

**Министерство образования и науки Украины
Донбасская государственная машиностроительная академия**

Фокин А.Г., Гетьман И.А.

**СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ
ACCESS В ЭКОНОМИЧЕСКИХ РАСЧЕТАХ**

**Учебное пособие
для студентов высших учебных заведений**

Утверждено
на заседании Ученого Совета ДГМА
Протокол №3 от 1 декабря 2005 г.

Краматорск 2005

УДК 004.65
ББК 22.18
Ф74

Рецензенты:

А.А.Зори, доктор технических наук, профессор, заведующий кафедрой электронной техники ДНТУ (Донецкий национальный технический университет)

С.О.Калоеров, доктор физико - математических наук, профессор кафедры теории упругости и вычислительной математики ДонНУ (Донецкий национальный университет)

Фокин А.Г., Гетьман И.А.

Ф74 Система управления базами данных Access в экономических расчетах: Учебное пособие для студентов высших учебных заведений. – Краматорск: ДГМА, 2005. – 136 с.

ISBN 966 – 379 -025 - 3

Приведены краткие теоретические сведения по теории баз данных. Дано краткое описание СУБД Access. Рассмотрены вопросы проектирования концептуальной модели базы данных. Подробнее описаны основные объекты Access: таблицы, запросы, формы, отчеты, а также порядок их построения и использования. Приведены практические задания для самостоятельной работы. Рассмотрен пример их выполнения.

ISBN 966 – 379 -025 - 3

ББК 22.18

© **А.Г.Фокин,**
И.А.Гетьман, 2005
© **ДГМА, 2005**

СОДЕРЖАНИЕ

Введение.....	5
1 Элементы теории баз данных.....	6
1.1 Понятие базы данных.....	6
1.2 Модели данных в БзД.....	8
1.3 Система управления базами данных.....	11
1.4 Архитектура СУБД.....	11
2 База данных Access.....	13
2.1 Объекты ACCESS.....	13
2.2 Файл базы данных.....	14
2.3 Начальное окно диалога.....	15
2.4 Окно базы данных.....	16
2.4.1 Создание объекта.....	17
2.4.2 Работа с объектом.....	18
3 Таблицы.....	19
3.1 Назначение таблиц.....	19
3.2 Проектирование концептуальной схемы базы данных.....	20
3.3 Создание таблицы.....	22
3.4 Использование свойств подстановки.....	30
3.5 Определение связей.....	31
3.6 Внесение изменений в определение таблицы.....	35
3.7 Просмотр и актуализация данных таблицы.....	37
(режим таблицы).....	37
4 Запросы.....	39
4.1 Назначение запросов.....	39
4.2 Типы запросов.....	41
4.3 Создание запроса на выборку.....	42
4.4 Выражения.....	45
4.5 Создание запроса на изменение.....	46
5 ФОРМЫ.....	48

5.1 Назначение форм	48
5.2 Элементы формы	49
5.3 Виды форм	51
5.4 Построение формы	52
5.5 Создание и внедрение подчиненных форм	57
5.6 Связывание форм с помощью командных кнопок	60
6 Отчеты	61
6.1 Назначение отчетов	61
6.2 Элементы отчета.....	62
6.3 Построение отчета.....	63
7 Макросы.....	66
7.1 Назначение макросов	66
7.3 Выполнение макросов	69
7.4 Ссылки на объекты форм и отчетов	71
7.5 Группы макросов. Условия в макросах.....	72
8 Практические задания к лабораторным работам	74
8.1 Лабораторная работа №1. Создание схемы базы данных	74
8.2 Лабораторная работа №2. Создание и использование запросов.....	93
8.3 Лабораторная работа №3. Создание и использование форм.....	100
8.4 Лабораторная работа № 4. Создание макросов.	104
Связывание форм	104
8.5 Лабораторная работа № 5. Создание и использование отчетов.....	104
9 Пример выполнения лабораторных работ	113
9.1 Указания к лабораторной работе №1	113
9.2 Указания к лабораторной работе №2	118
9.3 Указания к лабораторной работе №3	126
9.4 Указания к лабораторной работе № 4	130
10 Схемы баз данных	134
Список литературы	137

ВВЕДЕНИЕ

Для успешного функционирования разных организаций и предприятий необходимо наличие развитой информационной системы, которая могла бы реализовать автоматический сбор, актуализацию и обработку данных. Для организации таких систем широко используются системы управления базами данных (СУБД). Среди наиболее распространенных таких систем – система управления базами данных **Access**, которая входит в состав комплекса **Microsoft Office**. Однако следует знать, что **Access** не входит в базовый комплекс **Microsoft Office**. Это говорит о том, что стандартный комплекс **Microsoft Office** предназначен для широкого круга рядовых пользователей, а СУБД **Access** ориентирована на профессионалов. Таким образом, можно говорить про СУБД **Access** как профессиональную систему для создания и эксплуатации информационных систем предприятий.

СУБД **Access** находит широкое применение при решении многих экономических задач, при автоматизации систем управления экономическими системами.

1 ЭЛЕМЕНТЫ ТЕОРИИ БАЗ ДАННЫХ

1.1 Понятие базы данных

Долгое время для решения каждой задачи на ЭВМ использовалась только своя совокупность данных. Данные оформлялись в виде наборов данных или файлов. Использование файлов одной программы в других программах было затруднительным.

Например, в программе расчета зарплаты есть файл о сотрудниках. При использовании его в программе назначения субсидий (или, например, обрабатывающей результаты голосования по месту жительства) потребуются дополнительная информация, отсутствующая в этом файле (адрес, площадь квартиры и т.д.). Значит, необходимо либо создать новый файл, что приводит к дублированию информации (избыточности), либо расширить существующий, что влечет за собой необходимость корректировки всех использующих этот файл программ.

Недостатки традиционной (файловой) организации данных инициировали поиск новых методов организации данных, что привело к появлению в конце 60-х годов концепции баз данных (БзД).

База данных (БзД) – это единое централизованное хранилище данных определенной предметной области, к которому имеют доступ несколько приложений (программ или конечных пользователей).

При этом в максимальной степени должны соблюдаться следующие принципы:

- минимальная избыточность данных;
- целостность данных;
- независимость данных от используемых их приложений;
- возможность развития;

- высокая производительность (высокая пропускная способность, допустимое время реакции системы на обращения к ней);
- возможность многокритериального поиска данных приложениями различного уровня;
- безопасность и секретность;
- минимальные затраты на создание и эксплуатацию.

Функции организации БзД, обеспечения независимости данных от программ, организации доступа к данным и т.д. реализует система управления базой данных (СУБД).

СУБД – это комплекс программ, обеспечивающих создание, размещение на магнитном диске и поддержание в актуальном состоянии базы данных, а также обеспечение доступа к БзД прикладных программ и внешних пользователей.

Используется понятие банка данных. **Банк данных** – база данных и СУБД.

Базы данных имеют несколько уровней представления.

Самый низкий – *физический*, т.е. размещение данных на физическом носителе (диске). Доступен узкому кругу специалистов, связанных с вопросами производительности системы. В настоящее время реально физическую организацию базы данных представляют только разработчики СУБД.

Общая логическая организация данных (концептуальная схема или концептуальная модель) – описание данных администратором БзД, имеющим дело с глобальным описанием данных.

Самый верхний – *логический (логическая организация)*. Это описание данных в приложении, как оно видится конечному пользователю.

Итак, в базе данных данные организуются и хранятся в интересах решения всех задач предметной области. Каждая программа или пользователь выбирает только требуемые ей данные (рис. 1).

Обеспечивается независимость прикладных программ от данных. Это значит, что изменения в данных не влекут за собой изменения программ. Аналогично изменения в логике программы не вызывают необходимости изменения механизма доступа к данным.

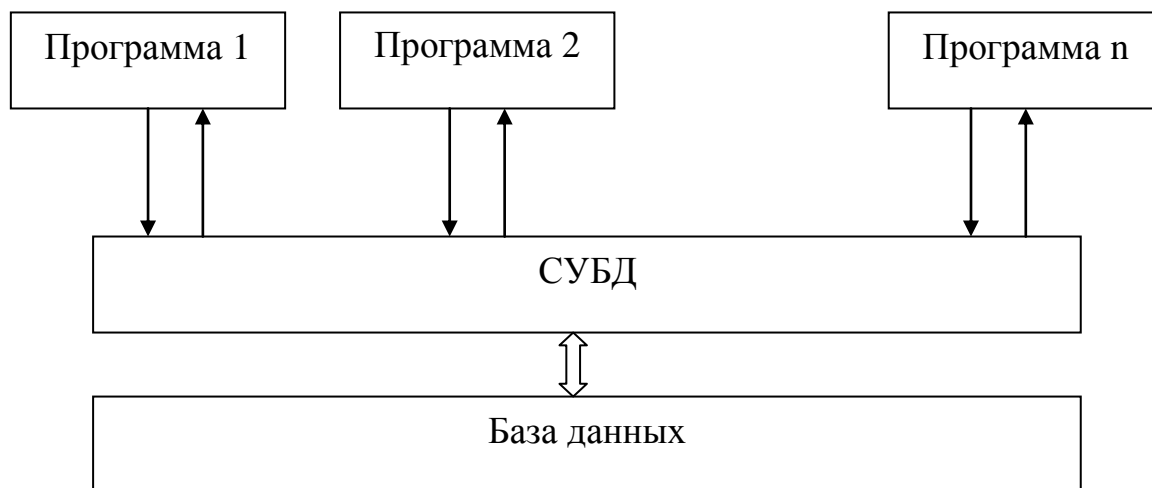


Рисунок 1 – Организация банка данных

1.2 Модели данных в БзД

Между элементами БзД (полями и записями) существуют определенные связи. Связи (отношения) между записями принято делить на три типа:

“*один-к-одному*” – каждой записи одного типа соответствует одна запись другого типа;

“*один-ко-многим*” – каждой записи одного типа соответствует несколько записей другого типа;

“*многие-ко-многим*” – каждой записи одного типа соответствует несколько записей другого типа и наоборот – каждой записи другого типа соответствуют несколько записей первого типа.

В зависимости от характера этих связей различают следующие типы организации данных (модели данных).

1 Линейная

Это, собственно говоря, обычный файл, описание которого вынесено из программы.

2 Иерархическая (древовидная)

Между отдельными элементами установлена связь «один-ко-многим».

Пример – структура предприятия, института, дерево объектов в WINDOWS 95 или просто файловая система в DOS (рис. 2).

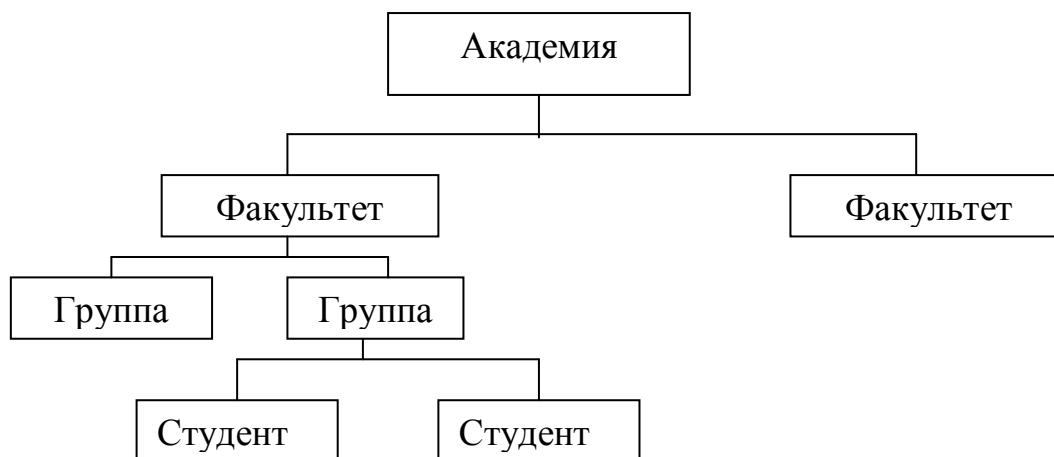


Рисунок 2 – Иерархическая структура данных

3 Сетевая

Связь “многие ко многим”. Пример – состав изделия. Изделие состоит из узлов, а те из других узлов и деталей и т.д. В свою очередь детали и узлы могут входить в другие узлы (рис. 3).

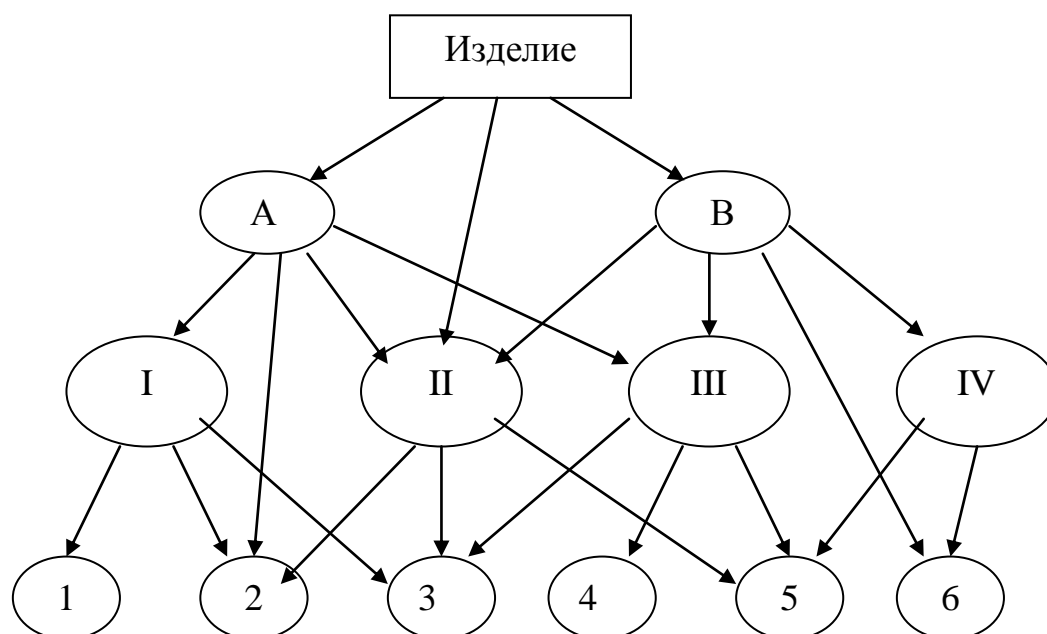


Рисунок 3 – Сетевая структура данных

4 Реляционная

Концепция реляционной БзД разработана Е.Коддом в 1970 г. В основе этой модели лежит понятие *отношение*, представляемое в виде двухмерной таблицы.

Отношение (таблица) представляется в компьютере в виде файла данных. Строка соответствует записи (компоненту), а столбец – полю. Часто (как в **Access**) все таблицы объединяются в один файл, тогда его структуру проследить нельзя: он имеет специальную организацию.

Некоторые термины реляционных баз данных:

Таблица – совокупная информация об объектах одного типа.

Кортеж (строка, запись) – информация об одном отдельном объекте.

Атрибут (поле, столбец) – часть информации об объекте.

Ключ – поле или группа полей, используемые для идентификации или определения местоположения записи.

Связь – способ, которым информация одной таблицы связывается с данными другой таблицы.

Объединение – объединение информации из нескольких таблиц или запросов на основе совпадающих значений определенных атрибутов.

Схема – описание общей логической структуры БзД.

5 Объектно-ориентированная

Отдельные компоненты базы данных представлены в виде объектов. Между записями базы данных и функциями их обработки создаются связи с помощью механизмов, подобных соответствующим средствам в объектно-ориентированных языках программирования. Эти модели объединяют особенности сетевой и реляционной структуры и используются для создания больших баз данных со сложной структурой данных.

1.3 Система управления базами данных

Почти все современные СУБД для ПЭВМ основаны на реляционной модели данных. В MS DOS наиболее распространенными были СУБД семейства **dBase – Ребус, FoxPro, Clipper** и т.п. Этот тип переработан и для Windows. Однако в Windows одним из наиболее распространенных является **Access (Доступ)**.

Основные функции СУБД:

- 1 Определение (описание) данных.
- 2 Создание БзД.
- 3 Внесение изменений в БзД.
- 4 Обработка данных: поиск, выборка данных по некоторому признаку, фильтр, сортировка, объединение с другой связанной информацией.
- 5 Вывод отчетов.
- 6 Управление доступом к данным.

1.4 Архитектура СУБД

Приложения, которые используют базы данных обычно принято относить к одной из программных архитектур.

1 Локальная архитектура

Как программа, так и база данных размещаются на одном компьютере.

2 Файл-серверная архитектура

База данных размещается на мощном отделенном компьютере (сервере), а персональные компьютеры подключаются к нему локальной сетью. На этих компьютерах установлены программы клиентов, которые обращаются к базе данных через сеть. Возможна одновременная работа с одной базой данных несколькими пользователями. Однако вся обработка выполняется на клиентских местах (компьютерах), где фактически формируется локальная копия данных. СУБД или клиентские программы могут находиться

на сервере, но при его запуске копируются на клиентский компьютер. В связи с тем, что управление базой данных осуществляется с разных компьютеров, в определенной мере усложнена организация контроля доступа, поддержка целостности данных.

3 Клиент-серверная архитектура

Сервер не только содержит базу данных, но и на нем выполняется программа СУБД, которая обрабатывает все запросы пользователя и возвращает им лишь результаты обработки этих запросов, причем только те данные, которые действительно были затребованы. Программа пользователя, которая называется клиентом, уже не работает напрямую с базой данных. Нагрузка с клиентских мест снимается, так как большая часть работы выполняется на сервере. Появляется возможность эффективно следить за целостностью данных, контролировать доступ к данным.

4 Распределенная архитектура

В сети работает несколько серверов, и таблицы базы данных распределены между ними. На каждом сервере функционирует своя копия СУБД. Кроме того, в такой архитектуре обычно используются специальные программы – *серверы приложения*. Они разрешают оптимизировать обработку запросов большого количества пользователей и равномерно распределить нагрузку между компьютерами сети. Если, например, нужно выполнить интенсивные вычисления (анализ сложной информации), программы для такой обработки могут выполняться на более мощных сетевых компьютерах. Эта архитектура – многоуровневая.

5 Интернет-архитектура

Доступ к базе данных и СУБД (размещенных на одном компьютере или в сети) осуществляется с одного из браузеров со стандартным протоколом.

2 БАЗА ДАННЫХ ACCESS

2.1 Объекты ACCESS

Microsoft Access – функционально полная реляционная СУБД. Основными объектами **Access** являются:

- *Таблица* - объект, определяемый и используемый для хранения данных. Каждая таблица содержит информацию об объектах определенного вида.

- *Запрос* - объект, позволяющий пользователю получить нужные данные из одной или нескольких таблиц или других запросов. Можно создать запрос на выборку, обновление, удаление или создание новых таблиц.

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

- *Отчет* - объект, предназначенный для создания документа, который можно распечатать или включить в документ другого приложения. Формы и отчеты получают данные как непосредственно из таблиц, так и через запросы.

- *Страница* – специальный объект базы данных, реализованный в последних версиях системы. Точнее – это страница доступа к данным. Этот объект выполняется в коде **HTML**, размещается на **WEB**-странице и передается клиенту вместе с ней. Содержит компоненты, которые позволяют установить связь **WEB**-страницы с базой данных, которая остается на сервере. Пользуясь этими компонентами, пользователь **WEB**-страницы может просмотреть и корректировать записи базы данных. Таким образом, страницы доступа к данным обеспечивают связь между клиентом, сервером и базой данных. Позволяют работать с другими базами данных, например с **Microsoft SQL Server**.

- *Макрос* - объект, являющийся структурированным описанием действий, выполняемых в ответ на определенное событие. *Событие* - это любое изменение состояния объекта **Microsoft Access**. Пример макросов: при

выборе данного - поиск в другой таблице; проверка значения поля при его вводе или изменении.

- *Модуль* - объект, содержащий программы, написанные на **Visual Basic for Applications (VBA)**. Модули могут быть независимыми объектами, содержащими функции, которые можно вызывать из любого места приложения: макроса, формы или отчета. Эти функции можно применять в выражениях, используемых в запросах, а также условиях на значение при работе с таблицами и отчетами. Но модули могут быть непосредственно “привязаны” к отдельным формам или отчетам для реакции на те или иные происходящие в них изменения, т.е. *события*.

Между таблицами и запросами можно устанавливать связи. **Access** поддерживает только иерархические связи (типа “один-к-одному” и “один-ко-многим”). Сетевые связи (типа “многие-ко-многим”) напрямую не реализуются. Их можно создать только с помощью дополнительной связующей таблицы.

2.2 Файл базы данных

База данных **Access** представляет собой один файл специальной структуры, содержащий как описание структуры базы данных, схемы базы данных, так и сами данные. В нем хранятся все вышеперечисленные объекты. По соглашению файл базы данных **Access** имеет расширение **.mdb (Microsoft Data Base)**. Присваивается по умолчанию.

Физическая организация базы данных (структура файла базы данных) в **Access** пользователю недоступна – он не может управлять размещением данных на носителе. Концептуальная схема базы данных представляется описанием всех таблиц и связей между ними. Описание запросов и связей между ними соответствует логической организации данных – это описание данных для конкретного приложения.

Система программ, которая управляет базой данных в **Access**, называется ядром базы данных **Microsoft Jet** или просто **Jet**.

2.3 Начальное окно диалога

После запуска **Access** на экране появится начальное окно диалога, содержащее три переключателя (рис.4).

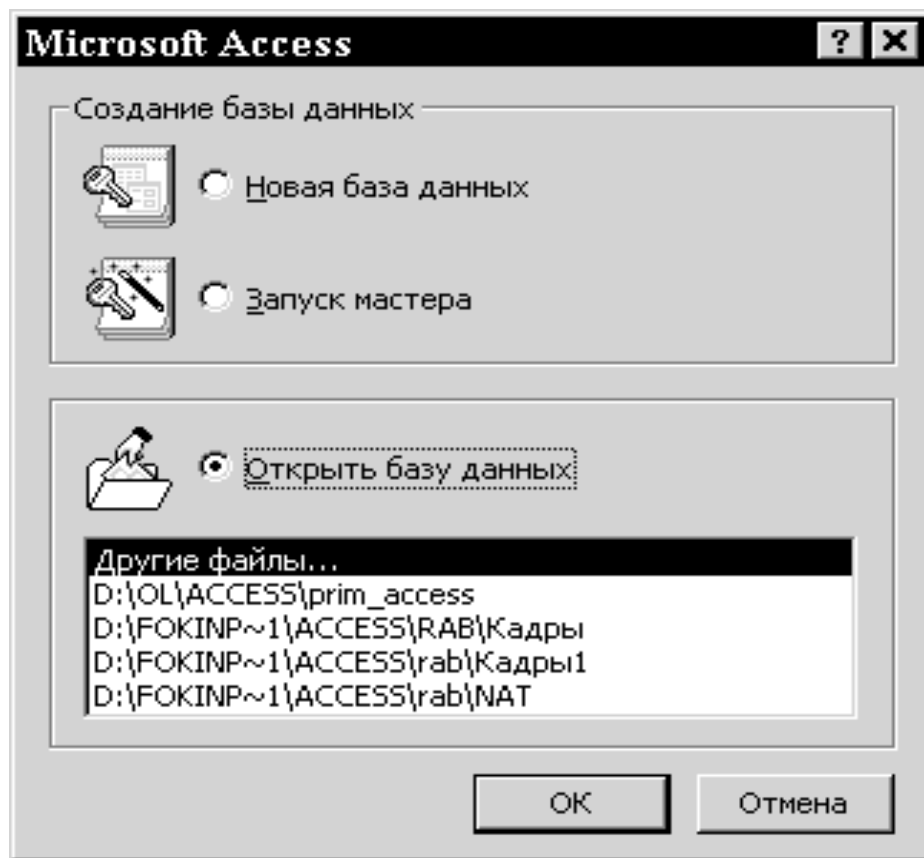


Рисунок 4 – Начальное окно диалога

1 Переключатель “*Новая база данных*”.

Выбор этого переключателя приводит к появлению диалогового окна **файл новой базы данных**. Структура этого окна стандартная в Windows и похожа на аналогичные окна других приложений. Создается пустая база данных. Дальнейшее ее проектирование осуществляется поэтапно, “вручную” в режиме конструктора.

2 Переключатель “*Запуск мастера*”.

Будет предложено более 20 типов структуры базы данных (шаблонов), используя которые можно легко создать конкретную структуру базы данных, включая таблицы, связи между ними, запросы, формы, отчеты и даже макросы и модули.

Проектированием базы данных мастер управляет посредством сменяющих друг друга диалоговых окон **Создание баз данных**.

3 Переключатель “Открыть базу данных”.

Появляется стандартное в Windows диалоговое окно открытия для выбора и открытия существующей базы данных.

2.4 Окно базы данных

После открытия существующей или создания новой базы данных на экране появится окно базы данных, содержащее 6 вкладок, соответствующих вышеописанным (см.п.2.1) типам объектов (рис.5).



Рисунок 5 – Окно базы данных

При первом открытии окна базы данных Access всегда активизирует вкладку **Таблицы**.

При выборе любой вкладки на экран выводится перечень доступных в базе данных объектов, соответствующих выбранной вкладке. В правой части окна базы данных расположены три командные кнопки.

2.4.1 Создание объекта

Командная кнопка **Создать** в окне базы данных позволяет с помощью одного из предлагаемых методов *создать новый объект*, соответствующий вкладке. Обычно при этом открывается диалоговое окно **Новый <объект>**. Здесь и далее <объект> означает наименование конкретного объекта (вкладки), например **Новая таблица**, **Новая форма** и т.д. Для объектов Макросы и Модули окно имеет заголовок <имя>:<объект>, где <имя> – имя объекта.

Открыть диалоговое окно **Новый <объект>**, а следовательно, и создать новый объект можно также и из любой вкладки окна базы данных с помощью кнопки **Новый объект** на панели инструментов или через операционное меню **Вставка**. На рис.6 представлен пример диалогового окна **Новая Форма**.

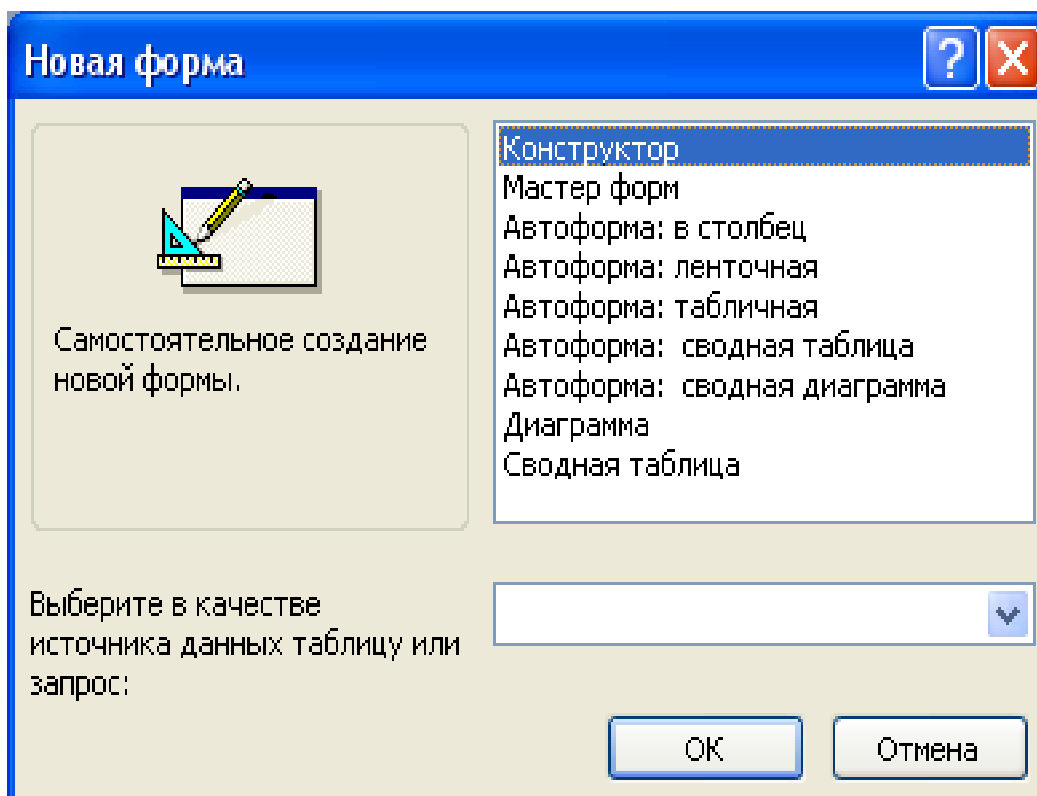


Рисунок 6 – Диалоговое окно **Новая Форма**

В нижней части окна диалога находится раскрывающийся список доступных таблиц или запросов. В нем можно выбрать базовую для построе-

ния объекта таблицу или запрос. Не все вкладки имеют этот список. Например, в окне **Новая таблица** такого списка нет, так как при построении таблицы никакие другие таблицы или запросы не требуются.

В правой части окна находится список способов формирования объекта. Перечень этих способов и действия по ним зависит от типа объекта. Отметим следующие способы:

- **Мастер <объект>** - автоматизированный способ. Сменяя друг друга, появляются диалоговые окна **Создание <объекта>**, в которых предлагается выбрать или подтвердить решение или ввести необходимые данные.

- **Специализированные мастера** - позволяют более быстро, чем общий мастер, создавать объект определенного вида (т.е. обладающий определенными свойствами, имеющий определенную структуру и т.п.).

- **Конструктор** - наиболее универсальный, но и более трудоемкий способ. Позволяет создавать объект в режиме конструктора, т.е. “вручную”.

Примечание. При создании объектов следует тщательно продумывать наименования (имена) объектов и их элементов, т.к. изменение имен в спроектированной базе может привести к конфликтам: в процессе разработки приложений ACCESS автоматически генерирует значительное количество ссылок на объекты и их элементы, которые при изменении их имен автоматически не меняются.

2.4.2 Работа с объектом

Две другие кнопки окна базы данных устанавливают один из двух *режимов работы с существующим объектом* (объект предварительно необходимо выделить):

1 **Режим конструктора** - кнопка **Конструктор**. В этом режиме можно просматривать и изменять структуру объекта (т.е. его описание).

2 **Рабочий режим. (Режим выполнения объекта)**. Объект открывается для работы с данными объекта. Характер этой работы, название кнопки и режима зависит от типа объекта (т.е. выбранной вкладки):

- для объектов **Таблица**, **Запрос** и **Форма** кнопка называется **Открыть** и позволяет просматривать и редактировать данные с помощью выбранного объекта в режиме таблицы или в режиме формы;

- для объекта **Отчет** кнопка называется **Просмотр**. Она активизирует режим предварительного просмотра, позволяющий увидеть отчет в том виде, как он будет выглядеть на бумаге;

- для объектов **Макросы** и **Модули** кнопка называется **Запуск** и позволяет запускать на выполнение макрос или модуль соответственно.

Из одного режима в другой можно быстро переходить с помощью кнопки **Вид** на панели инструментов в верхней части окна **Microsoft Excel**.

3 ТАБЛИЦЫ

3.1 Назначение таблиц

Таблица предназначена для хранения данных и содержит логически сгруппированные данные. Это основной объект базы данных. Все данные, которые находятся в базе данных, хранятся только в таблицах. В таблице сохраняется также ее структура. Каждая таблица имеет четкую структуру. Под этим подразумевается, что она содержит информацию про объекты заданного типа. Вся информация об одном конкретном объекте сосредоточена в записи (строке). Запись состоит из полей (столбцов), в которых хранятся значения атрибутов объекта. Обязательным правилом создания таблицы есть то, что в ее полях может находиться только фактическая, неизменная информация. В них принципиально не может быть расчетных значений, как, например, в MS EXCEL. Все расчеты возможны только через запросы.

Для каждой таблицы можно определить первичный ключ (одно или несколько полей, содержащих уникальные для каждой записи значения) и один или несколько индексов, помогающих ускорить доступ к данным. В большинстве приложений содержится несколько связанных таблиц, представляющих информацию в удобном виде. Многотабличность позволяет

упростить ввод данных и уменьшить количество дублируемых и неправильно введенных данных. Связи позволяют согласовывать или объединять данные нескольких таблиц, чтобы создать набор данных для форм и отчетов.

3.2 Проектирование концептуальной схемы базы данных

При разработке и внедрении баз данных одной из самых сложных проблем является проблема построения общей логической (концептуальной) модели базы данных. Возникает вопрос: как организовать информацию, чтобы использовать все преимущества реляционных моделей и избежать лишних затрат.

В Access эта проблема по существу сводится к разбиению всех данных на таблицы и установлению связей между ними.

Например, создается база данных для ведения портфеля заключенных предприятием договоров, т.е. перечень всех заключенных договоров на поставку изделий предприятия с описанием каждой позиции договора (табл.1).

Таблица 1 – Перечень реквизитов портфеля договоров

Номер договора Дата заключения договора Признак оплаты договора	Сведения о договоре
Наименование заказчика Город Улица	Сведение о заказчике
Дом Телефон	Сведение о заказчике
Наименование изделия Вес изделия Длина изделия	Сведения об изделии
Количество изделий Цена одного изделия	Сведения о составе договора

Можно было бы создать одну таблицу, включив в нее все приведенные в таблице 1 поля. Каждая строка такой таблицы описывает одну позицию договора.

Сразу обнаруживаются недостатки. Так как в одном договоре может быть несколько позиций (заказано сразу несколько изделий), то общие сведения о договоре будут дублироваться в каждой строке (т.е. в каждой позиции договора). Более того, один и тот же заказчик может заключить несколько договоров, и в каждом из них дублируется информация об этом заказчике. Аналогично одно и то же изделие могут заказать несколько заказчиков (т.е. встретиться в нескольких различных договорах). Теперь будет дублироваться информация об этом изделии. Любое дублирование – это расход памяти. Кроме того, это большие неудобства в обновлении информации. Предположим, у заказчика изменился адрес или телефон. Необходимо в базе данных найти все записи для этого заказчика и внести в них соответствующие изменения. К этому следует добавить, что при такой организации возрастает вероятность ошибок при вводе.

Для устранения вышеназванных недостатков используется процесс, называемый нормализацией. Он позволяет организовать поля данных в группы таблиц. Лежащая в основе нормализации математическая теория довольно сложна, но для практического применения ее можно сформулировать в виде довольно простых правил:

- 1 Каждое поле любой таблицы должно быть уникальным.
- 2 Каждая таблица должна иметь уникальный ключ.
- 3 Для каждого значения ключа значения в столбцах должны относиться к объекту таблицы и полностью его описывать.
- 4 Должна быть возможность изменения любого не ключевого поля без воздействия на данные других полей.

После нормализации нашей базы данных получим четыре таблицы, представленные в таблице 2 (ключи выделены шрифтом).

Таблица 2 – Структура нормализованных таблиц

Заказчик	Договор	Состав Догов	Изделие
КодЗак НаимЗак Город Улица Дом Телефон	НомДогов КодЗак Дата ПризОплаты	НомДогов КодИзд Количество Цена	КодИзд НаимИзд Вес Длина

В таблице **СоставДогов** для каждой позиции каждого договора по-прежнему создается одна запись. Однако вместо информации о договоре оставлен только его номер (**НомДогов**), а сама информация вынесена в отдельную таблицу **Договор**. Аналогично информация об изделии вынесена в таблицу **Изделие**, а для связи с ней появилось поле **КодИзд** – код изделия. Это может быть какой-либо идентификационный код или просто поле типа счетчик. В таблице **Договор** также появилось поле **КодЗак** - код заказчика, все сведения о котором хранятся в таблице **Заказчик**.

Следует помнить, что нормализацию, а также всю другую работу по анализу и проектированию, начиная с создания концептуальной модели и заканчивая моделированием логических структур данных, не обязательно выполнять вручную. Для этих целей используют специальные способы – так называемые *Case-технологии*, которые могут значительно облегчить работу по созданию информационных систем.

3.3 Создание таблицы

Создание таблицы осуществляется через окно диалога **Новая таблица** (см.п.2.4.1). Возможны следующие способы формирования таблицы.

1 Режим таблицы. Выводится бланк (форма) абстрактной таблицы, напоминающий таблицу **EXCEL**. Вместо имен столбцов А,В,С... стоят имена Поле1, Поле2,... Необходимо заполнить несколько строк этой таблицы. Имена столбцов можно изменять, дважды щелкнув на имени. Для со-

хранения **Файл** ⇒ **Сохранить макет** или кнопка **Сохранить** на панели инструментов. При этом **Access** предложит создать ключевое поле типа **Счетчик**. Можно отказаться и назначить в качестве ключа одно или несколько полей записи.

2 Мастер таблиц. Используются стандартные заготовки таблиц, из которых пользователь выбирает поля для своей таблицы. Конечно, предусмотренные заготовки облегчают работу пользователя, но они не решают всех проблем.

3 Импорт таблиц. Позволяет импортировать данные из других файлов, в том числе и из других баз данных (форматов, отличных от **Access**).

4 Связь с таблицами. Позволяет путем установления связей с файлом формата интегрировать две информационные системы.

5 Конструктор. Наиболее удобный и универсальный способ. Рассмотрим его подробнее.

На экране появится окно таблицы в режиме конструктора (рис.7).

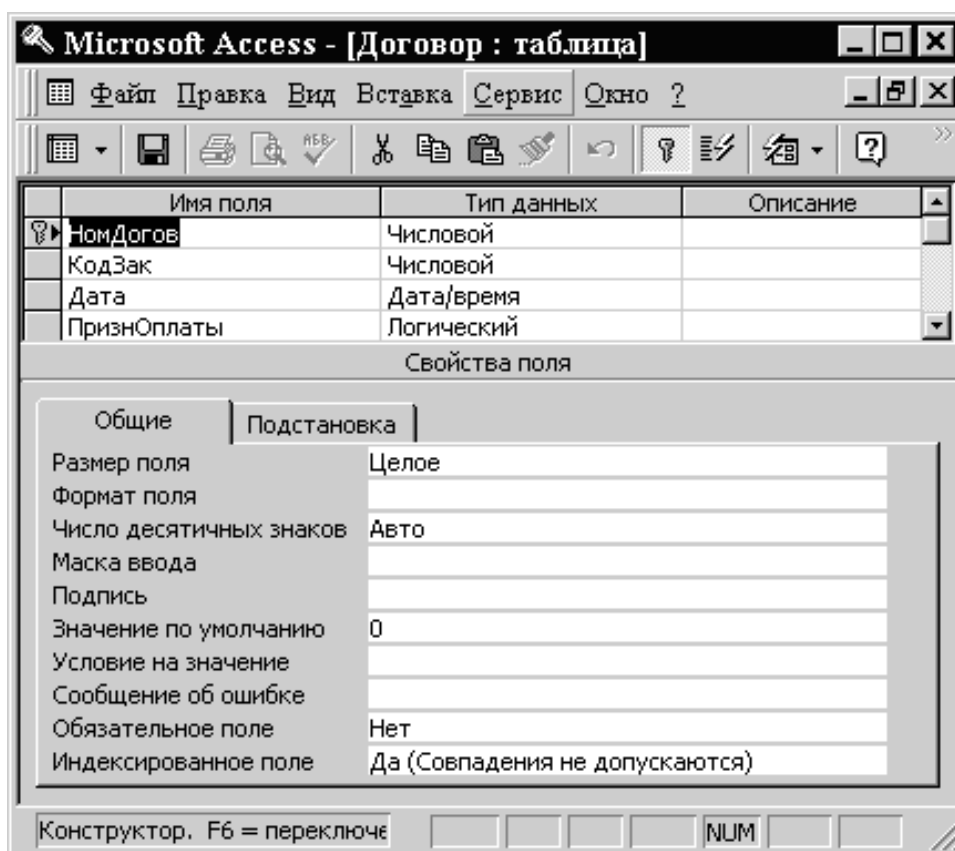


Рисунок 7 – Окно таблицы в режиме конструктора

Это окно разделено на две части. В верхней части располагается бланк таблицы, который необходимо заполнить. Каждая строка описывает одно поле.

Первый столбец (Имя поля) задает имя поля. На имя практически никакие ограничения не накладываются (в том числе можно использовать буквы кириллицы, пробелы). Однако лучше использовать краткие имена без специальных символов. Это обусловлено тем, что имя поля будет использоваться в дальнейшем в расчетах и других ссылках на это поле. Следует также помнить, что столбцы в таблице будут размещаться в таком порядке, в каком они описаны в бланке таблицы. Таким образом, нужно с самого начала продумать их последовательность с учетом удобства дальнейшей эксплуатации всей таблицы. Кроме того, нужно учитывать, что сортировка выполняется слева направо. То есть, сначала выполняется условие сортировки, заданное в первом столбце, затем – во втором и т.д.

Во втором столбце (Тип данных) задается тип данных. Тип данных можно или ввести с клавиатуры, или выбрать его из раскрывающегося списка заголовка, щелкнув на появляющейся стрелке в правой части ячейки, когда эта ячейка активизируется. Следует аккуратно отнестись к заполнению этого столбца. Так, если поле объявить как текстовое (т.е. оставить тип **Текстовый**, который присваивается автоматически), то над этим полем нельзя будет выполнять никакие вычисления, несмотря на то, что оно фактически может быть числовым. Кроме того, при вводе данных в это поле будет отсутствовать контроль правильности ввода, и **Access** не зафиксирует ошибку при попытке ввести в него не цифровой символ. Нельзя будет пользоваться функцией начисления процентов на остаток счета, если сумма счета описана любым типом, кроме денежного (**Денежный**). Для сохранения данных, которые означают дату удобно использовать тип **Дата/время**. В этом случае **Access** обеспечивает контроль правильности ввода. Кроме того, открывается возможность обработки дат. Например, если к дате добавить какое-то число **n**, то в результате будет получена также дата, которая насту-

пит через **n** дней. Аналогично, при вычитании из одной даты другой будет получено количество дней между этими датами. При вычислениях **Access** автоматически учитывает количество дней в каждом месяце. По любой дате можно определить, на какой день недели она приходится.

Допустимы следующие типы данных.

Текстовый – текст длиной не более 255 символов, или числа, которые не требуют расчетов, например номер телефона. Принимается по умолчанию.

Поле Мемо – текст длиной до 65535 символов. Физически сохраняется отдельно от поля. Поле содержит только ссылку на значение, которое находится в специальной структуре базы данных. Выбирается, когда размер текста превышает длину 255 символов или содержит символы форматирования. Следует помнить, что сортировка по этому полю невозможна.

Числовой – числовые данные, которые будут использоваться для расчетов. Конкретный тип данных уточняется с помощью Формат поля.

Дата/время – календарная дата или текущее время. Конкретный формат также устанавливается с помощью Формат поля.

Денежный – денежные данные, для которых, в отличие от обычных числовых данных, имеются некоторые особенности (например, правило округления).

Счетчик – уникальное целое число, которое **Access** генерирует автоматически при создании каждой новой записи. Значения такого поля начинаются с единицы и последовательно увеличиваются для каждой новой записи. Обычно используется для создания первичного ключа.

Логический – логические данные или поля, которые могут иметь только одно из двух возможных значений: “да - нет”, “истина - ложь”, “true - false” и т.п.

Поле объекта OLE – позволяет сохранять такие данные, как картинки, диаграммы, звуковые фрагменты и другие объекты OLE из приложений Windows. Как и поле Мемо, сохраняется в специальной структуре базы дан-

ных отдельно от записи, а в таблице находится только указатель на этот объект.

Гиперссылка – ряд букв и цифр, которые представляют собой адрес гиперпослания.

Мастер подстановок – позволяет осуществить замену (подстановку) значения выходного поля таблицы (кода) соответствующим ему значением из некоторой другой таблицы (подробнее см. п. 3.4). Выбор этого типа приводит к запуску Мастера подстановок, который и определяет тип поля.

В третьем столбце (**Описание**) бланка таблицы можно ввести любой поясняющий текст для каждого из полей. Во всех случаях при выборе поля в запросе или в форме **Access** выводит это описание в строке состояния (в нижней части окна **Access**). Это своего рода мини справка для пользователей.

После выбора типа данных **Access** в нижней части окна выводит список свойств, соответствующих выбранному типу. Нужно установить требуемые свойства (некоторые из них имеют значения по умолчанию). Можно оставить те значения свойств, которые присвоил **Access**. Однако присваивание конкретных значений отдельным свойствам позволяет значительно повысить уровень достоверности данных, которые будут вводиться, облегчить заполнение данных при вводе, сократить затраты памяти и др. Например, для текстовых полей (полей типа **Текстовое**) **Access** автоматически отводит 50 символов на каждое поле. На самом деле это очень много. Так, если поле содержит название города, то для сохранения этих названий достаточно памяти в два, а то и больше раз меньше (вряд ли найдется город, в названии которого больше чем 20 символов). При этом изменение значения свойства **Размер поля** на более точное (меньшее) позволит значительно сократить использование памяти. Особенно это будет заметно при больших объемах базы данных. При этом не лишним будет отметить, что в процессе эксплуатации системы размер поля можно изменять без каких-либо трудностей. Если окажется, что для ввода названия какого-либо города недостаточно

отведенного места и сократить название этого города невозможно, то в любой момент можно увеличить размер поля. Использование масок ввода позволяет сократить ввод данных, сделать ввод более дружелюбным. Например, при вводе номера телефона вводить одни цифры неудобно, а если вводить и разделители (круглые скобки и тире), то это приводит к увеличению объема ввода. Более удобным будет ввод данных, если **Access** автоматически проставит все разделители в процессе ввода.

Набор свойств зависит от типа поля: каждому типу соответствует свой набор. Основные свойства:

- **размер поля** – задает максимальную длину поля;
- **формат поля** – задает формат данных при их изображении на экране; например, для поля *Дата/время* можно указать только часы, или часы и минуты; **Access** подбирает допустимые для каждого типа варианты формирования данных и предлагает их при заполнении этого свойства;

- **маска ввода** – задает форму, в которой данные будут вводиться; например, можно определить, чтобы при вводе номера телефона **Access** автоматически брал в скобки междугородный код, а между группами цифр ставил тире (т.е. чтобы все цифры, которые вводятся с клавиатуры, сразу автоматически расставлялись бы в соответствие с заданной схемой);

- **подпись** – определяет заголовок столбца таблицы для поля, если это свойство не указано то берется имя поля; кроме того, эту особенность **Access** использует как “всплывающую подсказку” для элементов управления в формах: текст который выводится в белом прямоугольнике через небольшой промежуток времени после наведения курсора мыши на соответствующий объект;

- **значение по умолчанию** – значение, которое будет присвоено полю автоматически, если при вводе это поле не будет заполнено;

- **условие на значение** – задаются ограничения на значение поля для контроля правильности ввода; ограничение задается в виде отношения: знак отношения (например, “<”, “≤” и т.п., знак “=” можно опустить), по которо-

му идет граничное значение; несколько ограничений (отношений) можно объединить в одно с помощью логических операторов **or** (или), **and** (и) и **not** (нет); создать ограничения на значение поля можно с помощью мастера построения выражений, который вызывается с помощью кнопки, размещенной справа от поля для ввода ограничения;

- **сообщение об ошибке** – текст, который выводится на экран при невыполнении условия контроля, заданного в предыдущем свойстве;

- **обязательное поле** – определяет обязательность заполнения поля; если этому свойству присвоить значения “да”, то при обработке данных **Access** не завершит работу по формированию или корректировке строки, пока в это поле не будет введено какое-либо значение;

- **пустые строки** – позволяет вводить пустые текстовые данные;

- **индексированное поле** – создается индекс для быстрого доступа.

Переход в область свойств и обратно – **F6** или щелчок в нужном месте.

Каждая таблица в реляционной базе может иметь первичный ключ, который идентифицирует записи и позволяет отличать одну запись от другой. Ключ необходим для повышения эффективности доступа к данным таблицы (это связано с технологией работы СУБД). Кроме того, ключ используется для создания перекрестных связей между таблицами. Ключ может быть простым – одно поле таблицы, а может быть составным – состоять из некоторых полей таблицы. Его следует определить после завершения формирования таблицы. Для этого необходимо щелкнуть в области выделения (слева от имени поля). Если ключ состоит из нескольких полей – щелкать при нажатой **Ctrl**. Затем щелкнуть на кнопке **Ключевое поле** на панели инструментов. Можно также воспользоваться командой:

Правка⇒Ключевое поле.

Следует отметить, что в **Access** ключи создавать необязательно, если по ним не планируется доступ. Так, в нашем примере, при нормализации (п.3.2) мы определили ключи для всех таблиц (это было необходимо для

выполнения процесса нормализации). Однако при создании таблицы **СоставДогов** ключи не были объявлены.

Если при завершении создания таблицы **Access** выявил отсутствие ключей, об этом будет предупреждение с предложением создания этих ключей. Если на это предложение ответить подтверждение, **Access** создаст дополнительное поле типа **Счетчик**, которое и будет объявлено ключевым. Поэтому на такое предложение следует дать отрицательный ответ.

Созданную таблицу необходимо сохранить:

Файл ⇒ Сохранить;

Файл ⇒ Сохранить как...

Если это не будет выполнено, то при попытке закрыть таблицу **Access** напоминает, что таблица не сохранена, и предложить это сделать.

В каждой таблице можно определить несколько индексов.

Индекс - это внутренняя таблица, состоящая из двух полей: значение индекса (т.е. значения полей, включенных в индекс) и местоположение каждой записи таблицы с данным значением индекса. Позволяет быстро находить данные в таблицах (при больших объемах данных это особенно заметно), поскольку поиск ведется не в самой таблице, а в индексе, значительно меньшем по объему, чем сама таблица. Однако каждый индекс требует дополнительных затрат памяти. Кроме того, индексы снижают скорость при обновлении данных в индексированных полях, добавлении или удалении записей.

Если индекс состоит из одного поля, то его можно создать, указав свойство **Индексированное поле**. Для составных индексов используется кнопка **Индексы** на панели инструментов, или команда:

Вид⇒Индексы.

Открывается диалоговое окно **Индексы**, в котором необходимо последовательно указать все поля, которые составляют индекс. При этом задается порядок расположения полей в индексе и порядок их сортировки: по возрастанию или убыванию значения поля.

Для таблицы можно определить несколько индексов. Каждый индекс имеет свое уникальное имя.

3.4 Использование свойств подстановки

Вынесение повторяющейся информации в отдельные таблицы и замена ее в выходной таблице компактными кодами позволяет существенно сократить затраты памяти на магнитном диске, обеспечить целостность данных и удобство их обработки. Однако при пересмотре и актуализации данных неудобно пользоваться неинформативными кодами. Преимущественно имеют дело с первичными данными (например, Имя заказчика вместо его кода в таблице **Договор**). Для этого можно пользоваться свойством подстановки. В этом поможет **Мастер подстановок**, который создает поле, что позволяет выбрать значения из другой таблицы или из списка фиксированных значений, пользуясь полем со списком. Для запуска **Мастера подстановок** необходимо при создании поля в столбце **Тип данных** описание этого поля выбрать из списка. При выборе этого параметра в списке **Типов данных** и будет активизирован мастер для автоматического определения этого поля.

Появится последовательность диалоговых окон, в которых необходимо (каждый пункт отвечает одному окну диалога):

- Выбрать опцию **Объект “столбец подстановки” будет использовать значения из таблицы или запроса** для указания, откуда будут браться данные для подстановки, в данном случае – из таблицы.

- Выбрать конкретную таблицу или запрос для подстановки (в нашем случае – таблицу **Заказчик**).

- Выбрать одно или несколько полей, которые будут составлять столбец подстановки (в нашем случае **КодЗак, НаимЗак, Город**).

- Настроить ширину столбцов, которые составляют столбец подстановки. Если какое-то поле выводить не требуется (например, **КодЗак**),

установить нулевую ширину. Настройка осуществляется путем перемещения вправо – влево границы столбцов в заголовке.

- Сменить при необходимости подпись поля в основной таблице.

Теперь в таблице будет храниться значение первого столбца подстановки (в нашем случае это поле **КодЗак**), которое на экране будет показано в виде списка, который раскрывается. Когда список раскрыт, в нем выводятся все поля столбца подстановки, которые имеют не нулевую ширину (**НаимЗак, Город**), а в закрытом списке отображаются значения первого поля с ненулевой шириной (**НаимЗак**).

С помощью **Мастера подстановок** можно создать также поле, которое позволяет выбрать значения из фиксированного списка значений, используя поле со списком. В нашем случае это целесообразно задать для поля **ПризОплаты**. Для создания такого поля необходимо в первом окне диалога **Мастера подстановок** выбрать опцию **Будет введен набор фиксированных значений**. В следующем окне диалога необходимо ввести набор фиксированных значений, из которых можно будет выбрать значение при формировании поля. Так, для нашего примера это может быть набор значений **Оплачено, Не оплачено**. Применение такого поля позволит, во-первых, облегчить введение данных (вместо ввода вручную значение будет выбираться из списка), а, во-вторых, исключит ошибку ввода (так как могут быть выбраны заранее выбранные значения).

3.5 Определение связей

После определения таблиц между некоторыми из них следует установить **связи**.

В каждой **связи** принимают участие две таблицы: одна *главная* (отцовская), а другая – *связанная* (дочерняя).

Связь между таблицами устанавливает отношение между совпадающими значениями в ключевых полях, т.е. между полями, которые имеют одинаковые имена в обеих таблицах. В большинстве случаев с ключевым

полем одной таблицы, которое является идентификатором каждой записи, связывается внешний ключ другой таблицы.

Наиболее часто используется связь между таблицами типа отношения **“один-ко-многим”**. В этом случае каждой записи в главной таблице может соответствовать несколько записей в связанной таблице. Однако ни одна запись в связанной таблице не может иметь больше одной соответствующей ей записи в главной таблице.

Отношения типа **“один-к-одному”** используются не так часто, поскольку большая часть данных, связанная таким образом, может быть размещена в одной таблице. Такие отношения могут быть использованы для разделения широких таблиц на две, для выделения части таблицы для защиты.

Отношения типа **“многие-ко-многим”** средствами **Access** не используются. Для отображения такой связи необходимо создать вспомогательную таблицу. В этом случае отношения **“многие-ко-многим”** фактически сводятся к двум отношениям типа **“один-ко-многим”** с третьей таблицей.

Связи можно установить не только между таблицами, но и между запросами. Более того, связи могут быть смешанными: между таблицами и запросами одновременно. В дальнейшем **Access** сможет связывать эти таблицы при их использовании в запросах, формах или отчетах.

Для связи созданных таблиц необходимо щелкнуть кнопку **Схема данных** на панели инструментов или выполнить команду:

Сервис⇒Схема данных.

Откроется пустое окно **Схема данных**, а затем сразу же окно **Добавление таблицы**. Если окна **Добавление таблицы** на экране нет, то его можно активизировать через контекстное меню или с помощью команды

Связи⇒Добавить таблицу.

В этом окне следует выбрать необходимую вкладку. Обычно это вкладка **Таблица**. Однако если необходимо в связи включить и запросы, это будет другая вкладка. Далее необходимо выделить нужные таблицы (в

нашем случае четыре таблицы) или запросы и щелкнуть на кнопке **Добавить**. В окне **Схема данных** появятся все выделенные таблицы или запросы в виде прямоугольников с перечисленными в них именами полей каждой таблицы или запроса. Название таблицы размещается в верхней части прямоугольника.

Если наименования длинные или их много, то в прямоугольнике появляются линейки прокрутки, что не очень удобно. Поэтому лучше растянуть границы окон таким образом, чтобы сразу были видны полные названия всех полей. Порядок, в каком прямоугольники размещаются в окне **Схема данных**, также сначала соответствует очередности выбора объектов. Его можно в любой момент изменить простым “перетаскиванием” нужного прямоугольника в различные места (действия аналогичны работе с окнами в **Microsoft Office**).

Для установления связей между таблицами необходимо перетащить из родительской таблицы (стоящей на стороне “один”) поле, по которому устанавливается связь на соответствующее поле дочерней таблицы (стоящей на стороне “многие”). Откроется диалоговое окно **Изменение Связей**, в котором можно определить дополнительные поля, если требуется связь по нескольким полям. Здесь же можно установить опции:

- **целостность данных;**
- **каскадное обновление;**
- **каскадное удаление.**

Первая опция не позволит ввести в дочернюю таблицу не существующий в родительской таблице код или удалить из родительской таблицы строки, если на них есть ссылки из дочерней. Последние две опции позволяют синхронно обновлять ссылки или удалять все записи в связанных дочерних таблицах соответственно. Они доступны только после установки первой опции.

После щелчка на кнопке **Создать** связь будет создана, и в окне **Схема данных** появятся линии между таблицами, указывающие на наличие связи между ними (рис.8).

Кстати, если будет запрошена целостность данных **Access** покажет направление связи (“**один-ко-многим**”): на стороне “один” появится символ “**1**”, а на стороне “многие” - “**∞**”.

Если при создании таблиц для какого-то поля было установлено свойство подстановки, **Access** автоматически свяжет по этому полю обе таблицы – и основную, и таблицу подстановки. Целостность данных при этом не будет задана. Чтобы ее задать, следует скорректировать (изменить) связь.

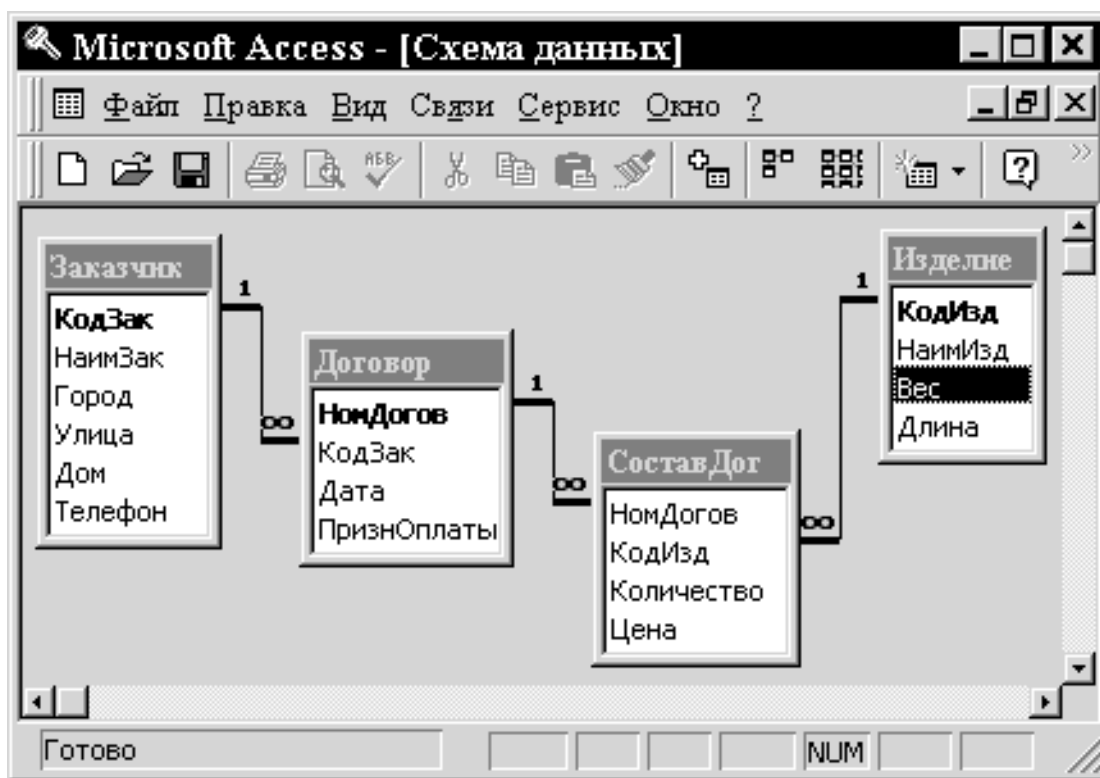


Рисунок 8 – Окно *Схема базы данных*

Для изменения связи щелкнуть на ее линии правой кнопкой мыши и в появившемся контекстном меню выбрать пункт **Изменить связь**. Появится диалоговое окно **Изменение Связей**.

Следует помнить, что **Access** допускает многократное использование одной и той же таблицы или запроса в разных системах связей, которые между тем отображаются в рабочем окне схемы данных, а значит если несколько раз добавлять одну и ту же таблицу (запрос), то и соответствующих им прямоугольников будет такое же число. В имени каждого прямоугольника появится индекс. Лишние прямоугольники в любой момент можно удалить, это никак не отразится на самих источниках данных. Кроме того, при переопределении свойства подстановки создается новая связь без уничтожения предыдущей. Соответственно, появится и новый прямоугольник. Поэтому старую связь и соответствующий прямоугольник необходимо удалить. В первую очередь необходимо удалить саму связь, а не прямоугольник. А уже после удаления связи можно удалять прямоугольник. Для удаления связи щелкнуть на ее линии, она станет жирной. Затем нажать на клавиатуре клавишу **Delete** или выбрать в контекстном меню команду **Удалить связь**.

Чтобы удалить из окна **Схема данных** таблицу, необходимо ее выделить и нажать на клавиатуре клавишу **Delete**. Таблица убирается из окна (становится невидимой), но ее связи сохраняются.

3.6 Внесение изменений в определение таблицы (режим конструктора)

Для изменения структуры таблицы ее следует открыть в режиме конструктора (п.2.4.2). Возможны следующие корректировки.

1 Переименование поля - щелкнуть по имени поля и внести изменения.

2 Вставка нового поля - выделить поле, перед которым необходимо вставить новое поле. Затем нажать клавишу **Insert** на клавиатуре. Или выполнить команду

Вставка ⇒ Поле.

Перед выделенным полем появится пустая строка, которую можно использовать для определения нового поля. Для вставки поля в конец необходимо воспользоваться следующей пустой строкой.

3 Удаление ненужного поля. Выделить ненужное поле и нажать клавишу **Del** на клавиатуре. Подтвердить удаление. Или выполнить последовательность команд

Правка ⇒ Удалить;

Правка ⇒ Удалить строку.

4 Перемещение полей. Выделить строку или группу строк (для этого щелкнуть по области выделения первой строки и при нажатии **Shift** - последней строки). Затем просто перетащить их на новое место.

5 Изменение типа поля. Щелкнуть ячейку “**Тип данных**” требуемого поля и изменить тип.

В некоторых случаях изменения могут коснуться нескольких таблиц. Пусть в нашем примере в базу данных необходимо внести дополнительно следующую информацию о заказчике: почтовый индекс, область, наименование ж/д станции, наименование ближайшего аэропорта, расстояние до аэропорта в километрах.

Анализ данных показывает, что все они, за исключением почтового индекса, относятся не к самому заказчику, а к городу, в котором он находится. Поэтому имеет смысл создать новую таблицу (**Город**), поместив в нее поля: **НазвГород**, **Индекс** (первые три цифры почтового индекса), **Область**, **Ж/дСтанция**, **Аэропорт**, **Расстояние**, а в таблицу **Заказчик** добавить только почтовый индекс (**Индекс**).

Естественно, следует также установить соответствующие связи (связать таблицы **Город** и **Заказчик**).

3.7 Просмотр и актуализация данных таблицы (режим таблицы)

Для просмотра, добавления, изменения или удаления данных необходимо воспользоваться режимом таблицы (см. п. 2.4.2). Это простейший способ просмотра данных по строкам и столбцам (рис.9).

Переход от поля к полю – клавиша **Tab** на клавиатуре, а переход от записи к записи – полоса прокрутки, стрелки. В нижней части экрана левее нижней полосы прокрутки находятся кнопки перемещения по записям (вверх, вниз на одну запись, к первой или последней записи таблицы, установка на новую запись). Здесь же выводится номер текущей записи и общее количество записей в таблице. Активизировав (щелчком или **F5**) поле номера текущей записи можно ввести номер требуемой записи для быстрого перехода к ней.

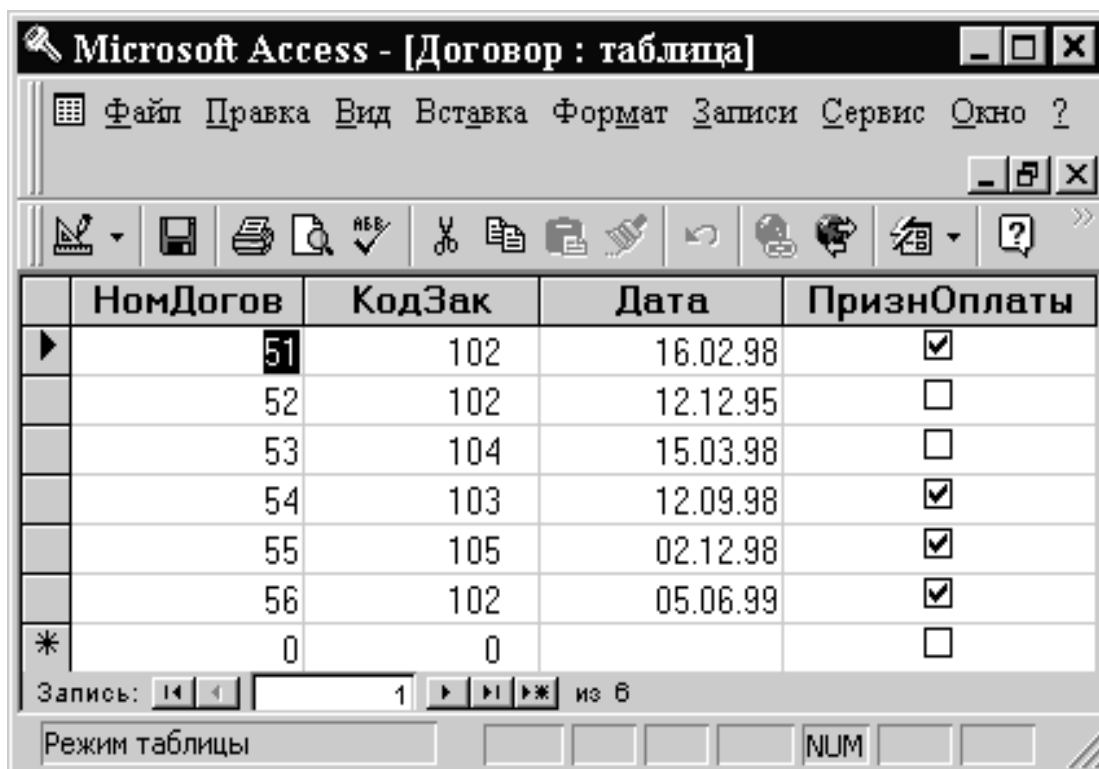


Рисунок 9 – Окно таблицы в режиме таблицы

Перейти к любому видимому полю можно щелчком на этом поле

1 Выделенное поле можно **изменить или удалить** клавишей **Del** на клавиатуре.

2 Для **удаления записи** выделить ее (щелчок слева в области выделения), затем нажать на клавиатуре клавишу **Del**.

3 **Вставка новой записи** осуществляется путем заполнения пустой строки в конце таблицы. Эта строка помечена маркером “*”.

4 **Изменение макета таблицы.** Можно изменять представление таблицы на экране: изменять высоту строк, ширину столбцов, шрифт, скрывать или выводить линии сетки и т.д. Эти изменения подобны **EXCEL**. Выполняются с помощью меню **Формат**, которое появляется, как только таблица будет открыта в рабочем режиме. Размеры можно изменить путем перемещения границ в заголовке или области выделения.

5 **Поиск записей.** Можно выполнить несколькими способами:

- **Правка ⇒ Найти**

- через контекстное меню

- кнопка **Бинокль** на панели инструментов.

Появится диалоговое окно **Поиск в поле**, в котором необходимо:

а) Заполнить поле **Образец**. Можно использовать символы “*” и “?”.

Не обязательно набирать все, можно вести поиск по части образца.

б) Установить необходимые операции:

- *просмотр*: все, вверх, вниз;

- *совпадение*: с началом поля, с любой частью, поле целиком;

- *с учетом регистра*: различать прописные и строчные или нет;

- *с учетом формата*: в том виде как выводиться на экран;

- *только в текущем поле*: поиск в текущем поле или во всех полях.

в) Нажать **Найти** или **Найти далее**. Для поиска от текущей можно использовать также комбинацию **Shift+F4**.

7 **Установка фильтра.**

а) **Фильтр на выделенном.** Отбираются записи, у которых значения определенных полей совпадают со значениями выделенного поля. Вызывается:

- контекстное меню
- кнопка на панели инструментов

Записи ⇒ Фильтр ⇒ Фильтр по выделенному

К отобранным данным можно применить новый фильтр.

б) **Исключить выделенное.** Отбираются записи за исключением тех, у которых значения полей совпадают с выделенными.

в) **Удалить фильтр.** Возобновляет содержимое всей таблицы.

г) **Расширенный фильтр.**

Записи ⇒ Фильтр ⇒ Расширенный фильтр

В появившемся диалоговом окне заполнить реквизиты именами полей (в первой строке) и условия отбора.

Сортировка таблицы. Выделить поле, по которому сортировать и нажать кнопку **Сортировка по возрастанию** или **Сортировка по убыванию**. Сортировка таблицы по нескольким полям выполняется с помощью расширенного фильтра.

Нужно отметить, что после ввода или корректирования данных нет необходимости их сохранять: **Access** сделает это автоматически.

4 ЗАПРОСЫ

4.1 Назначение запросов

Запросы позволяют довольно быстро отобрать необходимые данные из одной или нескольких связанных таблиц.

Запросы черпают информацию из базовых таблиц базы данных или из других запросов. В результате реализации запроса создается **набор записей**, с которым в большинстве случаев можно работать как с таблицей: вы-

бирать данные, печатать, просматривать и даже обновлять или удалять. Запросы являются удобным средством выделения данных, необходимых для решения определенной задачи.

В отличие от реальных таблиц, набор записей запроса в базе данных физически не существует. Он создается только в момент выполнения запроса. Однако если изменить данные в наборе записей, **Access** внесет изменения в таблицы, на базе которых построен запрос.

Следует отметить: отобразить данные можно и с помощью функций **Поиск** или **Фильтр**. Однако это будут разовые выборки. Запрос же сохраняется в базе данных и его можно использовать многократно. Кроме того, в запрос можно включать данные из нескольких таблиц или других запросов.

Набор записей может содержать **вычисляемые поля**, представляющие собой результат вычислений с использованием значений любых полей таблиц или запросов, арифметических операций и встроенных функций **Access**.

В запросе можно использовать **параметры** – данные, значения которых запрашиваются в момент выполнения запроса. Обычно параметры применяются для задания условий отбора записей из таблиц или запросов, иногда – для формирования вычисляемых полей (см. п.4.3, 4.4).

С помощью запросов можно выполнять выбор данных, их сортировку, фильтрацию, выполнять превращение данных в соответствии с определенными алгоритмами, создавать новые таблицы, наполнять их, выполнять вычисления и т.п.

Таким образом, полученный на основе запроса набор записей может не соответствовать ни одной реальной таблице базы данных. Его поля могут представлять собой набор полей из разных таблиц, а записи могут быть отфильтрованы, рассортированы.

4.2 Типы запросов

Запросы можно разбить на два типа.

1 *Запрос на выборку* предназначен для отбора необходимых для последующей обработки данных. Бывает следующих видов:

- **простой запрос** – осуществляет выбор данных из одной или нескольких (**многотабличный запрос**) таблиц или запросов. Этот запрос можно использовать не только для отбора данных, но и для внесения в базу данных единичных изменений (изменений конкретных полей в конкретных записях): можно вставлять отдельные строки, удалять их или обновлять данные в наборе записей. **Access** внесет необходимые изменения в соответствующие базовые таблицы запросов;

- **итоговый запрос** содержит итоговые значения по группам данных. Допускаются групповые операции: **Sum** (сумма), **Avg** (среднее арифметическое), **Max** (максимум), **Count** (количество записей) и др. Этот запрос используется для выборки и предварительной обработки данных;

- **перекрестный запрос** - особый тип итогового запроса, представляет вычисляемые значения в виде перекрестной таблицы. Для каждого поля перекрестного запроса может быть выбрана одна из четырех установок: **заголовки строк, заголовки столбцов, значение** (выводится в ячейках перекрестной таблицы) или **не отображается**.

2 *Запрос на изменение* предназначен для выполнения одних и тех же изменений одновременно над группой записей, удовлетворяющих некоторому критерию (условию):

- **запрос на обновление** осуществляет изменение значений одного или нескольких полей у группы записей. Например, изменение цены изделия сразу в нескольких договорах;

- **запрос на удаление** позволяет удалить из базы данных сразу группу записей, удовлетворяющих некоторому условию. Например, удаление из базы данных, не оплаченных договоров (или закрытых);

- **запрос на добавление** дает возможность скопировать выбранные данные или вычислить итоговые данные и сохранить их в другой таблице. Например, занести в архивную таблицу информацию по закрываемому договору (номер договора, код заказчика, общая сумма).

4.3 Создание запроса на выборку

Запрос на выборку создается через окно диалога **Новый запрос** (см. п. 2.4.1). В этом окне можно выбрать один из **Мастеров**.

Более универсальным является **Конструктор**. При работе с ним сначала появится диалоговое окно, в котором необходимо выбрать таблицы и запросы, являющиеся базовыми для составляемого запроса. Затем откроется **Окно конструктора запросов** (рис.10).

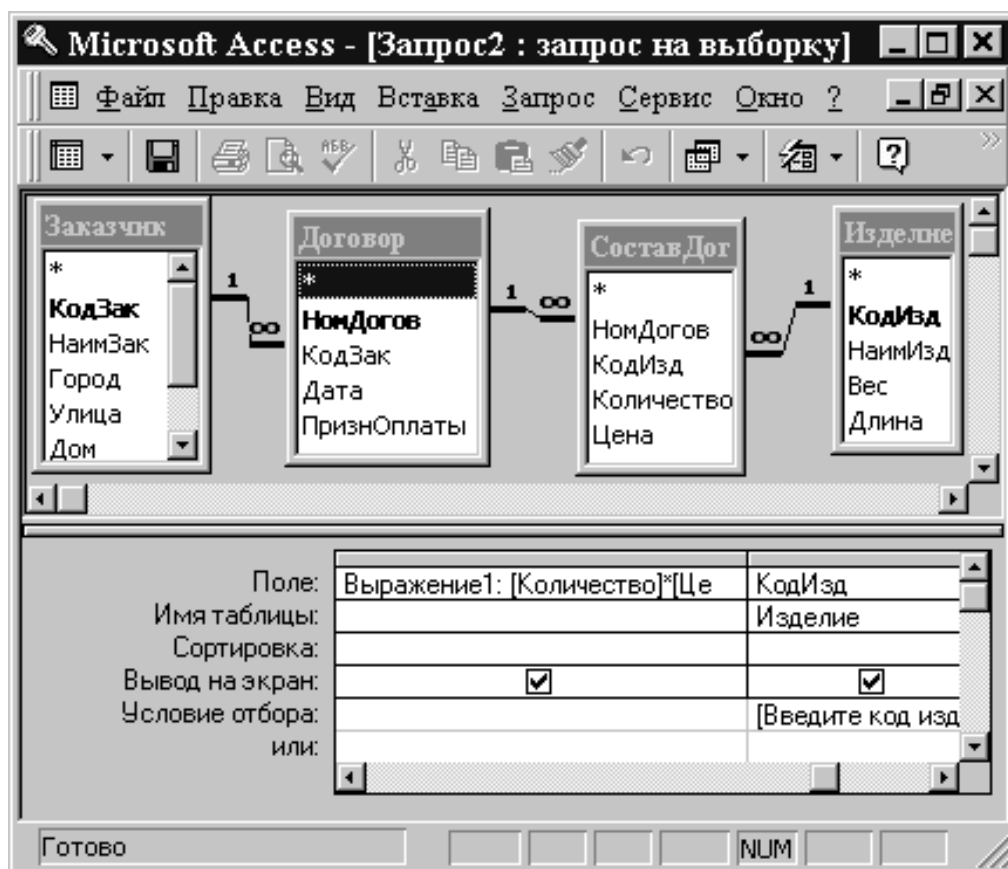


Рисунок 10 – Окно конструктора запросов при создании запроса на выборку

Это окно разделено на две части. В верхней части находятся сгруппированные по таблицам списки полей таблиц или запросов, на основе которых создается новый запрос (базовые таблицы и запросы). В нижней части располагается бланк, который необходимо заполнить. Каждый столбец бланка описывает одно поле, используемое в запросе. Это может быть поле из одной из таблиц или запросов (базового), быть вычисляемым (т.е. его значение рассчитывается на основе других полей таблиц или запросов) или итоговым (т.е. использующим одну из групповых функций **Access**).

Первая строка бланка запроса (**Поле**) служит для выбора полей, которые должны присутствовать в наборе записей, используются для сортировки данных или для выбора информации из таблицы. Чтобы выбрать поле из базовой таблицы или запроса, следует это поле перетащить из соответствующей таблицы (запроса) в верхней части окна в первую строку требуемой колонки. Чтобы создать вычисляемое поле, следует набрать в первой строке выражение для его вычисления.

Во *второй строке* (**Имя таблицы**) выводится имя таблицы или запроса, из которой выбрано поле. Если эта строка отсутствует – выполнить команду

Вид ⇒ Имена таблиц

В *третьей строке* (**Сортировка**) можно задать, нужно ли выполнять сортировку по этому полю, и указать направление сортировки: по возрастанию или убыванию значения поля.

Флажки в *четвертой строке* (**Вывод на экран**) позволяют исключить вывод в набор записей полей, включаемых в запрос только для отбора нужных записей.

Строка **Условие отбора** и строки **Или** используются для отбора записей. Если необходимо выбрать записи с конкретным значением поля, следует ввести это значение в ячейку **Условие отбора**. Можно использовать оператор **Or** для отбора по нескольким значениям. Можно задавать выражения

и использовать знаки сравнения (<, <=, >, >=, <>) для сравнения текущего значения поля и значения выражения (см. п. 4.4).

Например, если необходимо регулярно отслеживать некоторые изделия, можно составить запрос на выборку всех заказов на поставку интересующего изделия, включив в него данные по заказчику из таблицы *Заказчик*, по договору из таблицы *Договор*, а также поля **Количество** и **Цена** из таблицы *СоставДогов*. Для определения общей стоимости создается вычисляемое поле, определяемое как произведение полей **Количество** и **Цена** из таблицы *СоставДогов*. Для построения сложных выражений, включающих поля, знаки операций, функций можно использовать **Построитель выражений**. Для его вызова необходимо щелкнуть на кнопке **Построитель** на панели инструментов. Поле **КодИзд** используется для отбора записей. Условие отбора можно было бы указать непосредственно в бланке запроса (код требуемого изделия). Однако на этапе создания запроса не всегда можно определить, какие именно значения (в нашем случае – какие коды изделий) необходимо искать. В этом случае удобно использовать параметр запроса. Для этого в строку **Условие отбора** следует занести вместо конкретного значения имя или фразу, заключенную в квадратные скобки. Эта фраза воспринимается как имя параметра и выводится в окне диалога при выполнении запроса для ввода значения параметра. В этой связи разумно использовать содержательную фразу. В нашем случае определен параметр [Введите код изделия] для отбора информации по конкретному изделию (поле **КодИзд**). Параметров может быть несколько. По умолчанию **Access** присваивает параметрам запроса текстовый тип данных. Для изменения типа следует выбрать команду

Запрос ⇒ Параметры

Появится диалоговое окно со списком всех определенных параметров.

В **Access** принято, что любое имя, не совпадающее ни с одним именем поля или имени функции воспринимается как имя параметра и будет организован ввод его значения в диалоге при реализации запроса. Поэтому ча-

сто записанные с ошибками имена полей (например, в выражениях для расчета вычисляемого поля) воспринимаются как параметры.

4.4 Выражения

Выражения – это основной способ выполнения большого количества операций в **Microsoft Access**. Выражение представляет собой комбинацию символов, идентификаторов, операторов и значений, которая дает определенный результат. Например, следующее выражение позволяет вывести к элементу управления в форме или отчете сумму значений подготовительно-заключительного времени (**Тпз**) и единичного времени на изготовление партии деталей. Единичное время, в свою очередь, вычисляется как произведение единичного времени на изготовление партии деталей (**Тшт**) на количество деталей в партии (**Количество**)

$$= \mathbf{Тпз} + \mathbf{Тшт} \times \mathbf{Количество}.$$

Выражения используются:

- для формирования свойства **Условие на значение** при формировании структуры (описания) таблицы;
- создания рассчитываемых полей в запросах;
- указания условий выбора или обновления в запросах;
- создание рассчитываемых элементов в формах и запросах;
- указание условий выполнения макрокоманд и определение аргументов макрокоманд;
- определение аргументов для некоторых встроенных функций.

Некоторые выражения дают логический результат **true** или **false**. Это, к примеру, выражения, которые используются в условиях выбора записей. Другие выражения дают числовой или текстовый результат, как, например, выражения для вычисления рассчитываемых полей.

Признаком выражения есть знак равенства в первой позиции. В выражениях допустимо использование таких встроенных (внутренних) функций **Microsoft Access**:

Count([имя поля]) – количество записей.

Sum([имя поля]) – сумма значений.

Min, Max, Avg и др. – итоговые функции (минимальное, максимальное, среднее значения).

Page – номер текущей страницы в отчете.

Pages – общее число страниц в отчете.

Date() – текущая дата без компоненты часов.

New() – текущие время и дата.

Как уже говорилось, в **Microsoft Access** есть специальная встроенная программа для автоматизации формирования выражений – **Мастер выражений**. Запустить его можно с любого места, в котором необходимо ввести выражение, в первую очередь, выделив объект, в который необходимо вставить выражение. Для этого используется контекстное меню, командные кнопки и др.

Следует отметить, что если объект, в который вставляется выражение уже имеет какое-либо значение, то при построении выражения мастер копирует его в поле выражения.

4.5 Создание запроса на изменение

Для создания запроса на изменение (обновление, удаление, добавление) можно построить запрос на выборку данных, подлежащих изменению, и затем преобразовать его в требуемый вид. Для этого следует воспользоваться кнопкой “**Тип запроса**” на панели инструментов и выбрать необходимый тип запроса в раскрывающемся списке. Затем необходимо дополнить бланк запроса, заполнив дополнительно появившиеся строки. Так, в запросе на обновление появится строка **Обновление**, в которой следует за-

дать новое значение обновляемого поля. Кстати, это значение может быть задано выражением (см. п. 4.4).

На рис.11 приведено окно конструктора запросов при создании запроса на обновление. Этот запрос осуществляет увеличение цены изделия на заданный коэффициент. Значение коэффициента запрашивается при реализации запроса (параметр **[Введите коэффициент]**). Новое значение цены определяется как произведение старой цены на коэффициент: **[Цена]*[Введите коэффициент]** Цена меняется только в заданных изделиях. Код изделия также запрашивается через параметр **[Введите код изделия]**.

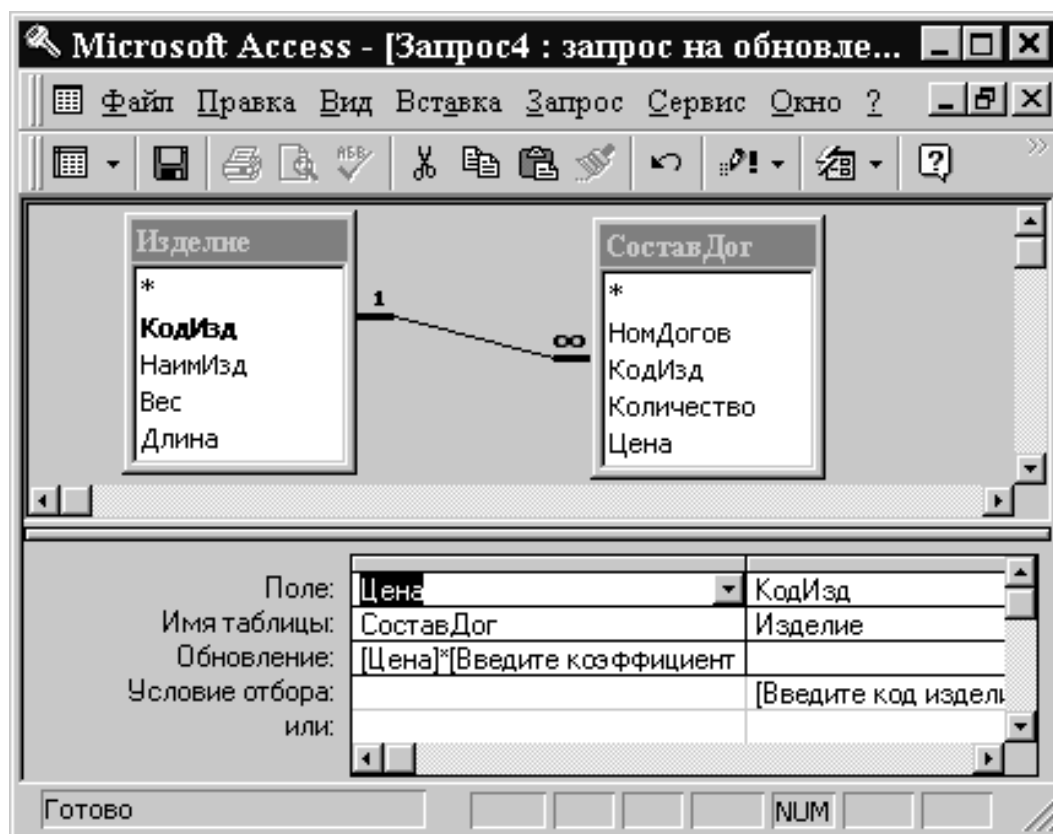


Рисунок 11 – Окно конструктора запросов при создании запроса на выборку

5 ФОРМЫ

5.1 Назначение форм

Работать с таблицами не всегда удобно, а представление данных в них не очень наглядно. Гораздо эффективнее формы. Если запрос – это способ для выбора, обработки и анализа данных, то форма – это способ отображения, даже больше – вывода данных. Форма дает возможность спроектировать работу так, чтобы конкретный пользователь имел возможность доступа и корректировки (ввода) только тех данных, которые находятся в его компетенции.

Таким образом, форма является одним из важнейших компонентов любой базы данных. Она обозначает действительные возможности самой СУБД и обеспечивает связь между самими данными и работающими с ними пользователями.

Основные возможности форм:

1 Можно управлять представлением данных на экране: изменять цветовое решение, добавлять тени, изменять форматы вывода и т.д.

2 В формы можно включать элементы управления: поля со списками, флажки, переключатели и т.д.

3 Форма может содержать объекты **OLE** (например, картинки, диаграммы и т.п.).

4 Используя макросы или процедуры **VBA**, в форме можно выполнять разнообразное редактирование данных.

5 Можно связать несколько форм или отчетов с помощью макросов или процедур **VBA**.

Формы используются:

- для вывода и редактирования данных;
- ввода данных;
- управления ходом выполнения приложения;

- вывода сообщений;
- печати информации.

Для работы с формами используется вкладка **Формы** в окне базы данных. Большинство форм создается на базе таблиц или запросов.

Обработка данных с помощью формы осуществляется в режиме формы (п.2.4.2). Этот режим предоставляет дружественную среду для работы с данными и удобный дизайн их представления на экране.

Форму можно просмотреть также в режиме таблицы (на экран выводится таблица, включающая все поля формы) или в режиме предварительного просмотра (позволяет увидеть форму в таком виде, в каком она будет выведена на печать).

5.2 Элементы формы

В форме можно выделить:

- заголовок;
- область данных;
- раздел примечаний;
- полосы прокрутки;
- область выделения;
- поле номера записи.

В *области данных* размещается информация, выбираемая из базовой таблицы или запроса или получаемая на ее основе.

Информация или элементы управления, которые не меняются от записи к записи, размещаются *в заголовке* или *разделе примечаний*. Заголовок и раздел примечаний необязательны и в новой форме по умолчанию отсутствуют.

Поле *номера записи* предназначено для идентификации номера текущей записи и перехода к требуемой записи и имеет вид, показанный на рис.12.

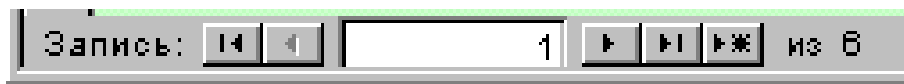


Рисунок 12 – Поле номера записи

Можно непосредственно набрать требуемый номер записи, перебирать записи по одной, щелкая на кнопках прокрутки (влево или вправо), выбрать первую или последнюю запись или добавить новую.

Область выделения – это вертикальная полоса в левой части формы. В ней против текущей записи помещается специальный маркер.

Полосы прокрутки служат для прокрутки данных, не поместившихся в окне.

Поле номера записи, область выделения и полосы прокрутки можно отключать, установив в **False** значения соответствующих свойств формы.

Информация в форме содержится в **элементах управления**. Основные из них:

- **Надпись** - постоянный текст, не связанный с другими элементами управления.

- **Поле** - содержит данные из базовых таблиц или запросов (**связанное поле**) или значение, вычисляемое непосредственно в самой форме (**свободное поле**).

- **Список** - содержит набор значений, который всегда остается открытым. Это может быть набор предварительно заданных (при описании элемента) значений или значений какого-либо поля таблицы или запроса. Выбирается одно из значений.

- **Поле со списком** можно рассматривать как объединение двух элементов: поля и списка. Чтобы раскрыть список, следует щелкнуть на стрелке в правой части поля.

- **Переключатели, флажки, выключатели** - используются для отображения на экране данных, имеющих только два допустимых значения: да/нет, включено/выключено и т.п. Особенно удобны графические изображения: флажки, кнопки и т.д.

- **Группа** - используется для обеспечения наглядности выбора из нескольких вариантов. В группе только один из элементов управления может иметь значение **ДА** (быть включенным).

- **Объект OLE** - предназначен для вывода в формы объектов **OLE**.

- **Командные кнопки** - обеспечивают вызов макросов или процедур **VBA**. С их помощью можно связать формы, чтобы создать законченное приложение.

Все элементы управления можно разделить на две большие группы. Первая (это, к примеру, информационные надписи) есть целиком автономная и ни с каким источником данных не связана. Вторая (например, поля ввода или переключения) без связи с таблицей вообще утрачивает всякий смысл. Это означает всю методику работы с определенными элементами.

5.3 Виды форм

Access может автоматически с помощью *Мастера форм* создавать формы таких видов:

а) *в один столбец*. Все поля размещаются в столбец и выводится только одна запись (как в **Excel** в режиме базы данных);

б) *ленточный*. Отформатированные записи выводятся на экран одна за другой, как в таблице, но более красочно. Каждая запись занимает отдельную ленточку. Отдельное поле может выводиться в несколько строк в пределах отведенного ему в ленточке места. Такая форма удобна для просмотра списка записей с небольшим числом полей;

в) *табличный*. Эта форма мало чем отличается от самой таблицы;

г) *выровненный*. Все поля размещаются в строку, одно за другим с переносом на другие строки значения тех полей, которые не поместились в текущем ряду;

д) *сводная таблица*. Применяется в тех случаях, когда возникает необходимость отображения не самих данных, а какого-то интегрированного суммарного результата;

е) *диаграмма*. Применяется в тех случаях, когда вместо конкретных цифровых величин необходимо показать пользователю более наглядную картинку. В этом случае, вместо цифр и ячеек, **Microsoft Excel** автоматически строит диаграмму по указанным полям, и выводит их на экран.

Кроме этих трех видов форм, создаваемых автоматически, можно создавать произвольные формы с помощью **Конструктора**:

а) *многостраничная форма*. Используется для вывода на экран записей, содержащих большой объем информации;

б) *подчиненная форма*. Используются для представления данных, находящихся на стороне “многие” отношения “*один-ко-многим*”;

в) *всплывающая форма*. Окно такой формы все время остается на переднем плане;

г) *монопольная форма*. Требуется как непременного условия дальнейшей работы приложения.

5.4 Построение формы

Формы создаются через окно диалога **Новая форма** (см.п.2.4.1). Имеется несколько способов создания формы.

1 Мастер форм.

Мастер форм можно вызвать через меню **Вставка**, с помощью командной кнопки Создать, или если в рабочей зоне вкладки **Форма** окна базы данных выбрать объект **Создание формы с помощью мастера**. Отдель-

ные функции этого мастера доступны непосредственно из окна **Новая форма**.

Сменяя друг друга, появится ряд диалоговых окон **Создание формы**, в которых необходимо:

- перенести в список **Выбранные поля** из списка **Доступные поля** (т.е. из выбранной таблицы или запроса) требуемые поля. При этом можно сменить таблицу или запрос и продолжить выбор из другого источника;

- выбрать вид создаваемой формы: в столбец, ленточный или табличный;

- выбрать стиль оформления формы (обычный, сумерки, камень, облака и т.п.);

- ввести заголовок (имя) формы.

2 Автоформа.

Мастера **Автоформа: в столбец**, **Автоформа: Ленточная** и **Автоформа: табличная** создают форму соответствующего типа, используя установки по умолчанию.

3 Конструктор.

Дает возможность “вручную” строить макет формы, создавая в нем требуемые элементы. То есть в этом режиме разработчик самостоятельно “рисует” новую форму. В режиме **Конструктора** можно также редактировать любую форму, в том числе и созданную с помощью мастера.

При вызове Конструктора **Access** откроет окно формы в режиме конструктора и выведет на экран окна **Панель элементов**, **Список полей**, **Окно свойств**. Некоторые из этих элементов (окон) могут отсутствовать. Для их вывода на экран необходимо воспользоваться соответствующей командой из меню **Вид**:

Вид ⇒ Панель элементов;

Вид ⇒ Список полей;

Вид ⇒ Свойства.

В новой форме будет только область данных с серой сеткой. Для добавления заголовка и раздела примечаний (п. 5.2.) необходимо выполнить команду

Формат ⇒ Заголовок /примечание формы.

Панель элементов содержит кнопки для всех элементов управления, которые можно использовать при создании формы. Для того чтобы создать на форме нужный элемент управления, необходимо щелкнуть на соответствующей кнопке на *Панели элементов*, поместить указатель мыши в точку, где будет находиться левый верхний угол элемента, и перетащить его в правый нижний.

Список полей содержит перечень полей базовой таблицы или запроса. Используется для создания присоединенных элементов управления. Для этого необходимо на панели элементов выбрать щелчком соответствующий тип и перетащить поле в требуемое место формы. При перетаскивании поля без предварительного выбора элемента на панели инструментов создается элемент управления по умолчанию. Обычно это элемент **Поле**. Если поле имеет свойства подстановки, создается элемент **Поле со списком**. Такой элемент можно создать для любого поля. Для этого на панели элементов нажать сначала кнопку **Мастер элементов**, а затем **Поле со списком** и перетащить поле из списка полей в таблицу. Появится серия сменяющих друг друга диалоговых окон **Создание полей со списком**, позволяющих построить соответствующий элемент управления. Эти окна похожи на окна, выводимые **Мастером подстановок** при определении свойства подстановки в процессе создания (корректировки) таблиц. Базовый запрос (а значит, и список полей) можно сменить. Для этого следует раскрыть **окно свойств для формы** и щелкнуть на свойстве **Источник записей** во вкладке **Данные**.

Окно свойств может открываться (кнопка **Свойства** или команда **Вид ⇒ Свойства**) для любого объекта: для всей формы, любого ее раздела (заголовка, области данных, раздела примечаний) или любого элемента

управления в зависимости от того, какой объект выделен. Если при открытом **Окне свойств** выделить какой-то другой элемент управления или раздел формы, то это окно автоматически изменится и будет отображать содержимое другого, т.е. выделенного объекта. Выделять форму в целом не очень удобно, поэтому для этого предусмотрен специальный элемент – утопленная рамка в левом верхнем углу формы. Когда форма активна (выделена) – эта рамка пуста, а когда активным – выделенным – является любой другой объект, кроме формы, в середину этой рамки помещается черный квадрат. Щелчок на этой рамке приводит к выделению формы в целом.

Набор свойств зависит от объекта (для формы в целом их более 70, для элемента управления более 30). Свойства сгруппированы по категориям на нескольких вкладках. Для многих свойств есть строители. Признаком их наличия является маленькая кнопка с многоточием, появляющаяся при выборе свойства.

Форма – это объект, с помощью которого пользователь общается с базой данных. Поэтому к дизайну формы ставятся повышенные требования. Для этого в **Access** имеются соответствующие способы.

Элементы управления можно перемещать по форме, изменять их размеры. Для придания элементам управления нужного вида их можно форматировать, используя панель инструментов форматирования (размер, шрифт, цвет, выравнивание и т.д.).

При щелчке на горизонтальной (или вертикальной) линейке выделяются все элементы управления, которые пересекает вертикальная (или горизонтальная) линия, проведенная из точки, в которой сделан щелчок. Выделить группу элементов можно также щелкая на каждом из них при нажатой клавише **Ctrl** на клавиатуре. Все выделенные элементы можно форматировать одновременно. В том числе выравнивать по левому или правому (верхнему или нижнему) краю, перемещать по форме.

Выровнять выделенные элементы можно с помощью команды:

Формат ⇒ Выровнять.

Для равномерного распределения элементов управления по полю формы используются команды:

Формат ⇒ Интервал по горизонтали;

Формат ⇒ Интервал по вертикали.

Элементы управления можно привязать к сетке с помощью команды:

Вид ⇒ Сетка.

Для управления последовательностью перехода при нажатой клавише **Tab** используется окно диалога **Последовательность перехода**, которое можно вызвать с помощью контекстного меню. В этом окне находится перечень всех элементов управления, в том порядке, в каком эти элементы будут активизироваться при каждом нажатии клавиши **Tab**. Этот порядок легко изменить путем перемещения соответствующих элементов в обозначенном окне.

Для улучшения внешнего вида формы можно использовать такие элементы, как линии и прямоугольники, а также цвет, шрифт и специальные эффекты оформления (вдавленное или приподнятое).

Созданные с помощью Мастера формы можно редактировать в режиме Конструктора. Так, можно изменить стандартный фон, который **Microsoft Access** автоматически выбирает для создания с помощью Мастера формы, на более подходящий. Для этого созданную форму необходимо перевести в режим конструктора и внести необходимые замены. В левом верхнем углу формы, как уже пояснялось, существует утопленная рамка с черным квадратом в середине. Если на нем осуществить двойной щелчок, то на экране появится окно свойств формы. Это окно можно также вызвать с помощью контекстного меню этого же элемента. На вкладке **Макет** находится свойство **Рисунок**, которое содержит имя файла с рисунком. Можно заменить рисунок на другой, или удалить его совсем. Между тем, при выделении этого свойства правее от его значения появится кнопка с тремя точками. Щелчок по этой кнопке открывает окно для поиска нужного файла. Вообще в окне свойств формы можно изменить разные свойства формы.

При выделении в форме какого-нибудь элемента окно свойств перестраивается на отображение свойств именно этого элемента.

5.5 Создание и внедрение подчиненных форм

Часто необходимо просматривать и редактировать информацию из различных таблиц, связанных отношением “*один-ко-многим*”. В этом случае удобно пользоваться подчиненными формами. **Подчиненная форма** – это форма, которая находится внутри другой формы. Первичная форма называется **главной формой**, а форма внутри формы называется **подчиненной формой**. Комбинацию “**форма/подчиненная форма**” часто называют также иерархической формой или комбинацией “родительской” и “дочерней” форм. Главная форма может иметь любое число подчиненных форм, если каждая подчиненная форма помещается в главную. Имеется также возможность создавать подчиненные формы до 7 уровней вложений. Это означает, что можно иметь подчиненную форму в середине главной формы, а другую подчиненную форму – в середине этой подчиненной формы и т.д.

При создании главной формы и подчиненной формы, созданных на основе таблиц, связанных отношением “*один-ко-многим*”, главная форма представляет сторону “один” отношения, а подчиненная форма – сторону “много”. Главная форма синхронизируется с подчиненной таким образом, что в подчиненной форме выводятся только записи, связанные с записью в главной форме. Если подчиненная форма создается с помощью мастера или путем перетаскивания формы, таблицы или запроса из окна базы данных в другую форму, **Microsoft Access** автоматически синхронизирует главную форму с подчиненной формой при соблюдении двух следующих условий:

- определенные отношения для выбранных таблиц или для базовых таблиц выбранных запросов;
- главная форма основана на таблице с начальным ключом, а подчиненная форма основана на таблице, которая имеет поле с таким же именем,

как и у этого первичного ключа, и с таким же или совместимым типом данных.

Как показано, подчиненные формы особенно удобны для вывода данных из таблиц или запросов, связанных отношением “*один-ко-многим*”.

Пусть, например, необходимо создать форму для ввода в базу данных нового договора или корректировки существующего. В одном договоре с конкретным заказчиком может быть заказано несколько изделий. В базе данных эти данные представлены двумя таблицами *Договор* и *СоставДогов*, связанных отношением “*один-ко-многим*”. Данные из таблицы *Договор* находятся на стороне “один”, а данные таблицы *СоставДогов* – на стороне “многo”. Такую информацию удобно представить с помощью подчиненной формы. Для этой цели сначала необходимо на базе таблицы *СоставДогов* составить подчиненную форму (рис.13).

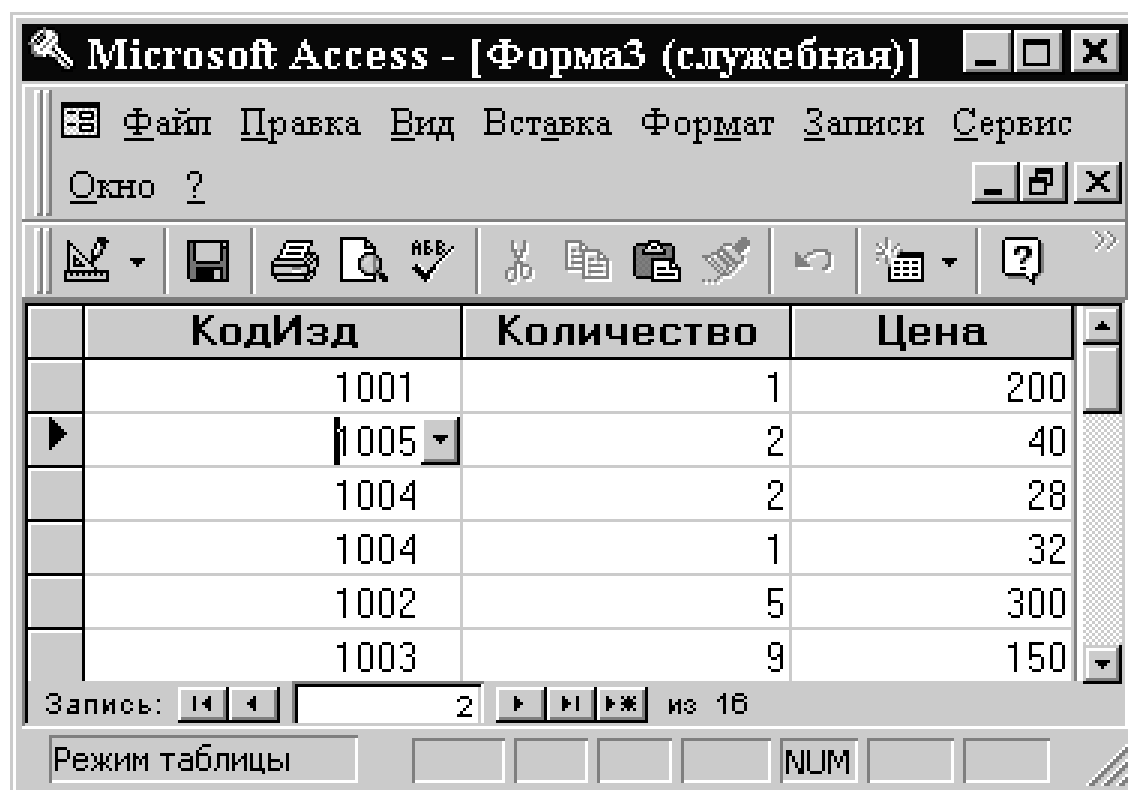


Рисунок 13 – Подчиненная форма

Данная форма не содержит никакой содержательной информации о договоре. Это просто список изделий с указанием количества и цены. А эти данные относятся к договору и имеют смысл только совместно с номером договора. Следовательно, самостоятельного значения такая форма не имеет, т.к. она бессмысленна.

Основная форма разрабатывается на базе таблицы *Договор* и содержит общую информацию о договоре.

Для внедрения подчиненной формы ее необходимо перетащить из вкладки **Формы** окна базы данных в требуемое место основной формы. Естественно, оба окна должны быть открыты и доступны.

Другой способ внедрения подчиненной формы – кнопка **Подчиненная форма/отчет** на панели элементов. Если при этом будет нажата кнопка **Мастер элементов**, то **Access** запустит мастера, чтобы помочь построить подчиненную форму с нуля. Если подчиненная форма создана ранее, услуги мастера не понадобятся, но после создания элемента управления **Подчиненная форма** в качестве значения его свойства **Объект-источник** ввести имя ранее созданной формы.

После внедрения подчиненной формы следует проверить ее свойства **Подчиненные поля** и **Основные поля**, чтобы убедиться, что **Access** правильно связал формы. (Эти свойства устанавливают связь между основной и подчиненной формами.) Обычно **Access** пытается автоматически установить такую связь. В результате будет получена форма, изображенная на рис.14.

Формирование договора

Заказчик:

Изделие:

Договор N от Призн.опл.

Код заказчика

Состав договора:

КодИзд	Количество	Цена
1001	1	200
1005	2	40
1004	2	28
1004	1	32
1002	5	300

Адрес:

Телефон:

Запись: 1 из 6

Режим формы: NUM

Рисунок 14 – Форма с внедренной в нее подчиненной

Теперь данные в подчиненной форме имеют смысл в контексте конкретного договора: они относятся к текущему договору, отображаемому в основной форме.

Главная и подчиненная формы в этом примере связаны таким образом, что в подчиненной форме выводятся только те записи, которые связаны с текущей записью в главной форме. То есть в подчиненной форме выводится информация, относящаяся к текущему договору. При смене договора в основной форме меняется его состав в подчиненной.

5.6 Связывание форм с помощью командных кнопок

Одной из особенностей Access является его способность запускать макросы или процедуры VBA в ответ на некоторые События в форме.

Перечень событий, на которые можно отреагировать таким способом, находится во вкладке События окна свойств конкретного объекта.

Большинство действий по обработке событий можно выполнить с помощью макросов. Они выполняют более 40 различных действий.

С помощью макросов и элементов управления **Командная кнопка** можно связывать формы. Например, если при работе с договором выясняется, что в базу данных не занесено изделие, которое заказчик хочет заказать, удобно было бы, не прекращая работы, вызвать соответствующую форму для ввода в базу данных нового (или корректировки существующего) изделия и после завершения работы с этой формой снова продолжить работу с договорами.

Это можно сделать с помощью командной кнопки, вызывающей макрос в ответ на событие **Нажатие кнопки**. Для ее создания необходимо из вкладки **Макросы** окна базы данных перетащить в нужное место формы соответствующий макрос (который открывает требуемую форму). **Макрос** – это набор с одной или более макрокоманд, которые выполняют определенные операции, такие как открытие форм или печать отчетов. Макросы могут быть полезны для автоматизации часто выполняемых задач. Например, при нажатии пользователем кнопки можно запустить макрос, который напечатает отчет.

При создании элемента управления **Командная кнопка** автоматически подключается Мастер для создания макросов, необходимых для обработки события **Щелчок на кнопке**. Обычно эта работа с формами или отчетами.

6 ОТЧЕТЫ

6.1 Назначение отчетов

Таблицу, запрос или форму можно вывести на печать. Однако форматирование таблиц и запросов ограничено изменением различных шрифтов. В формах возможности форматирования шире, но они плохо приспособле-

ны для проведения сложных вычислений, группировки данных и получения промежуточных и общих итогов. Формы ориентированы на отображение информации на экране.

Создать действительно полноценный выходной документ с использованием всех средств форматирования текста и дополнительной обработки данных (например, с включением в документ промежуточных и общих итогов) можно только с помощью отчета.

Отчет представляет собой эффективный способ представления данных в печатном формате. Имея возможность управлять размером и внешним видом всех элементов отчета, пользователь может отобразить сведения по своему желанию

В отчетах можно:

- для обеспечения иерархического представления осуществлять группировку данных;
- производить сложные вычисления не только внутри некоторой группы или набора строк, но и по нескольким группам одновременно;
- осуществлять разбивку документа на листы с формированием колонтитулов;
- вставить в любой раздел объекты **OLE**.

В основном отчеты применяются для просмотра “картины в целом“, поэтому в качестве основы часто используются многотабличные запросы.

6.2 Элементы отчета

Структура отчета во многом напоминает структуру формы. Однако формат отчета соответствует стандартам бумаги и имеет свою специфику.

Большая часть того, что было сказано про формы, относится и к отчетам.

Отчет, так же, как и форма, имеет **заголовок, область данных и раздел примечаний**. Кроме того, на каждом листе можно задавать **верхний и нижний колонтитулы**. Роль колонтитулов заметна, если отчет занимает

более одной страницы. В них выводится служебная информация, такая например, как номер страницы, общее число страниц и т.д.

В области данных для каждой строки (записи) из набора записей базового запроса (или таблицы) отводится одна строка (*информационная строка*). Данные можно **сгруппировать** по некоторым полям. Все строки, у которых совпадают значения группировочных полей, составляют группу. Для группы можно определить заголовок и область примечаний (в которой можно подвести итоги). Данные внутри группы, в свою очередь, можно снова сгруппировать (по другим полям). Это группировка второго уровня. И т.д. Всего допускается до 10 уровней группировки.

В отчете можно использовать такие элементы:

- надпись (свободный текст);
- поле (свободное или связанное с полем базы данных);
- объект **OLE**;
- линии, прямоугольники;
- подчиненные формы или отчеты.

6.3 Построение отчета

Чтобы построить отчет, необходимо вызвать диалоговое окно **Новый отчет** (см. п. 2.4.1). Возможны следующие способы построения отчета:

Мастер отчетов – запуск основного мастера по разработке отчетов, позволяющего выбрать поля для отчета, задать форматы, условия группировки и итоговые функции. На экране появляется ряд сменяющих друг друга диалоговых окон **Создание отчетов**, в которых необходимо:

1 Переместить в список **Выбранные поля** из списка **Допустимые поля** (т.е. выбранного запроса или таблицы) требуемые поля. Таблицу или запрос можно сменить.

2 Выбрать способ группировки данных. При этом мастер исследует исходные данные и попытается определить “естественные” группы. Он

предлагает сгруппировать информацию по ключевым полям таблиц, стоящих на стороне “один” отношения “*один-ко-многим*”. Можно принять предложенную мастером группировку или определить свою.

3 Задать условия сортировки в области данных (до четырех полей).

4 Сформировать для некоторых полей итоговые значения (**Sum, Avg, Min, Max**), а также долю сумм (процент) в общих итогах. Можно задать вывод только итоговых данных.

5 Выбрать вид макета отчета.

6 Выбрать один из шести встроенных стилей оформления отчета.

7 Задать имя отчета.

Автоотчет: в столбец – запуск мастера создания отчета в виде простого списка, содержащего названия и значение полей.

Автоотчет: ленточный – каждая запись выводится в отдельной строке. Если таблицы базового запроса связаны отношением “один-ко-многим”, мастер создает группу для данных, поступающих со стороны отношения “один”, но не вычисляет никаких итоговых данных.

Конструктор – создание отчета с помощью конструктора, т.е. “вручную”. В этом режиме можно также редактировать структуру отчета. Открывается окно формы в режиме конструктора и выводится на экран **Панель элементов, Список полей и Окно свойств**. Структура, состав и назначение этих элементов, а также порядок проектирования отчета такие же, как и в конструкторе форм. Некоторые из этих элементов могут отсутствовать. Для их вывода на экран необходимо воспользоваться соответствующей командой из меню **Вид**:

Вид ⇒ Панель элементов;

Вид ⇒ Список полей;

Вид ⇒ Свойства.

В новой форме будет только область данных с серой сеткой. Область данных предназначена для описания одной информационной строки. Размер этой области по вертикали (высота) задает высоту информационных

строк в отчете. Приемы формирования этой области такие же, как и при формировании формы. Размещение элементов управления выполняется с помощью **Панели элементов**. Элементы управления в этом случае играют роль элементов оформления, так как печатный отчет не является интерактивным объектом.

Для добавления заголовка и раздела примечаний (п.6.2) необходимо выполнить команду

Формат ⇒ Заголовок /примечание формы.

Для добавления или удаления колонтитулов:

Вид ⇒ колонтитулы.

Кнопка **Сортировка и группировка** на панели инструментов позволяет осуществлять сортировку и группировку информации. Появится диалоговое окно, в котором следует определить поля (или выражения) для группировки, а также некоторые свойства группы, в том числе наличие или отсутствие заголовка группы и раздела примечаний.

В качестве значения свойства **Данные** любого элемента управления, позволяющего выводить данные, можно использовать выражение. Признаком выражения является знак равенства в первой позиции. В выражении допустимо использование встроенных функций.

COUNT([имя-поля]) - количество записей.

SUM([имя-поля]) - сумма значений.

MIN, MAX, Avg и др. - итоговые функции.

Page - номер текущей страницы.

Pages - общее число страниц в отчете.

Date() - текущая дата без компонента времени.

New() - текущая дата и время.

7 МАКРОСЫ

7.1 Назначение макросов

В **Access** можно определить макрос, который выполнит практически те же действия, которые можно реализовать путем нажатия клавиш или при помощи мыши. Основные преимущества макросов в том, что они могут выполняться и отвечать на большое число видов событий. Событие – это замена данных, открытие и закрытие форм или отчетов, передача фокуса от одного элемента управления к другому и т.д. В макрос можно включить несколько макрокоманд и задать проверку условий таким образом, что в зависимости от значения данных в форме или отчете будут выполняться различные действия.

Access имеет разные типы макрокоманд, которые позволяют автоматизировать работу приложения. Можно использовать макросы для:

- открытия или закрытия какого-нибудь объекта **Access** в каком-либо доступном режиме;
- выполнения запроса на выборку или замену, пользуясь значениями элементов управления какой-либо открытой формы;
- выполнения действий в зависимости от значения в базе данных, формы или отчета;
- установления значений какого-нибудь элемента управления формой или отчета;
- обновления значений какого-нибудь элемента управления, источником которых является запрос;
- применения фильтра, перехода от какой-либо записи и поиск данных в базовой таблице ли запросе формы;
- выполнения какой-нибудь команды какого-либо меню **Access**;
- передачи фокуса любому из окон или какому-нибудь элементу управления в окне;

- вывода на экран информационных сообщений;
- перемещения, копирования, распечатки каких-либо объектов базы данных в текущую или другую базу данных **Access**;
- импортирования, экспортирования или связывания таблиц с другими базами данных.

7.2 Создание макросов

Создание макроса происходит через диалоговое окно макрос (п. 2.4.1). Структура окна показана на рисунке 15.

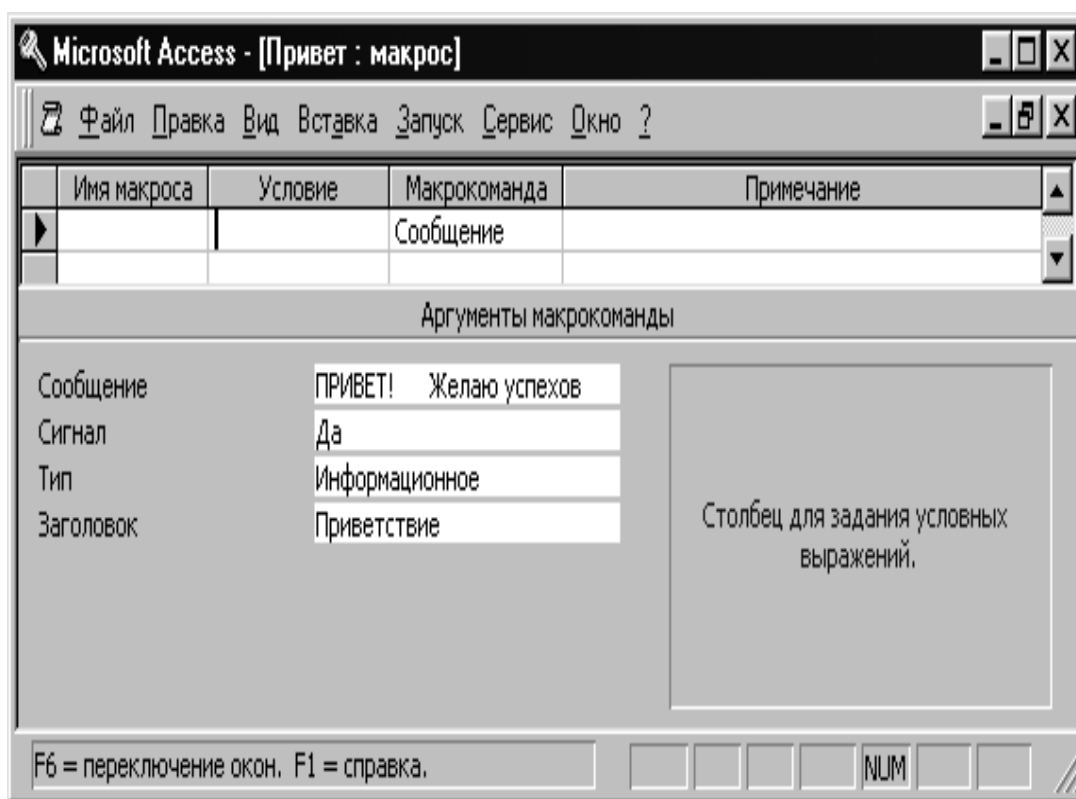


Рисунок 15 – Окно макроса

Верхняя часть окна используется для определения макроса, левая нижняя – для введения значений аргументов макрокоманд, включенных в макрос. Правая нижняя часть содержит короткую справку, которая зависит

от положения курсора (контекстную справку всегда можно получить путем нажатия клавиши <F1>).

Верхняя часть – это таблица из четырех столбцов. Первые два столбца могут быть отсутствующими. Для их появления необходимо щелкнуть на кнопки **Имена макросов** и **Условия** на панели инструментов соответственно. Можно воспользоваться соответствующим пунктом операционного меню **Вид**.

В столбце **Макрокоманда** задается одна из более чем 50 макрокоманд **Access**. Ее можно набрать непосредственно (распознается по нескольким первым буквам) или воспользоваться списком, который раскрывается в центре этого столбца.

Столбец **Примечание** используется для документирования макроса. Можно вводить дополнительные комментарии, размещая их на свободных строках, не придерживаясь правил макрокоманд.

Документирование созданных макросов – это хороший стиль программирования. Оно облегчает налаживание макросов, внесение изменений в будущем. Кроме того, облегчается чтение макросов. Примечания разъясняют работу каждой макрокоманды, устраняя необходимость запоминания аргументов.

После выбора макрокоманды, в нижней части окна макроса выводятся ее аргументы, как показано на рисунке 16.

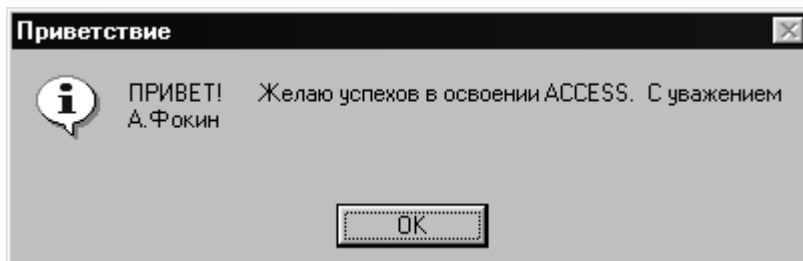


Рисунок 16 – Пример выполнения макроса **Привет**

Состав аргументов зависит от макрокоманды. В примере выбрана макрокоманда **Сообщение**, использованная для открытия монопольного окна диалога с сообщением. Значение аргумента **Сообщение** – это текст, который будет выведен в диалоговом окне. Аргумент **Сигнал** задает, будет ли появление диалогового окна сопровождаться звуковым сигналом. Аргумент **Тип** позволяет вставить в окно Сообщение значок, например, красный круг с белым крестиком и т.п. Аргумент **Заголовок** задает текст, который будет выведен в заголовок диалогового окна. Перед запуском макроса его необходимо сохранить и присвоить имя (в нашем примере **Привет**).

7.3 Выполнение макросов

Некоторые макросы (как, например, созданный выше) могут быть запущены на выполнение непосредственно из окна базы данных щелчком на командной кнопке **Запуск** или из окна макроса через меню **Запуск**. Пример выполнения созданного выше макроса **Привет** был представлен на рис. 16.

Однако такой способ выполнения макроса используется только для отладки, и то с ограничением (необходимо, чтобы были открыты все формы и отчеты, на которые из макроса имеются ссылки). При отладке можно пользоваться также пошаговым выполнением макроса (команда **Запуск** ⇒ **по шагам** или аналогичной кнопкой на панели инструментов). В этом случае перед выполнением каждого шага будет открываться диалоговое окно Пошаговое выполнение макроса (рис. 17), которое содержит имя макроса, название макрокоманды, условие ее выполнения и аргументы.

Похожее окно появится и при выявлении ошибок в макросе во время выполнения приложения (перед этим будет выведено окно, которое разъяснит ошибку).

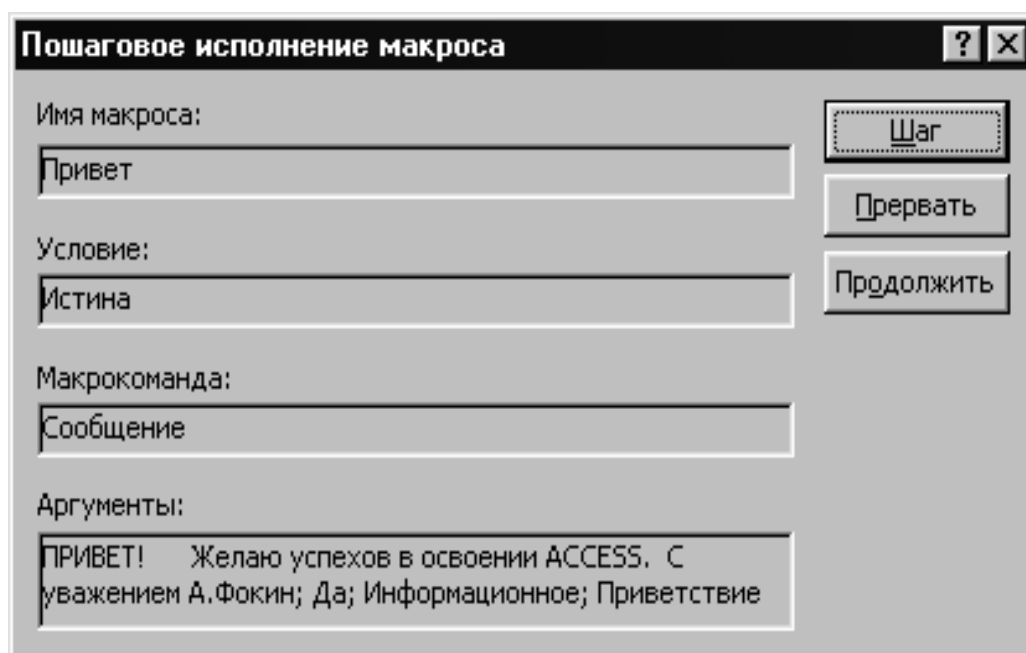


Рисунок 17 – Окно диалога *Пошаговое выполнение макроса*

Пошаговый режим распространяется на какой-нибудь макрос, поэтому его необходимо отключить, если необходимость в нем отпадает. Отключение происходит тем же путем, что и включение.

При работе приложений макросы на выполнение запускаются автоматически в ответ на возникновение определенных событий в форме или отчете. Для чего этот макрос необходимо связать с этим событием:

- открыть форму или отчет в режиме конструктора;
- активизировать соответствующий элемент управления, раздел или форму (отчет) в целом;
- открыть **Окно свойств** (щелчок на кнопке **Свойства** на панели инструментов, выполнить команду **Вид** ⇒ **Свойства** или воспользоваться контекстным меню);
- перейти на вкладку **События**;
- выбрать необходимое событие и задать для него имя необходимого макроса. При этом можно воспользоваться списком, который раскрывается, в центре свойства, которое содержит перечень всех макросов в группе мак-

росов, определенных в базе данных. Можно также обратиться к услуге **Мастера** для создания нового макроса.

Макрос можно запустить на выполнение и с другого макроса с помощью макрокоманды **Запуск мастера** или из модуля **VBA**.

7.4 Ссылки на объекты форм и отчетов

При создании макросов для автоматизации приложений часто приходится ссылаться на формы, отчеты или элементы управления этими объектами, чтобы установить свойства или значения.

Ссылки на открытую форму или отчет включают имя семейства (Collection)-Forms или Reports соответственно, за которым ставится восклицательный знак и имя объекта. Если имя содержит пробелы или специальные символы, его необходимо взять в квадратные скобки.

Например, ссылка на форму *Форма1* имеет вид

Forms!Форма1.

Ссылки на элемент управления или форму отчета создаются путем добавки до ссылки на объект восклицательного знака и имени управляемого элемента. Ссылка на элемент *Поле1* формы *Форма1* имеет вид

Forms!Форма1!Поле1.

Для ссылки на свойства объекта или элемента управления необходимо до ссылки на этот объект или элемент приложения добавить через точку имя свойства. Так, ссылка на свойство **ScrollBars** (линейки прокрутки) формы *Форма1* имеет вид

Forms!Форма1.ScrollBars.

Ссылка на свойство **Visible** (вывод на экран) элемента управления *Поле1* формы *Форма1* имеет вид

Forms!Форма1!Поле1.Visible.

На подчиненную (внедренную) форму или отчет можно ссылаться так же, как и на какой-либо другой элемент управления. Например, ссылка на свойство **Visible** формы *Форма1П*, внедренной в форму *Форма1*, имеет вид

Forms!Форма1!Форма1П.Visible.

Элемент управления **Подчиненная форма** имеет специальное свойство **Form** (**Подчиненный отчет – Report**), что позволяет ссылаться на объект, встроенный в этот элемент управления. Например, ссылка на свойство **Foutweight** (насыщенность) элемента *Поле3* формы *Форма1П*, встроенная в форму *Форма1*, имеет вид

Forms!Форма1!Форма1П.Форм!Поле3. Foutweight.

7.5 Группы макросов. Условия в макросах

В **Access** внутри одного макроса можно задать несколько макрокоманд и порядок их выполнения. На рис.18 приведен пример макроса **Autoexec**.

В этом макросе задано 8 макрокоманд:

- **Запуск Макроса** – запускает ранее созданный макрос **Привет**.
- **Песочные Часы** – выводит указатель мыши в виде песочных часов, чтобы показать пользователю, что следующие шаги могут занять несколько секунд.

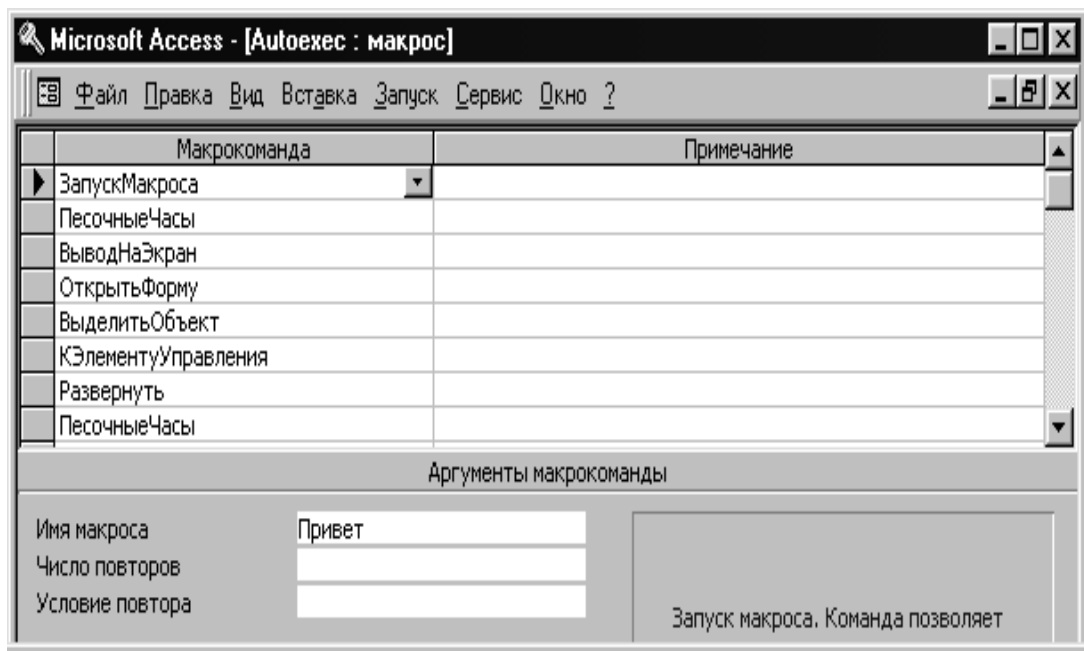


Рисунок 18 – Окно макроса *Autoexec*

- **Вывод на Экран** – предотвращает вывод на экран лишних изображений (аргумент **Включить вывод**, установленный в значение **Нет**).

- **Открыть Форму** – открывает форму *Форма3*. Аргумент **Режим** устанавливает режим, в котором форма будет открыта (допустим: **Форма**, **Конструктор**, **Просмотр** и **Таблица**). Аргумент **Режим данных** может принимать значения: **Добавление**, **Замена** и **Только чтение**. Аргумент **Режим окна** может бы **Невидимое**, **Значок**, **Окно диалога**.

- **Выделить Объект** – завершает открытие формы и передает ей фокус. **Access** не всегда ждет завершения работы макрокоманды перед выполнением следующей. Так, предыдущая макрокоманда только начинает открытие формы.

- **К Элементу Управления** – передает фокус конкретному элементу управления.

- **Развернуть** – увеличивает активное окно (т.е. окно, которое содержит объект, владеющий фокусом) до размеров рабочей области **Access**.

- **Песочные Часы** – обновляет предыдущее представление указателя мыши, информируя пользователя про окончание работы макроса.

В больших приложениях могут создаваться сотни макросов. Для удобства макросы можно группировать, объединять в одну группу несколько однотипных макросов. Каждому макросу в группе дается имя, которое заносится в столбец **Имя макроса** первой макрокоманды группы.

Для ссылки на конкретный макрос группы необходимо указать имя группы и через точку – имя макроса внутри группы. **Access** выполняет заданный макрос, начиная с его первой макрокоманды, пока не повстречает макрокоманду **Остановить Макрос** или имя следующего макроса. В начале группы может находиться несколько макрокоманд, вызванных при вызове группы без указания имени макроса.

8 ПРАКТИЧЕСКИЕ ЗАДАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ

8.1 Лабораторная работа №1. Создание схемы базы данных

1 По таблице 3 выбрать структуру (схему) базы данных, соответствующую номеру варианта.

2 Проанализировать схему. Исходя из смысла данных, самостоятельно установить типы полей в каждой таблице и их свойства, присвоить полям имена.

3 Создать пустую базу данных. В качестве имени файла базы данных использовать свою фамилию.

4 В режиме конструктора создать описание всех таблиц базы данных. При этом использовать свойство подстановки для замены неинформативных кодов соответствующими им значениями из других таблиц

Таблица 3 – Варианты структуры (схемы) базы данных

Вариант	Номер схемы	Таблица	Поля таблицы
1	2	3	4
0	1	Табл.1	Код заказчика
		Заказчик	Наименование заказчика
			Адрес: город
			Улица
			Дом
			Телефон
		Табл.2	Код изделия
		Изделие	Наименование изделия
			Вес изделия
			Максимальная длина
		Табл.3	Номер договора
		Договор	Код заказчика
			Дата заключения договора
			Призн.оплаты (перечисления денег по договору)
		Табл.4	Номер договора
		СоставДогов	Код изделия
			Количество изделий по договору
			Цена одного изделия
1	1	Табл.1	Код заказчика
		Заказчик	Наименование заказчика
			Банк
			МФО
			Расчетный счет

Продолжение табл. 3

1	2	3	4
		Табл.2	Код товара
		Товар	Стандарт
			Единица измерения
			Изготовитель
		Табл.3	Код заказа
		Заказ	Код заказчика
			Дата начала
			Дата завершения
			Признак завершения (да/нет)
		Табл.4	Код заказа
		СоставЗаказа	Код товара
			Количество товара
			Цена одной единицы
2	1	Табл.1	Код отдела
		Отдел	Наименование отдела
			ФИО начальника отдела
		Табл.2	Шифр темы
		Тема	Наименование темы
			Наименование заказчика
			Общая стоимость работ
			Дата завершения работ
		Табл.3	Табельный номер сотрудника
		Сотрудник	Код отдела
			ФИО сотрудника
			Часовая тарифная ставка
			Пол: муж/жен
		Табл.4	Табельный номер сотрудника

Продолжение табл. 3

1	2	3	4
		Работа	Шифр темы
			Время отработанное по теме (часов)
3	2	Табл.1	Код банка
		Банк	Наименование банка
			Адрес банка
		Табл.2	Код клиента
		Клиент	ФИО
			Адрес
			Признак: физическое/юридическое лицо
		Табл.3	Номер счета
		Счет	Код банка
			Код клиента
			Дата открытия счета
			Вид счета: простой/срочный
			Признак: счет закрыт/действующий
		Табл.4	Дата
		Операция	Номер счета
			Вид операции: взнос/расход
			Сумма операции
			Процент
4	1	Табл.1	Код вуза
		Вуз	Полное наименование вуза
			Продолжение табл. 1
		Табл.1	Сокращенное наименование вуза
		Вуз	Адрес
			Телефон

Продолжение табл. 3

1	2	3	4
		Табл.2	Код школы (техникума)
		Школа	Наименование школы (техникума)
			Адрес
			Признак: школа/техникум
		Табл.3	Шифр группы
		Группа	Код вуза
			Факультет
		Табл.4	Код студента
		Студент	Шифр группы
			ФИО
			Год поступления в вуз
			Год окончания школы (техникума)
			Код школы (техникума)
5	4	Табл.1	Шифр группы
		Группа	Факультет
			ФИО куратора
		Табл.2	Код преподавателя
		Преподаватель	ФИО преподавателя
			Признак наличия ученой степени: да/нет
		Табл.3	Код студента
		Студент	Шифр группы
			ФИО студента
			Год рождения
			Пол: муж/жен
		Табл.4	Наименование предмета

Продолжение табл.3

1	2	3	4
		Предмет	Шифр группы
			Код преподавателя
			Количество часов
			Признак: экзамен/зачет
6	1	Табл.1	Код предприятия
		Предприятие	Наименование предприятия
			ФИО директора
			Номер телефона
			Признак собственности: государственное/частное
		Табл.2	Код страховой компании
		СтрахКомпан	Наименование страховой компании
			Адрес
			Номер телефона
			Признак: государственная/частная
		Табл.3	Код предприятия
		Сотрудник	Код сотрудника
			ФИО сотрудника
			Адрес сотрудника
			Дата рождения
			Пол: муж/жен
		Табл.4	Номер договора страхования
		СтрахДогов	Код страховой компании
			Код сотрудника
			Общая сумма договора
			Дата начала договора
			Дата окончания договора

Продолжение табл. 3

1	2	3	4
7	1	Табл.1	Код цеха
		Цех	Наименование цеха
			ФИО начальника цеха
			Рабочий телефон
			Признак основной / вспомогательный Цех
		Табл.2	Код начисления/удержания
		СправНачисл	Наименование начисления/удержания
			Вид: начисление/удержание
		Табл.3	Табельный номер сотрудника
		Сотрудник	Код цеха
			ФИО сотрудника
			Признак: АУП/рабочий
			Домашний адрес
		Табл.4	Месяц
		Зарплата	Табельный номер сотрудника
			Код начисления/удержания
			Сумма
8	2	Табл.1	Код пункта
		Город	Страна
			Город
			Телефон справочной службы
		Табл.2	Код авиакомпании
		Компания	Краткое наименование авиакомпании
			Полное наименование авиакомпании
			Адрес головного офиса
			Телефон справочной службы

Продолжение табл.3

1	2	3	4
			Признак собственности: государственная/частная
		Табл.3	Номер рейса
		Рейс	Код пункта отправления
			Код пункта назначения
			Код авиакомпании
			Время отправления
			Признак: есть посадки в пути/нет
		Табл.4	Номер рейса
		Места	Дата
			Наличие билетов 1 класса
			Наличие билетов экономического класса
9	3	Табл.1	Код станции
		Станция	Наименование города
			Наименование железнодорожной станции
			Признак: узловая/неузловая станция
			Телефон справочной службы
9	3	Табл.2	Код владельца поезда
		Владелец	Наименование владельца поезда
			Наименование ж/д станции
		Табл.3	Номер поезда
		Поезд	Признак: скорый/пассажирский
			Признак: фирменный/обычный
			Код станции отправления
			Код станции назначения

Продолжение табл.3

1	2	3	4
			Код владельца поезда
		Табл.4	Номер поезда
		Билеты	Дата
			Наличие плацкартных билетов
			Наличие купейных билетов
			Наличие СВ билетов
10	5	Табл.1	Код изделия
		План	Наименование изделия
			Количество изделий на план
			Признак: включено/не включено в план
		Табл.2	Код группы материалов
		ГруппаМатер	Наименование группы материалов
		Табл.3	Код материала
		Материал	Наименование материала
			Код группы материала
			Единица измерения
			Количество материала на складе
		Табл.4	Код изделия
		Норма	Код материала
			Признак: основной/технужды
			Норма расхода
11	5	Табл.1	Код изделия
		План	Наименование изделия
			Квартал
			Количество изделий на план квартала
			Признак: план/проект
		Табл.2	Код группы оборудования

Продолжение табл.3

1	2	3	4
		ГруппаОбо- руд	Наименование группы оборудования
		Табл.3	Код станка (оборудования)
		Станок	Наименование станка
			Код группы оборудования
			Признак: основное/вспомогательное
			Фонд рабочего времени
		Табл.4	Код изделия
		Норма	Код оборудования
			Время Тшт (время на изготовление де- тали - штучное время)
			Время Тпз (время подготовительно- заключительное)
12	4	Табл.1	Код детали
		Деталь	Наименование детали
			Вес детали
		Табл.2	Код профессии
		Рабочий	Разряд
			Наименование профессии
			Признак: основной рабочий / вспомога- тельный
		Табл.3	Квартал
		План	Код детали
			Количество на квартал
			Признак: проект/план
			Признак: экспорт/внутренний
		Табл.4	Код детали

Продолжение табл.3

1	2	3	4
		Операция	Код профессии
			Разряд
			Время Тшт (время на изготовление детали - штучное время)
			Время Тпз (время подготовительно-заключительное)
13	6	Табл.1	Код изделия (детали)
		План	Наименование изделия
			Количество изделий на план
			Признак: план/проект
			Признак: экспорт/внутренний заказ
		Табл.2	Код материала
		Материал	Наименование материала
			Код единицы измерения
		Табл.3	Код оборудования
		Оборудование	Наименование оборудования
			Фонд рабочего времени
		Табл.4	Код изделия
		Операция	Код материала
			Код оборудования
			Норма расхода материала
			Норма времени
14	5	Табл.1	Код больного
		Больной	ФИО больного
			Домашний адрес
			Домашний телефон

Продолжение табл.3

1	2	3	4
			Год рождения
			Пол: муж/жен
		Табл.2	Код поликлиники
		Поликлиника	Наименование поликлиники
			Адрес поликлиники
			Признак: государственная/частная
		Табл.3	Код врача
		Врач	Код поликлиники
			ФИО врача
			Признак: основной/совместитель
		Табл.4	Код больного
		Посещения	Код врача
			Дата посещения
			Диагноз
15	5	Табл.1	Код цеха
		ЦехУчасток	Код участка
			Наименование цеха и участка
			Признак: основной/не основной
		Табл.2	Код группы оборудования
		ГруппаОбо- руд	Наименование группы оборудования
		Табл.3	Код оборудования
		Оборудова- ние	Наименование оборудования
			Код группы оборудования
		Табл.4	Инвентарный номер станка
		Станок	Код оборудования

Продолжение табл.3

1	2	3	4
			Код цеха
			Код участка
			Дата ввода в эксплуатацию
16	2	Табл.1	Код банка
		Банк	Наименование банка
			Признак: государственный/частный
			Адрес банка
			Телефон
		Табл.2	Код предприятия
		Предприятие	Наименование предприятия
			Адрес предприятия
			ФИО директора
			Телефон
			Признак: государственное/частное
		Табл.3	Номер договора о кредите
		ДоговКреди- та	Код банка
			Код предприятия
			Сумма кредита
			Дата начала погашения
			Дата окончания погашения
		Табл.4	Номер договора о кредите
		Погашение	Дата
			Сумма погашения кредита
17	1	Табл.1	Код цеха
		Цех	Наименование цеха
			Признак: основной / вспомогательный

Продолжение табл.3

1	2	3	4
		Табл.2	Код специальности
		Специальность	Наименование специальности
			Признак образования: высшее/нет
		Табл.3	Табельный номер сотрудника
		Сотрудник	Код цеха
			Год рождения
			Домашний адрес
			Пол: муж/жен
			ФИО сотрудника
		Табл.4	Табельный номер сотрудника
		Связь	Код специальности
			Дата получения специальности
			Номер документа
			Признак образования: дневное/нет
18	2	Табл.1	Код авиакомпании
		Авиакомпания	Адрес
			Признак: государственная/частная
			Телефон
			Наименование авиакомпании
		Табл.2	Код пункта назначения
		Пункт	Наименование пункта назначения
			Адрес
			Признак расположения аэропорта: в черте города/нет
		Табл.3	Номер рейса

Продолжение табл.3

1	2	3	4
		Рейс	Код авиакомпании
			Код пункта назначения
			Признак промежуточной посадки: есть/нет
		Табл.4	Номер рейса
		Билеты	Дата вылета
		Табл.4	Наличие билетов экономического класса
		Билеты	Наличие билетов 1 класса
19	5	Табл.1	Код материала
		Материал	Марка материала
			Наименование материала
			Стандарт на материал
			Единица измерения
		Табл.2	Код поставщика
		Поставщик	Наименование поставщика
			Адрес
			Телефон
			ФИО ответственного
			Признак: государственное/частное предприятие
		Табл.3	Номер договора поставки
		Договор	Дата заключения договора
			Код поставщика
			Срок поставки
		Табл.4	Номер договора
		Состав	Код материала

Продолжение табл.3

1	2	3	4
			Количество
			Цена одной единицы
			Признак: получено/не получено
20	6	Табл.1	Код страховой компании
		СтрахКомпа- ния	Наименование страховой компании
			Адрес
			Признак: государственная/частная
			Телефон
		Табл.2	Код вида страхования
		СпрВидСтра х	Наименование вида страхования
		Табл.3	Идентификационный код клиента
		Клиент	ФИО клиента
			Адрес клиента
			Телефон
			Пол: муж/жен
		Табл.4	Идентификационный код клиента
		Договор	Код страховой компании
			Код вида страхования
			Дата
			Сумма
			Дата окончания договора
21	4	Табл.1	Код детали
		Деталь	Наименование детали
			Вес детали
		Табл.2	Код материала

Продолжение табл.3

1	2	3	4
		Материал	Наименование материала
			Единица измерения
		Табл.3	Квартал
		План	Код детали
			Количество на квартал
			Признак: экспорт/внутренний заказ
		Табл.4	Код детали
		Норма	Код материала
			Норма расхода
22	4	Табл.1	Код детали
		Деталь	Наименование детали
			Вес детали
		Табл.2	Код станка
		Станок	Наименование станка
			Фонд рабочего времени
		Табл.3	Квартал
		План	Код детали
			Количество на квартал
			Признак: план/проект
		Табл.4	Код детали
		Норма	Код станка
			Время Тшт (время на изготовление детали - штучное время)
			Время Тпз (время подготовительно-заключительное)
23	4	Табл.1	Код изделия
		Изделие	Наименование изделия

Продолжение табл.3

1	2	3	4
			Признак исполнения: тропическое/обычное
		Табл.2	Код детали
		Деталь	Наименование детали
			Вес детали
		Табл.3	Квартал
		План	Код изделия
			Количество на квартал
			Признак: план/проект
			Признак: экспорт/внутренний заказ
		Табл.4	Код изделия
		СоставИздел	Код детали
			Количество деталей на изделие
24	5	Табл.1	Код оборудования
		Оборудование	Наименование оборудования
		Табл.2	Код цеха
		Цех	Наименование цеха
			ФИО начальника цеха
			Номер телефона
			Признак цеха: основной / вспомогательный
		Табл.3	Код цеха
		Участок	Номер участка
			Наименование участка
		Табл.4	Код цеха
		Станок	Номер участка

Продолжение табл.3

1	2	3	4
			Код оборудования
			Инвентарный номер
			Признак состояния: рабочее / законсервирован
25	6	Табл.1	Табельный номер сотрудника
		Сотрудник	ФИО сотрудника
			Домашний адрес
			Домашний телефон
			Год рождения
			Пол: муж/жен
		Табл.2	Код должности
		Должность	Наименование должности
			Признак: АУП/рабочий
		Табл.3	Код цеха
		Цех	Наименование цеха
			ФИО начальника цеха
			Признак: основной/вспомогательный цех
		Табл.4	Табельный номер сотрудника
		Работа	Код цеха
			Код должности
			Дата назначения
			Признак увольнения

5 Установить связи между созданными таблицами в соответствии со схемой базы данных. Установить опции, обеспечивающие целостность данных.

6 В режиме таблицы ввести в каждую таблицу несколько строк (10-15). При этом в подчиненные таблицы следует ввести несколько строк, соответствующих строкам вышестоящих таблиц. Изменить значения отдельных полей, вставить новые строки, удалить отдельные строки.

В отчете отразить: схему базы данных; подробное описание полей каждой таблицы (типы, длины и т.д.); описание связей между таблицами, подробное описание процесса создания описания базы данных (схемы).

8.2 Лабораторная работа №2. Создание и использование запросов

1 Из таблицы 4 выбрать соответствующее номеру варианта задание на разработку запросов.

2 Проанализировать задание и определить базовые таблицы для каждого запроса. Разработать структуру каждого запроса. В запросы включить поля из базовых таблиц, характеризующие указанные в задании объекты, в которых необходимо произвести расчеты.

3 Разработать запросы на выборку *Запрос1* и *Запрос2*.

4 Используя созданные запросы на выборку, добавить в каждую таблицу несколько записей (в режиме таблицы).

5 Создать запрос на изменение *Запрос3* в соответствии с табл.4.

6 Выполнить запрос на изменение.

7 В режиме таблицы просмотреть одну из таблиц или запрос. Отработать следующие приемы:

- переход к требуемой записи;
- изменение (удаление) содержимого отдельного поля;

- удаление записи;
- вставка новой записи;
- изменение макета таблицы;
- поиск записей;
- фильтр по выделенному ключу;
- исключить выделенное;
- удалить фильтр;
- сортировка таблицы (запроса).

Таблица 4 – Варианты запросов

Вариант	Запрос	Содержимое запроса
1	2	3
0	Запрос1	Перечень оплаченных договоров
	Запрос2	Информация о договорах на поставку заданного изделия
	Запрос3	Удаление неоплаченных договоров, заключенных ранее заданной даты
1	Запрос1	Перечень незавершенных заказов
	Запрос2	Информация о заказах заданного товара
	Запрос3	Удаление завершенных заказов, закрытых более года назад
2	Запрос1	Список сотрудников мужского пола по каждому отделу
	Запрос2	Список сотрудников заданного отдела, работавших по заданной теме с расчетом их зарплаты
	Запрос3	Изменение часовой тарифной ставки сотрудников заданного отдела на заданный коэффициент

Продолжение табл.4

1	2	3
3	Запрос1	Перечень срочных счетов
	Запрос2	Список операций по каждому счету заданного клиента (с указанием банка)
	Запрос3	Удаление закрытых счетов
4	Запрос1	Список студентов, окончивших техникум
	Запрос2	Список студентов заданного вуза, окончивших школу (техникум) не более 5 лет назад
	Запрос3	Удаление информации по заданной группе
5	Запрос1	Перечень предметов, по которым сдается экзамен и которые читают преподаватели с ученой степенью
	Запрос2	Список студентов, у которых занятия ведет заданный преподаватель
	Запрос3	Удаление информации по заданной группе
6	Запрос1	Список сотрудников, заключивших договора страхования с частными компаниями
	Запрос2	Список сотрудников заданного предприятия, заключивших договора на сумму, превышающую заданную
	Запрос3	Удаление информации о сотрудниках старше 65 лет
7	Запрос1	Список всех сотрудников основных цехов
	Запрос2	Информация о начислении зарплаты работникам заданного цеха
	Запрос3	По заданному коду начисления увеличить на заданный коэффициент начисления работникам с признаком "АУП"
8	Запрос1	Список всех беспосадочных рейсов
	Запрос2	Список всех рейсов в заданный пункт, на которые есть билеты на заданную дату

Продолжение табл.4

1	2	3
	Запрос3	Удаление информации о наличии мест на дату меньше текущей
9	Запрос1	Перечень скорых поездов
	Запрос2	Список поездов в заданный пункт, на которые есть места на заданную дату
	Запрос3	Удаление информации о наличии мест на дату меньше текущей
10	Запрос1	По каждой включенной в план позиции выдать нормы расхода материалов на технологические нужды
	Запрос2	Перечень включенных в план изделий, для которых требуется заданный материал
	Запрос3	Удалить информацию по не включенным в план изделиям
11	Запрос1	По каждой включенной в план позиции выдать нормы загрузки основного оборудования каждого вида
	Запрос2	Перечень включенных в заданном квартале в план изделий, обрабатываемых на заданном оборудовании
	Запрос3	Удаление информации по не включенным в план изделиям
12	Запрос1	План производства деталей на экспорт
	Запрос2	Загрузка рабочих заданной профессии планом производства
	Запрос3	Удалить из плана все позиции с признаком "Проект"
13	Запрос1	Потребность в материалах на выполнение экспортных заказов
	Запрос2	Использование заданного оборудования для выполнения плана
	Запрос3	Удалить из плана все позиции с признаком "Проект"

Продолжение табл.4

1	2	3
14	Запрос1	Информация о посещении врачей больными мужчинами
	Запрос2	Список больных, посетивших заданную поликлинику
	Запрос3	У всех больных диагноз ОРЗ заменить на ГРИПП
15	Запрос1	Перечень оборудования, установленного в основных цехах
	Запрос2	Список заданного (по коду) оборудования, эксплуатирующегося более 5 лет
	Запрос3	Во всех вспомогательных цехах удалить оборудование заданной группы
16	Запрос1	Список договоров кредита, заключенных государственными банками
	Запрос2	Информация о погашении кредитов, взятых частными предприятиями у заданных банков
	Запрос3	Удаление информации о договорах кредита, срок действия которых закончился более год назад
17	Запрос1	Список сотрудников основных цехов старше 50 лет
	Запрос2	Список сотрудников мужского пола, имеющих заданную профессию
	Запрос3	Удаление информации о сотрудниках старше 65 лет
18	Запрос1	Список рейсов, выполняемых частными авиакомпаниями без промежуточных посадок
	Запрос2	Список рейсов в заданный пункт, на которые есть билеты на заданную дату
	Запрос3	Удаление информации о наличии мест на дату меньше текущей

Продолжение табл.4

1	2	3
19	Запрос1	Список договоров и позиций в них, срок поставки которых истек и материалы не получены
	Запрос2	Список договоров на поставку заданного материала поставщиками - частными предприятиями
	Запрос3	По договорам, срок поставки которых истек, удалить позиции, по которым материал получен
20	Запрос1	Список договоров, заключенных частными страховыми компаниями
	Запрос2	Список договоров заданного вида страхования, заключенных женщинами с заданной страховой компанией
	Запрос3	Удалить все договора, срок действия которых истек год назад
21	Запрос1	Потребность в материалах на выполнение экспортных заказов
	Запрос2	Список деталей, включенных в план заданного квартала и для которых необходим заданный материал
	Запрос3	Удалить из плана все позиции с признаком "Проект"
22	Запрос1	План производства деталей
	Запрос2	Загрузка заданного станка деталями, включенными в план
	Запрос3	Увеличить на заданный коэффициент Тпз для всех деталей, изготавливаемых на заданном станке
23	Запрос1	Подетальный план производства экспортных заказов
	Запрос2	Список применяемости заданной детали в изделиях плана заданного квартала
	Запрос3	Удалить из плана все позиции с признаком "Проект"

Продолжение табл.4

1	2	3
24	Запрос1	Список рабочего оборудования
	Запрос2	Список оборудования, установленного в заданном цехе
	Запрос3	Удалить все законсервированное оборудование в заданном цехе
25	Запрос1	Список работающих работников старше 50 лет
	Запрос2	Список работников мужского пола, работавших в заданном цехе
	Запрос3	Для всех работников мужского пола старше 65 лет и женского пола старше 60 поставить признак "уволен"

8 Отработать установку расширенного фильтра.

9 В режиме конструктора отработать следующие приемы:

- вставка нового поля;
- переименование поля;
- изменение типа поля;
- перемещение поля;
- удаление поля.

10 В отчете отразить: описание структуры каждого запроса, описание приемов работы с таблицами и запросами в режиме таблицы и в режиме конструктора.

8.3 Лабораторная работа №3. Создание и использование форм

1 В соответствии с вариантом выбрать из табл.5 структуру форм *Форма1* и *Форма2*. Определить базовые таблицы или запросы для форм, перечень составляющих их полей. Нарисовать эскизы этих форм.

Таблица 5 – Варианты форм

Вариант	Форма	Содержимое формы
1	2	3
0	Форма1	Информация о заказчике
	Форма2	Информация об изделии
	Форма3	Сведения о составе заказа
1	Форма1	Сведения о заказчике
	Форма2	Сведения о товаре
	Форма3	Сведения о составе заказа
2	Форма1	Сведения о сотруднике
	Форма2	Сведения о теме
	Форма3	Сведения о работе сотрудника
3	Форма1	Сведения о банке
	Форма2	Сведения о клиенте
	Форма3	Сведения об операциях по счету
4	Форма1	Сведения о вузе
	Форма2	Сведения о школе
	Форма3	Сведения о студентах группы
5	Форма1	Сведения о преподавателях
	Форма2	Сведения о студентах
	Форма3	Сведения о предметах, читаемых в каждой группе
6	Форма1	Сведения о предприятиях
	Форма2	Сведения о страховых компаниях
	Форма3	Сведения о договорах, заключенных сотрудниками

Продолжение табл.5

1	2	3
7	Форма1	Сведения о цехе
	Форма2	Справочник начислений и удержаний
	Форма3	Сведения о зарплате сотрудников
8	Форма1	Сведения о пунктах назначения
	Форма2	Сведения о компаниях
	Форма3	Сведения о наличии билетов на рейсы
9	Форма1	Сведения о железнодорожных станциях
	Форма2	Сведения о владельцах поездов
	Форма3	Сведения о наличии билетов на поезда
10	Форма1	Сведения о плане
	Форма2	Сведения о материалах
	Форма3	Потребность в материалах на план
11	Форма1	Сведения о плане
	Форма2	Сведения об оборудовании (станках)
	Форма3	Загрузка оборудования позициями плана
12	Форма1	Сведения о плане
	Форма2	Сведения о рабочих
	Форма3	Технология (сведения об операциях) обработки деталей плана
13	Форма1	Сведения о материалах
	Форма2	Сведения об оборудовании (станках)
	Форма3	Сведения об операциях позиций плана
14	Форма1	Сведения о больном
	Форма2	Сведения о поликлинике
	Форма3	Сведения о посещениях к каждому врачу
15	Форма1	Сведения о цехе/участке
	Форма2	Сведения об оборудовании (станках)

Продолжение табл.5

1	2	3
	Форма3	Сведения о наличии (установке) в цехах каждого вида оборудования
16	Форма1	Сведения о банке
	Форма2	Сведения о предприятиях
	Форма3	Сведения о погашении кредитов каждым предприятием
17	Форма1	Сведения о сотрудниках
	Форма2	Сведения о специальностях
	Форма3	Сведения о полученных специальностях каждым сотрудником
18	Форма1	Сведения об авиакомпаниях
	Форма2	Сведения о пунктах назначения
	Форма3	Сведения о наличии мест на рейсы
19	Форма1	Сведения о материалах
	Форма2	Сведения о поставщиках
	Форма3	Состав договора поставки
20	Форма1	Сведения о страховых компаниях
	Форма2	Сведения о клиентах
	Форма3	Сведения о договорах, заключенных каждым клиентом
21	Форма1	Сведения о деталях
	Форма2	Сведения о материалах
	Форма3	Нормы расхода материалов на позиции плана
22	Форма1	Сведения о деталях
	Форма2	Сведения об оборудовании (станках)
	Форма3	Использования оборудования для выполнения позиций плана

Продолжение табл.5

1	2	3
23	Форма1	Сведения об изделиях
	Форма2	Сведения о деталях
	Форма3	Состав изделий, включенных в план
24	Форма1	Сведения об оборудовании (станках)
	Форма2	Сведения об участках
	Форма3	Наличие оборудования на каждом участке
25	Форма1	Сведения о должностях
	Форма2	Сведения о подразделениях (цехах)
	Форма3	Сведения о перемещениях каждого сотрудника (работе)

2 В режиме конструктора создать эти формы.

3 С помощью созданных форм осуществить корректировку базы данных.

4 Выбрать вариант формы **Форма3** и нарисовать ее эскиз. Отдельно вынести эскиз подчиненной формы **Форма3П**.

5 Составить базовый запрос для формы **Форма3** (или **Форма3П**).

6 Создать форму **Форма3П**.

7 Создать форму **Форма3**

8 Внедрить в форму **Форма3** подчиненную форму **Форма3П**.

9 С помощью созданной формы выполнить корректировку базы данных.

10 В отчете отразить: макет (эскиз) каждой формы, включая подчиненную, описание элементов форм, описание приемов создания форм в режиме конструктора, использование форм.

8.4 Лабораторная работа № 4. Создание макросов.

Связывание форм

1 Создать два макроса *Макрос1* и *Макрос2*, открывающие соответственно формы *Форма1* и *Форма2* (см. лабораторную работу № 3).

2 В разделе заголовка формы *Форма3*, созданной в лабораторной работе № 3, создать командные кнопки для запуска макросов *Макрос1* и *Макрос2*.

3 В каждой форме создать командные кнопки для закрытия форм.

Замечание. Выполнение лабораторной работы № 4 совместить с выполнением лабораторной работой № 3.

8.5 Лабораторная работа № 5. Создание и использование отчетов

1 В соответствии с вариантом выбрать из таблицы 4 структуру отчетов *Отчет1* и *Отчет2*. Определить перечень полей, составляющих отчеты. Нарисовать эскиз каждого отчета.

2 Разработать базовые запросы *ЗапросОтчет1* и *ЗапросОтчет2*, включив в них, где требуется, расчет промежуточных данных.

3 Создать отчеты *Отчет1* и *Отчет2* с помощью *Мастера отчетов*.

4 В режиме конструктора внести изменения в структуру отчетов для улучшения их внешнего вида.

5 С помощью разработанных отчетов выполнить выборку из базы данных информации и обработать ее.

6 В отчете отразить: макет (эскиз) каждого отчета; описание элементов отчета; описание приемов создания отчетов с помощью *Мастера отчетов* и корректировки в режиме конструктора; использование отчетов.

Таблица 6 – Варианты отчетов

Вариант	Отчет	Наименование отчета	Группировка	Область данных	Условие
1	2	3	4	5	6
0	Отчет1	Портфель заказов	Заказчик, Договор	Изделие, количество, цена, сумма	все
	Отчет2	План производства	Изделие	Договор, заказчик, количество, цена, сумма	Договор оплачен
1	Отчет1	Портфель заказов	Заказчик, Заказ	Товар, количество, цена, сумма	все
	Отчет2	План поставок	Товар	Заказчик, заказ, количество, цена, сумма	Заказ не выполнен (не завершен)
2	Отчет1	Распределение работ по теме	Тема	Отдел, Сотрудник, Работа, Вркмя	все
	Отчет2	Загрузка сотрудников	Отдел, Сотрудник	Тема, Работа, Время	Незавершенные работы
3	Отчет1	Операции банка	Клиент, Счет	Операции	Заданный банк
	Отчет2	Операции	Банк,	Операция	Заданный

		клиента	Счет		клиент
--	--	---------	------	--	--------

Продолжение табл.6

1	2	3	4	5	6
4	Отчет1	Список студентов вуза	Группа	Студент, Школа	Заданный вуз
	Отчет2	Поступаемость выпускников школ	Школа	Вуз, Студент	Только школы
5	Отчет1	Список студентов преподавателя	Предмет, Группа	Студент	Заданный преподаватель
	Отчет2	Список студентов для экзамена	Преподаватель, Предмет, Группа	Студент	Сдается экзамен
6	Отчет1	Клиенты страховой компании	Предприятие	Сотрудник, Страховой договор	Заданная страховая компания
	Отчет2	Перечень застрахованных сотрудников	Страховая компания	Сотрудник, Страховой договор	Заданное предприятие, Договор действует
7	Отчет1	Расчет заработной платы	Цех, Сотрудник	Зарплата, Справочник начислений и удержаний	все
	Отчет2	Расчет удержаний по цехам	Справочник удержания	Сотрудник, Удержание	Признак: удержание

			ний, Цех		
--	--	--	----------	--	--

Продолжение табл.6

1	2	3	4	5	6
8	Отчет1	Расписание	Город	Рейс, Компания	все
	Отчет2	Наличие билетов в заданный пункт	Компания, Дата вылета	Рейс, Наличие мест	Заданный пункт, Есть места
9	Отчет1	Расписание	Станция назначения	Поезд, Станция отправления	
	Отчет2	Наличие билетов в заданный пункт	Дата отправления	Поезд, Наличие мест	Заданная станция, Есть билеты
10	Отчет1		Изделие	Материал, норма расхода, общая потребность	Изделие включено в план
	Отчет2	Потребность в материалах на план	Материал	Изделие, Количество на план, норма расхода, общая потребность	Изделие включено в план
11	Отчет1		Изделие	Оборудование, норма времени, общая загрузка	Изделие включено в план
	Отчет2	Загрузка оборудования планом	Оборудование	Изделие, Количество на план, норма расхода, общая загрузка	Изделие включено в план

Продолжение табл.6

1	2	3	4	5	6
12	Отчет1	План производства	Деталь	Наименование, Количество	Изделие включено в план
	Отчет2	Загрузка рабочих планом	Рабочий	Деталь, Операция, норма времени	Изделие включено в план
13	Отчет1	Загрузка оборудования планом	Оборудование	Изделие, Количество на план, норма времени, общая загрузка	Изделие включено в план
	Отчет2	Потребность в материалах на план	Материал	Изделие, Количество на план, норма расхода, общая потребность	Изделие включено в план
14	Отчет1	Посещаемость поликлиники больными	Больной	Врач, Посещение	Заданная поликлиника
	Отчет2	Загрузка основных врачей	Поликлиника, Врач	Больной, посещение	Врач основной
15	Отчет1	Перечень оборудования завода	Группа оборудования	Цех, Станок	все

Продолжение табл.6

1	2	3	4	5	6
	Отчет2	Перечень оборудования по основным цехам	Цех	Оборудование, Станок	Основной цех
16	Отчет1	Список клиентов гос. банков	Банк	Предприятие, Номер договора	Гос.банк
	Отчет2	Погашение кредитов гос предприятия-ми	Предприятие	Банк, Договор, Погашение	Гос. Предприятие
17	Отчет1	Повышение квалификации сотрудниками	Отдел	Сотрудник, повышение квалификации	все
	Отчет2	Список сотрудников по специальностям	Специальность	Сотрудник, повышение квалификации	
18	Отчет1	Расписание	Авиакомпания	Пункт назначения, рейс	все
	Отчет2	Наличие билетов в заданный пункт	Рейс	Дата, наличие билетов	Заданный пункт
19	Отчет1	План поставок материалов	Материал	Поставщик, Договор, количество, Цена	Признак: не получено
	Отчет2	План поставок по поставщикам	Поставщик	Материал, Договор, Количество, Цена	Признак: не получено

Продолжение табл.6

1	2	3	4	5	6
20	Отчет1		Вид страхования	Клиент, Договор, Сумма	Договор не окончен
	Отчет2		Клиент, Компания	Договор, Вид страх, Сумма	Гос. компании
21	Отчет1	План производства	Деталь	Количество, Материал, потребность	Деталь включена в план
	Отчет2	Потребность в материалах на план	Материал	Деталь, потребность на деталь	Деталь включена в план
22	Отчет1	План производства	Деталь	Количество, Станок ,Загрузка станка	Деталь включена в план
	Отчет2	Загрузка оборудования по плану	Станок	Деталь, Загрузка станка	Деталь включена в план
23	Отчет1	План производства деталей	Изделие	Деталь, Потребность на изделие и план	Деталь включена в план
	Отчет2	Применяемость деталей	Деталь	Изделие, Потребность деталей на изделие и план	Деталь включена в план
24	Отчет1	Обеспечение оборудованием цехов	Цех	Оборудование, Станок	Оборудование рабочее
	Отчет2	Установленное оборудование	Оборудование	Цех, Станок	все

Продолжение табл.6

1	2	3	4	5	6
25	Отчет1	Движение со- трудников	Сотруд- ник	Цех, Долж- ность, Работа	все
	Отчет2	Штатное рас- писание	Цех	Должность, со- трудник	Признак: не уволен

9 ПРИМЕР ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

9.1 Указания к лабораторной работе №1

1 На основании анализа схемы базы данных разработать структуру каждой таблицы (табл. 7, 8, 9, 10).

Таблица 7

				Объект	Таблица
				Наименование	Заказчик
Поле	Тип	Размер	Примечание	Наименование поля	
КодЗак	числовой	целое	ключ	Код заказчика	
НаимЗак	текстовый	40		Наименование заказчика	
Город	текстовый	15		Адрес	
Улица	текстовый	15			
Дом	числовой	целое			
Телефон	текстовый	10		Телефон	

Таблица 8

				Объект	Таблица
				Наименование	Изделие
Поле	Тип	Размер	Примечание	Наименование поля	
КодИзд	числовой	целое	ключ	Код изделия	
НаимИзд	текстовый	20		Наименование изделия	
Вес	числовой	с плав.точкой		Вес изделия	
Длина	числовой	с плав.точкой		Максимальный размер	

Таблица 9

			Объект	Таблица
			Наименование	Договор
Поле	Тип	Размер	Примечание	Наименование поля
НомДогов	числовой	целое	ключ	Номер договора
КодЗак	числовой	целое	подстановка	Код заказчика
Дата	дата	краткий.дата		Дата заключения договора
ПризнОплаты	логичес.		флажок	Признак оплаты

Таблица 10

			Объект	Таблица
			Наименование	СоставДогов
Поле	Тип	Размер	Примечание	Наименование поля
НомДогов	числовой	целое		Номер договора
КодИзд	числовой	целое	подстановка	Код изделия
Количество	числовой	целое		Количество изделий
Цена	числовой	плав. точка		Цена одного изделия

2 Для создания пустой базы данных войти в начальное окно диалога и выбрать переключатель **Новая база данных**.

3 Для работы с таблицами в окне базы данных активизировать вкладку **Таблицы** (при первом входе она активна).

4 Для создания таблицы *Заказчик*:

4.1 Щелкнуть на кнопке **Создать**. Появится диалоговое окно **Новая таблица**.

4.2 Выбрать **Конструктор**. Появится окно таблицы в режиме конструктора.

4.3 Последовательно определить все поля таблицы. Для каждого поля в верхней части окна отводится одна строка, содержащая: **имя поля** (ввести с клавиатуры), **тип данных** (выбрать из раскрывающегося списка) и **описание** (ввести с клавиатуры). После выбора типа данных в левой нижней части окна можно задать значения свойств поля, набор которых зависит от выбранного типа данных, причем некоторые из значений устанавливаются по умолчанию. Следует установить свойство Размер поля (вкладка **Общие**):

для полей числового типа (**КодЗак** и **Дом**) – целое, а для полей текстового типа (все остальные) – размер поля в символах. Описание – это пояснительный текст, который выводится в строке состояния при выборе этого поля, т.е. это своего рода мини-справка при работе с данными.

4.4 Поле **КодЗак** определить как ключевое: выбрать его щелчком в области выделения, затем щелкнуть кнопку **Ключ** на панели инструментов или

Правка⇒ключевое поле

Заполненное окно таблицы в режиме конструктора приведено на рис.19.

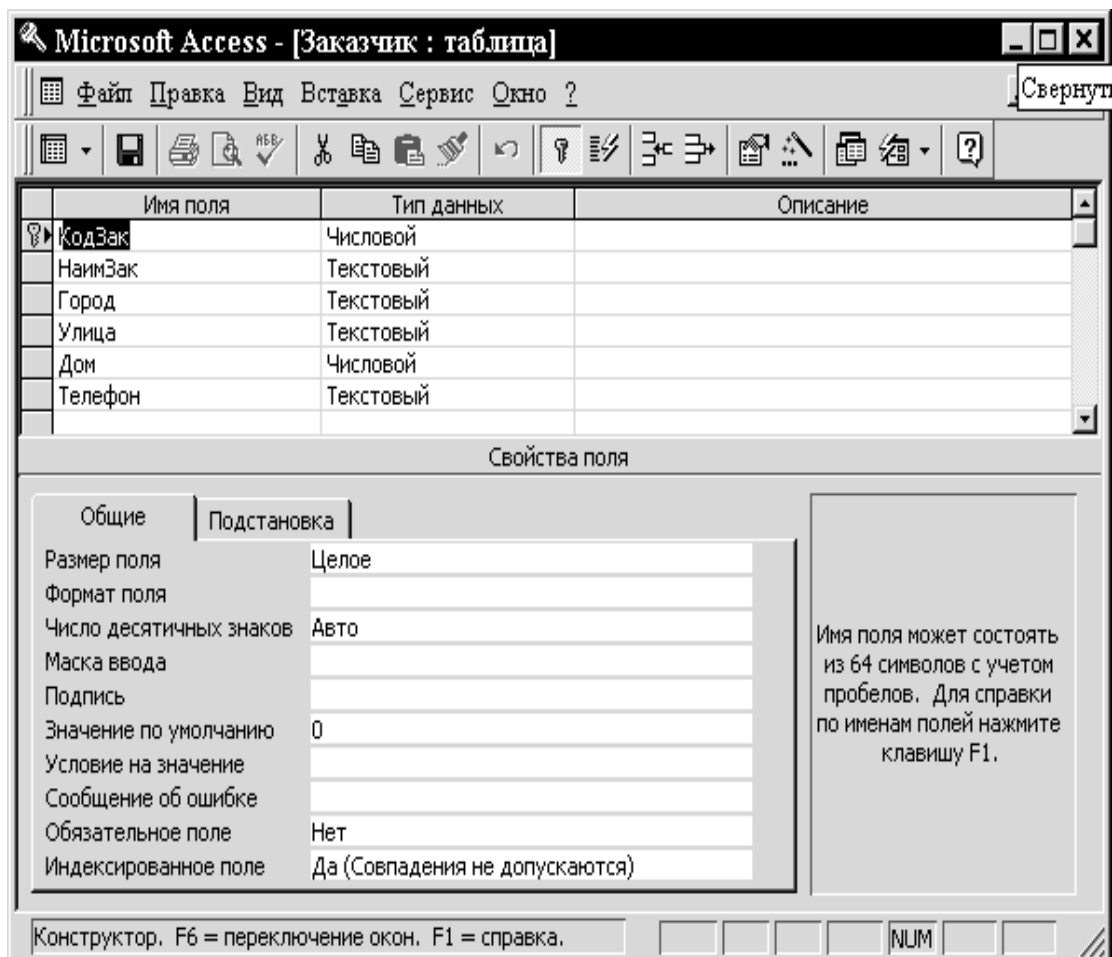


Рисунок 19 – Окно таблицы в режиме конструктора

4.5 Закрывать окно таблицы и сохранить описание таблицы, присвоив ей имя *Заказчик*.

5 Аналогично создать описания остальных таблиц.

6 В таблице *Договор* для поля **КодЗак** установить свойство подстановки. Для этого при определении типа данных выбрать **Мастер подстановок**. Появится ряд сменяющих друг друга диалоговых окон **Мастер подстановок**, в которых выбрать таблицу *Заказчик*, содержащую подстановочные значения, и включить в список подстановки поля **КодЗак**, **НаимЗак** и **Город** из этой таблицы.

7 Для поля **ПризнОплаты** таблицы *Договор* во вкладке **Подстановка** свойства поля выбрать для типа элемента управления значение **флажок** (устанавливается по умолчанию).

8 В таблице *СоставДогов* для поля **КодИзд** установить свойство подстановки. В список подстановки включить поля **КодИзд**, **НаимИзд** из таблицы *Изделие*.

9 Для определения связей между таблицами:

9.1 Щелкнуть кнопку **Схема данных** на панели инструментов или выбрать команду

Сервис⇒Схема данных.

9.2 Если не задано ни одного свойства подстановки (т.е. не выполнены пункты 6 и 8), появится окно диалога **Добавление таблицы** (при корректировке связей это окно вызывается через контекстное меню). Выбрать все таблицы, щелкнуть кнопку **Добавить** и закрыть окно. Появится окно **Схема базы данных**. В противном случае (если свойства подстановки заданы) **ACCESS** автоматически установит связи между таблицами, связанными свойством подстановки и сразу откроется диалоговое окно **Схема данных**. В нашем случае в нем будут указаны связи между таблицами *Заказчик* – *Договор* и *Изделие* – *СоставДогов*. Однако эти связи следует уточнить (скорректировать).

9.3 Для корректировки связи *Заказчик – Договор* щелкнуть правой кнопкой на линии этой связи и в появившемся контекстном меню выбрать пункт **Изменение связей**. Появится диалоговое окно **Связи**. Установить флажок **Обеспечение целостности данных**. Это не позволит ввести в таблицу *Договор* несуществующий в таблице *Заказчик* код заказчика или удалить из таблицы *Заказчик* запись о заказчике, на которую имеется ссылка с записей таблицы *Договор*. Для автоматического обновления поля **КодЗак** в таблице *Договор* при его изменении в таблице *Заказчик* установить флажок **Каскадное обновление полей**.

9.4 Аналогично скорректировать связь между таблицами *Изделие* и *СоставДогов*.

9.5 Для установления новой связи – между таблицами *Договор* и *СоставДогов* (по полю **НомДогов**) перетащить поле **НомДогов** из таблицы *Договор* на поле **НомДогов** в таблице *СоставДогов*. Установить флажок **Обеспечение целостности данных**, а затем **Каскадное удаление**. Это даст возможность при аннулировании заказа (т.е. удалении записи из таблицы *Договор*) удалять все относящиеся к этому заказу строки в *СоставДог* (т.е. дочерние строки).

9.6 Для более удобного представления таблицы можно перемещать в пределах окна Схема данных, изменять их размеры.

В результате определения связей будет получена схема, представленная на рис.20.

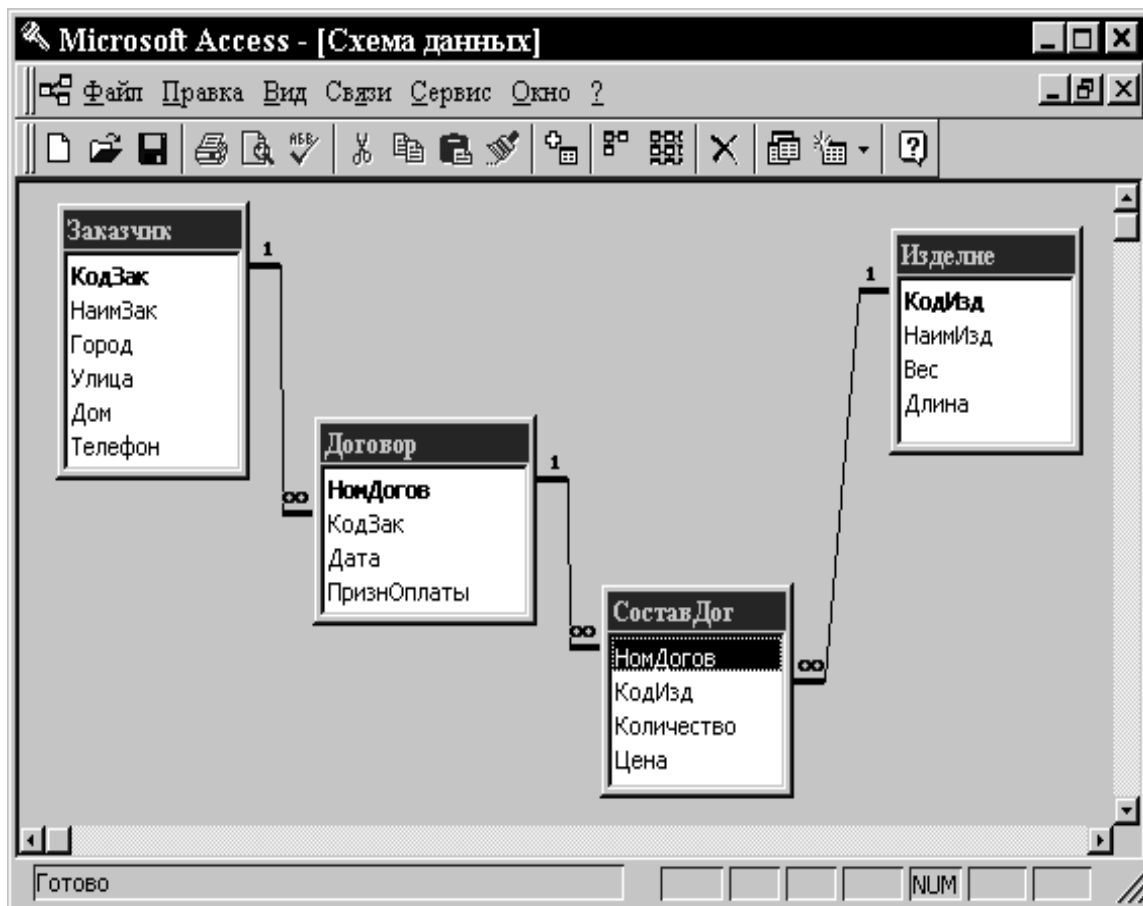


Рисунок 20 – Схема базы данных

9.7 Закрывать окно схемы базы данных и сохранить описание схемы.

10 Для входа в режим таблицы в окне базы данных (вкладка **Таблица**) щелкнуть на командной кнопке **Открыть**.

9.2 Указания к лабораторной работе №2

1 На основании анализа задания определить структуру запросов (табл. 11, 12, 13).

Таблица 11

		Объект	Запрос на выборку
		Наименование	Запрос1
Поле	Таблица/Запрос	Вывод на экран	Примечание
НомДогов	Договор		
Дата	Договор		
НаимЗак	Заказчик		
Город	Заказчик		
Телефон	Заказчик		
ПризнОплаты	Договор	нет	Выбор при "ДА"

Таблица 12

		Объект	Запрос на удаление
		Наименование	Запрос3
Поле	Таблица/Запрос	Вывод на экран	Примечание
НомДогов	Договор		
Дата	Договор		Выбор при < параметра
ПризнОплаты	Договор	нет	Выбор при равенстве "Нет"

Таблица 13

		Объект	Запрос на выборку
		Наименование	Запрос2
Поле	Таблица/Запрос	Вывод на экран	Примечание
НомДогов	Договор		
НаимЗак	Заказчик		Сортировка по возрастанию
Город	Заказчик		
Количество	СоставДогов		
Цена	СоставДогов		
Сумма	вычисляемое		Цена*Количество
КодИзд	СоставДогов	нет	Выбор при равенстве параметру

2 Для работы с запросами в окне базы данных активизировать вкладку **Запросы**.

3 Для создания запроса *Запрос2*:

3.1 Щелкнуть на кнопке **Создать**. Появится диалоговое окно **Новый запрос**.

3.2 Выбрать **Конструктор**. Откроется окно нового запроса в режиме конструктора и сразу появится окно **Добавление таблицы**.

3.3 Выбрать таблицы **Договор, Заказчик и СоставДогов**, которые будут использованы для построения запроса. Закрывать окно.

3.4 В строку **Поле** бланка запроса (нижнюю часть окна запроса) перетаскать из соответствующих таблиц верхней части окна поля **НомДогов, НаимЗак, Город, Колич, Цена**.

3.5 Для поля **НаимЗак** задать порядок сортировки **По возрастанию**.

3.6 Создать вычисляемое поле **Сумма**. Для чего в следующую пустую ячейку строки **Поле** ввести выражение **Количество*Цена**. Можно воспользоваться построителем выражений, вызвав его с помощью кнопки **Построить** на панели инструментов. **ACCESS** присвоит этому полю имя **Выражение1**. Заменить его на более содержательное имя **Сумма**. Для чего через контекстное меню, пункт **Свойства** открыть диалоговое окно **Свойства поля** и свойству **Подпись** определить значение **Сумма**.

3.7 Перетаскать в бланк запроса поле **НаимИзд** из таблицы **Изделие**. Поскольку это поле предназначено только для отбора записей, в строке **Вывод на экран** убрать флажок вывода поля на экран. В строку **Условие отбора** ввести фразу: **[Введите код изделия]**. Этим мы определили параметр для задания условия отбора при реализации запроса. Теперь каждый раз при выполнении запроса **ACCESS** будет запрашивать код изделия и отбирать только те записи, в которых значение поля **КодИзд** совпадает с введенным. Заполненное окно запроса приведено на рис.21.

3.8 Запрос создан. Закрывать окно запроса. Будет выдан запрос на сохранение и запрошено имя запроса.

4 Аналогично создать *Запрос1*. Он немного проще.

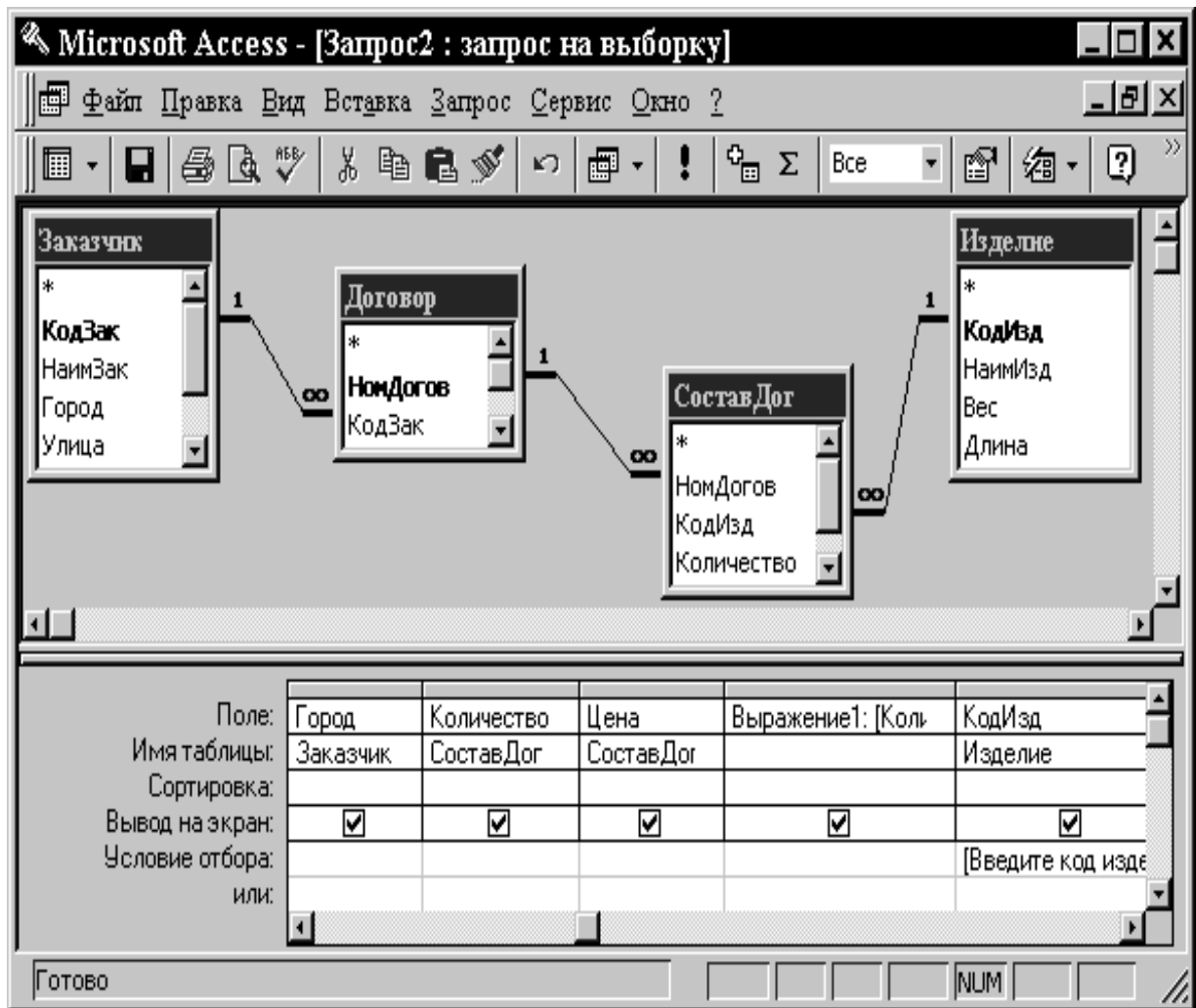


Рисунок 21 – Окно запроса на выборку в режиме конструктора

5 Запрос *Запрос3* – это запрос на изменение (удаление). Для его создания необходимо сначала построить запрос на выборку данных, подлежащих удалению. Это осуществляется аналогично созданию предыдущих запросов. Затем полученный запрос следует преобразовать к требуемому типу: щелкнуть кнопку **Тип запроса** на панели инструментов и в раскрывающемся списке выбрать тип **Удаление**. Заполненное окно запроса на удаление приведено на рис.22.

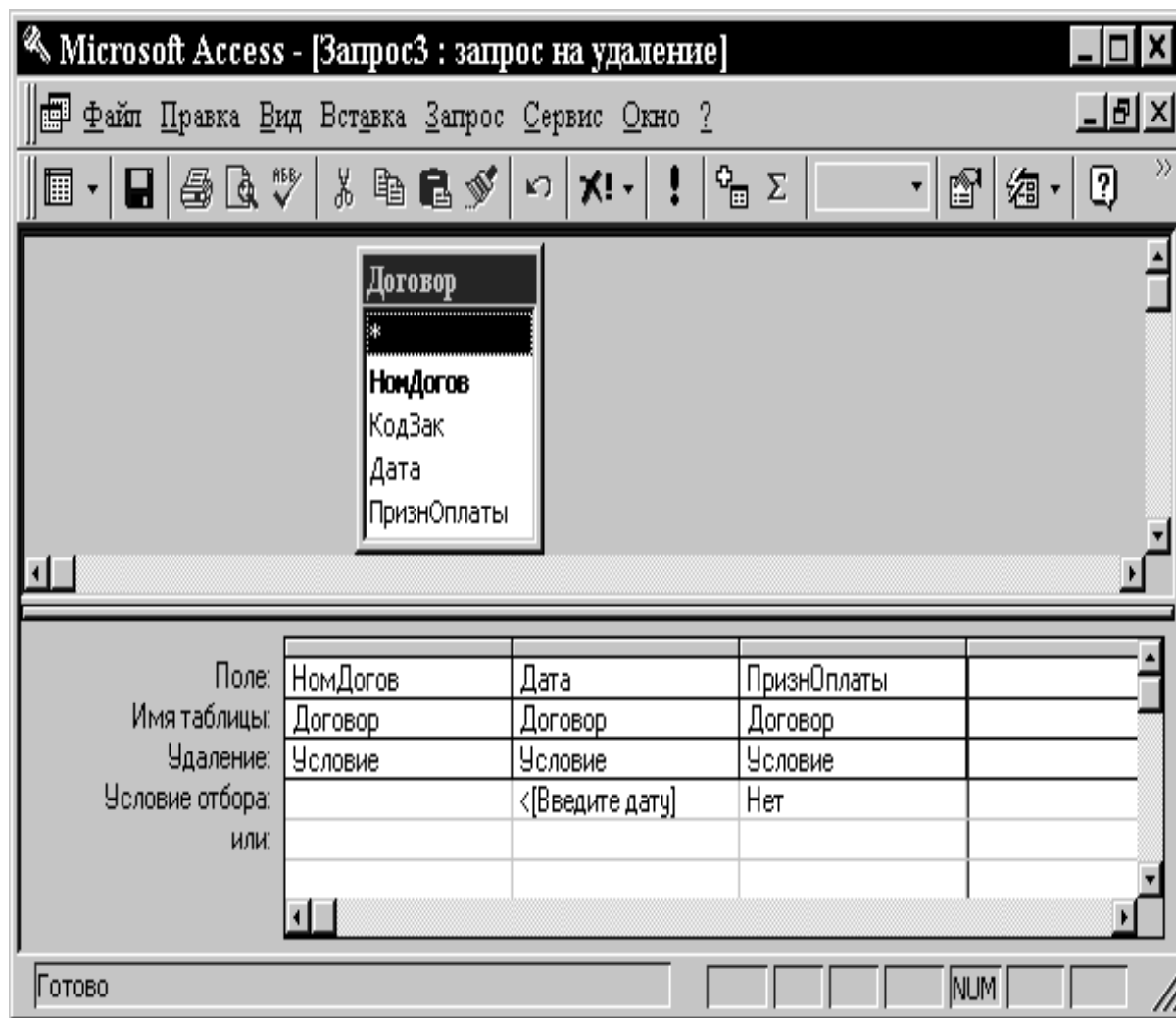


Рисунок 22 – Окно запроса на удаление в режиме конструктора

6 На рис.23 приведено заполненное окно запроса на обновление. Этот запрос осуществляет увеличение на заданный коэффициент цены заданного изделия во всех договорах.

7 Для входа в режим таблицы необходимо в окне базы данных активизировать соответствующую вкладку (**Таблицы** или **Запросы**), выбрать соответствующую таблицу или запрос и щелкнуть на командной кнопке **Открыть**. Оперативное переключение между режимами осуществляется кнопкой представления на панели инструментов (**Вид**).

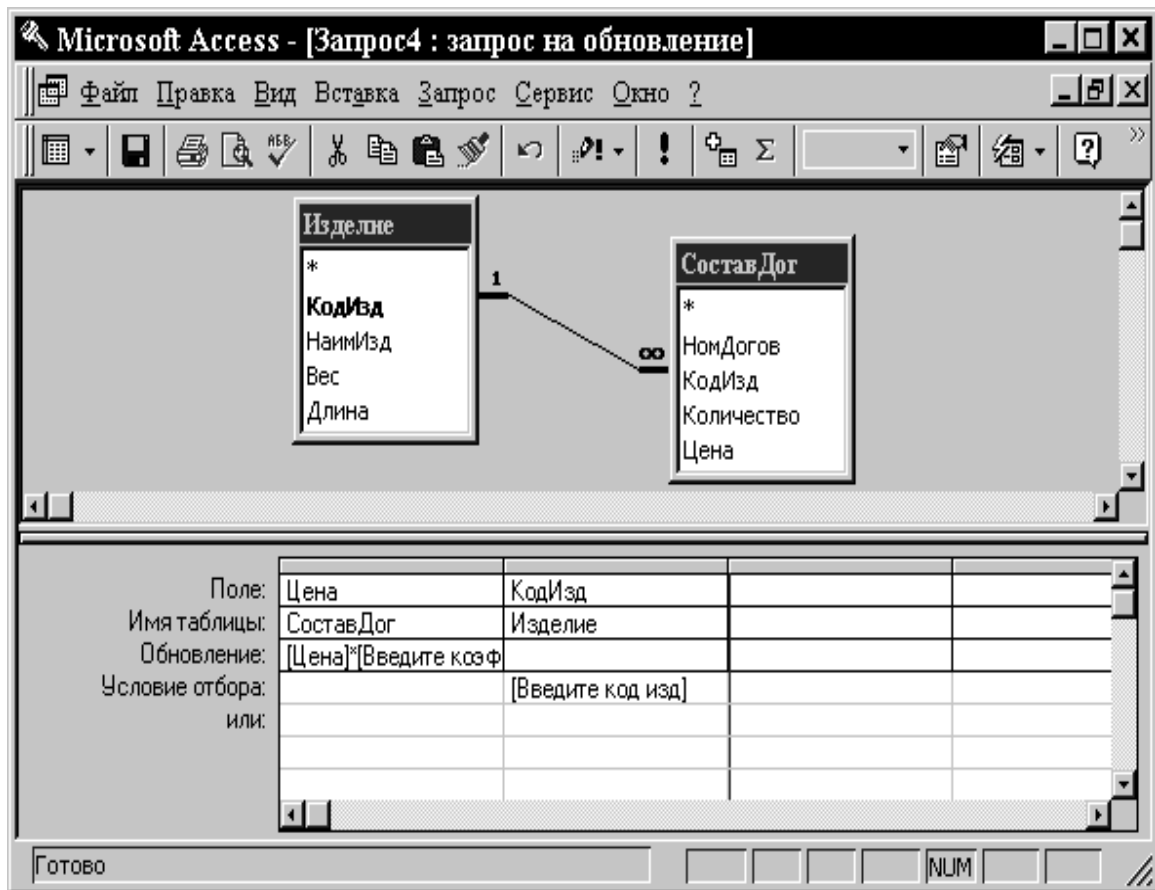


Рисунок 23 – Окно запроса на обновление в режиме конструктора

8 В результате работы с базой данных в режиме таблицы база данных содержит следующие данные (рис. 24, 25, 26, 27).

Microsoft Access - [СоставДог : таблица]

Файл Правка Вид Вставка Формат Записи Сервис Окно

?

НомДогов	КодИзд	Количество	Цена
51	1001	1	200
51	1005	2	40
51	1004	2	28
51	1004	1	32
51	1002	5	300
51	1003	9	150
52	1001	12	250
52	1004	14	25
53	1004	20	28
53	1005	20	42
54	1005	8	38
55	1002	15	280
55	1003	15	120
55	1005	15	40
55	1002	20	250
56	1004	10	30
*	0	0	0

Запись: 16 из 16

Режим таблицы

Рисунок 24 – Результаты работы с базой данных в режиме таблицы

Microsoft Access - [Заказчик : таблица]

Файл Правка Вид Вставка Формат Записи Сервис Окно ?

?

КодЗак	НаимЗак	Город	Улица	Дом	Телефон
101	Полет	Харьков	Клочковская	555	256-13-51
102	Аэлита	Харьков	Сумская	21	248-23-12
103	Мария	Краматорск	Шкадинова	111	6-21-54
104	Дружбан	Дружковка	Ленина	54	2-45-56
105	Столичный	Киев	Крещатик	12	755-98-21
*	0			0	

Запись: 3 из 5

Режим таблицы

Рисунок 25 – Результаты работы с базой данных в режиме таблицы

Microsoft Access - [Изделие : таблица]

Файл Правка Вид Вставка Формат Записи Сервис Окно ?

	КодИзд	НаимИзд	Вес	Длина
	1001	Швейная машинка	45,5	80
	1002	Оверлок бытовой	50,5	40
	1003	Стиральная машинка	22,5	95
▶	1004	Миксер	0,45	10
	1005	Электробритва	0,25	5
*	0		0	0

Запись: 4 из 5

Режим таблицы

Рисунок 26 – Результаты работы с базой данных в режиме таблицы

Microsoft Access - [Договор : таблица]

Файл Правка Вид Вставка Формат Записи Сервис Окно ?

	НомДогов	КодЗак	Дата	ПризнПлаты
▶	51	102	16.02.98	<input checked="" type="checkbox"/>
	52	102	12.12.95	<input type="checkbox"/>
	53	104	15.03.98	<input type="checkbox"/>
	54	103	12.09.98	<input checked="" type="checkbox"/>
	55	105	02.12.98	<input checked="" type="checkbox"/>
	56	102	05.06.99	<input checked="" type="checkbox"/>
*	0	0		<input type="checkbox"/>

Запись: 1 из 6

Режим таблицы

Рисунок 27 – Результаты работы с базой данных в режиме таблицы

9.3 Указания к лабораторной работе №3

На основании анализа задания разработать макеты (эскизы) форм (рис. 28, 29, 30).

The screenshot shows a Microsoft Access form titled "Информация о заказчике". The form has a menu bar with "Файл", "Правка", "Вид", "Вставка", "Формат", "Записи", "Сервис", and "Окно". Below the menu is a toolbar with various icons. The form content includes:

- Field "КодЗак:" with a text box containing "101".
- Field "НаимЗак:" with a text box containing "Полет".
- Field "Адрес:" with three text boxes containing "Харьков", "Клочковская", and "555".
- Field "Телефон:" with a text box containing "256-13-51".
- A text label: "При вводе адрес набирать в таком виде:".
- A text box containing: "город.", "улица.", "дом".
- Navigation buttons at the bottom: "Запись:" followed by navigation icons and "1 из 5".
- Form status bar: "Режим формы" and "NUM".

Рисунок 28 – Макет (эскиз) формы

The screenshot shows a Microsoft Access form titled "Формирование состава заказа". The form has a menu bar with "Файл", "Правка", "Вид", "Вставка", "Формат", "Записи", "Сервис", and "Окно". Below the menu is a toolbar with various icons. The form content includes:

- Buttons "Заказчик" and "Изделие".
- Field "Договор N" with a text box containing "51", "от" with a text box containing "16.02.98", and "Призн.опл." with a checked checkbox.
- Field "Код заказчика" with a text box containing "102".
- Section "Состав заказа по этому договору:".
- Table with columns "КодИзд", "Количество", and "Цена":

КодИзд	Количество	Цена
1001	1	200
1005	2	40
1004	2	28
1004	1	32
1002	5	300
- Field "Заказчик" with a text box containing "Аэлига".
- Field "Адрес:" with three text boxes containing "Харьков", "Сумская", and "21".
- Field "Телефон" with a text box containing "248-23-12".
- Navigation buttons at the bottom: "Запись:" followed by navigation icons and "1 из 6".
- Form status bar: "Режим формы" and "NUM".

Рисунок 29 – Макет (эскиз) формы

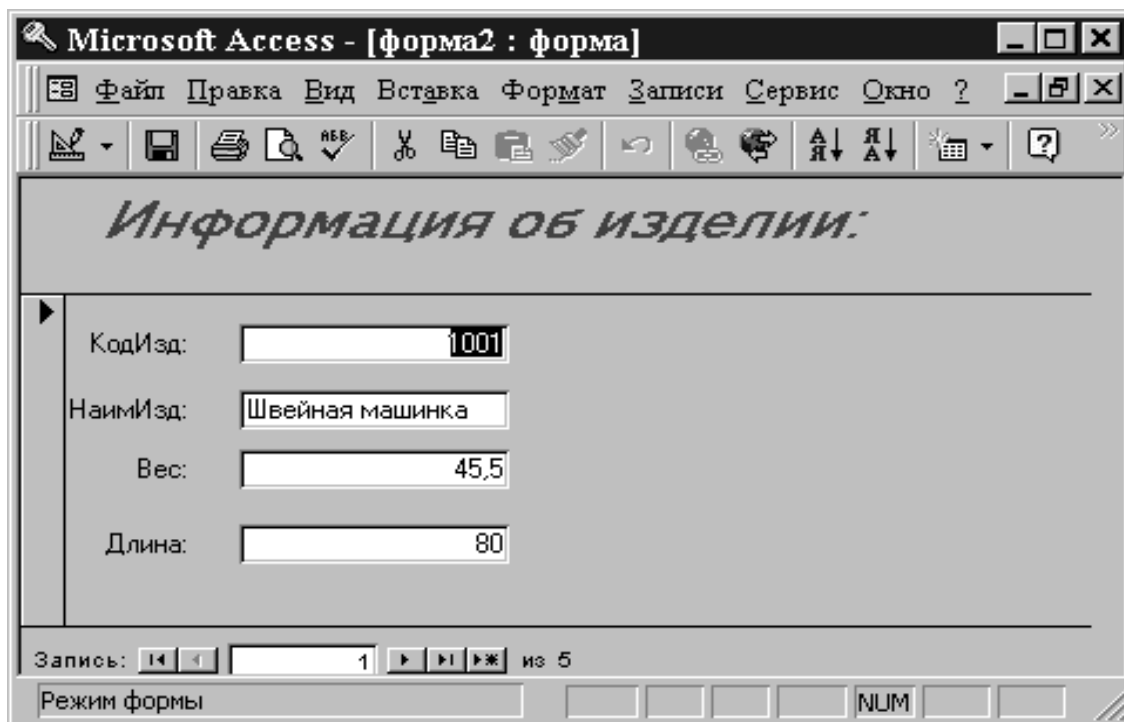


Рисунок 30 – Макет (эскиз) формы

Для работы с формами в окне базы данных активизировать вкладку **Формы**.

2 Для создания формы *Форма1*:

2.1 Щелкнуть на кнопке **Создать**. Откроется диалоговое **окно Новая форма**.

2.2 В раскрывающемся списке нижней части окна выбрать в качестве базовой таблицы (т.е. таблицы, используемой для создания формы) таблицу *Заказчик*.

2.3 Выбрать **Конструктор**. Откроется окно формы в режиме конструктора.

2.4 Для добавления заголовка и раздела примечаний выполнить команду

Формат⇒Заголовок/примечания формы.

2.5 Из списка полей в заголовок перетащить поле **КодЗак**. Остальные поля перетащить в область данных (при создании элемента управления типа

Поле нет необходимости в предварительном выборе инструмента на панели инструментов: он используется по умолчанию).

2.6 При создании поля **ACCESS** присоединяет к нему подпись, в которой отображается значение свойства **Подпись** (или имя поля, если это свойство не задано). Ее можно удалить, сначала выделив, а затем нажав клавишу **Del**. Удалить присоединенные подписи у полей **Город, Улица, Дом**.

2.7 Для создания свободных текстов, не связанных с другими элементами управления, используется кнопка **Надпись** на панели инструментов. Создать свободные тексты в заголовке, разделе примечаний, а также подпись **“Адрес”** в области данных.

2.8 Установить требуемые размеры элементов управления, используя для этого маркеры размера, находящиеся на сторонах, в нижних углах и в правом верхнем углу предварительно выделенного элемента.

2.9 Перемещая элементы управления, расставить их в требуемом порядке. Для перемещения невыделенного элемента его необходимо просто перетащить. Выделенный элемент перемещается перетаскиванием любой его стороны между маркерами (указатель принимает вид раскрытой ладони).

2.10 Выровнять элементы управления по вертикали: присоединенные подписи по правому краю, а поля – по левому. Для этого сначала необходимо выделить все выравниваемые элементы (столбец), щелкнув на первом из них и при нажатой *Shift* – на остальных. Щелчок в некоторой точке горизонтальной линейки выделяет все элементы управления, которые пересекает вертикальная линия, опущенная из этой точки. Аналогично – при щелчке на вертикальной линейке. После выделения выбранных элементов выбрать команду

Формат⇒Выровнять⇒По правому краю

или

Формат⇒Выровнять⇒По левому краю.

2.11 Аналогично выровнять по горизонтали элементы управления, описывающие адрес заказчика.

2.12 Для улучшения внешнего вида формы использовать цветное оформление, различные шрифты и т.д.

3 Форма *Форма2* создается аналогично.

4 Форма *Форма3* является более сложной и содержит подчиненную форму.

4.1 На базе таблицы *СоставДогов* разработать подчиненную форму *Форма3П*. Так как после внедрения в основную форму подчиненная форма будет использоваться в режиме таблицы, свойство **Режим по умолчанию** установить в значение **Таблица**.

4.2 Для разработки основной формы предварительно создать базовый запрос на выборку *ЗапросФорма3* следующей структуры (табл.14).

Таблица 14

		Объект
		Запрос на выборку
		ЗапросФорма3
Поле	Таблица/запрос	Примечание
НаимДогов	Договор	Сортировка по возрастанию
Дата	Договор	
ПризнОплаты	Договор	
КодЗак	Договор	
НаимЗак	Заказчик	
Город	Заказчик	
Улица	Заказчик	
Дом	Заказчик	
Телефон	Заказчик	

Порядок разработки запроса описан в указаниях к лабораторной работе № 2.

4.3 Разработать основную форму, зарезервировав в ней место для подчиненной формы. В правой части области данных расположены справочные данные о текущем заказчике. Необходимо запретить возможность

обновления этих полей. Для этого для каждого из них установить для свойства **Блокировка** значение “Да”.

4.4 Внедрить подчиненную форму в основную. Для этого, не закрывая окна формы, активизировать окно базы данных и перейти в нем во вкладку **Формы**. Выбрать в этом окне подчиненную форму и перетащить ее в требуемое место основной формы в окне формы.

9.4 Указания к лабораторной работе № 4

1 На основании анализа задания разработать структуру каждого отчета.

2 Для создания отчетов предварительно создать два базовых запроса на выборку *ЗапросОтчет1* и *ЗапросОтчет2* (табл. 15, 16), включив в них все необходимые для отчетов *Отчет1* и *Отчет2* соответственно поля:

Таблица 15

		Объект	Запрос
		Наименование	ЗапросОтчет1
Поле	Таблица/ запрос	Вывод на экран	Примечание
НаимЗак	Заказчик		
Город	Заказчик		
НомДогов	Договор		
НаимИзд	Изделие		
Количество	СоставДогов		
Цена	СоставДогов		
Сумма	Вычисляемое		Цена*количество
ПризнОплаты	Договор		

Таблица 16

		Объект	Запрос
		Наименование	ЗапросОтчет2
Поле	Таблица/ запрос	Вывод на экран	Примечание
НаимЗак	Заказчик		
Город	Заказчик		
НомДогов	Договор		
НаимИзд	Изделие		
Количество	СоставДогов		
Цена	СоставДогов		
Сумма	Вычисляемое		Цена*количество
ПризнОплаты	Договор	нет	Выбор при "Да"

Порядок разработки запросов аналогичен описанному в указаниях к лабораторной работе № 2. Поскольку оба запроса очень похожи, можно после создания первого запроса выделить его, щелкнув на его имени в окне базы данных, и скопировать в буфер обмена командой

Правка⇒Копировать.

А затем создать копию этого запроса (с новым именем) командой:

Правка⇒Вставить.

И после небольшой корректировки (в режиме конструктора) получить новый запрос.

3 Для работы с отчетами в окне базы данных активизировать вкладку **Отчеты.**

4 Для создания отчета *Отчет1* (с помощью **Мастера отчетов**):

4.1 Щелкнуть на кнопке **Создать**. Откроется диалоговое окно **Новый отчет.**

4.2 В раскрывающемся списке нижней части окна выбрать в качестве базового запроса (т.е. запроса, используемого для создания отчета) запрос *ЗапросОтчет1.*

4.3 Выбрать **Мастер отчетов** и щелкнуть кнопку **ОК**. Откроется первое диалоговое окно мастера отчетов **Создание отчетов.**

4.4 Из списка **Доступные поля** в список **Выбранные поля** перенести все поля. Щелкнуть кнопку **Далее**.

4.5 Выбрать тип представления данных **По Договор**. Щелкнуть кнопку **Далее**.

4.6 Следующее окно пропустить, щелкнув кнопку **Далее**

4.7 В следующем окне выбрать сортировку по возрастанию поля **НаимИзд**. Щелкнуть кнопку **Итоги...** и в раскрывшемся окне **Итоги** для полей **Количество** и **Выражение1** в графе **Sum** установить флажок. Это означает подведение итогов по этим полям. Закрыть окно **Итоги** и щелкнуть кнопку **Далее**.

4.8 В следующем окне выбрать вид макета для отчета **По левому краю 1**. Щелкнуть кнопку **Далее**.

4.9 Выбрать стиль **Строгий**. Щелкнуть кнопку **Далее**

4.10 Задать имя отчета **Портфель заказов предприятия**. Щелкнуть кнопку **Готово**. Формирование отчета завершено. На экране появится готовый отчет в режиме просмотра.

5 Для улучшения его внешнего вида перейти в режим конструктора. Для этого щелкнуть кнопку **Вид** на панели инструментов. Затем:

5.1 Надпись в заголовке отчета выровнять по центру. Для этого переместить до конца строки ее правый край, а затем щелкнуть кнопку **По центру** на панели инструментов.

5.2 В заголовке группы **НомДогов** для поля **НомДогов** установить выравнивание по левому краю, для чего щелкнуть на этом поле, затем на кнопке **По левому краю** на панели инструментов.

5.3 В заголовке группы **НомДогов** присоединенную надпись, передвинуть на правую сторону от элемента управления. Для этого выделить ее и перетащить маркер в верхнем левом углу. Выровнять элементы управления по вертикали и горизонтали. Это делается так же, как и выравнивание в форме.

5.4 В разделе **примечания для группы НомДогов** убрать надписи **Sum**, а вычисляемые поля с суммой перенести выше, установив их напротив вычисляемого текстового поля **=Итоги для ...**. Выровнять элементы управления по вертикали и горизонтали.

5.5 Закрывать отчет. В окне базы данных переименовать отчет.

Отчет **Отчет2** создается аналогично. В окне выбора типа представления данных выбрать **По изделие**. Макет для отчета выбрать **Ступенчатый**. В области данных добавить вычисляемое поле Средняя цена изделия, вычисляемое по формуле

$$=\text{Sum}(\text{Выражение1}) * \text{Sum}(\text{Количество}).$$

6 Для создания отчета **Отчет2** в режиме конструктора:

6.1 Щелкнуть на кнопке **Создать**. Откроется диалоговое окно **Новый отчет**.

6.2 В раскрывающемся списке нижней части окна выбрать в качестве базового запроса (т.е. запроса, используемого для создания отчета) запрос **ЗапросO2**.

6.3 Выбрать **Конструктор** и щелкнуть кнопку **ОК**. Откроется диалоговое окно отчетов в режиме конструктора.

6.4 Для добавления заголовка и раздела примечаний выполнить команду

Вид⇒Заголовок/примечания отчета.

6.5 В заголовке формы создать надпись.

6.6 Для создания группы выполнить команду

Вид⇒Сортировка и группировка.

В открывшемся окне в раскрывающемся списке выбрать поле **НаимЗак** и задать сортировку по возрастанию. Свойствам **Заголовок группы** и **Примечание группы** выбрать значения **Да** и закрыть окно.

6.7 В область заголовка группы из списка полей перетащить поля **НаимЗак** и **Город**. Присоединенные записи удалить.

6.8 В верхнем колонтитуле создать надписи

6.9 В область данных из списка полей перетащить поля **НомДогов**, **НаимИзд**, **Количество**, **Цена**, **Выражение1**. Присоединенные записи удалить.

6.10 Из списка полей в заголовок перетащить поле **КодЗак**. Остальные поля перетащить в область данных (при создании элемента управления типа **Поле** нет необходимости в предварительном выборе инструмента на панели инструментов: он используется по умолчанию).

6.11 При создании поля **Access** присоединяет к нему подпись, в которой отображается значение свойства **Подпись** (или имя поля, если это свойство не задано). Ее можно удалить, сначала выделив, а затем нажав клавишу **Del**. Удалить присоединенные подписи у полей **Город**, **Улица**, **Дом**.

6.12 Для создания свободных текстов, не связанных с другими элементами управления, используется кнопка **Надпись** на панели инструментов. Создать свободные тексты в заголовке, разделе примечаний, а также подпись **“Адрес”** в области данных.

6.13 Для добавления заголовка и раздела примечаний выполнить команду

Формат⇒Заголовок/примечания формы.

6.14 Для работы с отчетами в окне базы данных активизировать вкладку **Отчеты**.

10 СХЕМЫ БАЗ ДАННЫХ

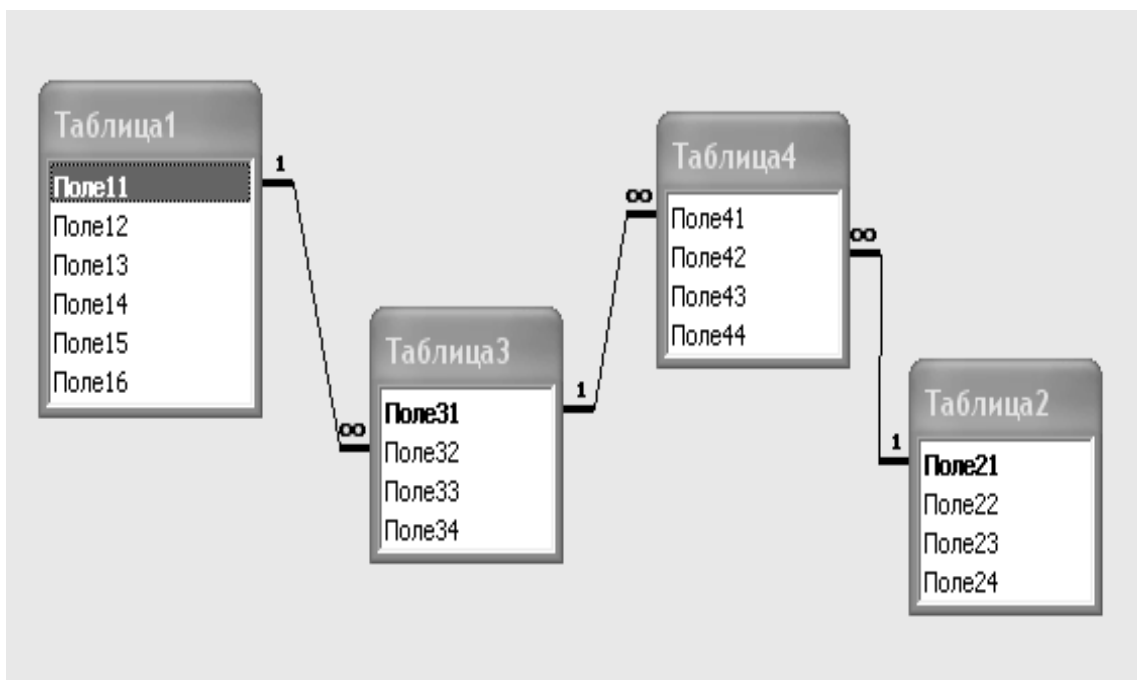


Схема № 1

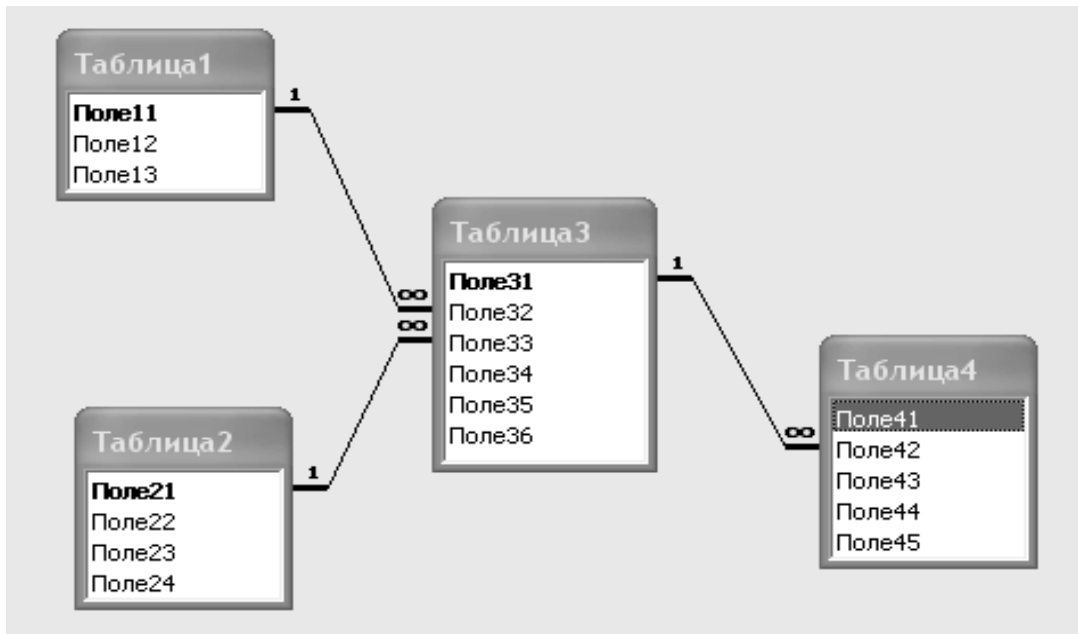


Схема № 2

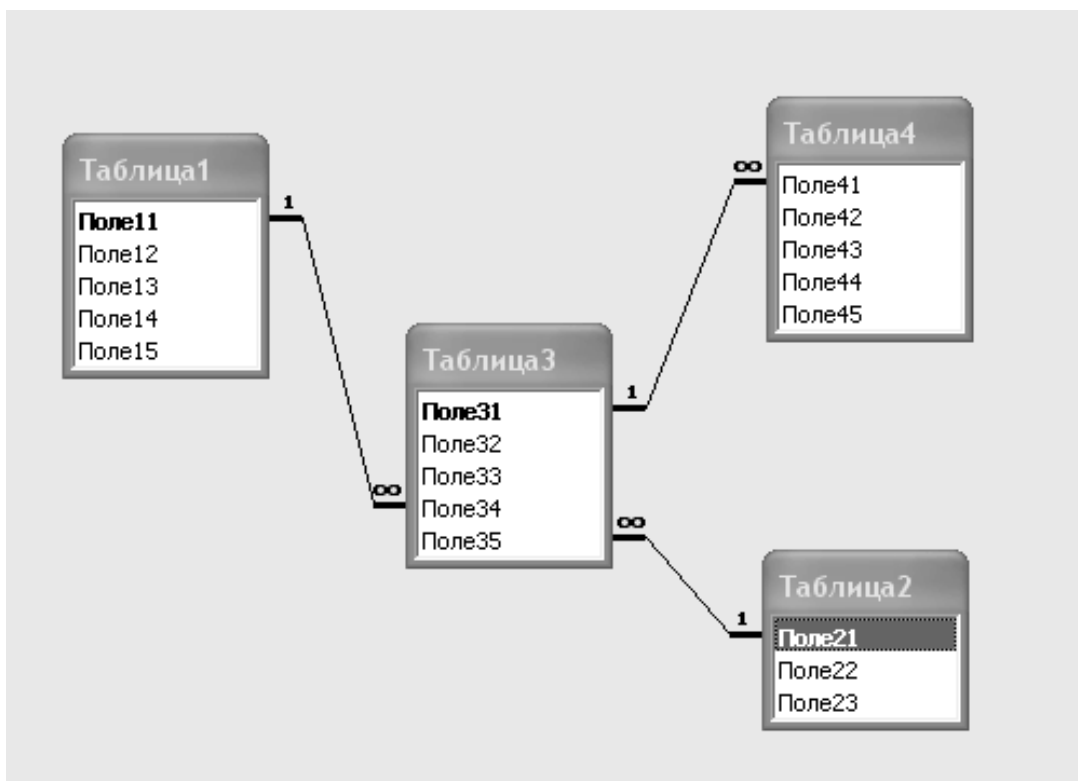


Схема № 3

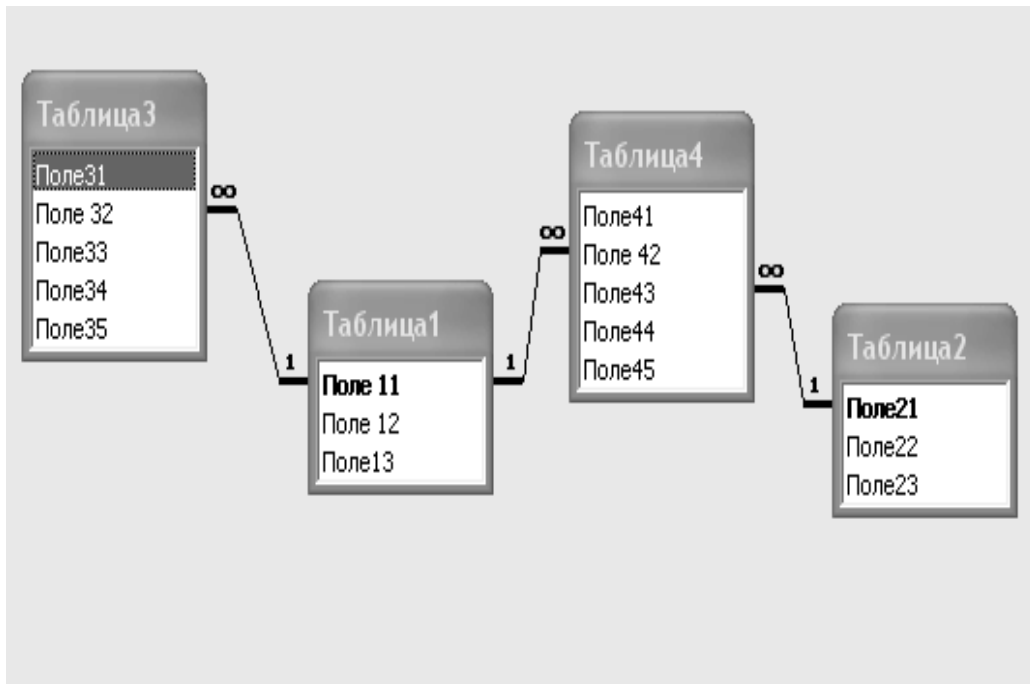


Схема № 4

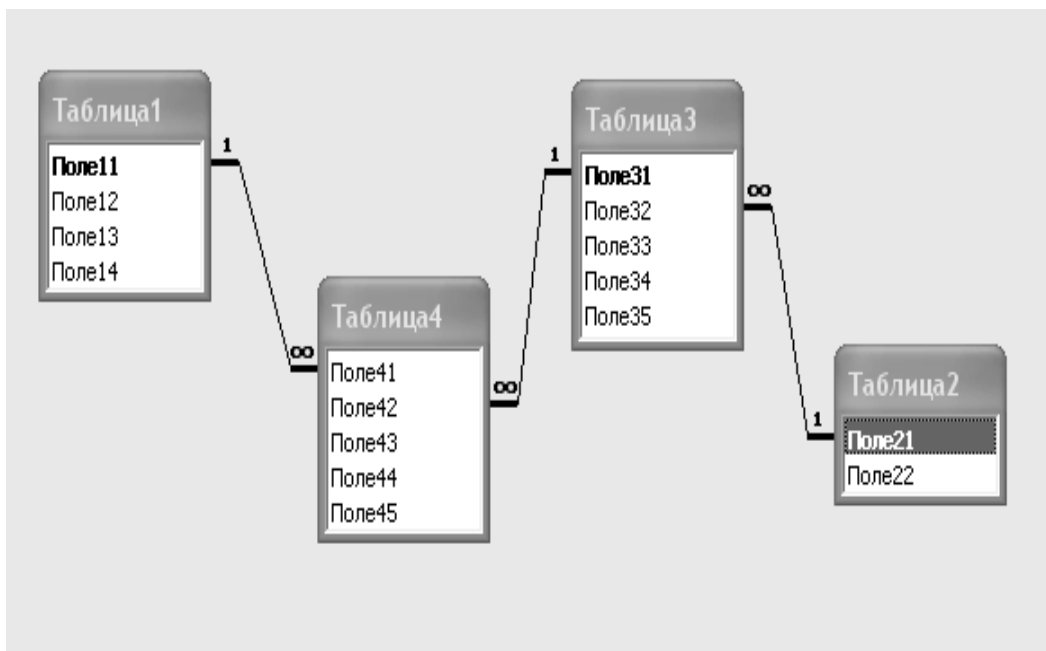


Схема № 5

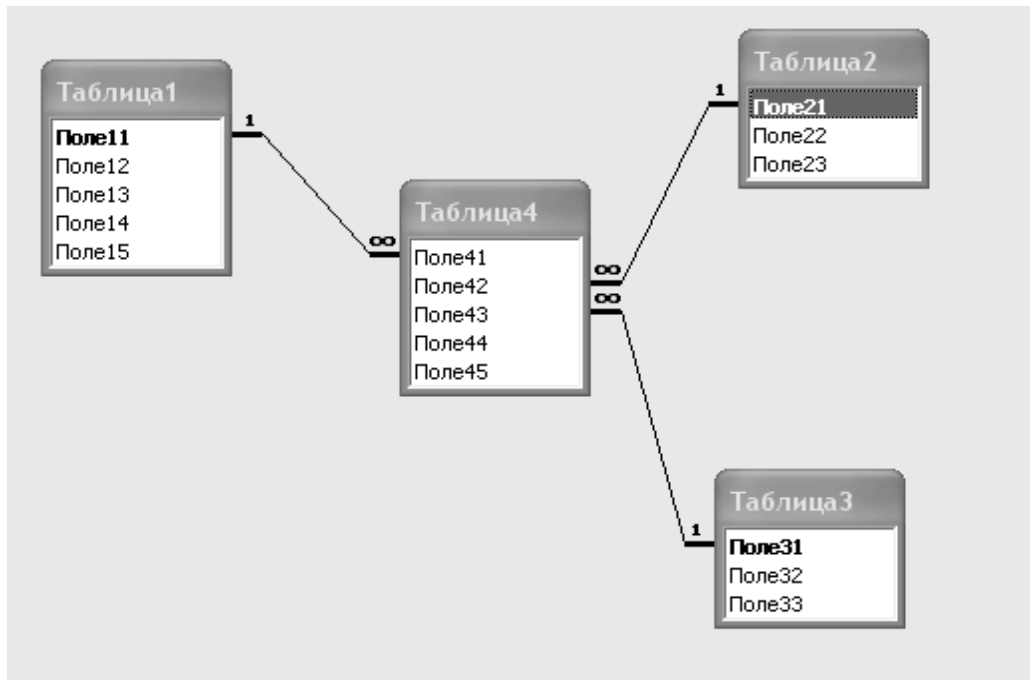


Схема № 6

СПИСОК ЛИТЕРАТУРЫ

1 Симонович С.В. Общая информатика: Учеб. пособие/ С.В.Симонович, Евсева Г.А. – М.: АСТ-ПРЕСС: Инфорком-Пресс, 1998. – 592 с.

2 Богумирский Б.Г. Энциклопедия Windows 98. – СПб.: Питер Ком, 1999. – 816 с.

3 Симонович С.В. Информатика: Базовый курс. – СПб.: Питер, 1999. – 640 с.

Анатолий Григорьевич Фокин

Ирина Анатольевна Гетьман

**СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ
ACCESS В ЭКОНОМИЧЕСКИХ РАСЧЕТАХ**

Учебное пособие

для студентов высших учебных заведений

Редактор

Дудченко Елена Александровна

Подп. в печать

Формат 60x84/16.

Ризограф. печать.

Усл.печ. л. 8,75

Уч.-изд.л. 6,36

Тираж 200 экз.

Заказ №