

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
**Федеральное государственное автономное образовательное учреждение  
высшего образования**  
**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**  
Институт сервиса, туризма и дизайна (филиал) СКФУ в г. Пятигорске

**Методические указания по выполнению лабораторных работ по  
дисциплине «Управление данными» - 5 семестр**

09.03.02 Информационные системы и технологии  
Квалификация бакалавр

Пятигорск 2020 г.

В методических указаниях представлен практикум лабораторных работ по курсу «Управление данными», даны краткие теоретические сведения, примеры и варианты заданий для лабораторных работ.

Целью изучения дисциплины является ознакомление с основными понятиями банков данных и знаний, рассмотрение роли и места банков данных в информационных системах, получение знаний о системах управления базами данных (СУБД), освоение принципов построения реляционных баз данных и знаний, тенденции развития банков данных.

## Содержание

Лабораторная работа 1 . Управление процессами.....	4
Лабораторная работа 2. Управление памятью .....	5
Лабораторная работа 3. Управление файлами. ....	6
Лабораторная работа 4. Файловые системы и диски. ....	8
Лабораторная работа 5. Управление пользователями и группами. Установка лимитов и квот.....	10
Лабораторная работа 6. Резервное хранение данных. Программирование командных файлов .....	11
Лабораторные работы 7. Управление данными реестра в операционной системе .....	12
Лабораторная работа №8. Проектирование структуры базы данных и создание таблиц.....	20
Лабораторная работа №9. Проектирование запросов для управления данными .....	33
Лабораторная работа №10. Проектирование форм представления и управления данными.....	38
Список литературы .....	<b>Error! Bookmark not defined.</b>

## Лабораторная работа 1 . Управление процессами.

*Цель работы: получение практических навыков управления процессами и самостоятельной работы с документацией команд.*

### Основные понятия

**Команды POSIX для работы с процессами** (должны быть во всех операционных системах)

**at** - запускает программы в определенное время.

**crontab** - файл содержащий таблицу расписаний запуска заданий.

**kill** - прекращение выполнения процесса по PID процесса.

**nice** - задает приоритет процесса перед его запуском.

**renice** - изменяет приоритет работающего процесса.

**ps** - выводит информацию о работающих процессах.

**fg** - перевод процесса из фонового режима.

**bg** - продолжение выполнения фонового процесса, если он приостановлен нажатием <Ctrl+Z>.

### Команды Windows для работы с процессами

Большую часть информации о процессах можно получить через **диспетчер задач**.

**at** - запуск программ в заданное время

**Schtasks** - настраивает выполнение команд по расписанию

**Start** - запускает определенную программу или команду в отдельном окне.

**Taskkill** - завершает процесс

**Tasklist** - выводит информацию о работающих процессах.

Для получения более подробной информации, можно использовать центр справки и поддержки или команду **help** (например: **help at**)

**command.com** - запуск командной оболочки MS-DOS

**cmd.exe** - запуск командной оболочки Windows

### Практические задания

#### Задание 1

1. Запустите Windows
2. Составьте справочник для выше приведенных команд (на русском языке), расписав какие параметры для чего нужны.
3. Поработайте с этими командами: запуская, запуская по времени, меняя приоритеты, уничтожая процессы.
4. Разберитесь как работать с диспетчером задач, и что означает информация в столбцах (кроме информации о памяти).
5. Что нужно уметь (с командной строки):

- задавать задание на запуск процессов в определенное время, показывать этот список, удалять из списка, изменять задания;
- запускает определенную программу или команду в отдельном окне;
- завершать процесс;
- выводить (и понимать) информацию (исключая память и ввод/вывод) о работающих процессах и потоках.

## **Задание 2**

1. Запустите Windows
2. Составьте справочник для выше приведенных команд (на русском языке), расписав какие параметры для чего нужны.
3. Поработайте с этими командами: запуская, запуская по времени, меняя приоритеты, уничтожая процессы.
4. Разберитесь как работать с диспетчером задач, и что означает информация в столбцах (кроме информации о памяти).
5. Что нужно уметь (с командной строки):
  - выводить (и понимать) информацию (исключая память и ввод/вывод) о работающих процессах и потоках

***К сдаче лабораторной предоставляются:** справочник команд управления процессами и умение работать с этими командами.*

## **Лабораторная работа 2. Управление памятью**

***Цель работы:** получение практических навыков управления памятью и самостоятельной работы с документацией команд.*

Поясняющая информация

**Команды POSIX для работы с памятью** (должны быть во всех операционных системах)

**ps** - выводит информацию о процессах и памяти

### **Команды для работы с Windows**

Большую часть информации о памяти можно получить через диспетчер задач.

**Tasklist** - выводит информацию о работающих процессах и памяти

Практические задания

## **Задание 1**

1. Запустите Windows
2. Составьте справочник для выше приведенных команд (на русском языке), расписав какие параметры для чего нужны.
3. Разберитесь как работать с диспетчером задач, и что означает информация в столбцах (все что относится к памяти).
4. Поработайте с этими командами.
5. Что нужно уметь:
  - показывать рабочий набор для каждого процесса и в целом для системы
  - изменение рабочего набора с момента последнего обновления для каждого процесса
  - невыгружаемую память для каждого процесса и в целом для системы
  - выгружаемую память для каждого процесса и в целом для системы
  - число обращений к диску для загрузки страниц, не найденных в ОЗУ для каждого процесса
  - изменение числа обращений к диску для загрузки страниц, не найденных в ОЗУ, с момента последнего обновления.
  - показывать максимальный рабочий набор для каждого процесса
  - размер адресного пространства, переданной процессу

### Лабораторная работа 3. Управление файлами.

*Цель работы:* получение практических навыков работы с файлами и самостоятельной работы с документацией команд.

#### Поясняющая информация

**Команды POSIX для работы с файлами** (должны быть во всех операционных системах)

**cd** - смена каталога

**chgrp** - смена группы пользователей для каталога или файла

**chmod** - смена уровня доступа пользователей для каталога или файла

**chown** - смена пользователя для каталога или файла

**cp** - копирование файлов

**ls** - выводит список файлов и каталогов текущей директории

**file** - указывает тип указанного файла

**find** - поиск файлов

**ln** - создание ссылок

**mkdir** - создание каталога

**mv** - перемещение файла или каталога

**pwd** - имя текущего каталога

**rm** - удаления файла

**rmdir** - удаление каталога

**du** - сводка использования дискового пространства

**cat** - слияние и вывод файлов

## Команды Windows для работы с файлами

**Attrib** - позволяет просматривать, устанавливать или снимать атрибуты файла или каталога, такие как «Только чтение», «Архивный», «Системный» и «Скрытый».

**Chdir (Cd)** - вывод имени текущего каталога или переход в другую папку

**Copy** - копирование одного или нескольких файлов

**Del (erase)** - удаление файлов.

**Dir** - вывод списка файлов и подкаталогов каталога

**Fc** - сравнение двух файлов и вывод различий между ними.

**Find** - поиск заданной строки текста в файле или нескольких файлах

**Findstr** - поиск образцов текста в файлах с использованием регулярных выражений.

**Ftype** - вывод или редактирование связи между типом файла и его расширением

**Mkdir** - создание папки

**Move** - служит для перемещения одного или нескольких файлов из одного каталога в другой.

**Rename (ren)** - изменяет имя файла или набора файлов

**Replace** - заменяет файлы в одном каталоге файлами с теми же именами из другого каталога

**Rmdir (rd)** - удаляет каталог.

**Tree** - представляет графически дерево каталогов заданного пути или диска.

**Xcopy** - копирует файлы и каталоги, включая подкаталоги

## Практические задания

### Задание 1

1. Запустите Windows .
2. Составьте справочник для выше приведенных команд (на русском языке), расписав какие параметры для чего нужны.
3. Поработайте с этими командами.
4. Что нужно уметь:
  - просматривать, устанавливать или снимать атрибуты файла или каталога.
  - выводить имя текущего каталога и переходить в другую папку.
  - копировать файлы.
  - удалять файлы.
  - выводить список файлов и подкаталогов каталога.
  - сравнивать файлы.

- искать заданные строки текста в файлах.
- выводить и редактировать связи между типом файла и его расширением.
- создавать папки.
- перемещать файлы.
- изменять имена файлов.
- заменять файлы в одном каталоге файлами с теми же именами из другого каталога.
- удалять каталог.
- выводить графически дерево каталогов заданного пути.
- копировать файлы и каталоги, включая подкаталоги.

#### **Лабораторная работа 4. Файловые системы и диски.**

*Цель работы:* получение практических навыков работы с файловыми системами, дисками и самостоятельной работы с документацией команд.

Поясняющая информация

**Команды POSIX для работы с ФС и дисками** (должны быть во всех операционных системах)

**df** - выводит информацию о подмонтированных дисках

#### **Команды для работы с Windows**

**Chkdsk** - вывод на экран отчета о состоянии диска

**Chkntfs** - просмотр или задание планирования автоматической проверки системы для томов файловых систем FAT, FAT32 или NTFS при запуске компьютера.

**Cipher** - отображение или изменение шифрования папок и файлов на томах NTFS

**Compact** - вывод сведений или изменение уплотнения файлов и каталогов в разделах NTFS

**Convert** - преобразование томов с файловой системой FAT и FAT32 в тома с файловой системой NTFS.

**Defrag** - доиск и объединение фрагментированных файлов

**DiskPart** - Программа DiskPart.exe — это работающий в текстовом режиме командный интерпретатор, который позволяет управлять объектами (дисками, разделами или томами) с помощью сценариев или команд, вводимых с командной строки.

**Format** - Форматирование диска

**Fsutil** (поддерживаются только с версии Windows 5.1) - является служебной программой командной строки, которая используется для выполнения связанных задач файловых систем FAT и NTFS. Подкоманды для нее:

**behavior** - запрашивает, изменяет, включает или отключает настройки для создания имен файлов с длиной 8.3 символа, для возможности отображения расширенных символов в имена файлов длиной 8.3 символа в томах NTFS, для обновления штампа времени последнего доступа к томам NTFS, частоты событий, записанных в системных журнал, и количества места на диске, зарезервированного зоной основной таблицы файлов.

**dirty** - запрос установки «грязного» бита тома. Задает «грязный» бит тома. Когда установлен «грязный» бит тома, autochk автоматически проверит том на наличие ошибок при следующей перезагрузке компьютера.

**file** - поиск файла по идентификатору безопасности, запрос файла в размещенных диапазонах, установка короткого имени файла, допустимой длины данных или нулевых данных для файла.

**fsinfo** - перечисляет все диски, запрашивает тип диска, сведения о томе, специальные сведения о томе NTFS или статистику файловой системы.

**hardlink** - создает жесткую ссылку.

**objectid** Обычно используется специалистами по технической поддержке. Управляет идентификаторами объектов, которые используются Windows XP для отслеживания объектов, таких как файлы и каталоги.

**quota** - управляет дисковыми квотами в томах NTFS.

**reparsepoint** - работа с точками монтирования  
**sparse** - управление разреженными файлами.

**usn** - управление журналом изменения, в котором сохраняется архив всех изменений файлов в томе.

**volume** - размонтирование тома и отображение свободного места на диске.

**Label** - служит для создания, изменения или удаления метки тома (т. е. имени) диска.

**Mountvol** - служит для создания, удаления и получения списка точек подключения тома.

**Subst** - устанавливает соответствие пути в файловой системе заданному диску.

**Vol** - отображает метку тома диска и серийный номер, если они существуют.

## Практические задания

### Задание 1

1. Запустите Windows

2. Составьте справочник для выше приведенных команд (на русском языке), расписав какие параметры для чего нужны.
3. Поработайте с этими командами.

## **Лабораторная работа 5. Управление пользователями и группами. Установка лимитов и квот.**

*Цель работы:* получение практических навыков управления пользователями, группами, управление лимитами, квотами и самостоятельной работы с документацией команд.

### Поясняющая информация

**Команды POSIX для работы с пользователями** (должны быть во всех операционных системах)

- newgrp** - переход в новую группу
- who** - выводит список пользователей системы
- id** - выводит информацию о пользователе

### **Команды Windows**

**Управление пользователями и группами**  
Смотрите панель управления.

### **Установка квот**

Через графический интерфейс - смотрите свойства тома (раздела).

**fsutil quota** (поддерживаются только с версии Windows 5.1) - управление квотами (для создания и изменения дисковых квот для конкретных пользователей, запроса дисковых квот для конкретных томов, поиска в журнале системы записей о нарушении квот).

**fsutil behavior** - для смены частоты записи событий квоты в системный журнал.

### Практические задания

#### **Задание 1**

1. Запустите Windows
2. Составьте справочник для выше приведенных команд (на русском языке), расписав какие параметры для чего нужны.
3. Поработайте с этими командами.
4. Разберитесь как добавлять и управлять пользователями.

## Лабораторная работа 6. Резервное хранение данных. Программирование командных файлов

*Цель работы:* получение практических навыков резервного копирования данных, программирования командных файлов и самостоятельной работы с документацией команд.

### Поясняющая информация

**Команды POSIX для работы по резервированию данных** (должны быть во всех операционных системах)

**tar** - архивирование данных

### Команды Windows

**Резервное хранение данных.**

**Ntbackup** - архивация может выполняться из командной строки или пакетного файла с помощью команды `ntbackup` и различных параметров командной строки.

#### Пример 4 (из HELP'а для Windows 5.1)

Следующие примеры иллюстрируют выполнение архивации из командной строки. Все три примера используют тип архива, параметры проверки, уровень ведения журнала, аппаратное сжатие и другие параметры, заданные в программе архивации. В первом примере выполняется архивация ресурса `\\iggy-multi\d$` в файл `D:\Backup.bkf`. Во втором примере архивация тех же данных в этот же файл. В третьем примере имеющийся архив заменяется новым архивом, содержащим те же данные. Во всех трех примерах букву диска можно заменить полным именем UNC (то есть, вместо файла архива `d:\backup.bkf` можно задать файл `\\iggy-multi\d$\backup.bkf`).

```
ntbackup backup \\iggy-multi\d$ /j "Созданный из командной строки архив 4" /f "D:\backup.bkf"
```

```
ntbackup backup \\iggy-multi\d$ /j "Созданный из командной строки архив 5" /f "D:\backup.bkf" /a
```

```
ntbackup backup \\iggy-multi\d$ /j "Созданный из командной строки архив 6" /f "D:\backup.bkf"
```

### Программирование командных (пакетных) файлов

Командный файл - текстовый файл с расширением `.bat` и `.cmd`

В командном файле можно записать последовательность команд на исполнение.

Кроме этого можно написать не большой скрипт.

Некоторые возможности по написанию скрипта:

- IF Оператор условного выполнения команд в пакетном файле

- FOR Запуск указанной команды для каждого из файлов в наборе.

- GOTO Передача управления в отмеченную строку пакетного файла.

- PAUSE Приостановка выполнения пакетного файла и вывод сообщения.

- REM Помещение комментариев в пакетные файлы

Более подробную информацию можно получить в справке Windows и на "[Bat файлы](#)".

Также в Windows можно использовать скриптовые языки **VBScript**, **JScript**, или другие скриптовые языки (Active Perl, Python). Поддерживается это с помощью Windows Script Host (WSH).

## Практические задания

### Задание 1

1. Запустите Windows
2. Составьте справочник для выше приведенных команд (на русском языке), расписав какие параметры для чего нужны.
3. Поработайте с этими командами.
4. Напишите программу в bat-файле для резервирования каталога (каким архиватором, какой каталог и куда спросите у преподавателя).
5. Сделайте автоматический запуск этого скрипта, для создания ежедневной, еженедельной и ежемесячной копии.

## Лабораторные работы 7. Управление данными реестра в операционной системе

**Цель работы:** получение практических навыков управления данными, которые определяют параметры работы, через реестр операционной системы.

### Настройка Windows XP через реестр

Системный реестр операционной системы — это сложный механизм, для надлежащего функционирования которого требуются постоянное диагностирование и грамотное управление. С момента выпуска компанией Microsoft первой операционной системы Windows 3.11 системный реестр претерпел значительные изменения, превратившись из простого набора текстовой информации (файлы \*.ini) в полноценную иерархическую базу данных с полным контролем над операционной системой. Во время работы с Windows XP Professional или с Windows XP Home Edition часто приходится оптимизировать настройки, изменение параметров которых возможно только путем ручного редактирования системного реестра. Редактирование параметров реестра предоставляет довольно большие возможности, и для полного их описания понадобится отдельная книга. В работе рассмотрим лишь наиболее часто используемые настройки,

позволяющие оптимизировать операционную систему, настроить пользовательский интерфейс, установить необходимые уровни сетевой и пользовательской безопасности.

Для начала имеет смысл объяснить, что представляет собой системный реестр. Все настройки операционной системы вместе с конфигурацией персонального компьютера собраны в единой базе данных, именуемой системным реестром. С момента запуска компьютера и вплоть до его отключения операционная система непрерывно использует эту базу данных, контролируя настройки профилей всех пользователей, параметры программ, типы документов, сетевые настройки и т.д. В отличие от предыдущих версий ОС Microsoft Windows, в семействе Windows XP системный реестр не имеет ограничений по объему. Для работы с системным реестром в операционной системе Microsoft Windows XP пользователю предлагается использовать встроенную утилиту Regedit (рис. 1).

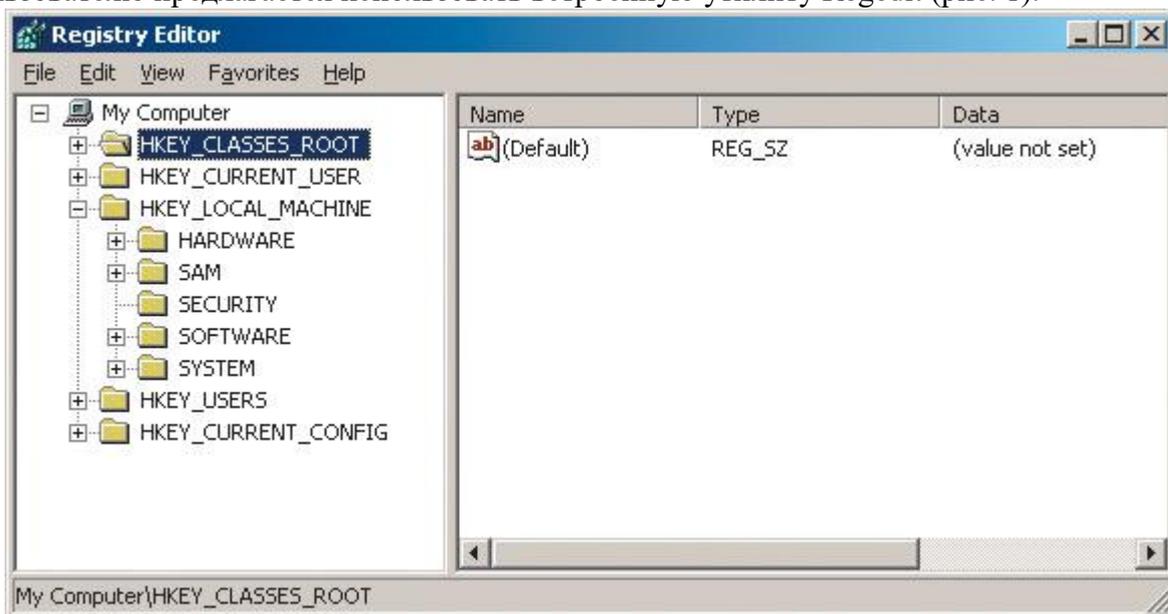


Рис. 1. Утилита редактирования системного реестра Regedit

Основываясь на иерархической системе, реестр предоставляет для работы максимально удобную форму, состоящую из разделов, подразделов, и параметров (ключей реестра). Реестр Windows XP состоит из пяти главных разделов:

- HKEY\_CLASSES\_ROOT (HKCR) — является подразделом HKEY\_LOCAL\_MACHINE\Software, параметры которого хранят информацию соответствия между приложениями и типами файлов, обеспечивая запуск необходимой программы во время открытия файлов с помощью проводника;
- HKEY\_CURRENT\_USER (HKCU) — является подразделом HKEY\_USERS, настройки параметров которого соответствуют потребностям активного пользователя, выполнившего вход в систему;
- HKEY\_LOCAL\_MACHINE (HKLM) — хранит настройки персонального компьютера для всех пользователей; помимо этого в данной папке содержится информация об аппаратной конфигурации и программном обеспечении;
- HKEY\_USERS (HKU) — содержит настройки профилей для всех пользователей Windows XP на персональном компьютере;
- HKEY\_CURRENT\_CONFIG (HKCC) — включает в себе ссылки на подраздел HKEY\_LOCAL\_MACHINE\SYSTEM\ CurrentControlSet\Hardware Profiles\ Current и содержит информацию о текущей конфигурации оборудования.

Значения ключей реестра (параметров) соответствуют определенному типу. REG\_BINARY — двоичные данные; информация хранится в двоичном виде и отображается в шестнадцатеричном формате. REG\_DWORD представляет собой целые числа размером в четыре байта и отображается в двоичном, шестнадцатеричном или

десятичном формате. REG\_EXPAND\_SZ представляет собой строку данных переменной длины. REG\_MULTI\_SZ — многострочный текст, удобный для чтения. REG\_SZ — текстовая строка фиксированной длины. REG\_FULL\_RESOURCE\_DESCRIPTOR представляет собой последовательность вложенных массивов, с помощью которой хранятся списки ресурсов оборудования или драйверов.

### **Оптимизация Windows XP**

До того как начать работу с реестром, настоятельно рекомендуем сделать его резервную копию, создать точку отката в Windows XP (System Restore) или создать образ диска с операционной системой, например с помощью программы PowerQuest DriveImage. Эту процедуру необходимо осуществить в том случае, если при неправильном редактировании реестра произошли изменения, повлекшие за собой некорректную работу ОС, и вернуть исходные настройки представляется весьма затруднительным. Имея резервную копию, вы всегда сможете восстановить исходные значения всех ключей реестра. К сожалению, почти в 99% случаях как неопытные, так и продвинутые пользователи персональных компьютеров игнорируют эту процедуру, что, естественно, приводит к плачевной ситуации и вызывает непредсказуемые последствия, например потерю важной информации. Для тех, кто заранее не заботился о сохранности реестра, единственный выход из сложившейся ситуации — переустановка операционной системы с нуля.

Погружение в практическое исследование недокументированных возможностей Windows XP следует начать с описания предварительных рекомендуемых процедур. Известно, что со временем системный реестр становится похожим на мусорную корзину, и это в первую очередь связано с тем, что пользователь постоянно устанавливает различные программы, деинсталляция которых очень часто производится некорректно или не полностью. Такую процедуру можно выполнить с помощью специальных утилит, которые автоматически сканируют реестр с последующей очисткой или удалением ненужных ключей реестра. В этих целях можно использовать упомянутую выше утилиту RegCleaner.

Дополнительные настройки, позволяющие оптимизировать операционную систему Windows XP через реестр, представлены в табл. 1.

Таблица 1. Общая оптимизация системы

Ключ	Параметр	Тип данных	Значение	Описание
HKEY_CURRENT_USER\ControlPanel\Desktop	WaitToKillAppTimeout	REG_SZ	5000	Настройка служит для уменьшения временного интервала, в течение которого Windows XP ожидает перед закрытием приложения. Меньшее значение ускоряет процесс завершения работы. Рекомендованное значение — 5000. По умолчанию — 20 000
HKEY_CURRENT_USER\ControlPanel\Desktop	HungAppTimeout	REG_SZ	2000	Уменьшение значения этого ключа реестра сокращает время, которое диспетчер задач Windows XP ожидает перед закрытием зависшей, не отвечающей на запросы программы. Рекомендованное значение — 2000. По умолчанию — 5000
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System	DisableTaskMgr	REG_DWORD	1	Отключение вызова диспетчера задач
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer	NoInstrumentation	REG_DWORD	1	Позволяет отключить слежение (по умолчанию) Windows XP за действиями пользователя, включая запускаемые программы и открываемые документы
HKEY_CURRENT_USER\ControlPanel\Desktop	MenuShowDelay	REG_SZ	0	Уменьшение значения этого ключа реестра сокращает паузы при раскрытии окон
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer	AlwaysUnloadDLL	REG_SZ	1	Позволяет динамически выгружать данные из памяти DLL-библиотек сразу после завершения работы приложения
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion	RegisteredOrganization RegisteredOwner	REG_SZ	Owner. Organization Owner	Используется для изменения информации о зарегистрированном владельце и организации, которая отражается в свойствах системы
HKEY_CURRENT_USER\ControlPanel\Desktop	AutoEndTasks	REG_SZ	1	Опция используется для автоматического закрытия без предупреждения всех зависших программ
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\CrashControl	AutoReboot	REG_DWORD	1	Используется для перезагрузки Windows XP после неожиданной остановки, краха
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\CrashControl	LogEvent	REG_DWORD	0	Используется для запрещения создания log-файла событий
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\8042prt\Parameters	CrashOnCtrlScroll	REG_DWORD	1	Позволяет вызывать «синий экран смерти» при нажатии комбинации клавиш: удерживая правый Ctrl, два раза нажмите на Scroll Lock
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement	SecondLevelDataCache	REG_DWORD	0	Опция позволяет вручную установить в реестре величину кэша второго уровня. Для этого добавьте ключ — значение в байтах. Если значение не задано или равно 0 (по умолчанию), то размер L2-кэша устанавливается автоматически
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement	DisablePagingExecutive	REG_DWORD	1	Для ускорения производительности оставляет исполняемый код ядра резидентным в памяти вместо вытеснения его по мере необходимости в виртуальную память на диск, то есть в swap. Такие ситуации случаются очень редко — только когда запущено очень много программ. Значения ключа: 0 — выключено (по умолчанию), 1 — включено. Для относительно безопасного изменения этого ключа рекомендуется иметь не менее 256 Мбайт памяти. Не применяйте эту опцию, если вы используете ждущий (standby) и спящий (hibernate) режимы
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement	IOPageLockLimit	REG_DWORD	RAM — значение ключа	Эта установка увеличивает число страниц, которые система будет читать или писать на жесткий диск за один раз. Исходя из этого должна вырасти производительность. Значения ключа задаются в шестнадцатеричной системе от 0×1000 до 0×20 000 Кбайт, то есть от 4 до 128 Мбайт. По умолчанию — 0, что соответствует 512 Кбайт
			128 Мбайт — 16 384 Кбайт	
			160 Мбайт — 32 768 Кбайт	
			256 Мбайт — 65 536 Кбайт	
			512 Мбайт — 13 1072 Кбайт	
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement	LargeSystemCache1	REG_DWORD	1	Включение этой опции увеличивает общую производительность Windows XP. Рекомендовано иметь 256 Мбайт памяти или больше. Измените значение ключа с 0 на 1
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement	DisablePagingExecutive	REG_DWORD	1	Запрещает записывать в файл подкачки код (драйверы и т.д.) и требует оставлять их всегда в физической памяти, если объем памяти больше 256 Мбайт. Установка значения равного 1 может существенно ускорить работу системы

## Оптимизация работы оборудования

После настройки операционной системы на максимальное быстродействие и минимальную загрузженность в том объеме, который допускает процесс редактирования реестра, пришла пора перейти к рассмотрению таких проблем, как оптимизация работы дополнительно установленного оборудования. При этом сама операционная система не позволяет изменить подобные функции в своих настройках, так что сделать это можно только путем редактирования реестра вручную. Настройки, позволяющие оптимизировать работу оборудования через реестр, представлены в табл. 2.

Таблица 2. Настройка оборудования

Ключ	Параметр	Тип данных	Значение	Описание
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem	NtfsDisableLastAccessUpdate	REG_DWORD	1	При использовании этой опции файловая система NTFS не будет обновлять метки времени последнего доступа к каждому файлу или папке. Попробуйте использовать эту возможность для увеличения производительности вашей дисковой подсистемы. Добавьте или измените значение ключа с 0 на 1
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem	NtfsDisable8dot3NameCreation	REG_DWORD	1	При использовании этой опции ОС Windows XP не будет создавать в разделе NTFS специальную таблицу для совместимости со старыми приложениями, содержащую все имена файлов и папок (данного логического диска), представленные в формате MS-DOS (8 символов в имени и 3 символа для расширения файла). В настоящее время подобная специфика NTFS не является актуальной, поэтому эту особенность желательно отключить для увеличения производительности
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4D36E96A-E325-11CE-BFC1-08002BE10318}\0000	EnableUDMA66	REG_DWORD	1	Позволяет включить поддержку UDMA-66 для вашего жесткого диска, что увеличивает производительность винчестера и уменьшает нагрузку на процессор при дисковых операциях
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdrom	AutoRun	REG_DWORD	0	Предотвращает автозапуск вставленного CD-ROM
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem\CDFS	CacheSize	REG_BINARY	ff ff 00 00	Опции позволяют устанавливать размер кэша, резервируемого для CD-ROM. Установка данных значений повышает скорость чтения
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem\CDFS	Prefetch	REG_DWORD	4000 (hex)	
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem\CDFS	PrefetchTail	REG_DWORD	4000 (hex)	

## Сетевые настройки

Редактор реестра позволяет производить настройку сети под нужды пользователя. В частности, имеется возможность редактировать параметры ключей для изменения максимального размера окна во время приема TCP-пакетов, включения поддержки больших TCP-окон и т.д. Подробное рассмотрение сетевых настроек в реестре следует начать с изучения ключей, которые хранят указанную информацию. В разделе HKEY\_LOCAL\_MACHINE\System\Controset001\Services\Tcpip можно обнаружить данные о текущих настройках TCP/IP. Кроме того, в разделе HKEY\_LOCAL\_MACHINE находится информация о зарегистрированном пользователе, его имени, сетевом провайдере и пр. Настройки, позволяющие оптимизировать работу в сети, представлены в табл. 3.

Таблица 3. Сетевые настройки

Ключ	Параметр	Тип данных	Значение	Описание
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters	TCPWindowSize	REG_DWORD	2238 (для модемных соединений)	Параметр определяет максимальный размер окна для приема TCP-пакетов, предлагаемый операционной системой. Окно приема определяет количество байтов, которые отправитель может передать, не получая подтверждения. Вообще, чем больше окно, тем лучше работа в сетях с высокой пропускной способностью. Значение по умолчанию составляет 0×2238 и для модемных соединений не рекомендуется устанавливать и изменять его
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters	EnablePMTUDiscovery	REG_DWORD	1	Включение этой опции заставляет TCP автоматически определять MTU
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters	Tcp1323Opts	REG_DWORD	0/1/2/3	Совместно с опцией TCP/IP Window Size, описанной выше, включает поддержку больших TCP-окон. Это особенно актуально в сетях с большой пропускной способностью

## Настройки интерфейса

Настройка и оптимизация интерфейсной части операционной системы всегда вызывает интерес у большинства пользователей. Преобразить внешний вид рабочего стола, удалить ненужные иконки и папки, оптимизировать цвета и запретить их дальнейшее изменение, да и просто создать удобную рабочую обстановку помогут возможности по редактированию реестра. При этом следует помнить, что изменение таких настроек не увеличивает быстродействия и не улучшает работы операционной системы.

## Примеры.

Приведем несколько примеров. Смена «обоев» на рабочем столе с последующим блокированием возможности возврата — путем удаления закладки в окне Display Properties. Включить автозагрузку путь к файлу выключения или перезагрузки компьютера, что повлечет за собой его мгновенное выключение сразу после загрузки. Техническая реализация этих возможностей представлена в табл. 4.

Таблица 4. Нестандартные настройки

Ключ	Параметр	Тип данных	Значение	Описание
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	Необходимо создать ключ с любым именем, например Shutdown	REG_SZ	C:\WINDOWS\system32\shutdown.exe	После загрузки операционной системы компьютер автоматически выключается
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	Необходимо создать ключ с любым именем, например Video	REG_SZ	C:\video.avi	Одновременно с программами автозагрузки открывается любой файл в соответствующем приложении, например видеофайл
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System	NoDispBackgroundPage	REG_DWORD	1	Ставим «обои» на рабочем столе и запрещаем их изменение
HKEY_LOCAL_MACHINE\ дальше System\CurrentControlSet\control\Shutdown	FastReboot	REG_DWORD	1	Меняем значение Shutdown на Restart
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer	NoDesktop	REG_DWORD	1	Скрываем на рабочем столе все, кроме «Пуска»
HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet Explorer\Restrictions	NoNavButtons	REG_DWORD	1	Блокировка кнопок «Вперед» и «Назад» в Internet Explorer
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ создайте раздел Winlogon	LegalNoticeCaption	REG_SZ	Заголовок окна	Всплывающее окно с произвольным текстом перед вводом пароля для входа в Windows
	LegalNoticeText	REG_SZ	Текст	
HKEY_CURRENT_USER\Software\Microsoft\Office\8.0\Word\Options	LiveScrolling	REG_SZ	(0=выкл., 1=вкл.)	Заставить MS Word реально прокручивать скроллинг

Вопросы практического применения процесса редактирования ключей реестра вручную становятся актуальными почти для всех владельцев персональных компьютеров, как только те начинают переходить в разряд продвинутых пользователей и у них появляются потребности в более высокой производительности операционной системы. Нужно отметить, что работа с реестром необходима, как говорится, для образования и серьезно расширяет представления о работе Windows XP в целом, поскольку реестр является сердцем операционной системы. Поняв и осмыслив его структуру, вы всегда сможете поддерживать ОС в работоспособном состоянии.

#### Практические задания

Задание 1. Изменить в реестре параметр, который выводит текущую версию ОС на рабочий стол

Ключ	Параметр	Тип данных	Значение
HKEY_CURRENT_USER\Control Panel\Desktop	PaintDesktopVersion	REG_DWORD	1

Задание 2. Изменить в реестре параметр, который изменяет надпись папки «Мой компьютер»

Ключ	Параметр	Тип данных	Значение
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}	Default	REG_SZ	Введите новую надпись

Задание 3. Изменить в реестре параметр, который изменяет надпись папки «Корзина»

Ключ	Параметр	Тип данных	Значение
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\CLSID\{645FF040-5081-101B-9F08-00AA002F954E}	Default	REG_SZ	Введите новую надпись

Задание 4. Изменить в реестре параметр, который изменяет надпись папки «Мой компьютер»

Ключ	Параметр	Тип данных	Значение
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\CLSID\{871C5380-42A0-1069-A2EA-08002B30309D}	Default	REG_SZ	Введите новую надпись

Задание 5. Добавить значок «Корзина» в папку «Мой компьютер»

Ключ	Параметр	Тип данных	Значение
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\MyComputer\NameSpace	Default	REG_SZ	Добавить подраздел {645FF040-5081-101B-9F08-00AA002F954E}

Задание 6. Скрыть и восстановить вкладки окна «Свойств экрана» - «Темы оформления», «Рабочий стол», «Заставка», «Параметры».

Ключ	Параметр	Тип данных	Значение
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System	NoDispAppearancePage	REG_DWORD	1 ( если 0 восстанавливает параметр)
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System	NoDispBackgroundPage	REG_DWORD	1 ( если 0 восстанавливает параметр)
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System	NodispScrSavPage	REG_DWORD	1 ( если 0 восстанавливает параметр)
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System	NoDispSettingsPage	REG_DWORD	1 ( если 0 восстанавливает параметр)

Задание 7. Скрыть все значки на рабочем столе кроме «пуск»

Ключ	Параметр	Тип данных	Значение
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer	NoDesktop	REG_DWORD	1 ( если 0 восстанавливает параметр)
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System	NoDispBackgroundPage	REG_DWORD	1 ( если 0 восстанавливает параметр)

**К сдаче лабораторной предоставляются:** справочник команд управления процессами и умение работать с этими командами.

## Лабораторная работа №8. 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* - эти 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 непрограммиста 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, то есть 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.

### **Проектировании 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 изданных 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 прочее.

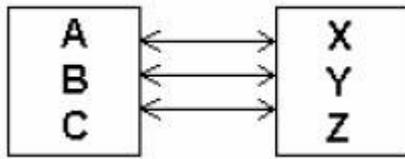
Ключ является идентификатором, уникально определяющим запись об объекте. Ключ в наборе записей может включать в себя один (простой ключ) или более (составной) элементов данных, совокупность которых однозначно идентифицирует запись об объекте.

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**

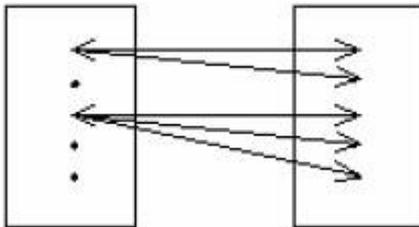
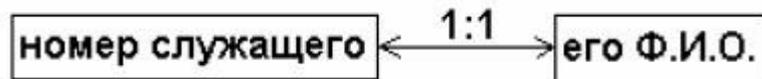
Традиционными средствами для представления характера взаимосвязей между парами связанных элементов данных являются отображения (двусторонние связи) и ассоциации (односторонние связи).

## Отображения



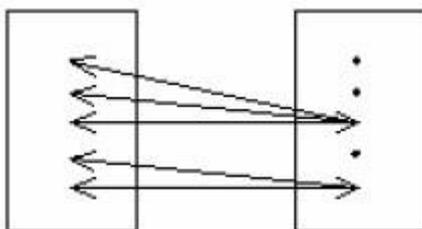
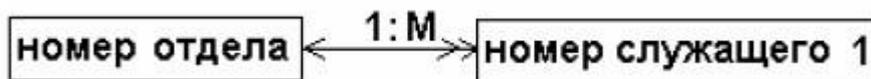
Рассмотрим основные типы отображений и их изображение на диаграммах.

Отображение 1:1. Один экземпляр элемента, от которого направлена связь, идентифицирует один экземпляр элемента, к которому направлена связь. Связь уникальна как слева направо, так и в обратном направлении.

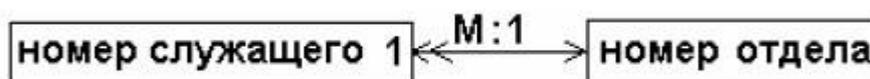


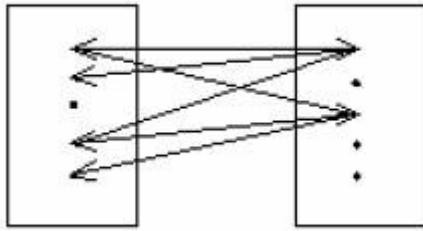
Отображение 1:M. Один экземпляр элемента с выходящей связью идентифицирует несколько экземпляров элемента данных принимающих эту связь.

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

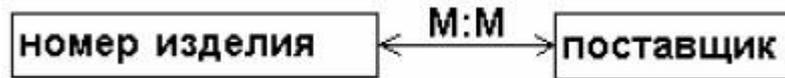


Отображение M:1. Это отображение подобно предыдущему с той лишь разницей, что слева направо связь уникальна, а в обратном направлении не является таковой.





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.



### 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*.

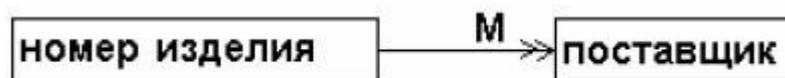
Существует три типа ассоциаций:

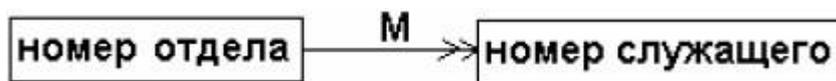
- простая (тип 1);
- сложная (тип М);
- условная (тип С).

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

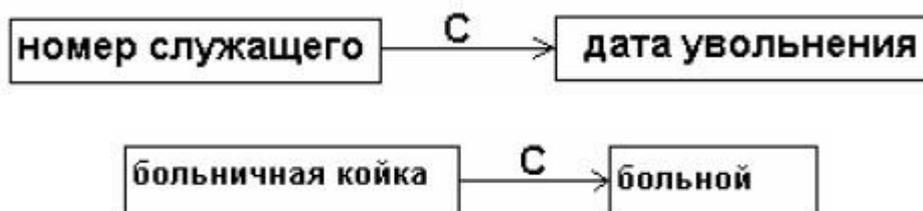


Сложная ассоциация, в которой экземпляр элемента данных с выходящей связью идентифицирует несколько (один или более) экземпляров элементов данных, к которым эта связь направлена. Идентификация необязательно является уникальной и может представлять многозначную зависимость. На диаграмме сложную ассоциацию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





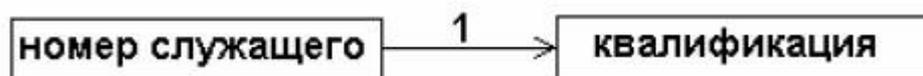
Условная ассоциация, в которой для элемента данных с выходящей связью не идентифицируется ни одного экземпляра элемента данных, к которому эта связь направлена, но она существует и является уникальной.



### 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.



Полностью составной ключ включает два 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,

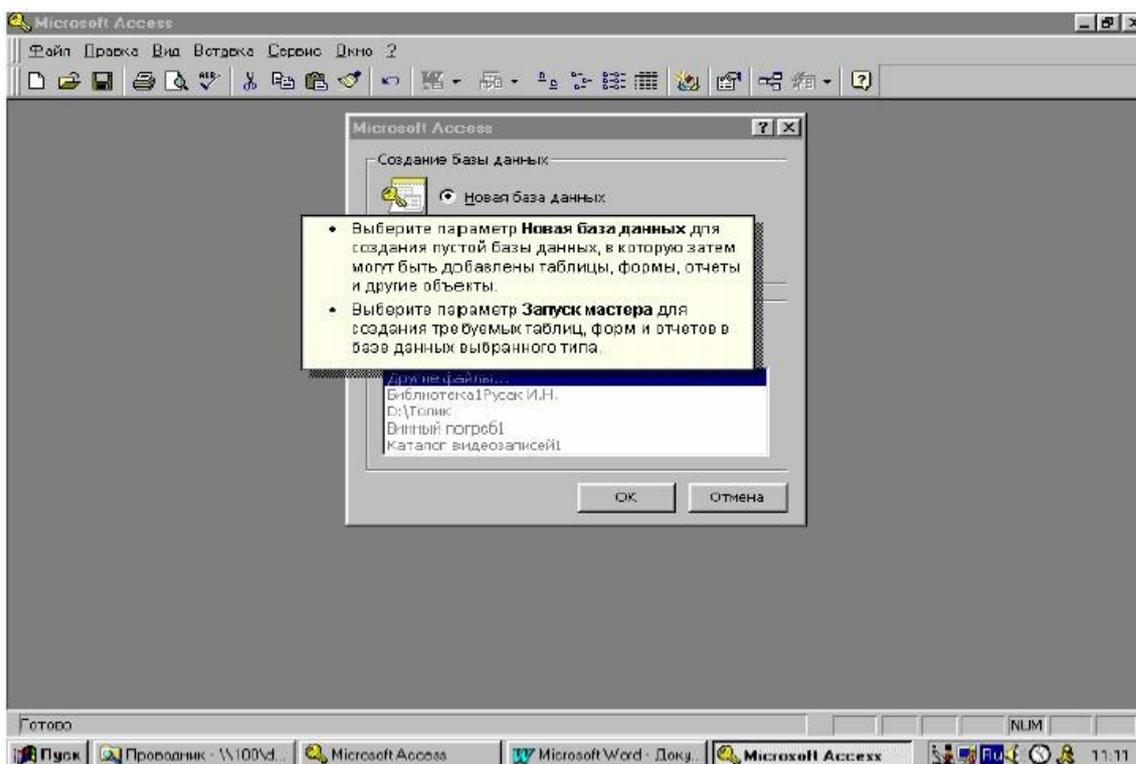


Рис. 1.1. Окно начального диалога в MS Access

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 поставщика, заказах, а также сведения о предприятии.



Рис. 1.2. Окно «Файл новой базы данных»

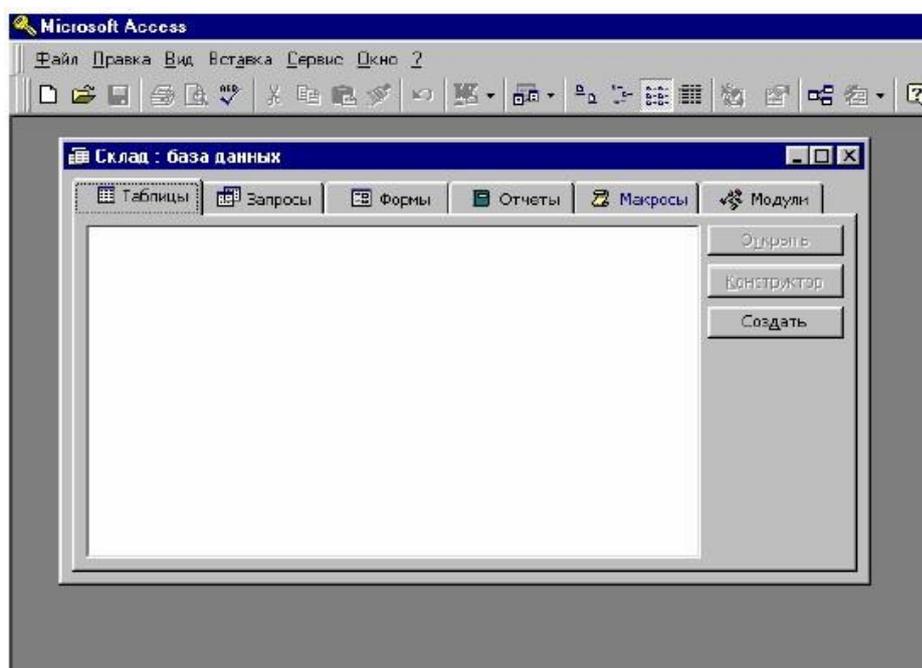


Рис. 1.3. Окно базы данных

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 **Ok**. 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"**, 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

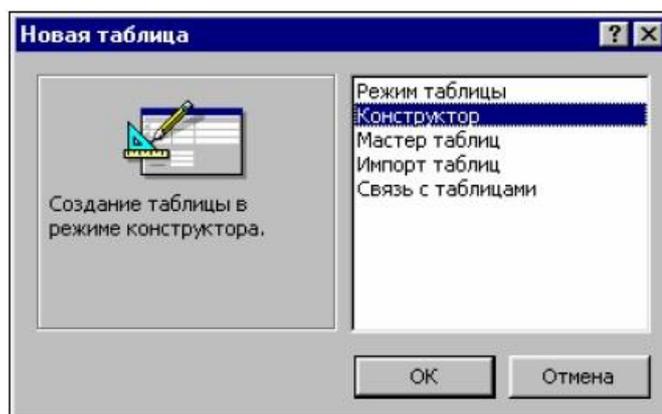
Создать таблицу, содержащую сведения о товарах.

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:

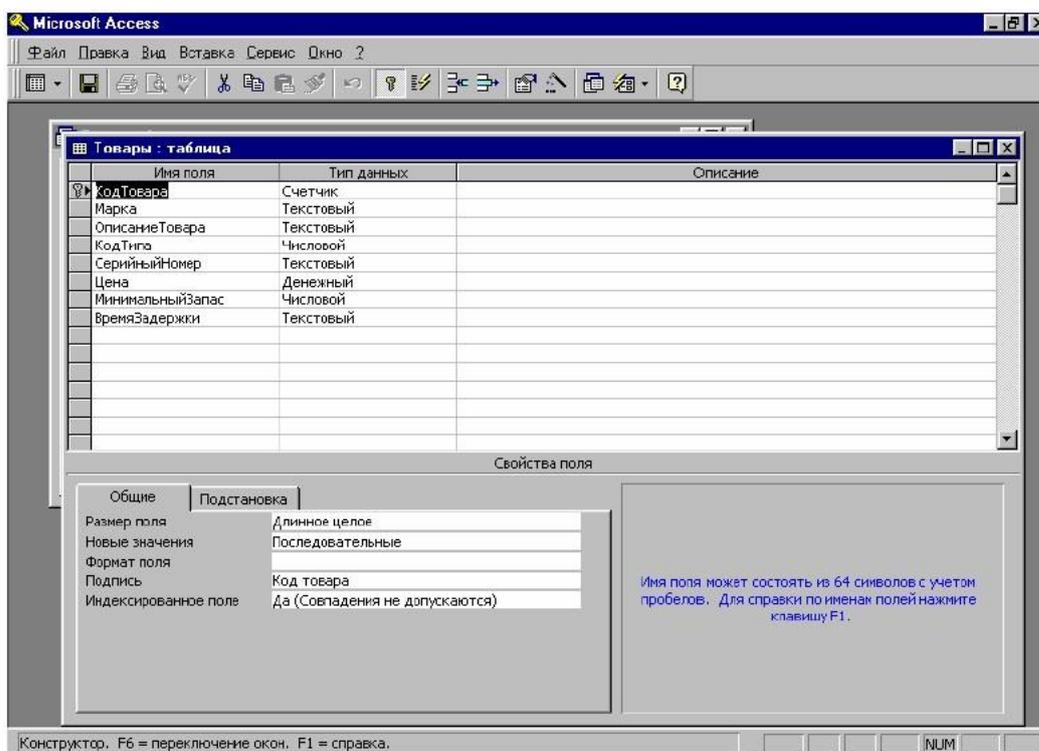
Код товара	код, однозначно определяющий конкретный товар
Марка	марка товара (указанная на упаковке)
Описание товара	описание, уточняющее состав данного товара
Код типа	код типа товара
Серийный номер	серийный номер, присвоенный партии товара
Цена	стоимость единицы товара
Минимальный запас	минимальный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



**Рис. 1.4. Выбор режима создания новой 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. 1.5. 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":

Свойства			
Число записей:	5	Дата изменения:	10.03.98 9:19:07
Дата создания:	10.03.98 9:19:03	Допускаются обновления:	Истина
Столбец			
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).

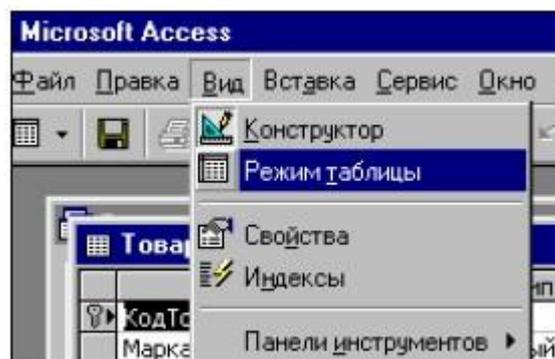


Рис. 1.6. Выбор режима таблицы

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 сведениях 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) С помощью **Мастера таблиц** создать таблицу "Товары" ввести сведения о товарах:

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

3) В режиме конструктора создать таблиц "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 (**Сервис – Схема Данных**), создать связи между таблицами.

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

#### **Вопросы для самопроверки**

1. База данных. Физическое и логическое представления данных.
2. Модель данных. Атрибут. Отношение. Ключ отношения.
3. Нормализация. Свойства реляционных структур.
4. Предметная область. Информационный объект. Элемент данных. Идентификатор.
5. Отображения. Один к одному. Один к М. М к М.
6. Ассоциация. Виды ассоциаций.
7. Реляционные ключи.

## Лабораторная работа №9. 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 критерии 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 будет предложено выбрать один из вариантов создания запроса.

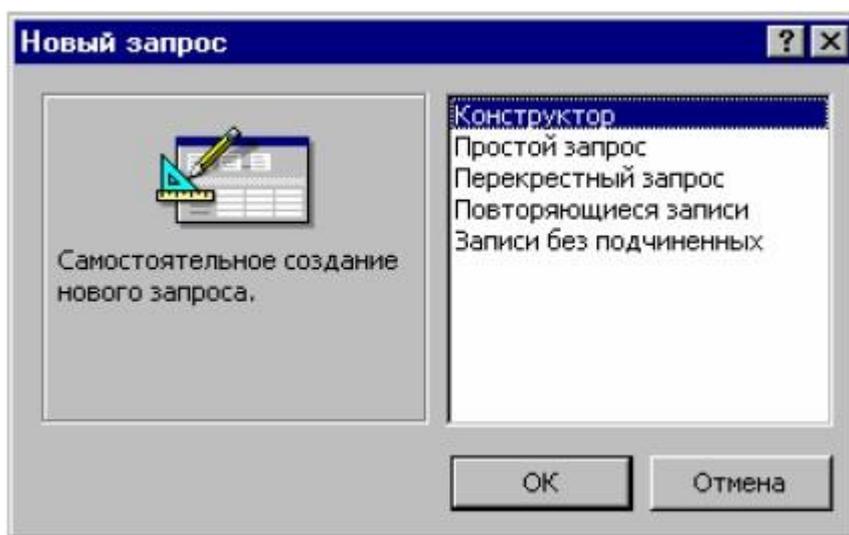


Рис. 2.1. Окно нового запроса

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.

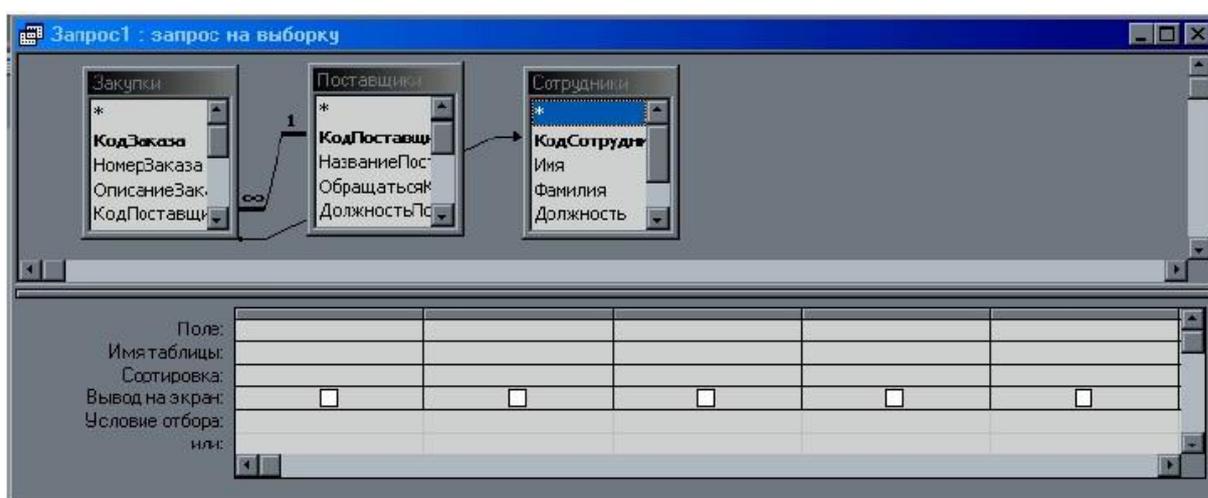


Рис. 2.2. Конструктор запроса

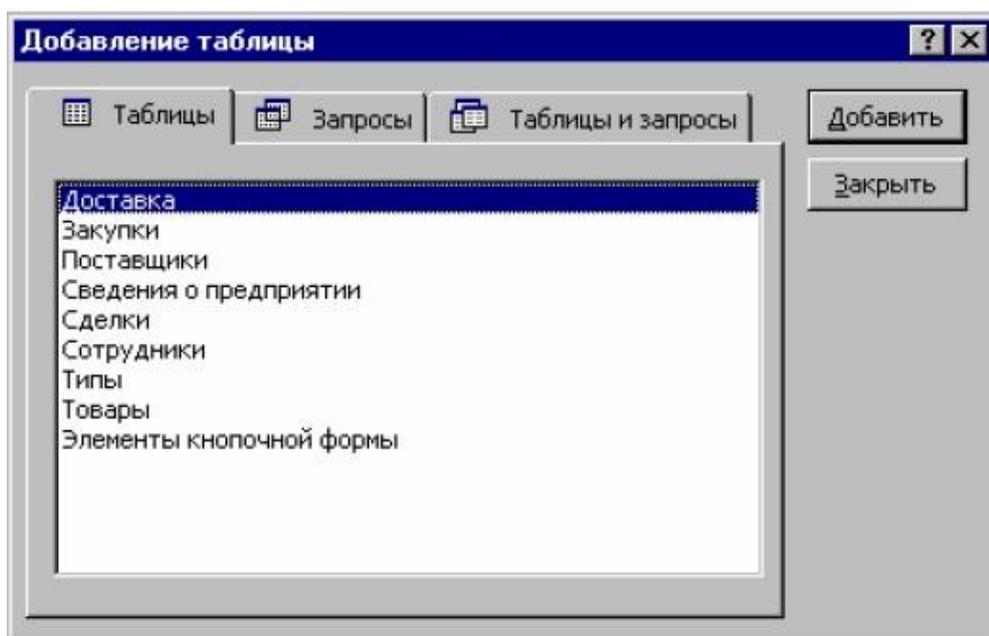


Рис. 2.3. Окно добавления таблиц

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.

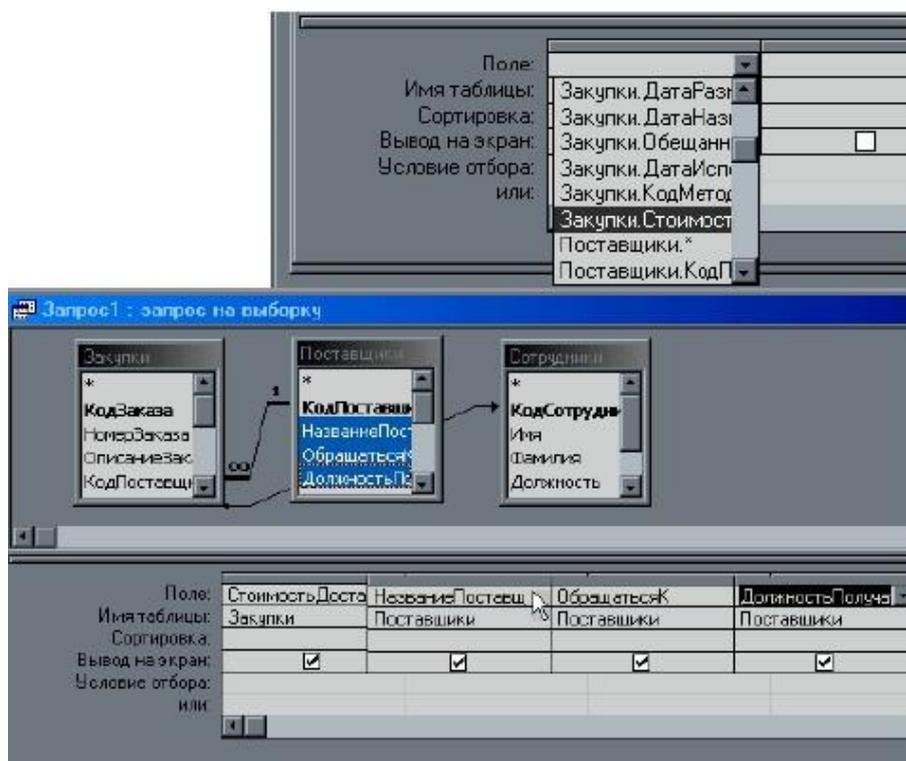


Рис. 2.4. Работа с бланком запроса в режиме конструктора

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").

### Выполнение работы (варианты)

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 России), оформивших заказ на сумму, превышающую 300\$;
- 6) вывести записи о поставщиках, расположенных в Москве и Санкт-Петербурге;
- 7) получить данные о клиентах, которые проживают в Екатеринбурге и Хабаровске или которые сделали заказ на поставку компьютеров;
- 8) выдать информацию о клиентах, проживающих в городах, названия которых начинаются на букву "Д", или в Московской области, но не заказавших телевизоры;
- 9) получить сведения о заказах, срок поставки которых истекает после 01.06.2005;
- 10) вывести информацию о поставщиках города Москвы, которые должны выполнить заказы в период времени с 15.04.2001 по 18.12.2003.
- 11) получить сведения о клиентах, заказавших товары стоимостью от 200 до 400\$, получающих заказы 14.06.2005 или 20.09.2005.

### **Вопросы для самопроверки**

1. Запросы СУБД MS Access. Создание запросов на выборку.
2. Создание параметрического запроса.
3. Создание перекрестного запроса.

## Лабораторная работа №10. 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 (в столбец, ленточная и 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 окна.

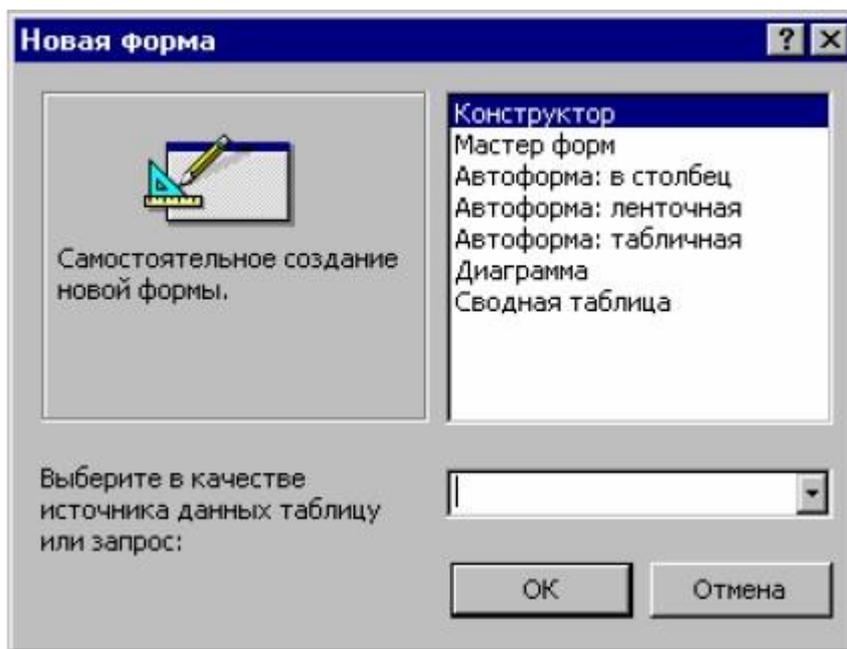


Рис. 3.1. Выбор полей в мастере форм

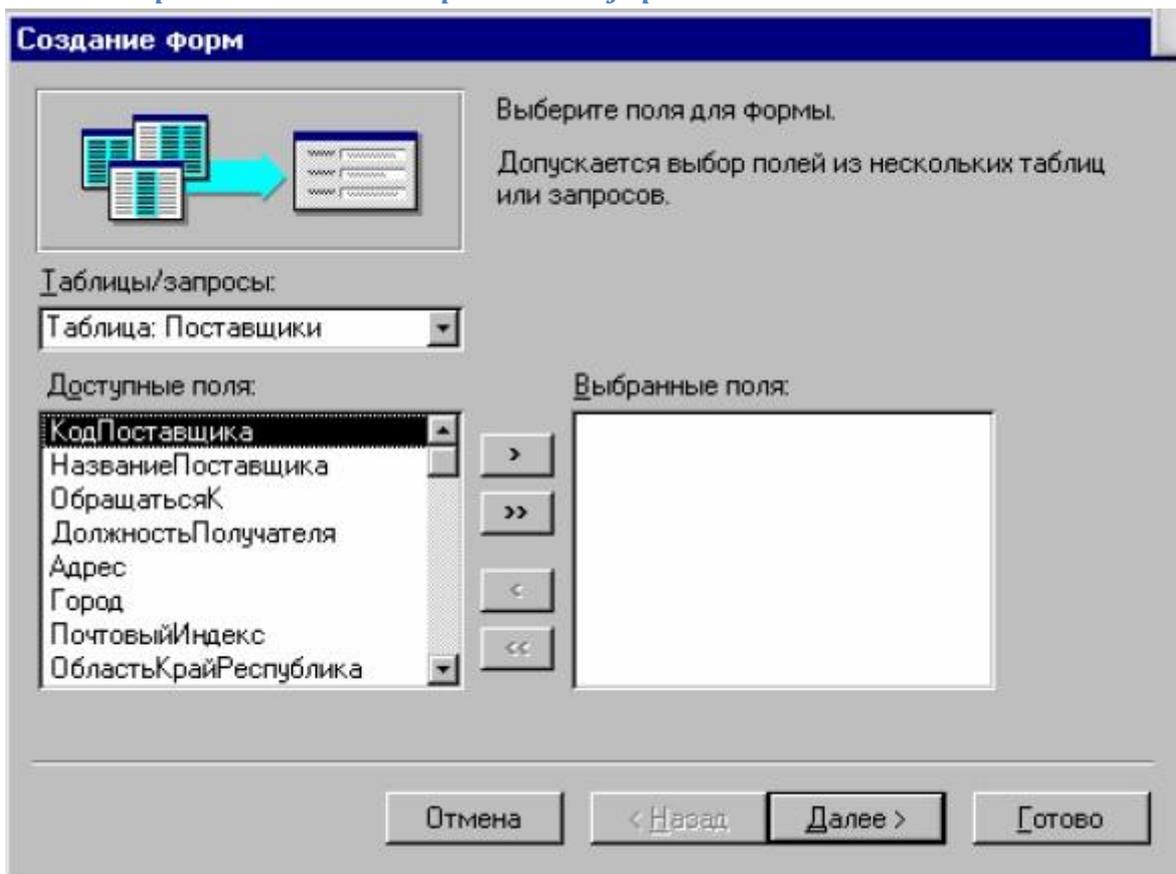
Наиболее оптимальным представляется следующий прием. Сначала при помощи **Мастера форм** создать заготовку формы, затем в режиме **Конструктора** внести необходимые изменения и дополнения. Например, изменить порядок расположения полей, добавить вычисляемые поля, элементы управления и т.д. Впрочем, часто 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) выбираем режим **Мастер форм**. В раскрывающемся списке в нижней части окна следует указать источник данных создаваемой формы. В нашем случае это таблица **Поставщики**. После подтверждений (нажатием кнопки **Ок**) запускается мастер. Рассмотрим последовательно шаги построения формы с помощью **Мастера**.

*Шаг 1. Выбрать поля для отображения в форме.*



**Рис. 3.2. Выбор полей в мастере форм**

Выберем следующие поля:

**КодПоставщика;**

**НазваниеПоставщика;**

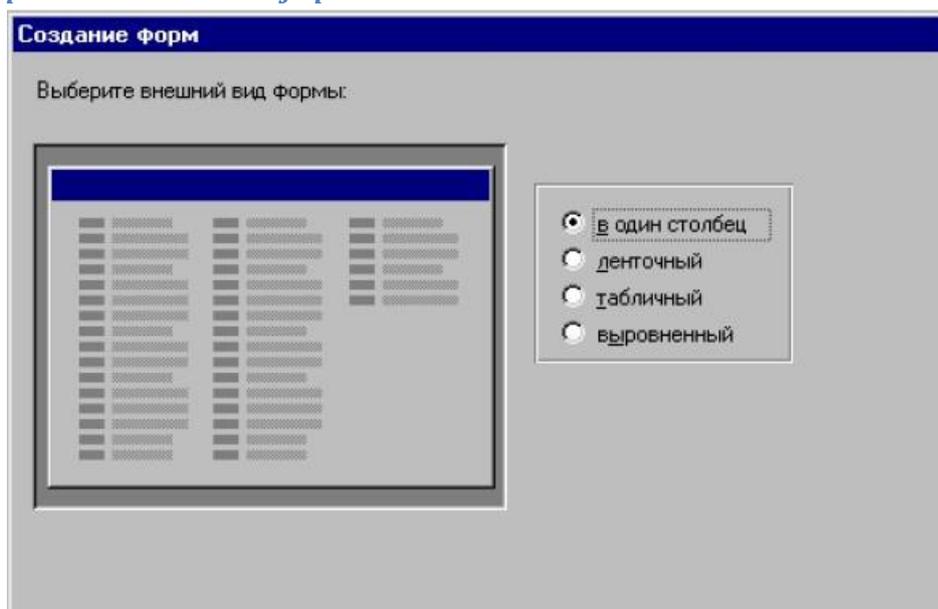
**Город;**

**ОбластьКрай Республика;**

**Страна;**

**НомерТелефона.**

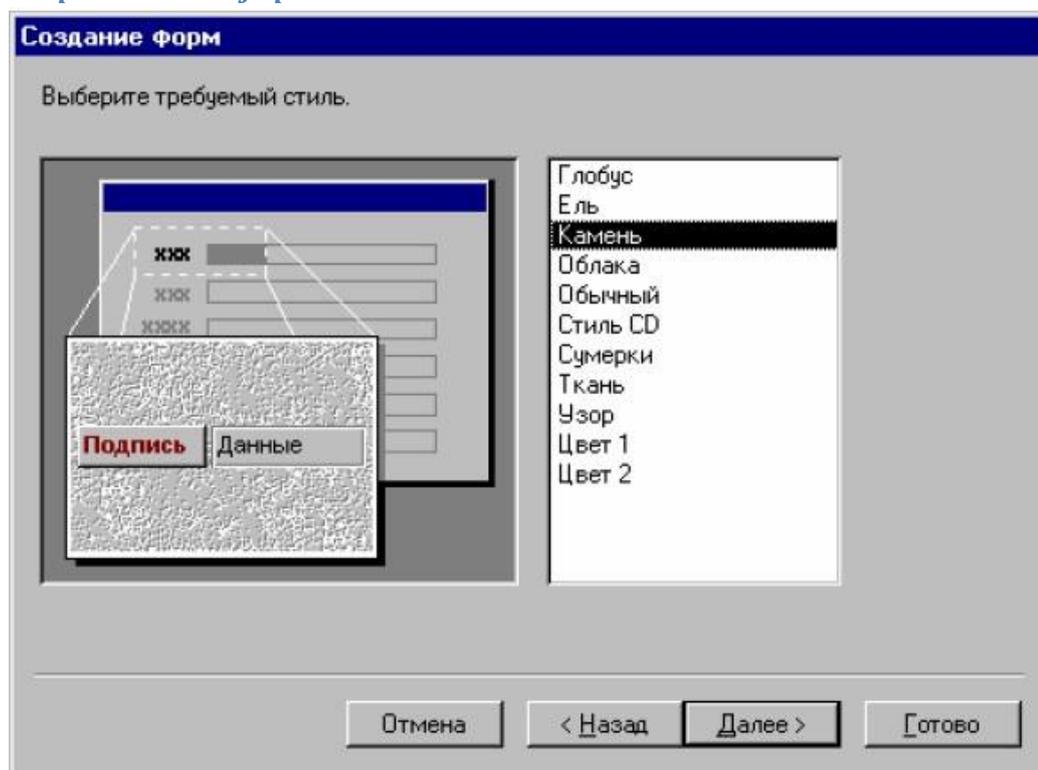
*Шаг 2. Выбрать внешний вид формы.*



**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.

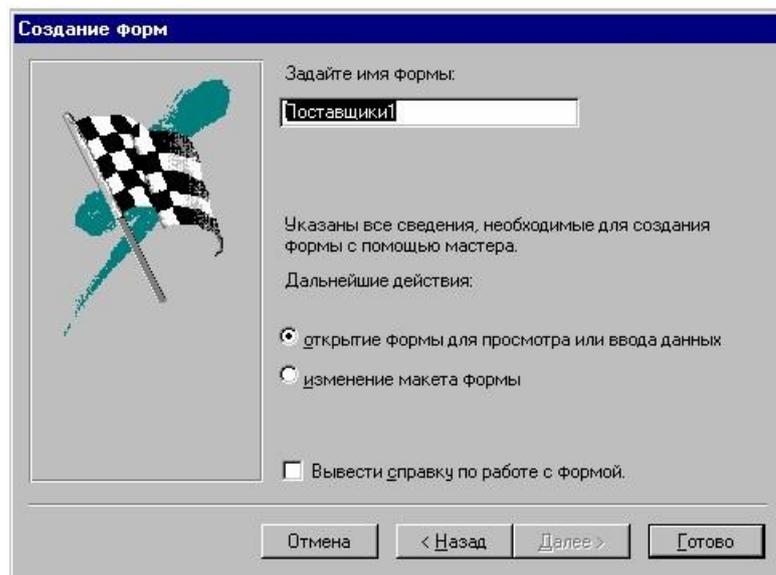


Рис. 3.5. Указание имени формы и действия по завершении процесса создания

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).

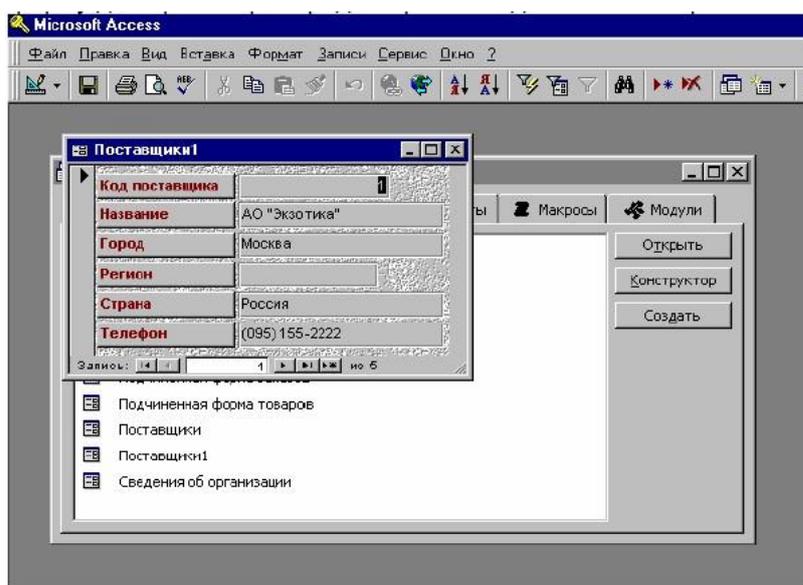
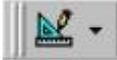


Рис. 3.6. Форма «Поставщики», полученная с помощью Мастера

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

Для редактирования формы в режиме **Конструктора** можно воспользоваться одним из следующие 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).

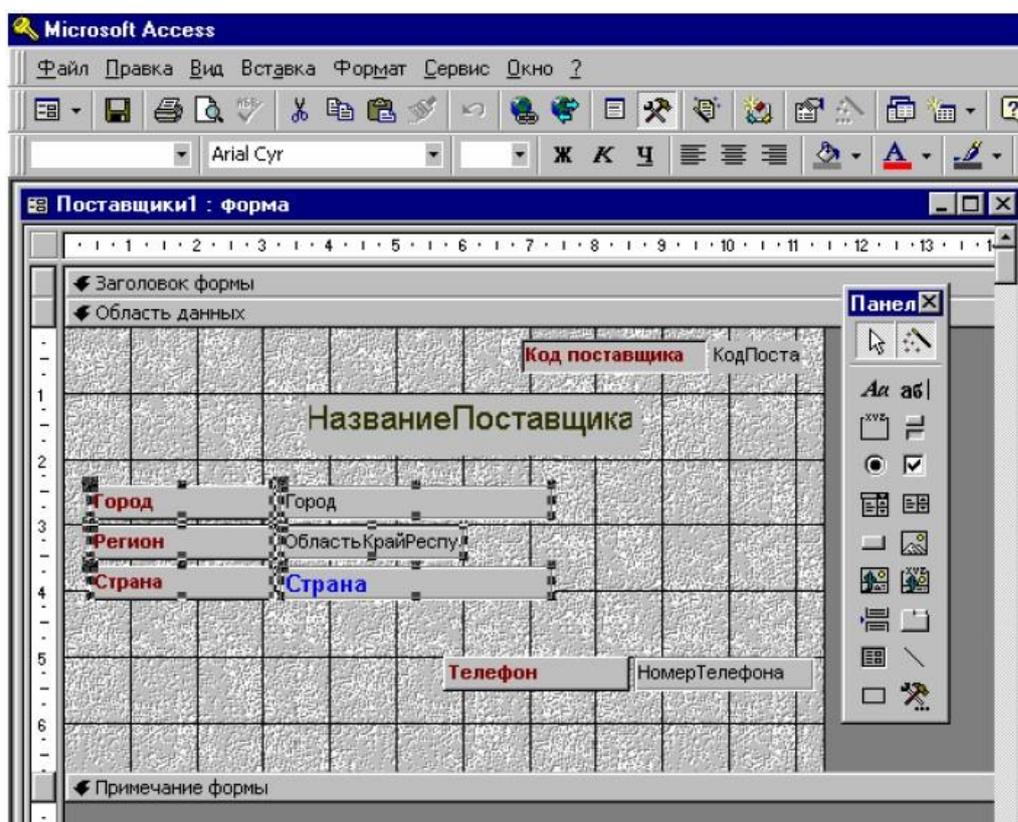


Рис. 3.7. Форма в режиме конструктора

### Выполнение работы (варианты)

Создать форму в соответствии с вариантом задания.

1 Форму «**Клиенты**», отображающую следующие сведения о клиентах:

- **Наименование**



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 указанной 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.

### **Вопросы для самопроверки**

1. Формы СУБД MS Access. Как создаются формы. Шаги построения формы.
2. Отчеты и элементы управления СУБД MS Access.
3. Типы данных СУБД MS Access.

**Перечень основной литературы:**

1. Васильев, В. В. Информационные технологии в библиотечном деле : учеб.-метод. пособие / Отв. ред. серии О. Р. Бородин. – М. : Либерия-Бибинформ, 2013. – 368 с.
2. Ларсон, Б. Разработка бизнес-аналитики в Microsoft SQL Server. – СПб. – М. – Н. Новгород : Питер, 2011. – 688 с.
3. Макин, Дж. К. Проектирование серверной инфраструктуры баз данных Microsoft SQL SERVER : учебный курс : пер. с англ. / Дж. К. Макин, Майк Хотек. – М. : Русская редакция, 2012. – 560 с.

**Перечень дополнительной литературы:**

1. Избачков, Ю. С. Информационные системы: учебник / Ю. С. Избачков, В. Н. Петров. - 3-е изд. - СПб.: Питер, 2011. - 656 с.
2. Томас, О. Оптимизация и администрирование баз данных Microsoft SQL Server 2008 : официальное пособие для самоподготовки : пер. с англ. : учебный курс Microsoft. – М. : Русская редакция, 2012. – 601 с.
3. Основы проектирования и разработки реляционных баз данных : (Спец. 075200 – Компьютерная безопасность) : учеб. пособие / авт.-сост.: О. М. Лепешкин, Д. Л. Осипов ; Федеральное агентство по образованию, Ставроп. гос. ун-т. – Ставрополь : Изд-во СГУ, 2010. – 203 с.