запроса на выборку, выводящего записи, содержащие сведения о товарах, срок поставки которых совпадает с указанной датой. Дату указывать в поле формы.

11. Списки клиентов по поставщикам. С помощью кнопки запускать запрос на выборку, выводящий сведения о клиентах, проживающих в указанном городе.
12. Бланк регистрации заказа. Поставщика выбирать из раскрывающегося списка. Условия оплаты указывать с помощью переключателей.

Библиотека БГУИР

Лабораторная работа №4. Макросы MS Access

Определение и назначение.

Макрос - это последовательность, состоящая из одной или нескольких макрокоманд, выполняющих определенные действия над записями данных базы данных. Это могут быть такие наиболее часто выполняемые действия как, например, открыть форму, выполнить запрос, вывести отчет на печать и другие. Макрокоманда, в свою очередь, - это основной компонент макроса; замкнутая инструкция, самостоятельно или в комбинации с другими макрокомандами определяющая выполняемые в макросе действия. Иногда их называют просто командами.

Предназначен макрос главным образом для автоматизации выполнения последовательности часто повторяющихся действий. Например, имеется база данных "Сотрудники". При работе с этой базой данных могут возникать следующие задачи: прием на работу нового сотрудника, увольнение сотрудника, поиск в базе данных сведений об определенном сотруднике, печать отчетов. При этом каждый раз приходится выполнять одну и ту же последовательность определенных действий. К примеру, чтобы в нашу базу данных ввести сведения о новом сотруднике, необходимо открыть форму "Сотрудники" в режиме добавления новой записи, ввести посредством данной формы новые записи, закрыть форму и вывести отчет о новых сотрудниках. Если подобную задачу приходится решать довольно часто, то имеет смысл создать макрос, включающий в себя макрокоманды, выполняющие перечисленные действия. Макрос можно назвать "Зачисление нового сотрудника". Теперь для ввода данных о новом сотруднике достаточно щелчком мыши запустить наш макрос, и все макрокоманды, одна за другой, последовательно будут выполнены в том порядке, в каком они записаны в макросе.

Приведем пример макроса.

Макрокоманды

Имя	Условие	Макрокоманда	Аргумент	Значение
		OpenForm	Имя формы: Режим: Имя фильтра: Условие отбора: Режим данных: Режим окна:	Сотрудники Форма Добавление Обычное
		OpenQuery	Имя запроса: Режим: Режим данных:	Продажи по сотрудникам и странам Таблица Изменение
		OpenReport	Имя отчета: Режим: Имя фильтра: Условие отбора:	Продажи по сотрудникам и странам Просмотр

В этом макросе выполняется линейная последовательность команд. Первой командой открывается форма «Сотрудники» для добавления записей о новых сотрудниках. Вторая команда запускает запрос «Продажи по сотрудникам и странам». Последняя макрокоманда выдает в режиме просмотра отчет «Продажи по сотрудникам и странам». После чего макрос завершает свою работу.

Создание макросов

Линейные макросы

Как указывалось выше, для некоторой последовательности часто повторяющихся простых действий используют макросы линейной структуры, представляющие собой последовательность определенных макрокоманд. Эти макрокоманды выполняются в том порядке, как они записаны в макросе, и последовательность выполнения не изменяется в зависимости от условий. Отсюда и условное название макросов такого типа.

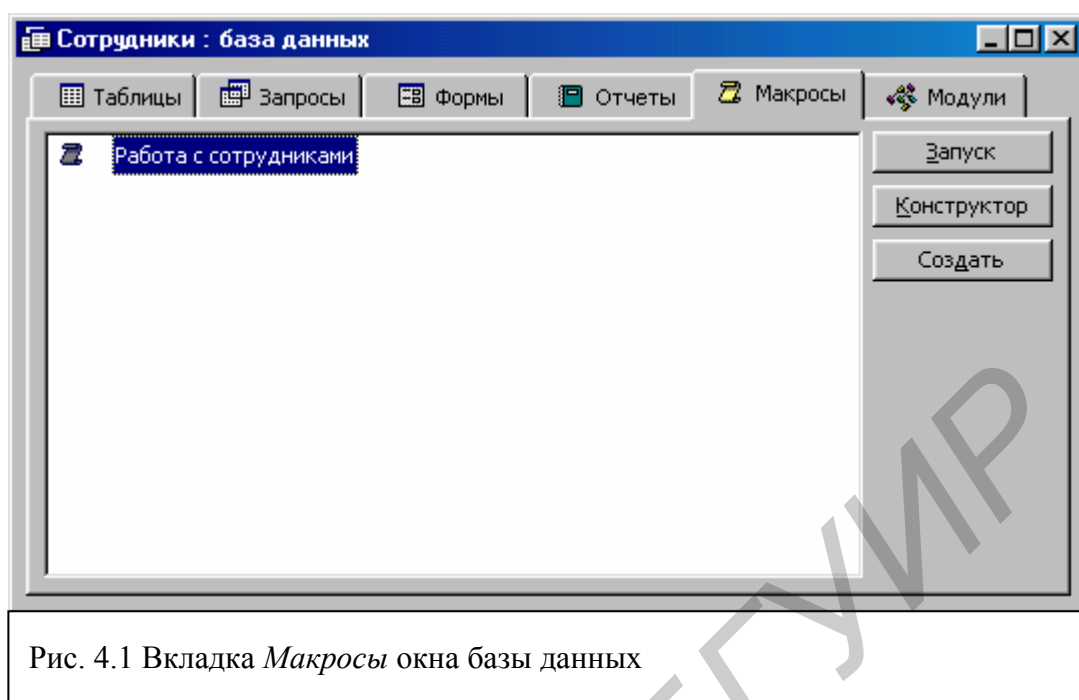


Рис. 4.1 Вкладка *Макросы* окна базы данных

Рассмотрим способы создания макросов. Для начала следует напомнить, что макрос является объектом базы данных MS Access. Для того чтобы создать макрос, необходимо перейти на вкладку **Макросы** (Рис. 4.1) окна базы данных. Затем нажать кнопку **Создать** для создания нового макроса. В результате откроется окно конструктора, (макросы создаются только в режиме конструктора), в котором можно формировать макрокоманды с соответствующими аргументами, если необходимо. Как создавать собственно макрокоманды, мы и рассмотрим на примере простейшего линейного макроса.

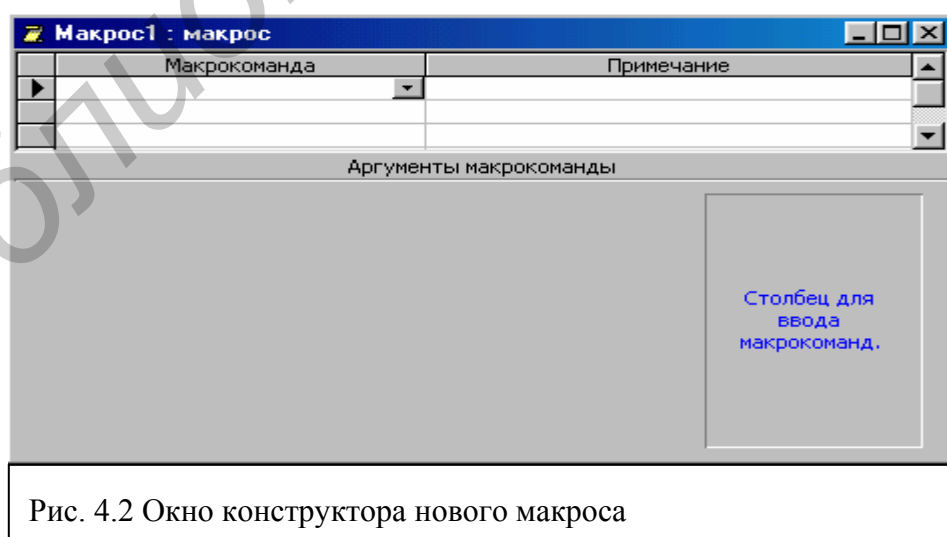


Рис. 4.2 Окно конструктора нового макроса

Первый способ предполагает выбор макрокоманд из поля со списком **Макрокоманда**. (Рис. 4.2) При выборе макрокоманды из списка автоматически становится доступным раздел **Аргументы** макрокоманды, в котором имеется возможность вы-

брать аргументы, соответствующие выбранной команде. Под аргументами макрокоманды понимается уточняющие параметры. Например, для макрокоманды *Открыть форму* в качестве аргумента должны быть указаны следующие аргументы: имя открываемой формы, режим, в котором необходимо открыть форму, имя фильтра и условие отбора, режим ввода записей и режим окна вывода. Каждую макрокоманду можно также снабдить примечаниями. Для этого нужно ввести соответствующий текст в поле *Примечание*. Примечание никак не влияет на работу макрокоманды, но значительно упрощает понимание цели ее применения и помогает быстрее разобраться в макросах сложной структуры.

Второй способ заключается в использовании объектов базы данных (Рис. 4.3). Так например, если требуется создать иакрос, открывающий форму, можно воспользоваться следующим приемом.

- Открыть вкладку **Макросы** окна базы данных
- Нажать кнопку **Создать**
- Используя соответствующие команды меню **Окно**, расположить окно базы данных и окно конструктора макроса таким образом, чтобы оба окна были видны полностью
- В окне базы данных перейти на вкладку с объектами требуемого типа
- В окне базы данных выбрать нужный объект и мышью перетащить его в окно **Конструктора** макроса в поле столбца **Макрокоманда**.

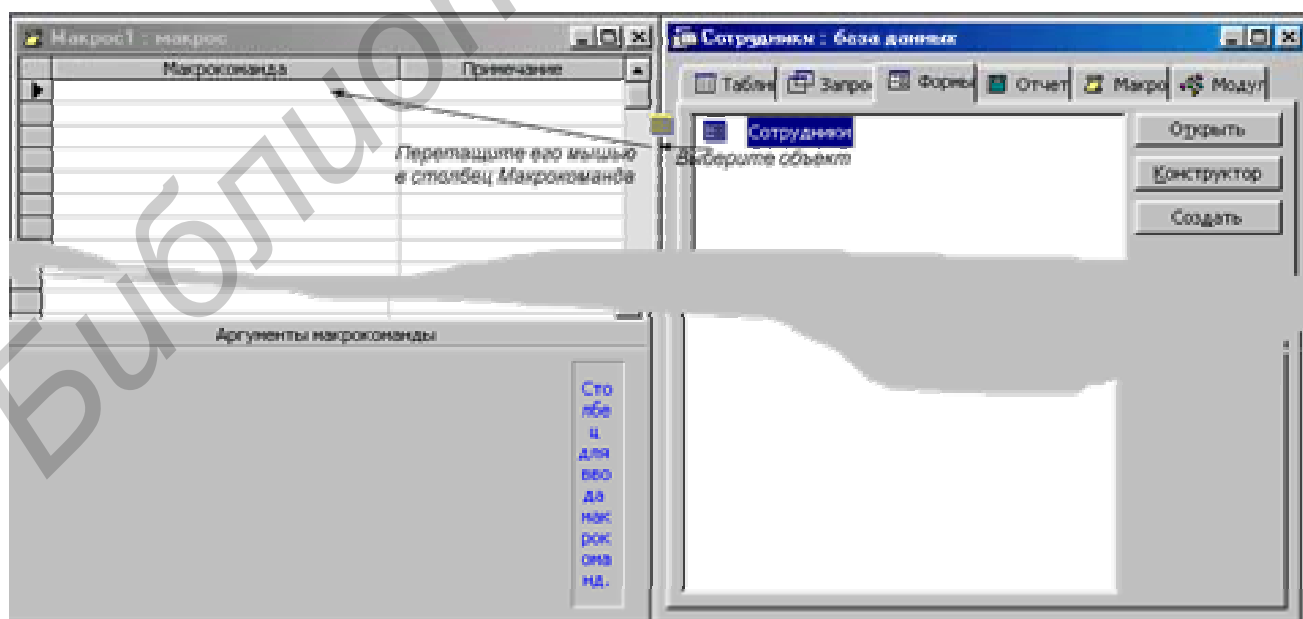
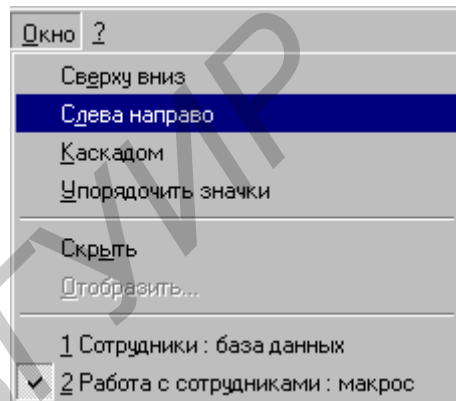


Рис. 4.3 Перетаскивание объекта базы данных в поле макрокоманд

В результате выполненных действий в окне Конструктора будет сформирована команда Открыть форму со всеми необходимыми аргументами, принятыми по умолчанию.

Пример 1. Предположим, при работе с базой данных «Сотрудники», требуется выводить форму для просмотра или ввода сведений о сотрудниках, запрос «Продажи по сотрудникам и странам» и отчет по сотрудникам и продажам для просмотра. Для автоматизации выполнения этих действий создадим макрос «Работа с сотрудниками».

1. На вкладке *Макросы* нажать на кнопку **Создать**.
2. С помощью команды *Слева направо* меню *Окно* расположить рядом окна конструктора макросов и окна базы данных.
3. В окне базы данных перейти на вкладку *Формы*.
4. В колонку *Макрокоманда* конструктора макросов из окна базы данных перетащить мышью значок формы *Сотрудники*.
5. Изменить аргументы макрокоманды, выбрав режим данных *Добавление*.



Аргументы макрокоманды	
Имя формы	Сотрудники
Режим	Форма
Имя фильтра	
Условие отбора	
Режим данных	Добавление
Режим окна	Обычное


6. По аналогии с пп. 3, 4 сформировать макрокоманды *ОткрытьЗапрос* и *ОткрытьОтчет*.
7. Изменить аргумент макрокоманды *ОткрытьОтчет* *Режим*, выбрав значение *Просмотр*.

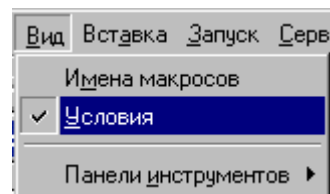
Созданный макрос решает поставленную нами задачу – последовательно открывает форму «Сотрудники» для добавления новых данных, запрос «Продажи по сотрудникам и странам» в режиме таблицы для изменения данных и одноименный отчет для просмотра.

Макросы с условиями

В ряде случаев требуется, чтобы некоторые действия макроса могли выполняться лишь при выполнении определенных условий. Так, например, в нашей базе данных при достижении некоторого среднего значения стоимости заказа должен выводиться итоговый отчет по продажам.

Процесс создания макроса остается таким же, как и в случае с линейным макросом, необходимо лишь добавить условие проверки. Для того, чтобы добавить условие, можно воспользоваться одним из нижеперечисленных действий:

-  Нажать кнопку **Условия** на панели **Конструктор макросов**
 - В меню **Вид** включить щелчком мыши **Условия**.
- В результате в бланке конструктора макросов появится



еще одна колонка – *Условие* (Рис. 4.4). В этой колонке с помощью логических операторов и функций формируется логическое выражение проверки *Условие*. Если условие выполняется (значение логического выражения – «ИСТИНА»), то выполняются команды, записанные в строке с условием и все последующие, в которых в ячейке *Условие* содержится символ «..» (две точки подряд) до строки с другим условием или до конца макроса. В том случае, если условие не выполняется (значение логического выражения – «ЛОЖЬ»), эти команды пропускаются, и выполнение макроса продолжается с команды, указанной в следующей строке или команды содержащей пустую ячейку *Условие*.

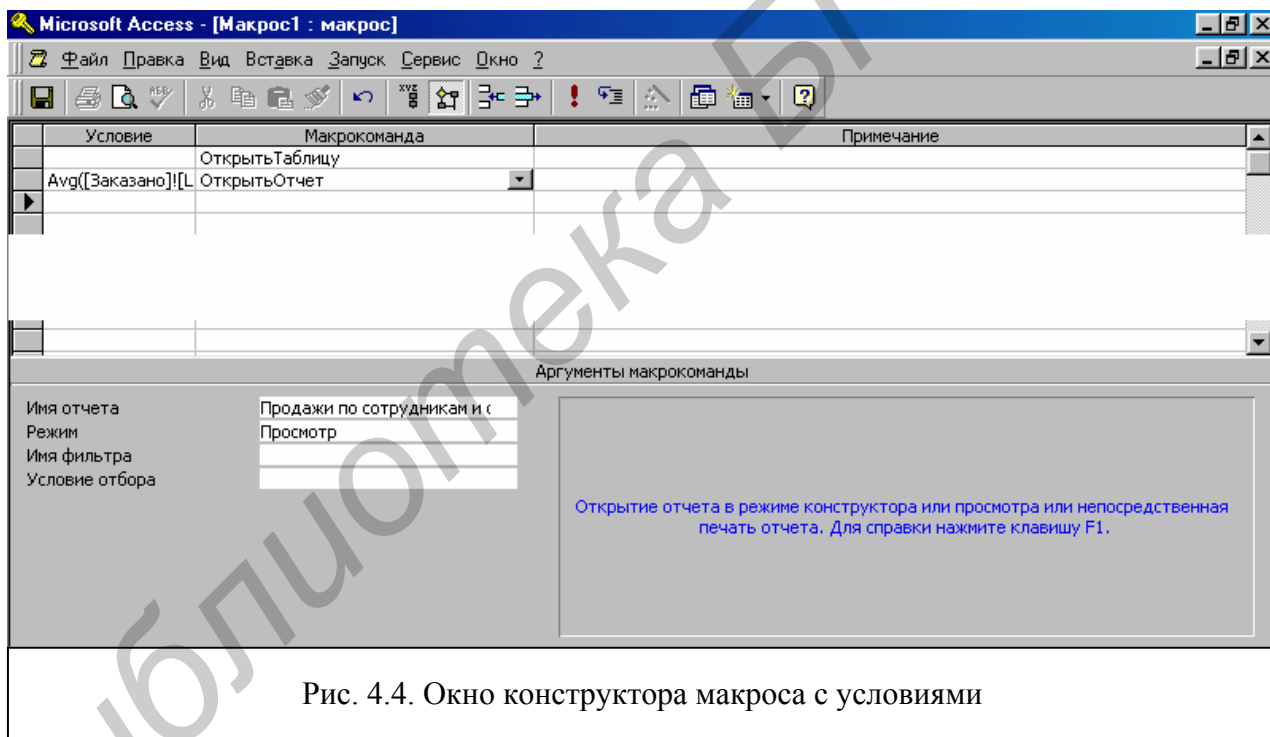

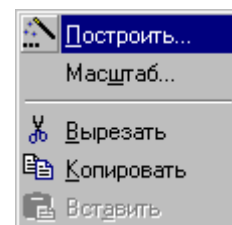


Рис. 4.4. Окно конструктора макроса с условиями

Пример 2. Пусть требуется выводить отчет по сотрудникам и продажам в том случае, если средняя стоимость заказов превысила сумму 2000р. Для решения задачи создадим макрос с условием. Ниже приведем последовательность действий, принятых при создании этого макроса.

1. Создать новый макрос.
2. В окне конструктора макроса нажать кнопку **Условия**.
3. Описанным выше способом вставить макрокоманду *ОткрытьФорму*, которая будет открывать форму *«Заказано (средняя цена)»*, содержащую вычисляемое мое поле для определения среднего значения цены заказов.
4. В колонке *Условие* следующей строки ввести логическое выражение для проверки условия. При этом можно воспользоваться **Построителем вы-**

ражений (рис.3.5). Для этого необходимо нажать кнопку **Построить**  в панели инструментов **Конструктор макросов** или воспользоваться контекстным меню в ячейке **Условие** строки соответствующей макрокоманды. Используя элементы **Построителя выражений**, сформировать условие проверки значения средней цены заказанных товаров:



товаров:

[Forms]![Заказано (средняя цена)]![Avg _ Цена]>20000.

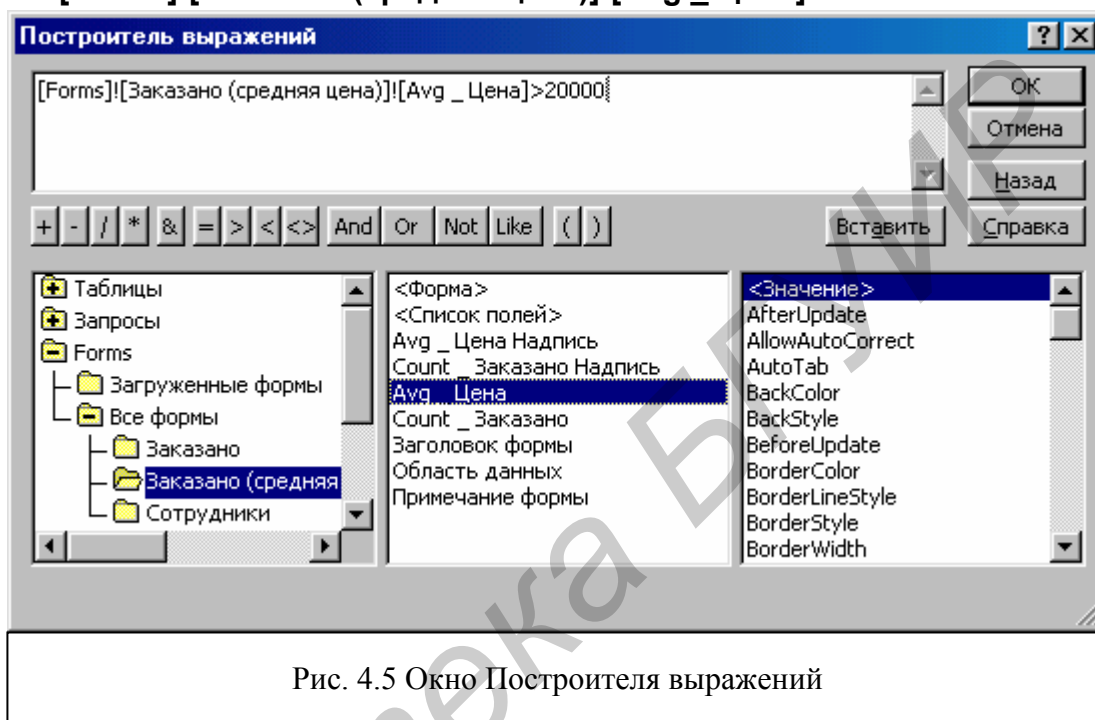


Рис. 4.5 Окно Построителя выражений

5. В ячейке **Макрокоманда** этой же строки, где задано условие, ввести макрокоманду **ОткрытьОтчет** - эта макрокоманда будет выполняться всякий раз, когда средняя цена заказа будет превышать сумму 20000 руб., отображая отчет **«Продажи по сотрудникам и странам»**.


В результате выполненных действий мы создали простейший макрос с условием, позволяющий решить конкретную задачу – выдача отчета по продажам при средней цене заказов, превышающей 20000р. Данный макрос выдает отчет лишь в том случае, когда выполняется заданное условие.

Группа макросов

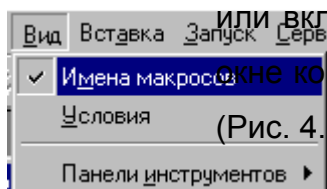
В сложных формах базы данных может содержаться большое количество различных элементов управления, поля с особыми свойствами, может осуществляться обработка нестандартных событий. Как правило, эти элементы связаны с соответствующими макросами. Иногда база данных может содержать несколько десятков различных макросов. Ориентироваться в таком большом множестве бывает не очень удобно. В таких случаях для того, чтобы упростить управление базой данных, создают **группу макросов**.

Группа макросов – это набор связанных макросов, сохраняемых под общим именем. При этом каждая группа может «специализироваться» либо по объектам, либо по определенным действиям. Например, может быть создана группа макросов для активных элементов формы «Сотрудники» или группа макросов для вывода на печать из различных объектов базы данных.

Группа макросов представляет собой объединение различных макросов (чаще родственных по своему назначению) в одном объекте. Причем в одной группе могут быть как простые линейные макросы, так и сложные макросы с условиями разветвленной структуры. Макросы внутри каждой группы различаются по их именам. Имя макроса группы состоит из двух частей, разделенных точкой – **Имя_Группы.Имя_Макроса**. Например, имя макроса, выводящего отчет на печать и который содержится в группе «Печать», может выглядеть так: *Печать.ПечатьОтчета_Продажа*. Для запуска конкретного макроса из группы необходимо указать его полное имя. Для идентификации каждого макроса, включаемого в группу, конструктор макросов имеет столбец *Имя макроса*.

Создание группы макросов создается в том же самом окне конструктора макросов и технология создания практически не отличается от создания уже рассмотренных макросов. Отличие состоит лишь в необходимости включения имени макроса. Для того чтобы включить колонку с именами макросов в окне конструктора, следует нажать кнопку  **Имена макросов** панели инструментов **Конструктор макросов**

или включить **Имена макросов** в меню **Вид**. В результате в окне конструктора появится еще одна колонка – *Имя макроса* (Рис. 4.6).



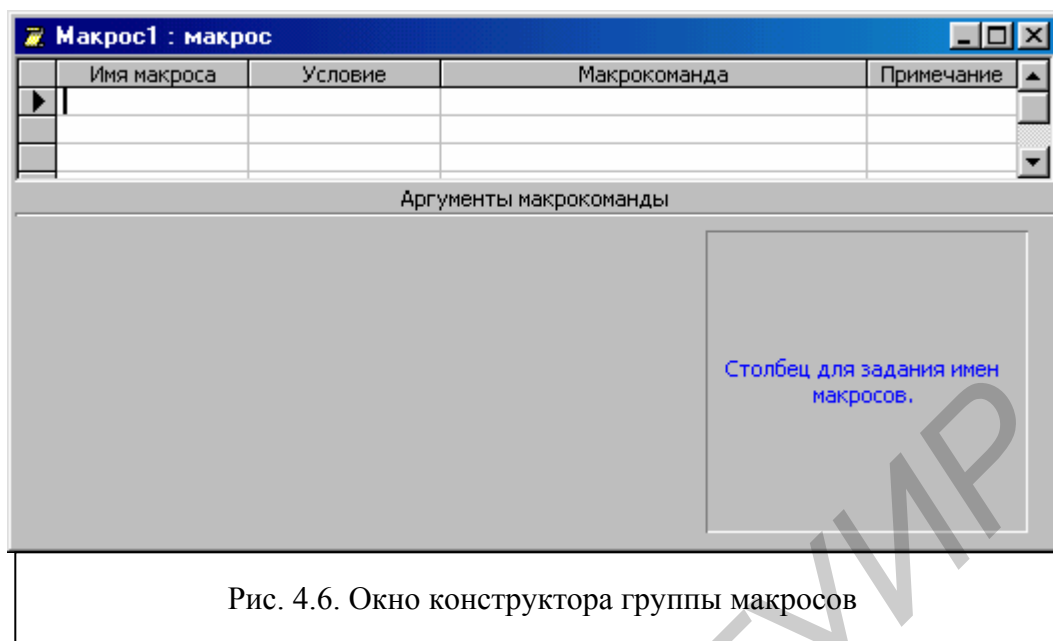


Рис. 4.6. Окно конструктора группы макросов

Теперь можно приступить к формированию группы макросов. Для каждого макроса внутри группы должно быть введено имя в столбце *Имя макроса*. Затем следует сформировать макрокоманды данного макроса (способы формирования макрокоманд и их аргументов описаны выше). При этом в ячейки *Имя макроса* макрокоманд остаются пустыми. Для того чтобы начать ввод макрокоманд другого макроса, необходимо в строку первой макрокоманды ввести имя. Процесс повторяется для всех макросов группы.

Когда производится запуск макроса группы, макрокоманды выполняются до тех пор, пока не произойдет одно из следующих событий:

- в столбце *Имя макроса* встретилось имя другого макроса;
- встретилась пустая строка;
- была выполнена макрокоманда *ОстановитьМакрос*;
- была выполнена макрокоманда *ОстановитьВсеМакросы*;
- достигнут конец группы макросов.

Пример 3. Пусть требуется создать макрос, обеспечивающий взаимодействие с базой данных по всем требуемым действиям: просмотр необходимых форм и отчетов, выдача отчетов при достижении определенного значения средней цены заказов и печать данных, полученных из запросов.

Для решения поставленной задачи создадим группу макросов, включающую ранее созданные макросы.

1. Создать новый макрос.
2. В окне конструктора макроса нажать кнопку *Имена макросов*.
3. В ячейке *Имя макроса* первой макрокоманды ввести имя первого макроса группы – «*Вход*».
4. Выполнить действия, описанные в пп. 2 – 7 примера 1 для формирования макрокоманд макроса «*Работа с сотрудниками*».

Совет: для включения в макрос группы макрокоманд из обычного макроса можно воспользоваться следующим приемом:

- Открыть существующий макрос в режиме конструктора;
 - Выделить строки макрокоманд обычного макроса;
 - Скопировать выделенные макрокоманды в буфер обмена;
 - Открыть окно группы макросов в режиме конструктора;
 - Установить курсор в первую строку, начиная с которой предполагается вставить макрокоманды;
 - Вставить макрокоманды из буфера обмена.
5. Ввести имя следующего макроса - «Отчетность».
 6. По аналогии с п. 4 ввести макрокоманды макроса «Отчет периода».
 7. В следующей строке ввести имя макроса «Печать сведений».
 8. Сформировать макрокоманды печати отчетов «Сотрудники», «Промежуточная сумма заказа» и «Продажи по сотрудникам и странам».
 9. Сформировать макрос «Выход», включающий макрокоманды, обеспечивающие подготовку к завершению работы и выходу из MS Access.
 10. Сохранить макрос, присвоив ему имя «Учет деятельности»

Группа макросов «Учет деятельности», полученная в результате выполнения вышеперечисленных действий, приведена в табл. 1.

Таблица 1. Макрокоманды группы макросов «Учет деятельности»

Имя	Условие	Макрокоманда	Аргумент	Значение	
Вход		ОткрытьФорму	Имя формы:	Сотрудники	
			Режим:	Форма	
			Имя фильтра:		
			Условие отбора:		
			Режим данных:	Добавление	
			Режим окна:	Обычное	
			ОткрытьЗапрос	Имя запроса:	Продажи по сотрудникам и странам
				Режим:	Таблица
				Режим данных:	Изменение
			ОткрытьОтчет	Имя отчета:	Продажи по сотрудникам и странам
			Режим:	Просмотр	
			Имя фильтра:		
			Условие отбора:		
Отчетность		ОткрытьФорму	Имя формы:	Заказано (средняя цена)	
			Режим:	Форма	
			Имя фильтра:		
			Условие отбора:		
			Режим данных:		
			Режим окна:	Обычное	
		[Forms]![Заказано (средняя цена)]![Avg_Цена]>20000	ОткрытьОтчет	Имя отчета:	Продажи по сотрудникам и странам
				Режим:	Просмотр
				Имя фильтра:	
				Условие отбора:	
Печать сведений		ОткрытьОтчет	Имя отчета:	Сотрудники	
			Режим:	Печать	
			Имя фильтра:		
			Условие отбора:		

		ОткрытьОтчет	Имя отчета:	Промежуточная сумма заказа
			Режим:	Печать
			Имя фильтра:	
			Условие отбора:	
	[Forms]![Заказано (средняя цена)]![Avg_Цена]>20000	ОткрытьОтчет	Имя отчета:	Продажи по сотрудникам и странам
			Режим:	Печать
			Имя фильтра:	
			Условие отбора:	
Выход		Закрыть	Тип объекта:	
			Имя объекта:	
			Сохранение:	Подсказка
		ОстановитьВсеМакросы		
		Выход	Параметры:	Подтверждение

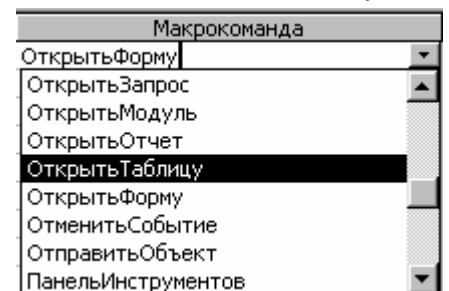
Редактирование и отладка макросов

Редактирование макросов



В процессе создания или после сохранения, макрос можно редактировать – изменять, удалять или добавлять макрокоманды и их аргументы, и другие элементы макроса. Редактирование осуществляется в режиме конструктора. В процессе редактирования обычно используют клавиатуру, панель инструментов **Конструктор макросов** и набор команд и функций из строки меню (которое автоматически настраивается на текущий режим работы). При этом возможны следующие случаи редактирования.


Редактирование макрокоманд. Если необходимо изменить введенную макрокоманду, то следует воспользоваться раскрывающимся списком в поле макрокоманд. Прокрутив открывшийся список, выбрать необходимую макрокоманду. Выбранная макрокоманда заменит ранее прежнее содержимое поля.




Можно также непосредственно вводить название макрокоманды с клавиатуры, предварительно выделив поле макрокоманд (выделяется автоматически, если в это поле входить при помощи клавиш управления курсором). При этом в поле будет автоматически введена та макрокоманда, название которой включает первые введенные символы.

Удаление макрокоманд. Для удаления одной или нескольких макрокоманд из макроса следует выделить соответствующие строки при помощи маркера выделения строк. (для выделения нескольких строк необходимо протащить

мышью, удерживая нажатой левую кнопку, по маркерам выделяемых строк). Для удаления выделенных строк:

- выбрать команду **Удалить строки** в меню **Правка**;
- выбрать команду **Удалить строки** в контекстном меню;
- нажать на кнопку **Удалить строки**  в панели инструментов конструктора макросов;
- нажать клавишу **** на клавиатуре.

Вставка макрокоманд. Чтобы вставить пропущенную макрокоманду, необходимо добавить строку в окне конструктора макроса. Для этого требуется вначале выделить строку, перед которой будет вставлена макрокоманда. Можно выделить также сразу столько строк, сколько требуется вставить, начиная от той строки, перед которой эти строки должны быть вставлены. Чтобы вставить указанные строки, остается выполнить одно из нижеперечисленных действий:

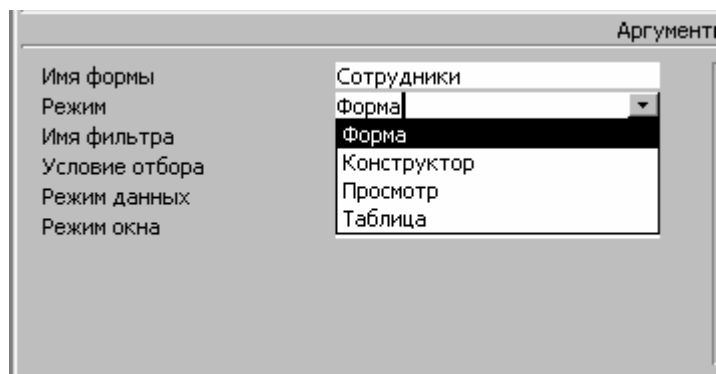
- выбрать команду **Строки** в меню **Вставка**;
- выбрать команду **Добавить строки** в контекстном меню;
- нажать на кнопку **Добавить строки**  в панели инструментов конструктора макросов;
- нажать клавишу **<Ins>** на клавиатуре.

В результате выполненных действий в макрос будет вставлено указанное количество строк. Вставленные строки будут расположены перед той строкой, которая была выделена первой.

Для вставки макрокоманд можно также перетащить мышью соответствующие объекты базы данных из окна базы данных, как это было описано в разделе «Линейные макросы». При этом соответствующая макрокоманда вставляется перед той, в строке которой был оставлен объект (отпущена кнопка мыши).

Изменение последовательности выполнения макрокоманд. Изменение последовательности выполнения макрокоманд заключается в перестановке соответствующих строк в окне макроса. Для этого необходимо выделить последовательность строк при помощи маркера выделения, а затем перетащить выделенные строки мышью в требуемую позицию. При этом жирной линией отмечается начало вставки выделенных строк.


Редактирование аргументов макрокоманды. Чтобы изменить аргументы макрокоманды, необходимо в окне конструктора установить курсор в строку, содержащую макрокоманду, затем перейти в раздел *«Аргументы*



макрокоманды». Затем изменить соответствующий аргумент, выбрав соответствующее значение аргумента из списка.

Аргумент *Имя объекта* можно изменить также, перетащив соответствующий объект из окна базы данных. Остальные аргументы автоматически принимают значения по умолчанию для данного типа объектов. При этом не имеет значения, в каком именно месте окна аргументов отпускать кнопку мыши.

Некоторые из аргументов могут быть заданы в виде вычисляемых выражений. В конце поля такого аргумента появляется кнопка **Построить**. При нажатии на эту кнопку вызывается окно **Построителя выражений** (см. рис.3.5). используя панели инструментов и элементы **Построителя выражений**, можно построить выражение для вычисления требуемого значения.

Редактирование условий в макросах. Для формирования условия в макросах с условиями необходимо в окно макроса добавить столбец *Условие* (если он еще не добавлен), как это было описано в разделе макросы с условиями. Затем следует ввести логическое выражение проверки требуемого условия. Кроме «ручного» ввода с клавиатуры имеется возможность воспользоваться **Построителем выражений**. Для этого следует нажать кнопку **Построить**  панели инструментов **Конструктор макросов**.


Для редактирования условия можно также воспользоваться **Построителем выражений**. При этом выражение, введенное в столбце *Условие*, будет отображено в окне **Построителя выражений**. Чтобы изменить какой-либо операнд в выражении или оператор, необходимо выделить соответствующую часть в самом выражении, а затем, используя панели инструментов и элементы **Построителя выражений**, ввести в выражение другой операнд или оператор.

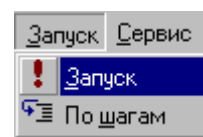
Редактирование имен макросов в группе. Для ввода имен макросов в группе необходимо добавить столбец *Имя*, а затем ввести имя макроса с клавиатуры. Для редактирования имени макроса используют стандартные приемы редактирования текста с помощью клавиатуры.

Отладка макросов

После проектирования макроса наступает следующий этап - отладка макроса. На стадии отладки происходит выявление ошибок и их устранение. Для простейших макросов, состоящих из небольшого количества макрокоманд, ошибки можно отслеживать непосредственно при выполнении макроса. Если же макрос состоит из боль-

шого количества макрокоманд и имеет сложную структуру, то может потребоваться отладка по шагам.

Запуск макроса. Чтобы запустить макрос из окна конструктора на выполнение, можно нажать кнопку **Запуск**  панели инструментов Конструктор макросов или воспользоваться командой **Запуск** меню **Запуск**.



Отладка макроса. Процесс отладки заключается в контроле за ходом выполнения макрокоманд макроса, обнаружении ошибок и их устранении.

В процессе отладки выполняют пробный запуск, и, если макрос выполняется, сравнивают результат выполнения с ожидаемым. Если в результате выполнения макроса должны быть проведены какие-либо вычисления, то обычно разрабатывают тестовый пример, результат вычисления которого заранее точно известен или предопределен, и сравнивают результаты, полученные при выполнении пробного запуска с результатами теста. Если результаты различаются, то следует отыскать ошибку в вычислениях или последовательности действий (алгоритме) самого макроса.


Ошибки, возникающие во время выполнения какой-либо последовательности инструкций (программе, модуле, макросе) бывают двух основных видов – *фатальные* и *нефатальные*. *Фатальные* ошибки приводят к немедленному прекращению выполнения макрокоманд макроса (операторов программы). *Нефатальные* ошибки не приводят к завершению выполнения макроса, выполнение продолжается, но результат, как правило, искажается.

Фатальные ошибки обусловлены, как правило, синтаксическими ошибками в написании макрокоманд, применением некорректных макрокоманд или значений аргументов (например, введено имя несуществующего объекта). *Фатальные* ошибки проще обнаружить в процессе пробного запуска. В случае возникновения ошибки, макрос останавливается на той макрокоманде, где она была обнаружена.

Нефатальные ошибки являются результатом некорректного проектирования последовательности действий макрокоманд в макросе или ошибками в формулах вычислений. Такие обнаружить труднее, так как они не приводят к остановке макроса.

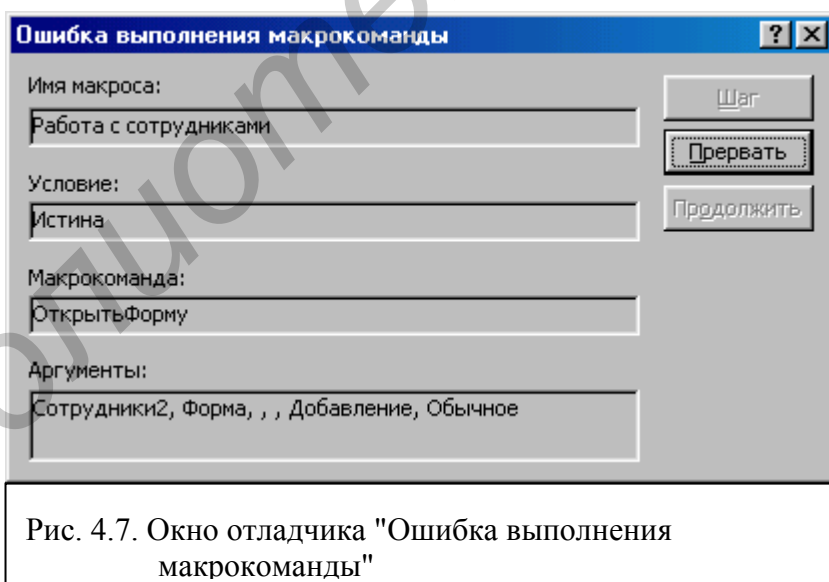
Фатальные ошибки устраняются путем изменения ошибочных макрокоманд и (или) значений их аргументов на корректные. *Нефатальные* устраняются только при изменении последовательности действий в макросе или в результате замены выражения, по которому вычисляется результат.

Для отладки сложных макросов целесообразно использовать пошаговый режим запуска. В этом режиме выполняется только одна макрокоманда и макрос останавливается. Затем переходят к следующей макрокоманде и так далее до конца макроса. В таком режиме проще отследить последовательность выполняемых действий и контролировать промежуточные результаты.

Для выполнения макроса в пошаговом режиме следует нажать кнопку **По шагам**  панели инструментов **Конструктор макросов** или воспользоваться командой **По шагам** меню **Запуск**.

Обработка ошибок. Если макрос содержит некорректную макрокоманду или макрокоманду с ошибочными аргументами, то при попытке запустить такой макрос возникает ошибка выполнения.

Например, в макросе «*Работа с сотрудниками*», который был создан при выполнении примера 1, случайно (или не случайно) был изменен аргумент Имя формы макрокоманды *ОткрытьФорму* – вместо имени существующей формы «*Сотрудники*» было введено имя «*Сотрудники2*». Так как такой формы на самом деле не существует, то при попытке выполнения данной макрокоманды при запуске макроса возникнет ошибка. В ответ на возникновение ошибки выдается диалоговое окно **Ошибки выполнения макрокоманды** с именем и аргументами макроса (Рис. 4.7).



Исходя из информации, содержащейся в этом окне, пользователь имеет возможность обнаружить ошибочно введенную макрокоманду или аргумент. Остается лишь отредактировать ошибочную макрокоманду либо изменить аргумент (или вообще отказаться от использования этой макрокоманды). В пошаговом режиме выполнения макроса имеется возможность перейти на один шаг выполнения вперед, т. е. выпол-

нить следующую макрокоманду или продолжить выполнение макроса (игнорируя ошибку).

Варианты заданий

Создать макрос, автоматизирующий выполнение следующих задач:

1. Открыть форму «Клиенты» для просмотра. После закрытия формы открыть соответствующую таблицу для добавления.
2. Запустить запрос на выборку, выводящий сведения о поставщиках по указанному заказу. Если такого заказа не существует, открыть форму бланка регистрации заказа на добавление.
3. Вывести список поставщиков. Для выбранного поставщика отобразить форму в режиме просмотра.
4. Зарегистрировать клиента – открыть соответствующую форму для добавления.
5. Оформить заказ. Если в форме бланка заказа введен несуществующий клиент, зарегистрировать его. После регистрации вывести список всех клиентов.
6. С помощью запроса на выборку вывести всех поставщиков, обслуживающих заказы в указанном городе. В случае отсутствия таковых, добавить в таблице *Поставщики* соответствующие записи.
7. Вывести список товаров с истекшим сроком доставки. После закрытия таблицы удалить соответствующие записи.
8. Определить среднюю стоимость заказов. Если она превысила определенное значение – вывести отчет.
9. Макросы в вариантах 3 – 5 объединить, разбив макрокоманды на группы.
10. Связать макрос варианта 3 с формой регистрации поставщиков.

Список литературы

1. Фигурнов В.Э. Программное обеспечение персональных ЭВМ. – М.: Наука, 1988.
2. Бемер С., Фратер Г. MS Access для пользователя: Пер. с нем. – Киев.: Торгово-издат. бюро ВНУ, 1994.
3. Артре Ш. Структурный подход к организации баз данных./Пер. с англ. - М.: Финансы и статистика, 1989.
4. Гиффорд Д. и др. Access 97. Энциклопедия пользователя. /Пер. с англ. - К.: Диа Софт, 1997.
5. Змитрович А.И. Базы данных. Учеб. пособ. для вузов.- Мн.: Университетское, 1991.
6. Мейер Д. Теория реляционных баз данных.- М.: Мир, 1987.
7. Хансен Г., Хансен Дж. Базы данных: разработка и использование./ Пер. с англ. - М.: БИНОМ, 1999.