

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Нижегородский государственный университет  
им. Н.И. Лобачевского**

**В.Л. Тарасов**

**Работа с базами данных в Access 2010**

**Часть 1**

**Учебно-методическое пособие**

Рекомендовано методической комиссией механико-математического факультета для студентов ННГУ, обучающихся по направлениям подготовки 010200 "Математика и компьютерные науки", 010400 "Прикладная математика и информатика", 010800 «Механика и математическое моделирование»

**Нижний Новгород  
2014**

УДК УДК 681.3.06  
ББК 32.973.233  
Т19

Т19 Тарасов В.Л. РАБОТА С БАЗАМИ ДАННЫХ В ACCESS 2010. ЧАСТЬ 1: Учебно-методическое пособие. – [электронный ресурс]. – Нижний Новгород: Нижегородский госуниверситет, 2014. – 126 с.

Рецензент: заведующий кафедрой информационных систем и технологий ННГАСУ д.ф.-м.н., профессор **А.Н. Супрун**

В пособие изложены основные понятия, используемые при работе с базами данных и работа с базами данных в СУБД Access версии 2010. Описаны разработка таблиц, создание запросов на выборку и изменение данных, перекрестных запросов. Рассмотрено создание форм, отчетов, импорт данных из Access в другие приложения, создание документов слияния, макросов. Учебно-методическое пособие предназначено для студентов, изучающих базы данных

Ответственный за выпуск:  
председатель методической комиссии  
механико-математического факультета ННГУ,  
к.ф.м.н., доцент **Н.А. Денисова**

УДК УДК 681.3.06  
ББК 32.973.233

© Тарасов В.Л., 2014  
© Нижегородский государственный  
университет им. Н.И. Лобачевского, 2014

# Содержание

<b>СОДЕРЖАНИЕ</b> .....	<b>3</b>
<b>ПРЕДИСЛОВИЕ</b> .....	<b>5</b>
<b>1. ОСНОВНЫЕ ПОНЯТИЯ</b> .....	<b>6</b>
1.1. БАЗА ДАННЫХ.....	6
1.2. АРХИТЕКТУРА СИСТЕМЫ БАЗ ДАННЫХ.....	7
1.2.1. Три уровня архитектуры.....	7
1.2.2. Языки работы с базами данных.....	7
1.2.3. Функции администратора БД.....	8
1.2.4. Функции СУБД.....	8
1.2.5. Архитектура клиент/сервер.....	9
1.2.6. Распределённая обработка.....	9
1.3. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ.....	11
1.4. РЫНОК СУБД.....	12
<b>2. СРЕДА ACCESS</b> .....	<b>14</b>
2.1. ЗАГРУЗКА.....	14
2.2. ИНТЕРФЕЙС ACCESS.....	16
2.3. ОБЪЕКТЫ БАЗЫ ДАННЫХ ACCESS.....	17
<b>3. ТАБЛИЦЫ</b> .....	<b>18</b>
3.1. ТИПЫ ДАННЫХ.....	18
3.2. СТРУКТУРА БАЗЫ ДАННЫХ.....	19
3.3. СОЗДАНИЕ ТАБЛИЦ.....	20
3.3.1. Конструктор таблиц.....	21
3.3.2. Задание ключевого поля.....	21
3.3.3. Свойства полей.....	22
3.4. ДОБАВЛЕНИЕ ТАБЛИЦ В РЕЖИМЕ КОНСТРУКТОР.....	23
3.4.1. Мастер подстановок.....	23
3.5. СОЗДАНИЕ ТАБЛИЦ В РЕЖИМЕ ТАБЛИЦА.....	26
3.5.1. Создание маски ввода.....	28
3.6. СВЯЗИ МЕЖДУ ТАБЛИЦАМИ.....	30
3.6.1. Установление связей между таблицами.....	30
3.7. ВВОД И ПРОСМОТР ДАННЫХ В РЕЖИМЕ ТАБЛИЦЫ.....	33
3.7.1. Ввод и редактирование данных.....	33
3.7.2. Ввод графической информации в поля таблицы.....	34
3.7.3. Подстановка данных.....	36
3.7.4. Управление видом информации.....	37
3.7.5. Подтаблицы.....	38
3.7.6. Сортировка информации.....	39
3.7.7. Поиск и замена информации.....	39
3.8. ФИЛЬТРАЦИЯ ЗАПИСЕЙ.....	40
3.8.1. Фильтр по выделенному.....	40
3.8.2. Использование обычного фильтра.....	42
3.8.3. Расширенный фильтр.....	43
<b>4. ЗАПРОСЫ</b> .....	<b>47</b>
4.1. СОЗДАНИЕ ПРОСТОГО ЗАПРОСА.....	47
4.1.1. Кнопка Режим.....	49
4.2. КОНСТРУКТОР ЗАПРОСОВ.....	50
4.3. ЗАПРОСЫ С ВЫВОДОМ НАБОРОВ ЗНАЧЕНИЙ.....	52
4.4. ПОДВЕДЕНИЕ ИТОГОВ ПО ЗАПИСЯМ.....	53
4.5. ВЫЧИСЛЯЕМЫЕ ПОЛЯ.....	54
4.6. ЗАПРОСЫ С ПАРАМЕТРАМИ.....	56
4.7. ПЕРЕКРЕСТНЫЕ ЗАПРОСЫ.....	59
4.7.1. Создание перекрестного запроса на основе таблицы.....	59
4.7.2. Создание перекрестного запроса на базе простого.....	62

4.8.	ЗАПРОСЫ НА ИЗМЕНЕНИЕ ДАННЫХ .....	65
4.8.1.	<i>Создание таблицы из запроса</i> .....	65
4.8.2.	<i>Запрос на удаление данных</i> .....	67
4.8.3.	<i>Запрос на добавление</i> .....	68
4.8.4.	<i>Запрос на обновление</i> .....	69
<b>5.</b>	<b>ФОРМЫ</b> .....	<b>71</b>
5.1.	ИНСТРУМЕНТ ФОРМА .....	71
5.2.	ИСПОЛЬЗОВАНИЕ ФОРМЫ .....	72
5.3.	МАСТЕР ФОРМ .....	73
5.4.	МОДИФИКАЦИЯ ФОРМ .....	75
5.4.1.	<i>Режим макета</i> .....	76
5.4.2.	<i>Режим конструктора</i> .....	77
5.4.3.	<i>Добавление связанных элементов управления</i> .....	80
5.4.4.	<i>Свойства элементов управления</i> .....	81
5.4.5.	<i>Форматирование элементов управления</i> .....	82
5.4.6.	<i>Перемещение</i> .....	82
5.4.7.	<i>Изменение размеров и выравнивание</i> .....	82
5.4.8.	<i>Цвет и шрифт</i> .....	84
5.5.	ПОДЧИНЕННЫЕ ФОРМЫ .....	84
<b>6.</b>	<b>ОТЧЕТЫ</b> .....	<b>87</b>
6.1.	ИНСТРУМЕНТ ОТЧЕТ .....	87
6.2.	МАСТЕР СОЗДАНИЯ ОТЧЕТОВ .....	89
6.3.	ИСПОЛЬЗОВАНИЕ ОТЧЕТА .....	95
6.4.	МОДИФИКАЦИЯ СТРУКТУРЫ ОТЧЕТА .....	96
6.4.1.	<i>Работа в конструкторе отчетов</i> .....	96
6.4.2.	<i>Управление группировкой</i> .....	103
6.4.3.	<i>Форматирование элементов управления</i> .....	105
6.5.	СОЗДАНИЕ НАКЛЕЕК .....	105
6.6.	СВЯЗЬ С ДРУГИМИ КОМПОНЕНТАМИ OFFICE .....	108
6.6.1.	<i>Создание документа слияния Word</i> .....	108
6.6.2.	<i>Публикация в Word</i> .....	115
6.6.3.	<i>Анализ в Excel</i> .....	116
<b>7.</b>	<b>МАКРОСЫ</b> .....	<b>119</b>
7.1.	СОЗДАНИЕ МАКРОСА .....	119
7.2.	ВЛОЖЕННЫЕ МАКРОСЫ.....	121
7.3.	УСЛОВНЫЕ МАКРОКОМАНДЫ.....	122
7.4.	ЗАПУСК МАКРОСА.....	123
7.5.	НАЗНАЧЕНИЕ КЛАВИШ ДЛЯ ЗАПУСКА МАКРОСОВ .....	124
	<b>ЛИТЕРАТУРА</b> .....	<b>126</b>
	Основная .....	126
	Дополнительная .....	126

## Предисловие

В данном пособии рассматривается работа с системой управления базами данных (СУБД) Access, выпускаемой компанией Microsoft. С помощью Access можно разрабатывать базы данных различной сложности: от индивидуальной, предназначенной для использования на отдельном компьютере, до систем масштаба предприятия, работающих в компьютерных сетях. Access, как одно из приложений пакета Microsoft Office, установлена на многих компьютерах, использующих Office.

Access обладает большим набором инструментов для выполнения различных манипуляций с базами данных. Для начинающих пользователей большую помощь оказывают *мастера*, которые предлагают последовательность шагов с подробными подсказками, для создания таблиц, форм, отчетов и выполнения других действий. *Конструкторы* предоставляют опытным пользователям полные возможности по созданию объектов базы данных. Все это позволяет быстро освоить работу с базами данных, используя только визуальные средства.

# 1. Основные понятия

## 1.1. База данных

Существование каждого предприятия, организации, отдельного человека сопровождается накоплением сведений о его деятельности, которые оформляются в виде записей на различных материальных носителях. Такими носителями могут быть различные предметы: глиняные таблички, папирус, пергамент, бумага, перфокарты, магнитные ленты, магнитные диски, оптические диски и т.п. Всю совокупность записей о деятельности какого-либо предприятия можно назвать *базой данных* этого предприятия. Могут создаваться и локальные базы данных для хранения сведений об отдельной сфере деятельности предприятия или человека, например, в базе данных могут храниться сведения о чьих-либо знакомых, любитель музыки может вести базу данных об имеющейся у него коллекции альбомов. В настоящее время базы данных ведутся с помощью компьютеров.

Применение компьютеров для хранения и обработки данных началось примерно с середины 1950-х годов. Компьютерные базы данных имеют ряд преимуществ по сравнению с традиционным бумажным способом ведения записей. Отметим некоторые из них.

*Компактность.* Нет необходимости в многочисленных журналах и картотеках.

*Скорость.* Обеспечивается быстрый поиск нужной информации.

*Низкие трудозатраты.* Отпадает необходимость в утомительной, рутинной ручной работе по заполнению бумажных форм и карточек.

*Оперативность.* Всегда доступна точная актуальная информация.

Указанные преимущества привели к широкому распространению компьютерных баз данных, хотя их использование требует достаточно больших затрат, связанных с приобретением компьютеров, программного обеспечения, обучением работников.

Приведем несколько терминов, используемых, когда речь идет о базах данных.

*Система баз данных* – это компьютеризованная система хранения записей. Система баз данных включает в себя *данные*, непосредственно сохраняемые в *базе данных (БД)*, *аппаратное обеспечение*, *программное обеспечение (ПО)* и *пользователей*.

*СУБД (система управления базами данных)* – это программная система, предназначенная для создания на компьютере базы данных для многих приложений, поддержания её в актуальном состоянии и обеспечения эффективного доступа пользователей к содержащимся в ней данным в рамках предоставленных им полномочий.

*Пользователи* подразделяются на прикладных программистов, конечных пользователей и администраторов баз данных (АБД).

*Прикладные программисты* отвечают за написание прикладных программ, использующих БД.

*Конечные пользователи* работают с СУБД непосредственно через рабочую станцию или терминал.

*Администраторы баз данных* – это технические специалисты, отвечающие за администрирование БД и СУБД в соответствии с требованиями, устанавливаемыми *администратором данных*.

Если предприятие хранит свои данные в централизованной базе данных, должен существовать человек, который несёт основную ответственность за данные предприятия. Это *администратор данных*. Он принимает решения на уровне высшего руководства предприятия о том, какие данные необходимо вносить в базу данных в первую очередь, над какими данными и какие операции можно осуществить.

Системы баз данных могут реализовываться как на малых персональных компьютерах (ПК), так и на больших машинах или системах машин, объединенных в сеть. В зависимости

от мощности используемых машин и своего назначения, системы баз данных могут быть однопользовательскими и многопользовательскими.

*Однопользовательская система* (single-user system) – это система, в которой к базе данных в одно и то же время может получить доступ не более одного человека.

*Многопользовательская система* (multi-user system) – это система, в которой к базе данных могут получить доступ сразу несколько пользователей.

С точки зрения пользователя между этими системами нет разницы, так как в обоих случаях ему предоставляется одинаковая возможность работы с интересующими его данными в рамках предоставленных полномочий.

## 1.2. Архитектура системы баз данных

### 1.2.1. Три уровня архитектуры

Под архитектурой системы баз данных понимают общий подход к ее описанию. Различают три уровня архитектуры: *внутренний*, *концептуальный* и *внешний* (рис.1.1).

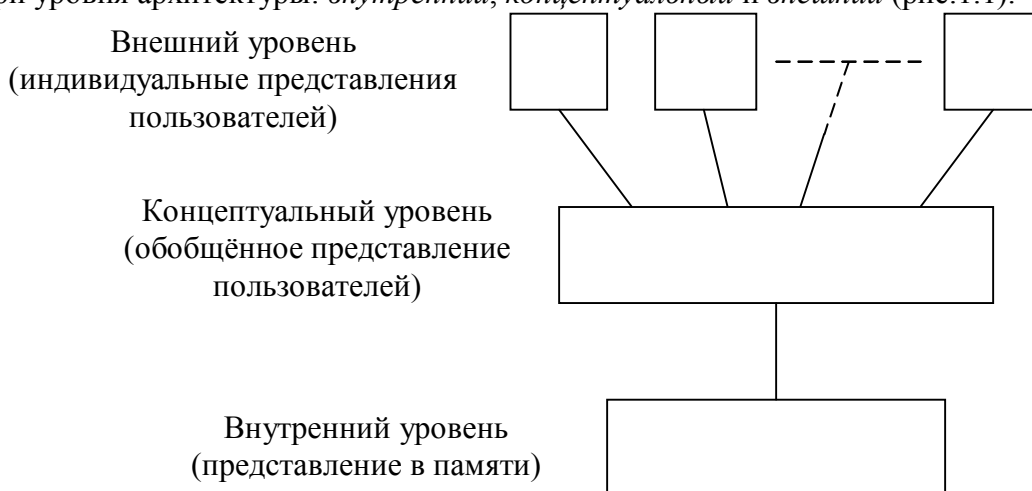


Рис. 1.1. Три точки зрения на базу данных

Представление отдельного пользователя называется *внешним представлением*. Например, пользователь из отдела кадров может рассматривать БД как набор записей с информацией об отделах плюс набор записей с информацией о служащих и ничего не знать о деталях, проектах, поставщиках и других вещах, сведения о которых есть в базе.

*Концептуальное представление* – это представление всего содержимого БД. На концептуальном уровне данные определяются с помощью *концептуальной схемы*. Концептуальная схема содержит перечень всех данных, хранящихся в базе данных, правила *безопасности* и *целостности*. Под целостностью понимается непротиворечивость сведений, хранящихся в БД.

*Внутреннее представление* описывается с помощью *внутренней схемы*, которая описывает типы хранимых полей, существующие индексы, способы представления хранимых полей, физическую последовательность хранимых записей. Внутренний уровень наиболее близок к физическому хранению.

### 1.2.2. Языки работы с базами данных

Для работы с данными можно использовать универсальные языки программирования: C, Pascal, Basic, C++ и специализированные языки, например, FoxPro. Универсальные языки называются базовыми языками. Они могут включать подязык данных. В настоящее время в качестве подязыка данных используется SQL (Structured Query Language, структурирован-

ный язык запросов). Подъязык данных включает язык определения данных и язык обработки данных.

### 1.2.3. Функции администратора БД

Администратор базы данных (АБД) отвечает за нормальное функционирование БД. В его функции входит:

- *Определение концептуальной схемы.*
- *Определение внутренней схемы.*
- *Взаимодействие с пользователями.*
- *Определение правил безопасности и целостности.*
- *Определение процедур резервного копирования и восстановления.* После того, как предприятие доверило свои данные СУБД, оно стало критически зависимым от успешного функционирования системы. В случае повреждения какой-либо части БД важно иметь возможность восстановить данные с минимальной задержкой.
- *Управление производительностью и реагирование на изменяющиеся требования.*

### 1.2.4. Функции СУБД

СУБД – это программное обеспечение, которое управляет работой с базой данных. СУБД должна обеспечивать выполнение ряда функций. Перечислим их.

- *Определения данных.* СУБД должна обеспечивать определение структуры БД, то есть создание внешней, концептуальной и внутренней схемы.
- *Обработка данных.* СУБД должна обрабатывать запросы пользователей на выборку, изменение или удаление существующих данных, а также на добавление новых данных в базу. Запросы к данным могут быть планируемыми, которые заранее предусмотрел АБД, и не планируемыми.
- *Безопасность и целостность данных.* СУБД должна контролировать пользовательские запросы и пресекать попытки нарушения правил безопасности и целостности.
- *Восстановление и дублирование.* СУБД должна иметь компонент, называемый *администратором транзакций*, который осуществляет контроль над изменением данных, обеспечивает их восстановление и дублирование.
- *Словарь данных.* Он содержит описание данных, входящих в данную БД. Словарь данных является системной базой данных, содержащей данные о данных.
- *Производительность.* СУБД должна обеспечивать быстрое выполнение запросов пользователя к БД. Заметим, что в своё время система FoxPro стала лидером среди СУБД для персональных компьютеров именно благодаря своей высокой производительности.

Как правило, СУБД реализует внутренний и концептуальный уровни архитектуры, то есть обеспечивает хранение данных на дисках и предоставляет возможность описания логической структуры данных. Для реализации внешних представлений пользователей пишется отдельная программа с удобным для конечного пользователя интерфейсом и возможностью выполнения типовых запросов как надстройка над СУБД. Опишем последовательность обработки запроса пользователя к БД.

- Пользователь выдаёт запрос на доступ;
- СУБД перехватывает этот запрос и анализирует его;



- Затем СУБД просматривает внешнюю схему для этого пользователя, соответствующее отображение внешний–концептуальный, концептуальную схему, отображение концептуальный–внутренний и определение структуры хранения;
- Выполняет необходимые операции над хранимой базой данных.

### 1.2.5. Архитектура клиент/сервер

Систему баз данных можно рассматривать как систему, состоящую из двух частей – *сервера* (или машины базы данных) и набора *клиентов* (или внешнего интерфейса) (рис.1.2). Сервер – это собственно СУБД. Он поддерживает все основные функции СУБД. Клиенты – это различные приложения, которые выполняются “над” СУБД – приложения, написанные пользователем и встроенные приложения, предоставляемые поставщиками СУБД или сторонними поставщиками.



Рис. 1.2. Схема архитектуры клиент-сервер

### 1.2.6. Распределённая обработка

Распределенная обработка предполагает использование нескольких компьютеров, соединенных между собой. Программу-сервер можно запускать на одном компьютере, а программу-клиент на другом (рис.1.3).

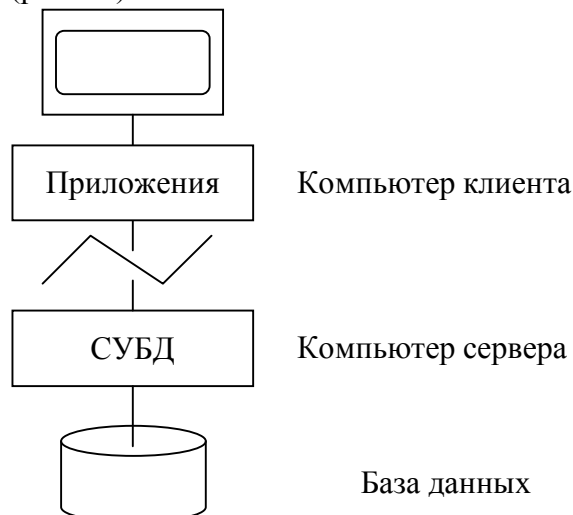


Рис. 1.3. Клиент и сервер запускаются на разных машинах

Компьютер, на котором запускается программа-сервер, часто называют также *сервером*.

Один сервер может предоставлять доступ нескольким клиентам (рис.1.4).

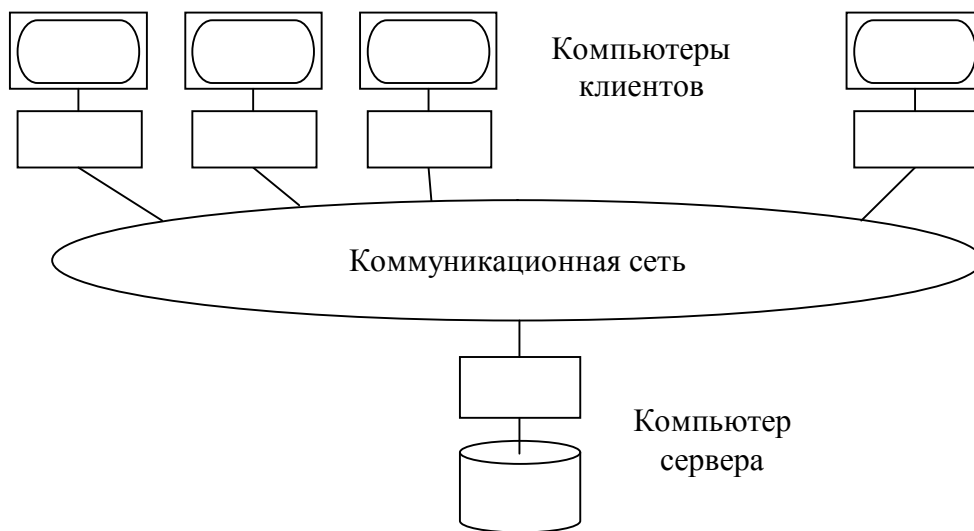


Рис. 1.4. Один сервер, много клиентов

Каждая машина может одновременно являться и клиентом и сервером (рис.1.5.). Такая схема взаимодействия компьютеров характерна для сети Интернет.

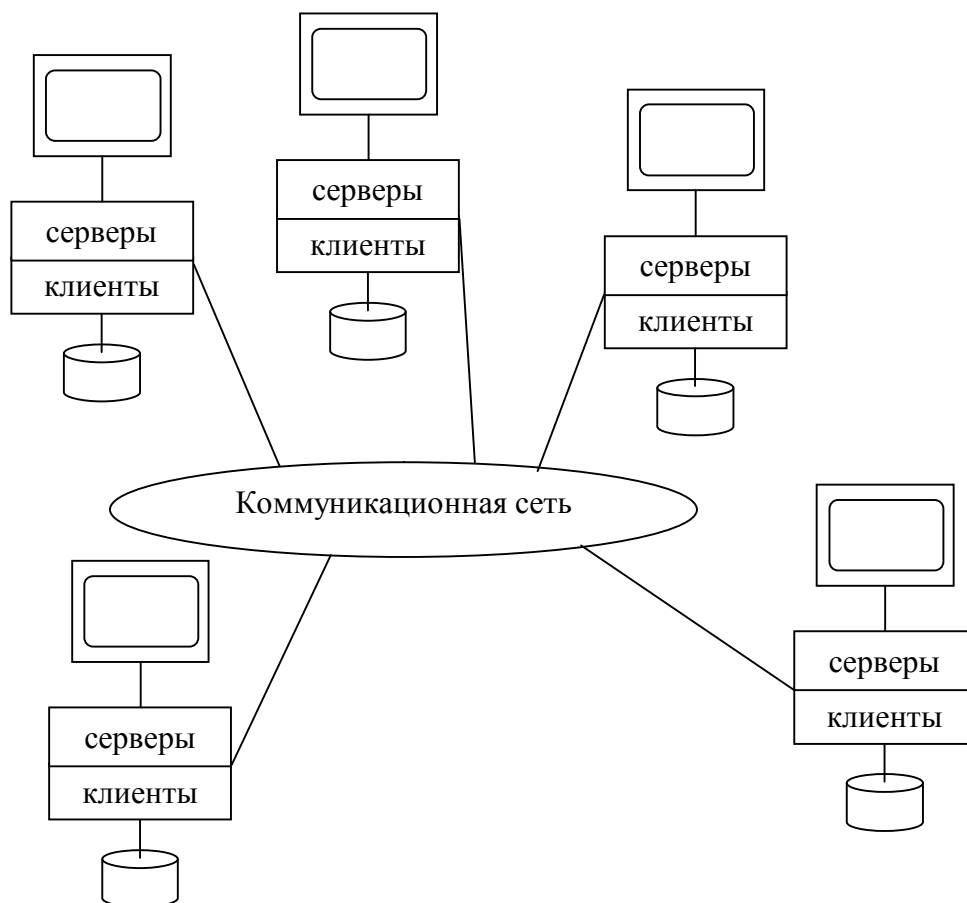


Рис. 1.5. Каждый компьютер является и клиентом и сервером

### 1.3. Проектирование базы данных

В настоящее время информация в базах данных хранится, как правило, в виде *таблиц*. Такой способ представления информации называется *реляционной моделью данных*. Таблицы состоят из *строк* и *столбцов*. Отдельная строка таблицы содержит информацию об одном конкретном объекте, например, об одной из фирм, с которой взаимодействует данное предприятие. Столбцы таблицы называются *полями*, то есть строка состоит из полей.

При проектировании базы данных нужно учитывать три уровня архитектуры: *внешний, концептуальный и внутренний*.

На *внешнем* уровне ориентируются на требования пользователей. Эти требования обычно состоят в том, чтобы пользователю была легко доступна необходимая для работы информация о предметной области, описываемой с помощью базы данных.

Пусть, например, некоторой фирме постоянно требуется работать с информацией о своих *поставщиках* и о поставляемых ими *деталях*. Подобные сведения содержатся в специальных документах – *накладных*, которые сопровождают каждую *поставку* деталей от поставщика. В накладной содержатся реквизиты поставщика, то есть его название и адрес, название и некоторые характеристики детали, например, вес и цвет, а также количество поставленных деталей. Конечный пользователь, занимающийся снабжением, привык работать с накладными и с его точки зрения БД должна обрабатывать накладные.

На *концептуальном* уровне проводится логический анализ предметной области. Состав поставщиков и деталей может изменяться, причем изменения в составе и характеристиках поставщиков и деталей независимы друг от друга, то есть поставщики и детали есть независимые *объекты* или, как говорят в теории баз данных, *сущности*. Таким образом, в каждой накладной на поставку упоминаются два различных объекта или сущности, это *поставщик* и *деталь*. Каждая из этих сущностей характеризуется некоторым набором свойств или *атрибутов*.

В упрощенном варианте поставщик имеет следующие атрибуты: *название, город* (место расположения), *статус* или *рейтинг*. Рейтинг может характеризовать поставщика в глазах данной фирмы с точки зрения стоимости и качества поставляемых деталей, аккуратности в выполнении договоров и помогает выбрать наиболее выгодного поставщика.

Деталь имеет атрибуты: *название, вес, цвет*. Для полноты картины неплохо хранить в базе данных рисунок или чертеж детали.

Атрибуты каждой сущности следует хранить в базе данных в одном экземпляре, чтобы не было разночтений из-за возможных ошибок.

Из приведенных рассуждений следует, что в рассматриваемой предметной области имеются две сущности *Поставщик* и *Деталь*, информацию о которых будем хранить в двух таблицах со следующей структурой.

**Таблица Поставщики**

Имя поля	Назначение
<i>КодПоставщика</i>	Уникальный номер
<i>Поставщик</i>	Название, не обязательно уникальное
<i>Город</i>	Город, где расположен поставщик
<i>Рейтинг</i>	Значение рейтинга

**Таблица Детали**

Имя поля	Назначение
<i>КодДетали</i>	Уникальный номер
<i>Название</i>	Название детали
<i>Цвет</i>	Цвет
<i>Вес</i>	Вес
<i>Эскиз</i>	Рисунок, чертеж или эскиз

Накладная фиксируют факт поставки определенного количества какой-либо детали некоторым поставщиком. Эту информацию можно хранить в виде таблицы *Поставки*, структура которой приведена ниже.

**Таблица Поставки**

Имя поля	Назначение
<i>КодПоставщика</i>	Номер поставщика
<i>КодДетали</i>	Номер детали
<i>Количество</i>	Количество в партии
<i>Дата</i>	Дата поставки

В таблице *Поставки* хранятся только числовые коды поставщиков и деталей. Зная эти коды, можно легко найти полную информацию о поставщике и детали из соответствующих таблиц.

Таким образом, после анализа предметной области приходим к решению, что база данных о поставщиках и деталях должна состоять из трех таблиц *Поставщики*, *Детали*, *Поставки*. Будем считать, что на этом закончено концептуальное проектирование.

На *внутреннем* уровне решаются вопросы о выборе СУБД и о конкретном представлении таблиц БД в выбранной СУБД, о типе данных для полей таблиц, об установлении связей между таблицами. Современные СУБД берут на себя вопросы о размещении таблиц в файлах на дисках, раньше такие вопросы приходилось решать программисту, который разрабатывал БД. Обсуждение этого уровня проектирования отложим до знакомства с конкретной СУБД.

## **1.4. Рынок СУБД**

В настоящее время существует много СУБД, предлагаемых различными фирмами. Все СУБД можно разделить на два класса: СУБД для настольных систем и промышленные или «тяжелые» СУБД.

Настольные системы предназначены для использования на одном компьютере или в небольшой сети. Они обладают ограниченными возможностями по количеству хранимых записей, у них упрощен контроль за целостностью данных. К таким системам относятся Fox-Pro и Access фирмы Microsoft, Paradox фирмы Borland и другие.

Промышленные СУБД предназначены для использования на больших машинах и в крупных сетях. Они способны хранить практически неограниченные объемы данных, отличаются высоким быстродействием, имеют развитые механизмы контроля целостности данных и восстановления после сбоев. Здесь следует прежде всего упомянуть систему DB2 фирмы IBM, Oracle фирмы Oracle, Microsoft SQL Server фирмы Microsoft.

Для использования в сети Интернет часто применяют СУБД MySQL, которую можно использовать бесплатно.

В настоящее время все более популярной становится СУБД Access. Это полноценная реляционная СУБД, с помощью которой можно создавать достаточно сложные приложения, работающие на одном компьютере или в сети. Access является одним из компонентов инте-

грированного пакета Microsoft Office, который включает также текстовый редактор Word, электронные таблицы Excel, редактор презентаций Power Point и другие программы. Access имеет развитый интерфейс, похожий на интерфейс других приложений пакета Office, интегрирована с ними, то имеется возможность передавать данные в редактор Word, например, для составления типовых писем, или в Excel для анализа числовых данных из БД.

Далее будет описываться работа в СУБД Access. Данная программа постоянно развивается, появляются новые версии. В настоящей работе рассмотрена версия Access 2010. Номер версии 2010 соответствует году ее выхода.

## 2. Среда Access

### 2.1. Загрузка

После запуска Access нужно открыть меню **Файл** (рис.2.1) и выбрать пункт меню **Создать**. Новую базу данных можно создать по существующим шаблонам и образцам, значки которых представлены в окне.

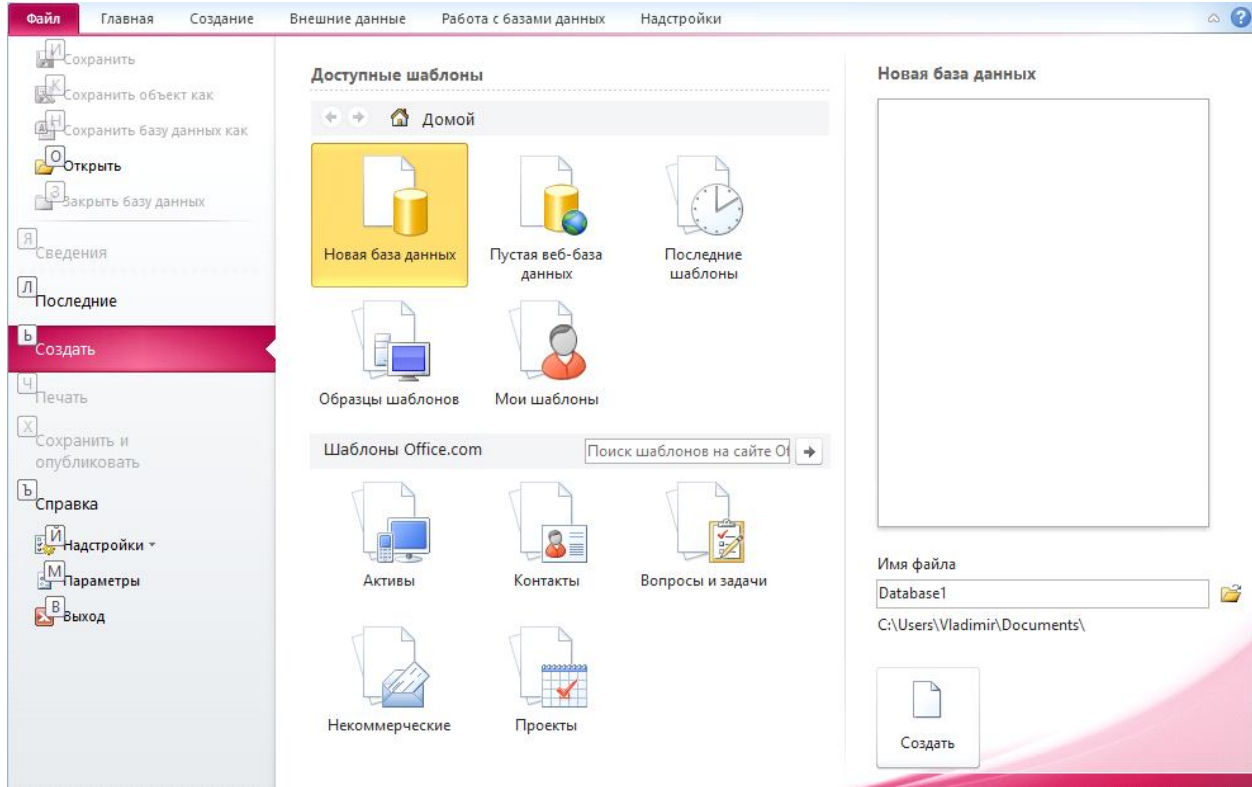



Рис. 2.1. Диалог загрузки Access

Для создания новой пустой базы данных выберем **Новая база данных**. Справа от поля **Имя файла** нажмем значек , откроется диалог сохранения файла (рис.2.2), в котором следует выбрать папку для файла БД и указать имя сохраняемого файла, например, для БД поставщиков и деталей можно задать имя *ПоставщикиДетали*. Расширение имени файла будет добавлено автоматически. В дальнейшем эту базу данных будем называть *ПоставщикиДетали* по имени файла.

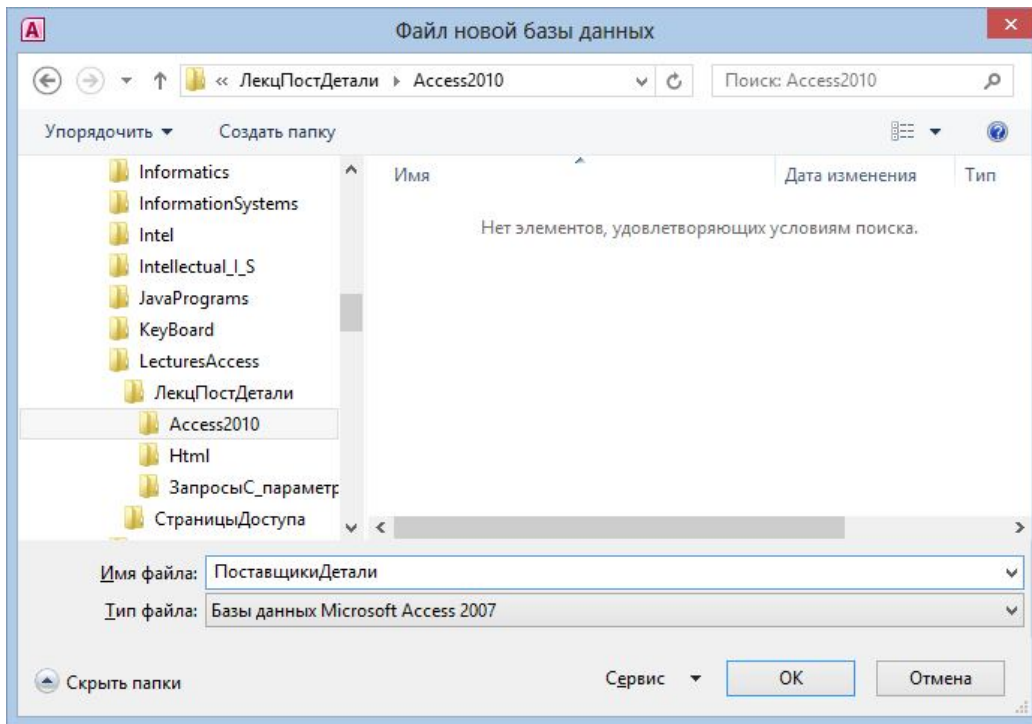


Рис. 2.2. Диалог сохранения файла базы данных

После назначения папки и имени файла базы данных, этот выбор отразится в окне создания БД (рис.2.3).

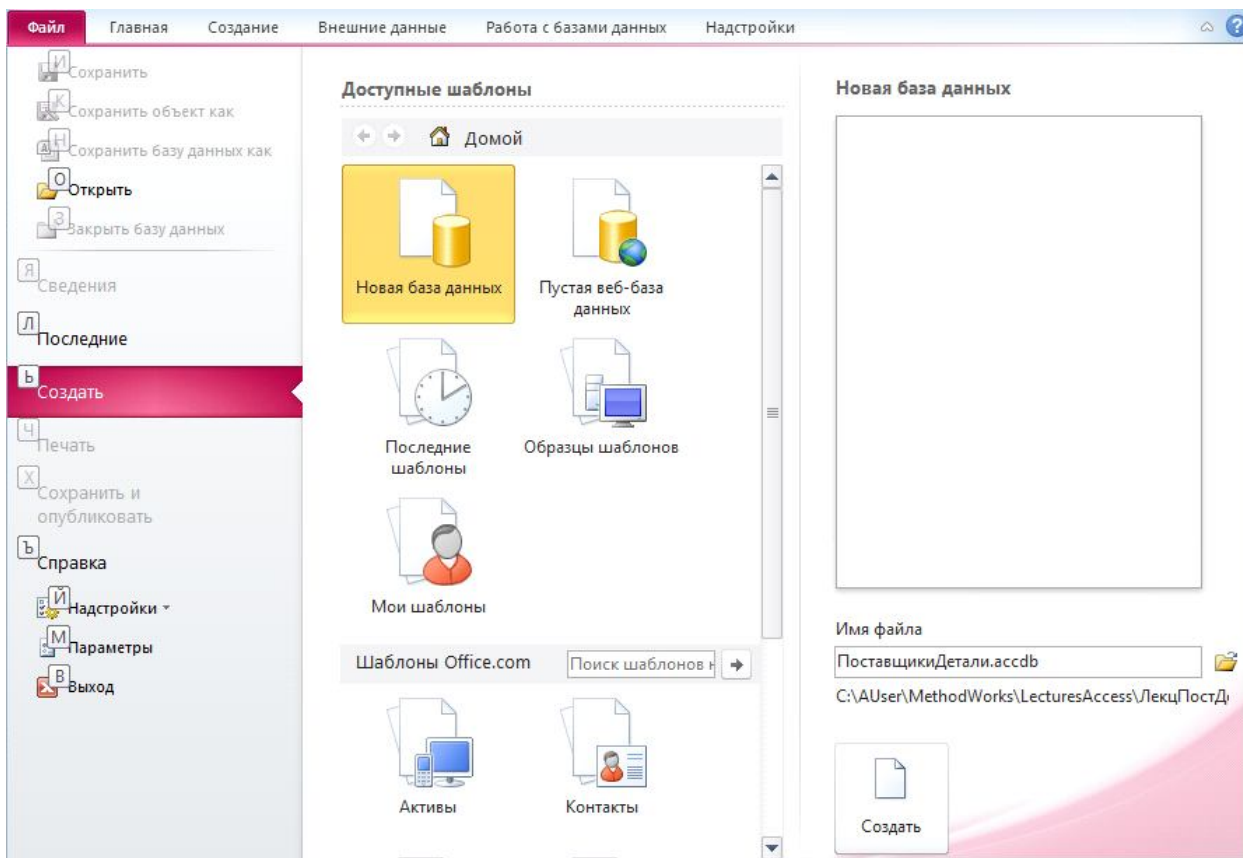


Рис. 2.3. Выбрана папка для создаваемой БД и имя файла БД

Новая БД с выбранным именем создается после нажатия кнопки **Создать**. Окно программы после создания новой БД показано на рис. 2.4.

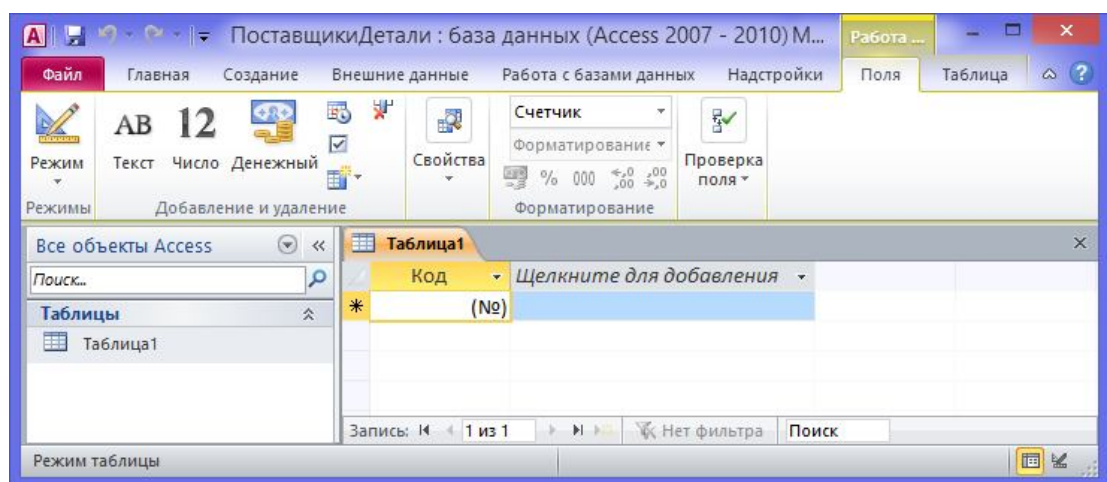


Рис. 2.4. Основное окно Access

Access создает единственный файл с расширением .accdb, в котором размещается вся информация базы данных.

Перейти от работы с текущей БД к созданию новой можно, выполнив команду **Файл, Создать**, при этом текущая БД будет закрыта. Access может работать одновременно только с одной базой данных, в отличие от, например, текстового редактора Word, который может работать сразу с несколькими открытыми документами. Если нужно держать открытыми несколько баз данных, можно запустить несколько экземпляров Access.

## 2.2. Интерфейс Access

Интерфейс Access показан на рис.2.5. В верхней части основного окна находятся закладки, на которых расположены кнопки инструментов для работы с базой данных.

Слева расположена панель **Все объекты Access**, позволяющая переключаться на работу с различными объектами, которые могут входить в состав базы данных.

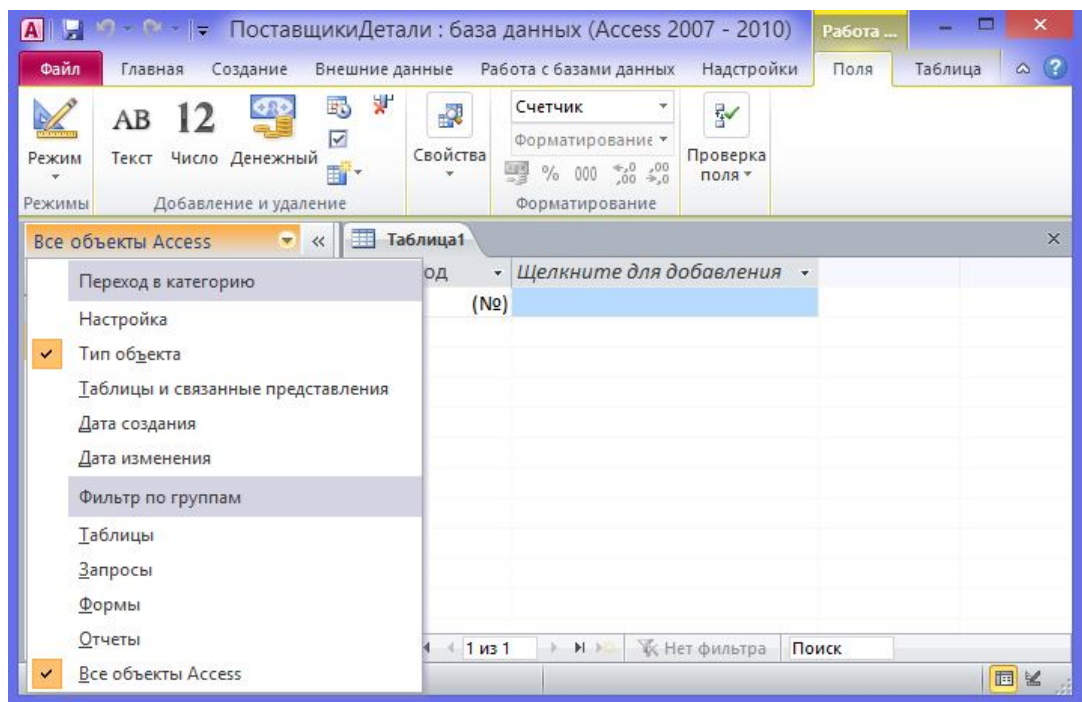


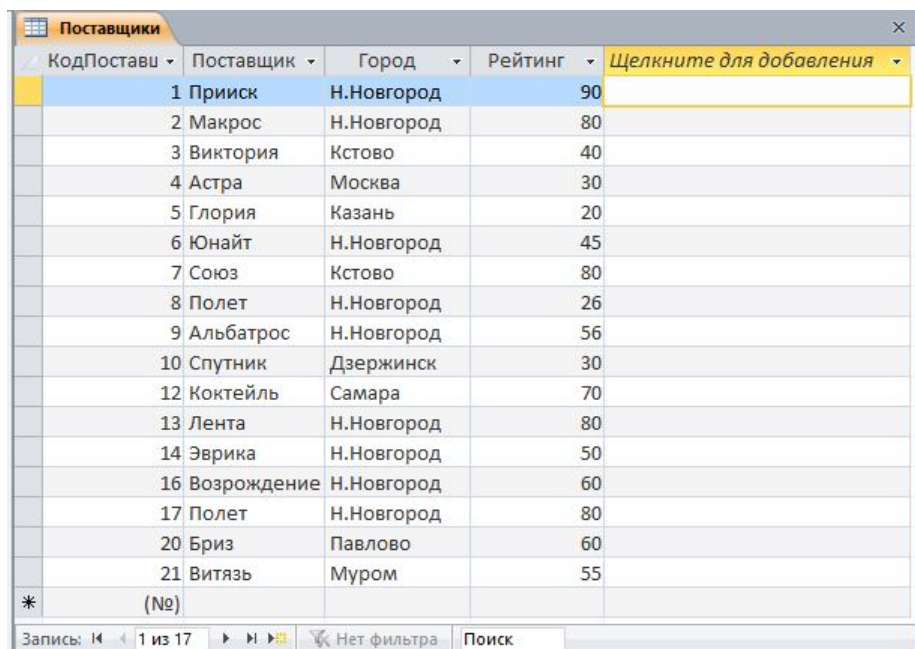
Рис. 2.5. Список объектов Access



## 2.3. Объекты базы данных Access

Рассмотрим вкратце роль и назначение различных объектов базы данных.

*Таблицы* – это основа БД, именно в них хранится информация. Таблицы состоят из строк и столбцов. Каждая строка таблицы – это *запись*, каждый столбец – *поле*. Таким образом, запись состоит из полей. Все элементы какого-либо столбца имеют один и тот же тип, например, числовой, текстовый и т.п. Тип поля задаётся в процессе создания структуры таблицы. Забегая вперед, рассмотрим содержимое таблицы (рис.2.6), содержащей сведения о фирмах-поставщиках. Таблица называется *Поставщики*. Каждый поставщик имеет уникальный числовой номер *КодПоставщика*, текстовое имя *Поставщик*, место расположения *Город* текстового типа и числовую характеристику *Рейтинг*.



КодПостав	Поставщик	Город	Рейтинг	Щелкните для добавления
1	Прииск	Н.Новгород	90	
2	Макрос	Н.Новгород	80	
3	Виктория	Кстово	40	
4	Астра	Москва	30	
5	Глория	Казань	20	
6	Юнайт	Н.Новгород	45	
7	Союз	Кстово	80	
8	Полет	Н.Новгород	26	
9	Альбатрос	Н.Новгород	56	
10	Спутник	Дзержинск	30	
12	Коктейль	Самара	70	
13	Лента	Н.Новгород	80	
14	Эврика	Н.Новгород	50	
16	Возрождение	Н.Новгород	60	
17	Полет	Н.Новгород	80	
20	Бриз	Павлово	60	
21	Витязь	Муром	55	
*	(№)			

Рис. 2.6. Содержимое таблицы *Поставщики*

*Запрос* обеспечивает выборку данных, удовлетворяющих некоторым условиям, из одной или нескольких таблиц или ранее созданных других запросов.

*Форма* – это окно, содержащее элементы управления – такие как надписи, текстовые поля, флажки и списки для просмотра, ввода и редактирования информации. Формы могут отображать одну или несколько записей из одной или нескольких таблиц или запросов.

*Отчёты* предназначены для вывода на печать информации из БД в обработанном виде. Отчёт может содержать элементы управления и данные из одной или нескольких таблиц.

*Страница доступа к данным* напоминает форму, но хранится в отдельном файле формата HTML, поэтому её можно просматривать в Web-браузере, например, Internet Explorer и, следовательно, иметь доступ к данным из Интернета или корпоративной сети. (HTML, Hypertext Markup Language – язык разметки гипертекстов – используется для создания Web-страниц.)

*Макрос* представляет собой набор нескольких макрокоманд, имеющий имя, использование которого позволяет ускорить выполнение часто встречающихся действий.

*Модули* представляют собой программы, написанные на встроенном языке Visual Basic for Application (VBA). Они разрабатываются для выполнения действий, которые трудно осуществить с помощью визуальных инструментов или для автоматизации обработки данных, хранящихся в БД.

## 3. Таблицы

### 3.1. Типы данных

Информация, хранящаяся в базе данных, может быть самой разной по смыслу и по форме, например, это может быть текстовая информация о названиях фирм, числовая информация о количестве поставленных деталей, о весе отдельной детали или информация о дате поставки. СУБД предоставляют возможность использовать различные типы данных для полей таблиц.

Access поддерживает следующие типы данных для полей таблиц.

**Текстовый.** Может содержать строку символов длиной до 255 символов.

**Поле Мемо.** Может содержать произвольный многострочный текст размером до 64000 символов.

**Числовой.** Данное поле может содержать некоторое числовое значение. Конкретный числовой тип определяется свойством «размер поля». Возможные значения числового типа приведены в табл. 3.1.

Таблица 3.1. Типы числовых данных

Тип	Размер, байт	Диапазон значений
<b>Байт</b>	1	от 0 до 255
<b>Целое</b>	2	от -32768 до 32767
<b>Длинное целое</b>	4	от -2147483648 до 2147483647
<b>Одинарное с плавающей точкой</b>	4	От $-3.402823 \cdot 10^{38}$ до $-1.401298 \cdot 10^{-45}$ для отрицательных от $1.401298 \cdot 10^{-45}$ до $3.402823 \cdot 10^{38}$ для положительных
<b>Двойное с плавающей точкой</b>	8	от $-1.79769313486231 \cdot 10^{308}$ до $-4.94065645841247 \cdot 10^{-324}$ для отрицательных, от $4.94065645841247 \cdot 10^{-324}$ до $1.79769313486231 \cdot 10^{308}$ для положительных
<b>Код репликации</b>	16	Глобальный уникальный идентификатор (GUID).
<b>Действительное</b>	12	От $-10^{38} + 1$ до $10^{38} - 1$

Код репликации размером в 16 байт используется в базе данных Microsoft Access для создания уникального идентификатора при репликации. *Репликация* – это создание копий баз данных (реплик), которые могут обмениваться обновляемыми данными или реплицированными формами, отчетами или другими объектами. Такой обмен называют синхронизацией. Каждая копия базы данных называется репликой, и каждая реплика содержит общий набор таблиц, запросов, отчетов, форм, макросов и модулей. Каждая реплика может также содержать локальные объекты, которые существуют только в этой реплике. В Microsoft Access коды GUID называются кодами реплик.

Поля типа **Действительное** могут хранить целые числа, задаваемые десятичными цифрами. Количество десятичных цифр может устанавливаться свойством **Точность** в пределах от 1 до 28.

**Дата/время.** Содержит дату и время в диапазоне от 100 до 9999 года.

**Денежный.** Применяется для значений валют. Предотвращает округления при проведении вычислений. Может иметь до 15 цифр в целой части и до 4 в дробной.

**Счётчик.** Обеспечивает автоматическую вставку последовательных (увеличивающихся на 1) или случайных чисел при добавлении записи. Гарантируется, что значения счетчика не будут повторяться.

**Логический.** Содержит только одно из двух значений: "Да/Нет", "Истина/Ложь", "Вкл/Выкл".

**Поле объекта OLE.** Может содержать двоичные объекты, например, документы MS Word, MS Excel, рисунки, звуки и другие двоичные данные, созданные в программах, использующих протокол OLE (Object Linking and Embedding – связывание и встраивание объектов). Технология OLE обеспечивает обмен данными между различными программами, например, с помощью этой технологии в текстовый документ можно вставить рисунок, созданный в каком-либо графическом редакторе. Объекты, хранимые в базе данных, могут быть связанными или внедренными. Для отображения объекта OLE на форме или в отчете необходимо использовать присоединенную рамку объекта.

**Гиперссылка.** Гиперссылка хранит путь к месту назначения, например, к объекту, документу или Web-странице в Internet. Гиперссылка может представлять адрес в Интернете или в интрасети (URL, Uniform Resource Locator – единообразный указатель ресурса) или сетевой маршрут к файлу в локальной сети (UNC, Universal Naming Convention – универсальное соглашение об именах). Адрес гиперссылки может также содержать некоторую специальную адресную информацию (например, объект базы данных, закладку Microsoft Word или диапазон ячеек Microsoft Excel, на которые указывает адрес). При выборе гиперссылки Web-обозреватель или Microsoft Access используют адрес гиперссылки для перехода по заданному адресу. Для настройки гиперссылки можно использовать команду контекстного меню **Гиперссылка**, со списком действий по настройке гиперссылки (рис.3.1).

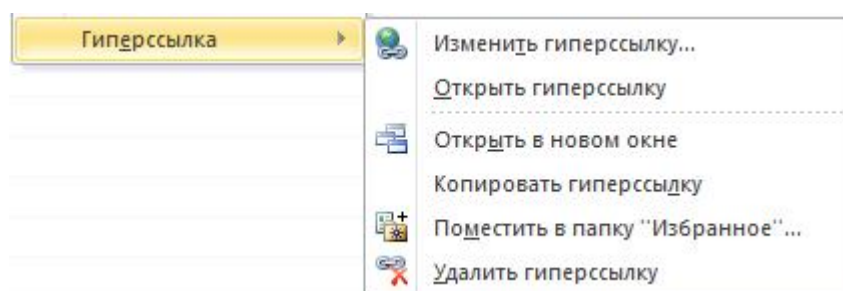


Рис. 3.1. Команды для настройки гиперссылок

**Мастер подстановок.** Этот режим не является самостоятельным типом. Его выбор запускает мастер, который позволяет создать список или назначить таблицу или запрос, из которого можно будет выбирать значения для поля. Тип данных устанавливается по значениям, выбранным в процессе работы мастера.

## 3.2. Структура базы данных

Прежде, чем создавать базу данных, нужно продумать её структуру, т.е. решить вопрос о том, какие таблицы с какими полями должны в нее входить.

Вернемся к примеру базы данных *Поставщики и Детали*, рассмотренному в гл.1. Там было установлено, что в этой базе данных должны быть три таблицы: *Поставщики*, *Детали* и *Поставки*. Назначим для полей этих таблиц подходящие типы данных.

Таблица <i>Поставщики</i>		
Имя поля	Тип поля	Назначение
<u>КодПоставщика</u>	Счетчик	Уникальный номер
<i>Поставщик</i>	Текстовый	Имя, не обязательно уникальное;
<i>Город</i>	Текстовый	Город, где расположен поставщик
<i>Рейтинг</i>	Числовой	Значение рейтинга

Таблица <i>Детали</i>		
Имя поля	Тип поля	Назначение
<u>КодДетали</u>	Счетчик	Уникальный номер
<i>Название</i>	Текстовый	Название детали
<i>Цвет</i>	Текстовый	Цвет
<i>Вес</i>	Числовой	Вес
<i>Эскиз</i>	Поле OLE	Изображение детали

Таблица <i>Поставки</i>		
Имя поля	Тип поля	Назначение
<u>КодПоставщика</u>	Числовой	Номер поставщика
<u>КодДетали</u>	Числовой	Номер детали
<u>Дата</u>	Дата/Время	Дата поставки
<i>Количество</i>	Числовой	Количество

При проектировании таблиц нужно обеспечить наличие *ключей*. Ключ состоит из одного или нескольких полей, которые однозначно идентифицируют каждую запись в таблице. В таблице *Поставщики* таким полем может служить *КодПоставщика*, в таблице *Детали* – *КодДетали*. В таблице *Поставки* ключом могут быть одновременно три поля: *КодПоставщика*, *КодДетали* и *Дата*, если один и тот же поставщик одну и ту же деталь в один и тот же день поставляет не более одного раза. Ключевые поля при описании структур таблиц принято подчеркивать.

### 3.3. Создание таблиц

Для работы с таблицами следует выбрать **Таблицы** из списка объектов (рис. 2.5). С таблицами можно работать в режиме **Конструктор** и в режиме **Таблица**, переключение между которыми можно выполнять с помощью кнопки **Режим** (рис. 3.2)

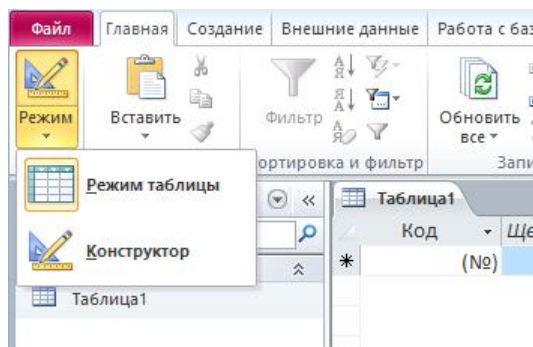


Рис. 3.2. Переключение режимов работы

Как видно, например, на рис. 2.5, при создании новой базы данных в нее добавлена одна таблица, названная по умолчанию Таблица1. Выберем режим **Конструктор**. Появится окно для ввода имени таблицы (рис.3.3). Назовем таблицу *Поставщики*.

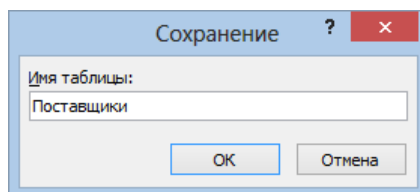


Рис. 3.3. Ввод имени таблицы


После задания имени создаваемой таблицы откроется бланк конструктора таблиц (рис.3.4).


### 3.3.1. Конструктор таблиц

Режим конструктора позволяет полностью контролировать процесс создания таблицы.

На рис.3.4. показан бланк конструктора таблиц в процессе создания структуры таблицы *Поставщики*.

Новое поле добавляется путем ввода его имени в свободную ячейку столбца **Имя поля**.

Поле можно удалить, для чего его нужно выделить и нажать кнопку  **Удалить строки**. Нельзя удалить поле, которое используется для связи с другой таблицей (см. п. 3.10). Для удаления такого поля сначала нужно удалить связь.

Для вставки поля между существующими полями нужно выделить нижележащее поле и выполнить команду **Вставить строки** .

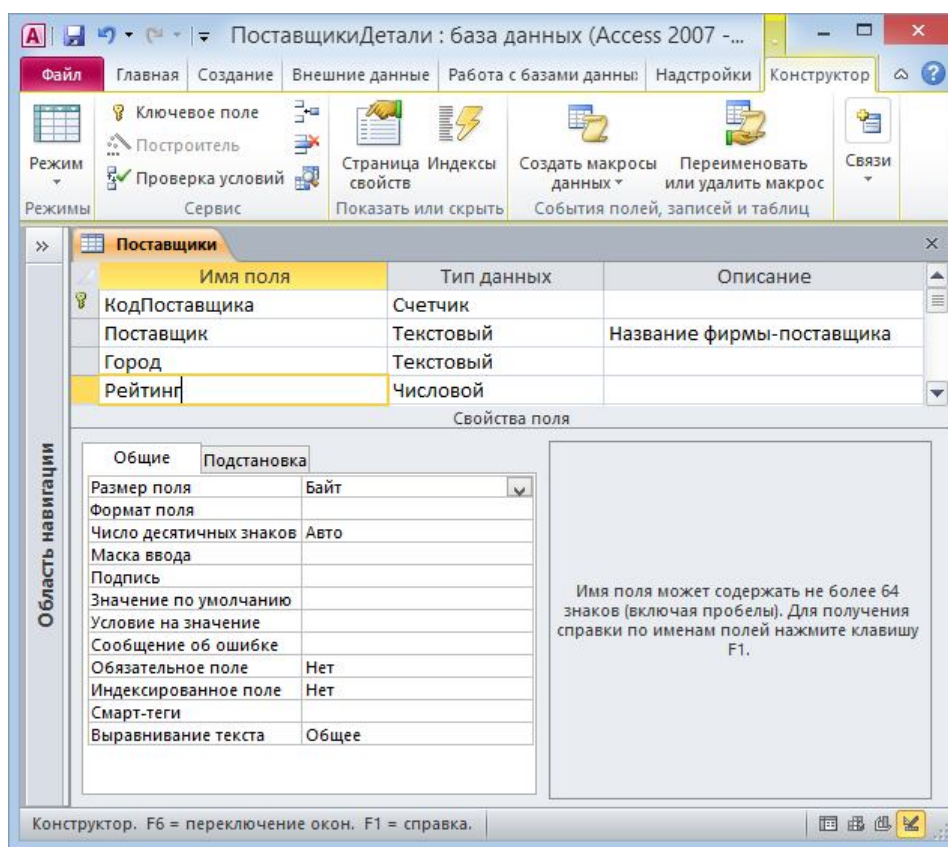



Рис. 3.4. Конструктор таблиц

### 3.3.2. Задание ключевого поля

*Ключевое поле* позволяет однозначно идентифицировать запись, то есть отличить одну запись от другой. По умолчанию при создании таблицы автоматически создается поле *Код*, имеющее тип *Счетчик*, и оно назначается ключевым, о чем говорит изображение ключа слева. При создании таблицы *Поставщики* заменим название поля *Код* на *КодПоставщика*.

Щелкая по кнопке  **Ключевое поле** на закладке **Конструктор** можно сделать поле ключевым или отменить это назначение (рис.3.5).

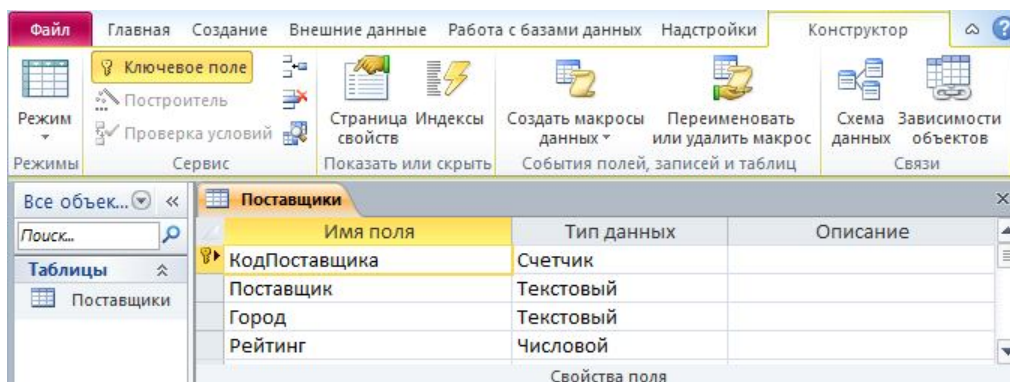


Рис. 3.5. Назначение ключевого поля

При вводе данных в ключевые поля осуществляется проверка, чтобы не было совпадения значений в различных записях. По ключевому полю производится автоматическая индексация.


### 3.3.3. Свойства полей

Свойства полей устанавливаются с помощью вкладок **Общие** и **Подстановка** в нижней половине окна конструктора таблиц (рис.3.4.). Рассмотрим эти свойства.

**Размер поля.** Для текстового поля это максимальное число символов (до 255). По умолчанию устанавливается размер 255 символов. При выборе размера надо учитывать, что в поле размером 20 символов нельзя будет ввести текст длиной 30 символов. С другой стороны, установка слишком большой длины может привести к бесполезному увеличению размера файла базы данных, если значения, хранимые в поле будут существенно меньше его установленной длины. Для числовых полей размер выбирается из списка возможных числовых типов данных, которые перечислены в табл. 3.1.

**Формат** устанавливает вид данных на экране, например, для типа **Дата/Время** можно выбрать представление даты вида *30 июня 1999 г.* или *6/30/99*.

**Число десятичных знаков** определяет количество цифр после десятичной точки для дробных чисел. Это значение влияет только на вид представления числовых величин, а не их способ хранения в памяти.

**Маска ввода** позволяет задать шаблон для ввода, который дает некоторую гарантию правильности ввода данных, она применяется для полей типа **Дата/Время** и текстовых. Маска ввода выводит на экран символы – местоименители, показывает, сколько символов нужно ввести, включает разделительные символы (дефис, скобки). Например, для даты маска ввода может выглядеть так: *--.---.---*. Данная маска кодируется набором символов *99/99/00*. Заполнитель 9 означает, что разрешается вводить только цифры, причём её ввод не является обязательным, заполнитель 0 требует обязательного ввода цифры. При выборе свойства **Маска ввода** справа появляется кнопка , по которой запускается мастер, помогающий создать маску.

**Подпись** используется в качестве заголовка столбца в режиме таблицы.

**Значение по умолчанию** позволяет автоматически вводить в поле какое-либо значение.

Свойство **Обязательное поле** имеет два значения **Да** и **Нет**. Если установить значение **Да**, Access потребует обязательного ввода какого-либо значения в данное поле.

Свойство **Индексированное поле** определяет, будет или нет проводиться индексация по данному полю. Индексация состоит в создании списка номеров записей, упорядоченных в соответствии со значениями поля. Наличие индекса ускоряет операции поиска и сортировки, но требует дополнительного места на диске.

### 3.4. Добавление таблиц в режиме Конструктор

Для создания новой таблицы перейдем на закладку **Создание** (рис.3.6). Заметим, что создание любых объектов базы данных выполняется с помощью инструментов, собранных на этой закладке. Нажмем кнопку **Конструктор таблиц**.

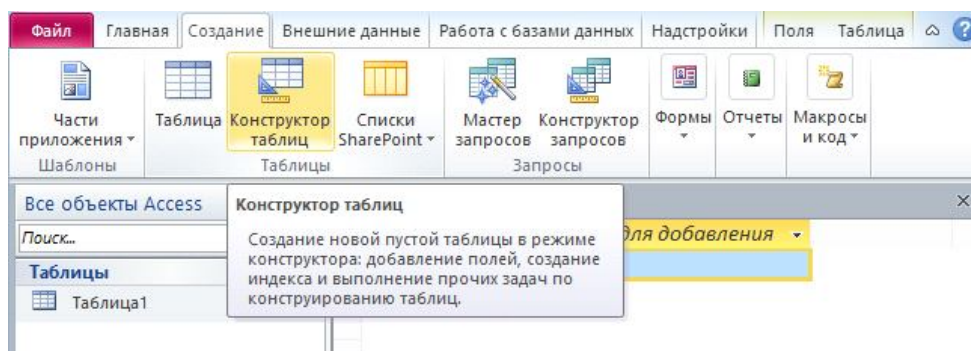


Рис. 3.6. Закладка **Создание**. Создание таблицы

Создадим таблицу *Детали* (рис.3.7).

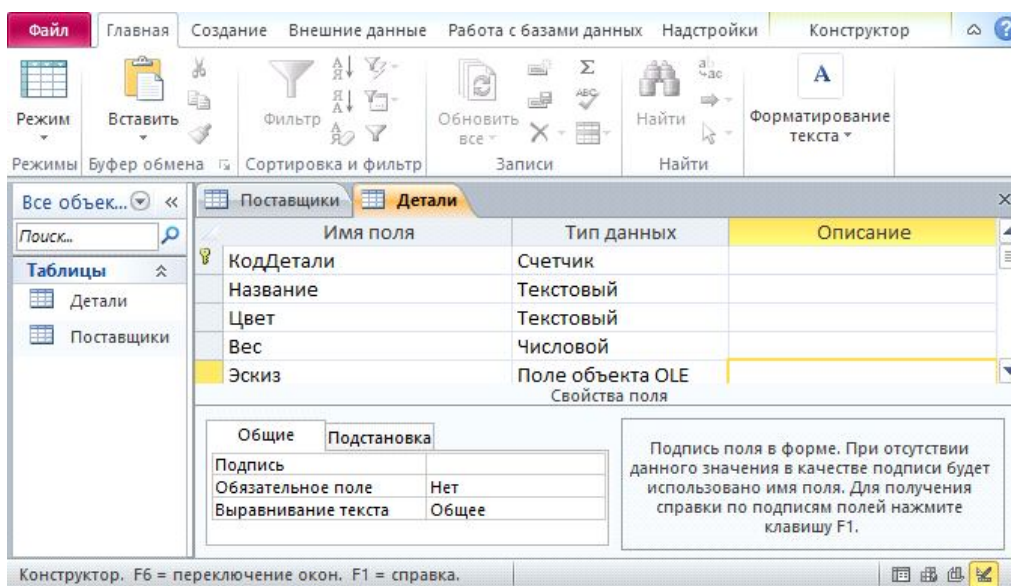


Рис. 3.7. Создание таблицы *Детали*

#### 3.4.1. Мастер подстановок

Сделать более простым ввод данных в таблицу позволяет *подстановка*. Например, обеспечим возможность ввода значения поля *Цвет* таблицы *Детали* путем выбора нужного цвета из списка. Перейдем в режим конструктора для таблицы *Детали* и выберем для типа данных поля *Цвет* значение **Мастер подстановок...** (рис.3.8).

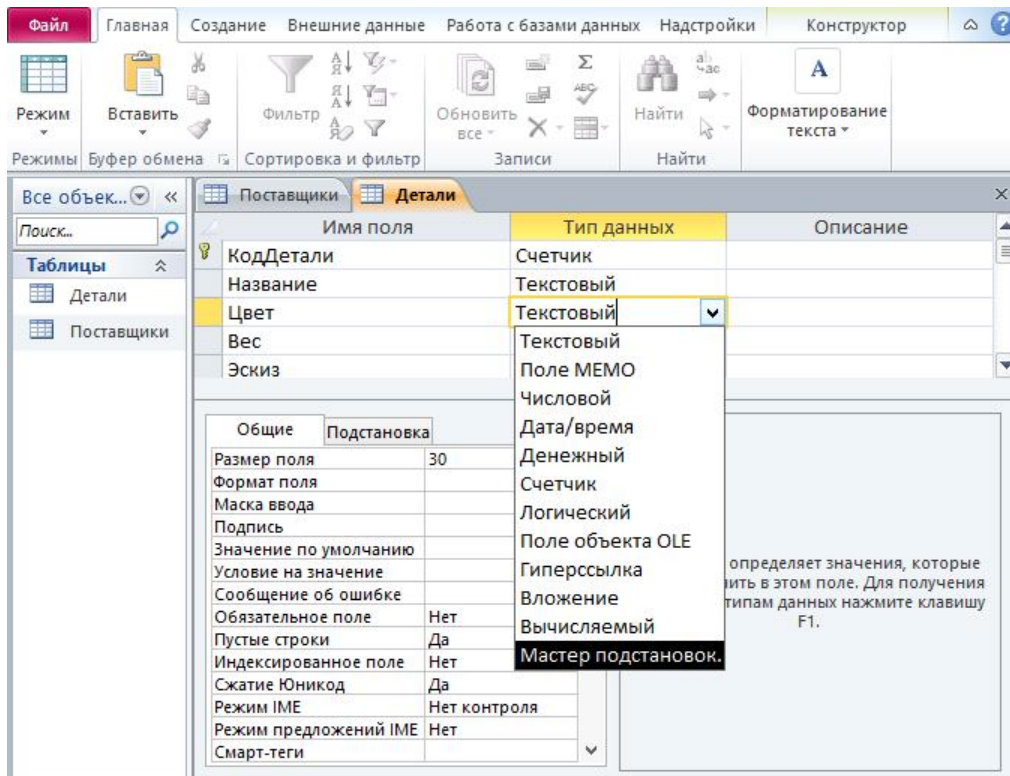


Рис. 3.8. Выбор режима **Мастер подстановок**

При использовании режима **Мастер подстановок** предьявляется несколько диалоговых окон. Первый диалог мастера показан на рис.3.9.

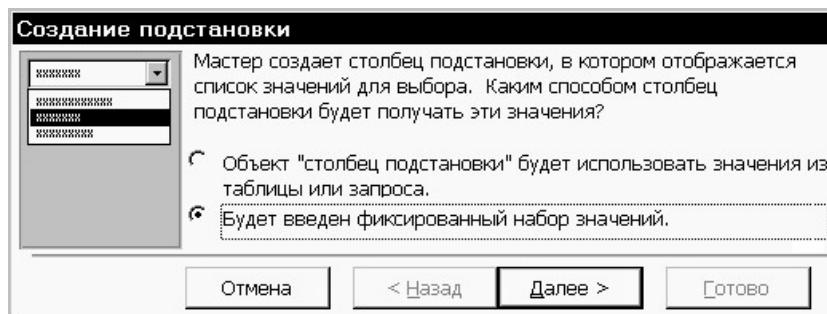


Рис. 3.9. Первый диалог мастера подстановок. Выбор источника значений

Выберем способ **Будет введен фиксированный набор значений**. На втором диалоге (рис.3.10), вводим значения цветов в столбец.



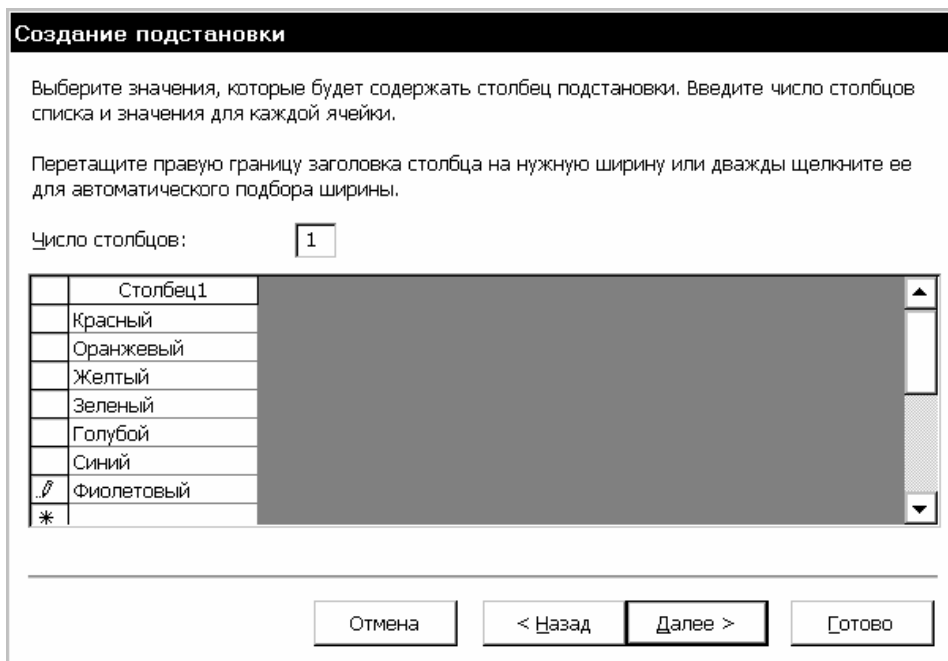


Рис. 3.10. Второй диалог мастера подстановок. Ввод возможных значений  
 На последнем диалоге задаем название столбца подстановки (рис.3.11).

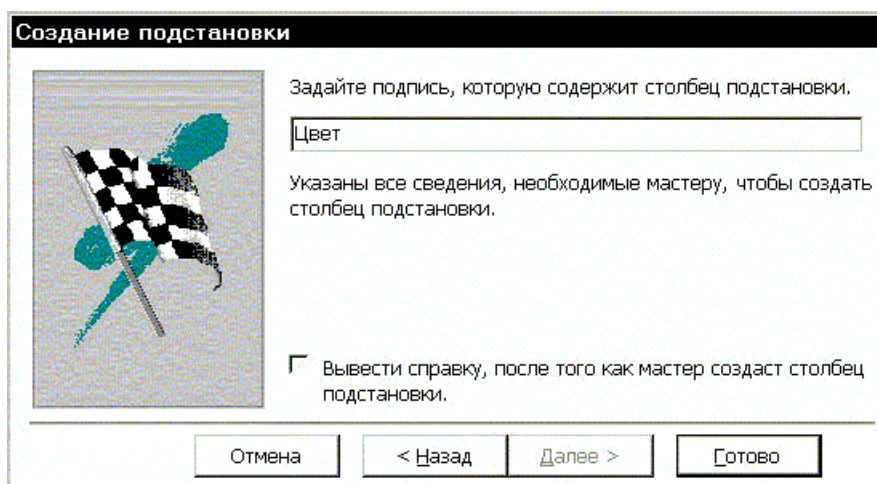


Рис. 3.11. Третий диалог мастера подстановок. Ввод подписи

Таблица с использованием столбца подстановки показана на рис.3.12. Надпись, введенная на третьем диалоге (рис.3.11), совпадает с названием поля таблицы, поэтому название столбца не изменилось.

	КодДетали	Название	Цвет	Вес	Эскиз
▶	1	Гайка	Красный	1	hoto Editor 3.0
	2	Болт	Красный	2	hoto Editor 3.0
	3	Винт	Оранжевый	3	hoto Editor 3.0
	4	Винт	Желтый	1	
	5	Муфта	Зеленый	2	
	6	Шпилька	Голубой	1	
*		(Счетчик)	Синий	0	
			Фиолетовый		

Запись:

Рис. 3.12. Использование столбца подстановок в режиме таблицы

Если создать форму для таблицы *Детали* после добавления столбца подстановки для поля *Цвет*, то автоматически на форме будет создан список для выбора значения. Пример такой формы показан на рис.3.13.

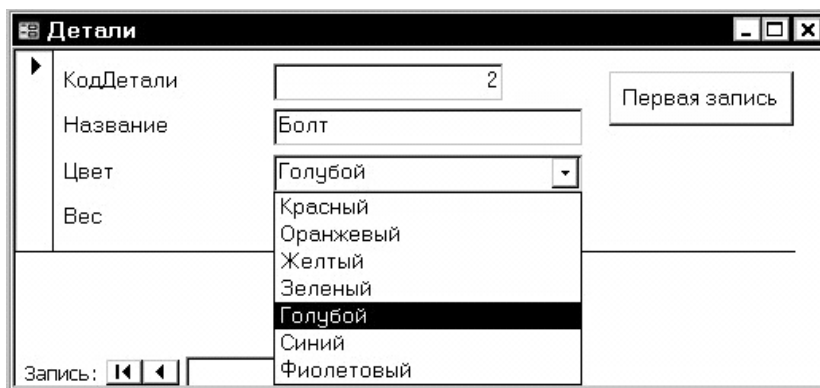


Рис. 3.13. Использование столбца подстановок в форме

Столбец подстановки можно связать с таблицей или запросом, задав соответствующий режим на первом диалоге мастера (рис. 3.9). Это более удобно, так как при изменении значений в таблице, связанной со столбцом подстановки, автоматически изменяются предлагаемые для выбора значения.

### 3.5. Создание таблиц в режиме Таблица

Таблицы можно создавать в режиме **Таблица** (рис.3.14).

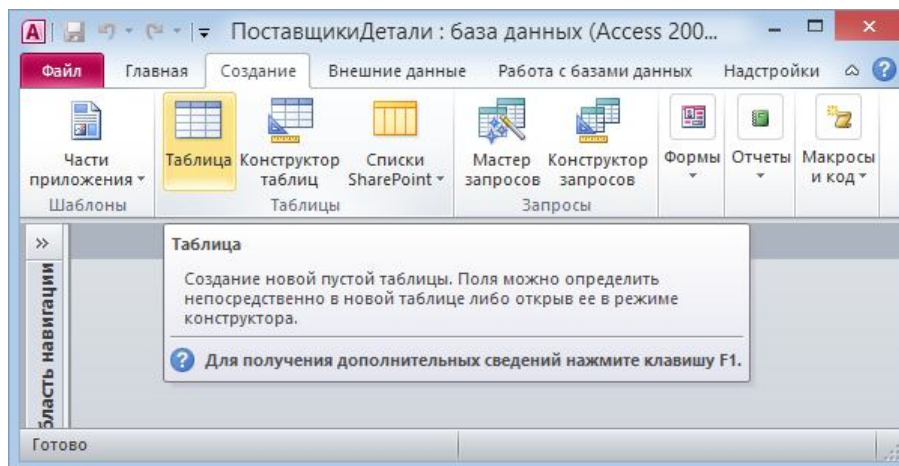


Рис. 3.14. Выбор режима **Таблица** для создания таблицы

На экране появиться пустая таблица (рис.3.15), в ячейки которой можно вводить значения. Access по введенному значению определит тип данных и включит его в структуру таблицы. При добавлении нового поля можно выбирать тип данных для этого поля (рис.3.15).

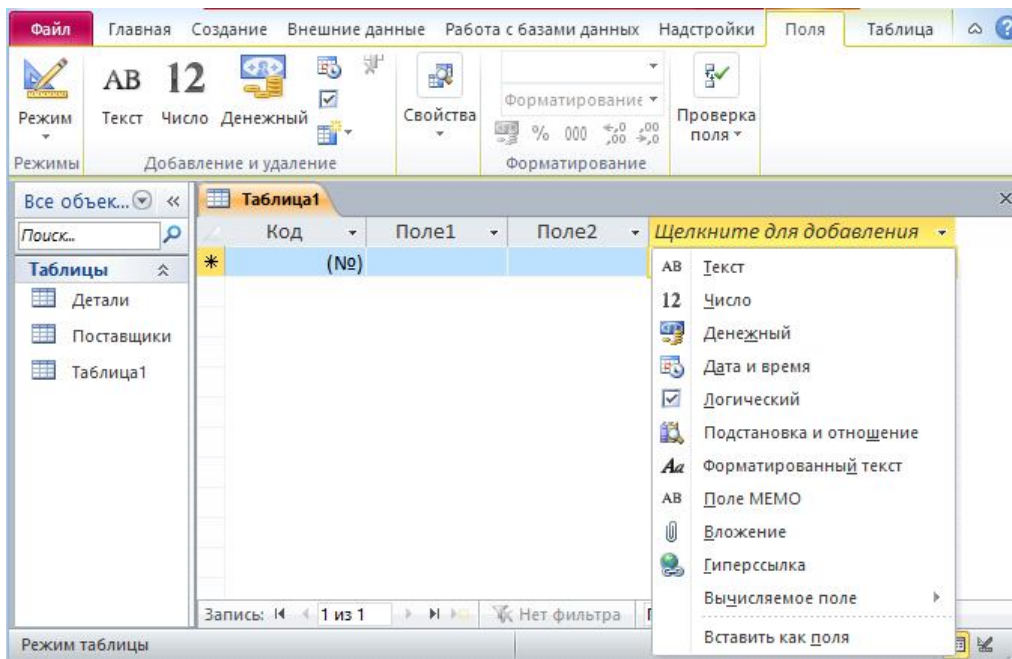


Рис. 3.15. Выбор типа данных для поля

В соответствии со структурой, описанной в п. 3.2, создадим таблицу *Поставки*. Для переименования столбцов таблицы используем команду **Переименовать поле** контекстного меню, вызываемого щелчком правой кнопки мыши (рис.3.16).

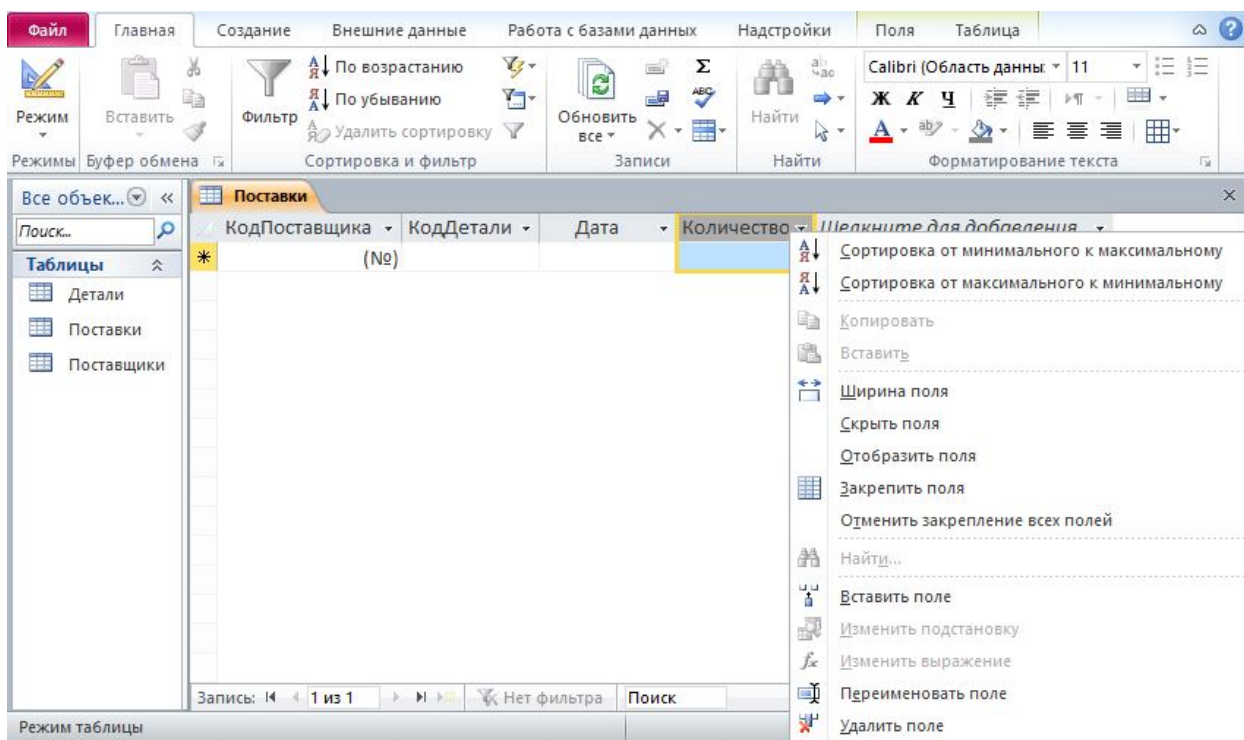


Рис. 3.16. Изменение имени поля таблицы

Переключимся на режим конструктора (рис.3.17), при этом сохраним таблицу под именем *Поставки*.

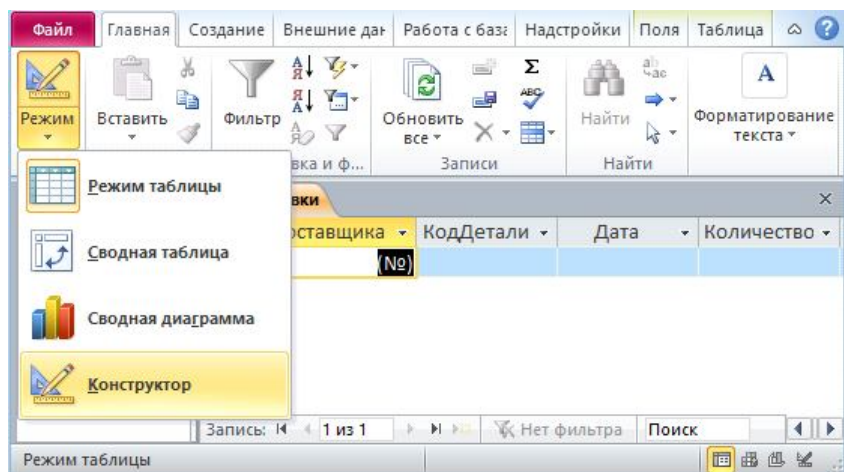


Рис. 3.17. Переход в режим конструктора

В режиме **Конструктор** выделим три поля *КодПоставщика*, *КодДетали* и *Дата* и сделаем группу из этих трех полей ключем (рис.3.18). Назначение группы из нескольких полей означает, что никакие две строки таблицы не должны иметь одинаковые комбинации значений этих трех полей. Для таблицы *Поставки* это значит, что один и тот же поставщик одну и ту же деталь в один в тот же день поставляет единственный раз, не бывает двух или более поставок от одного поставщика одной детали в один день.

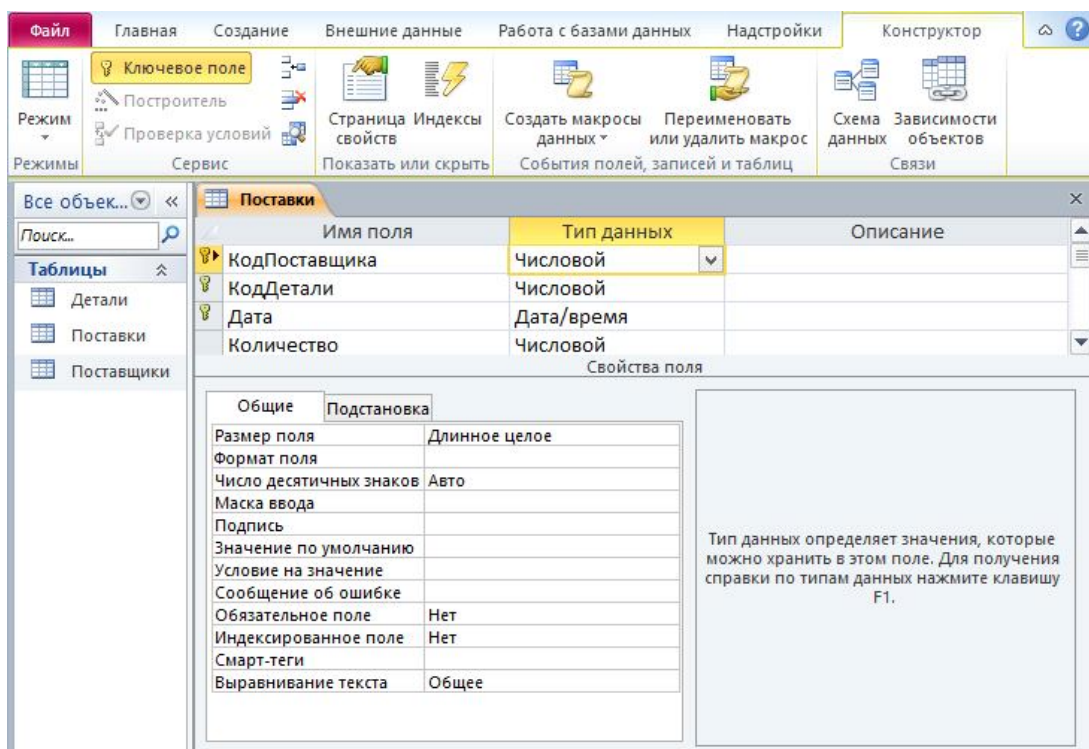


Рис. 3.18. Ключ из трех полей

### 3.5.1. Создание маски ввода

Для поля *Дата* таблицы *Поставки* выберем свойство **Маска ввода** (рис.3.19). Выберем, далее, одну из предустановленных масок ввода - **Краткий формат даты**. В поле **Проба** можно попробовать ввести дату.

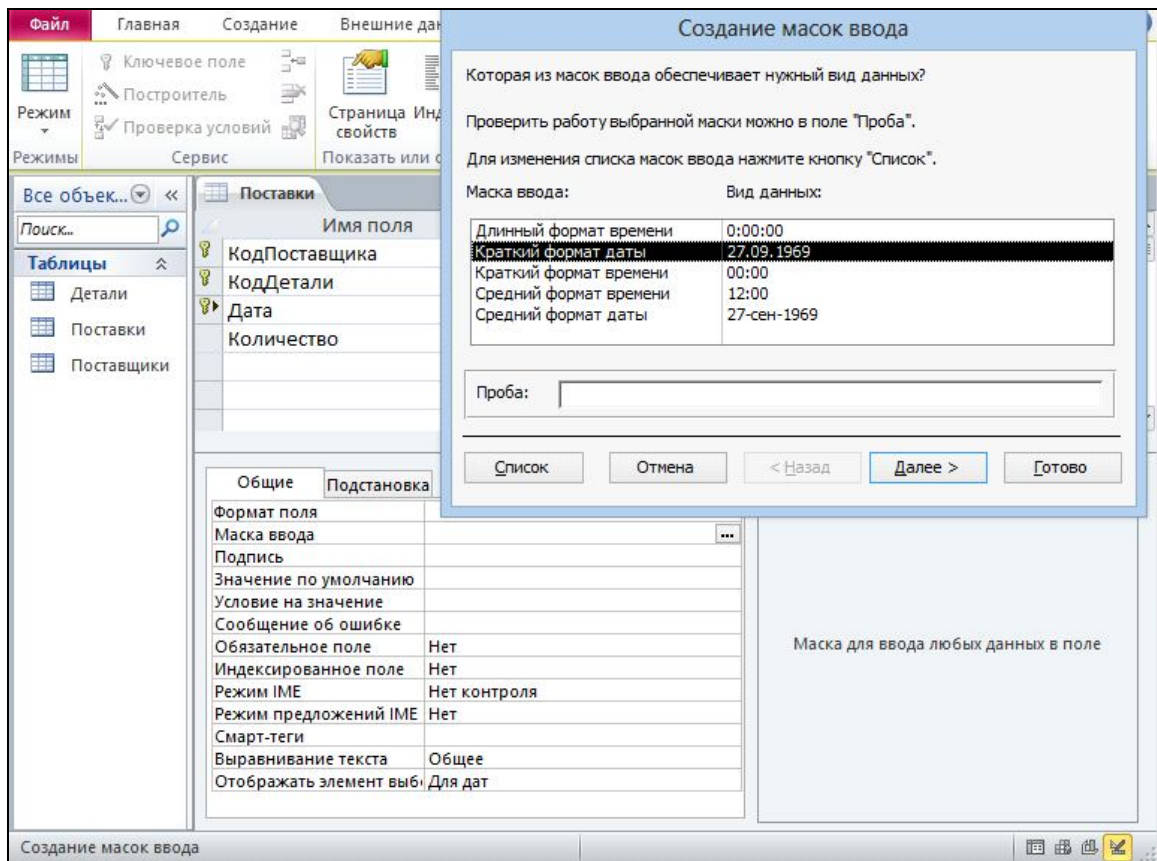


Рис. 3.19. Создание маски ввода

Нажав кнопку **Далее** перейдем в окно (рис.3.20), где можно сделать самостоятельные настройки маски ввода.

В поле **Маска ввода** приведены символы, кодирующие маску ввода. Цифра ноль (0) разрешает ввод в соответствующей позиции только цифры. То есть восемь нулей разрешают ввод восьми цифр для даты.

В поле **Заполнитель** указывается символ, который будет отображаться в поле таблицы при его заполнении. Пример поля таблицы показан в поле **Проба**.

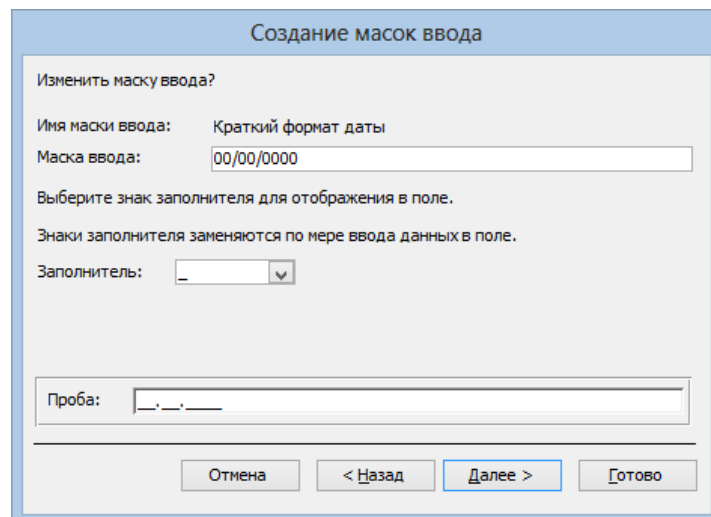


Рис. 3.20. Настройка маски ввода

## 3.6. Связи между таблицами

Если между таблицами установлены связи, возможен контроль за данными, хранящимися в базе.

### 3.6.1. Установление связей между таблицами

Связь возможна между таблицами, у которых имеются поля с *одинаковыми значениями*. Установка связей между таблицами позволяет обеспечить *целостность данных*.

Для установления связей нужно перейти на закладку **Работа с базами данных** и выбрать инструмент **Схема данных** (рис.3.21).

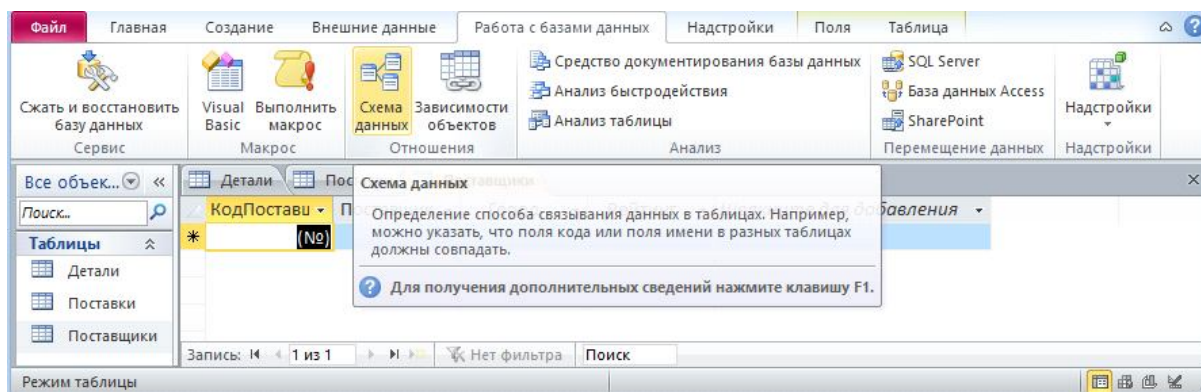


Рис. 3.21. Переход к установке связей между таблицами

Откроется окно **Добавление таблицы** (рис.3.22), в котором надо выбрать все три таблицы нашей базы данных.

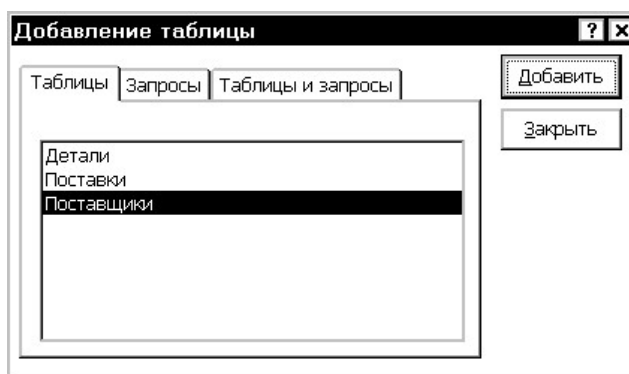


Рис. 3.22. Добавление таблиц в схему данных

После закрытия окна **Добавление таблицы** становится активным окно **Схема данных** (рис. 3.23).

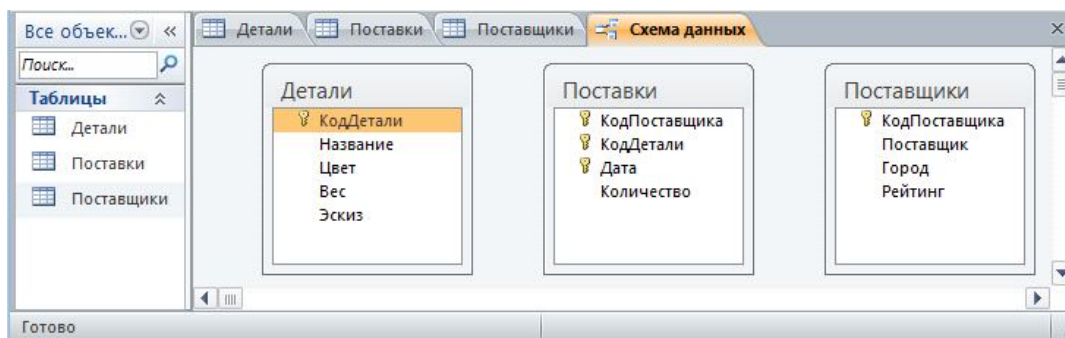


Рис. 3.23. Окно **Схема данных**

При установке связи между таблицами возможно появления информационного окна (рис.3.24)

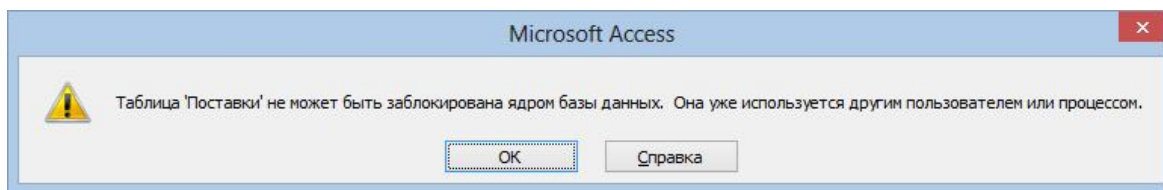


Рис. 3.24. Информация об использовании таблицы

Так бывает, когда в устанавливаемой связи участвует открытая таблица. В ситуации, показанной на рис. 3.23 открыты все три таблицы базы данных, о чем говорят видимые ярлыки закладок с названиями таблиц. Для закрытия таблицы нужно щелкнуть правой кнопкой по ее ярлыку и выбрать команду **Закреть** (рис.3.25).

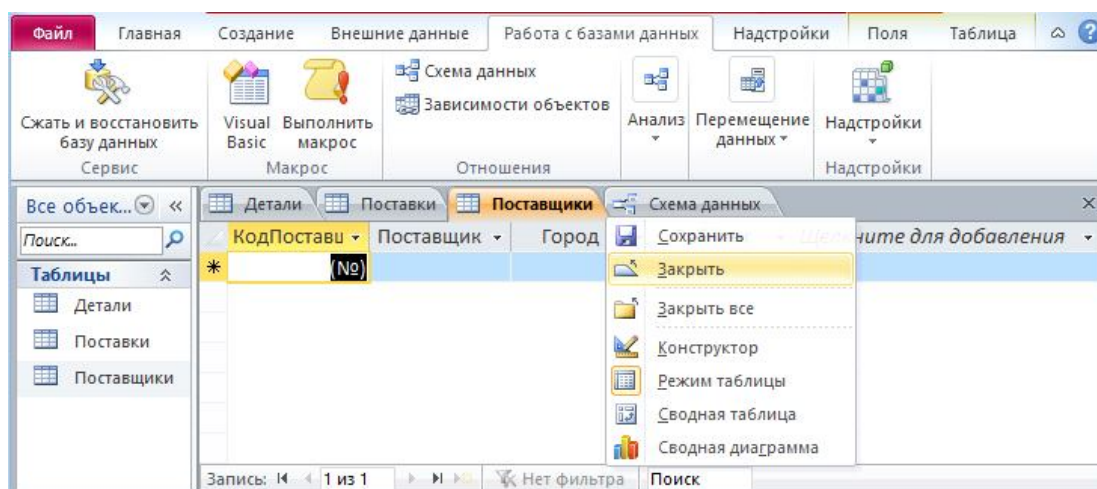


Рис. 3.25. Закрытие таблицы

Связи между таблицами устанавливаются путём перетаскивания мышью поля из одной таблицы на соответствующее поле другой таблицы. При установке связи появляется окно (рис.3.26), в котором можно задать параметры создаваемой связи.

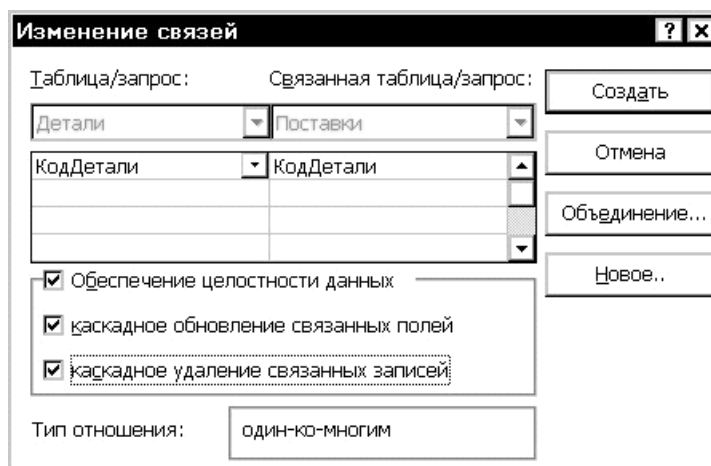


Рис. 3.26. Настройка связи между таблицами

Как правило, одно из полей в связанных таблицах имеет *уникальные значения*. Это может быть ключевое поле или поле, для которого свойство **Индексированное поле** имеет значение **Да (совпадения не допускаются)**. Таблица, содержащая уникальное поле, называется *базовой*, а вторая таблица – *подчинённой*. В рассматриваемой связи базовой является таблица *Детали*, а подчиненной - таблица *Поставки*. Поскольку одной записи в базовой таблице мо-

жет соответствовать нескольким записей в подчинённой (одна и та же деталь может постав- ляться много раз), подобная связь называется «*один-ко-многим*». Как видно из рис. 3.26, связь *Детали*→*Поставки* имеет вид «один-ко-многим».

Поле подчиненной таблицы, которое связано с ключевым полем базовой таблицы, называется *внешним ключом*. Например, внешним ключом в таблице *Поставки* являются поле *КодДетали*.

Если в окне **Изменение связей** установить флажок **Обеспечение целостности данных**, Access обеспечит проверку соответствия между связанными полями при вводе или измене- нии данных. Ограничения целостности данных ограничивают возможность произвольного изменения и удаления данных:

- Нельзя будет ввести значение в поле подчинённой таблицы, если такого значения нет в связанном поле базовой таблицы. Нельзя также будет изменить значение в поле базовой таблицы, если с ним связано значение в подчинённой. Если же в диалоге **Изменение свя- зей** установить флажок **Каскадное обновление связанных полей**, то будет обеспечено ав- томатическое обновление записей в поле подчиненной таблицы при изменении значения в соответствующем ключевом поле базовой таблицы.
- Нельзя удалить запись из базовой таблицы, если в подчинённой таблице имеются соот- ветствующие записи. Однако в диалоговом окне **Изменение связей** можно установить флажок **Каскадное удаление связанных записей**, что приведет к удалению всех записей подчиненной таблицы при удалении соответствующей записи из базовой таблицы.

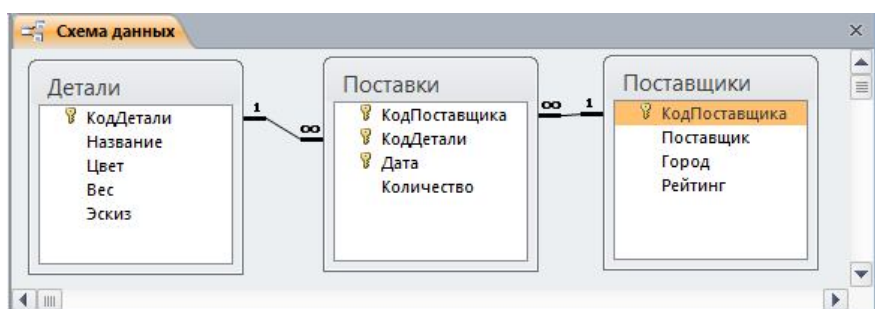


Рис. 3.27. Связи между таблицами

Установленные связи между таблицами можно изменять или удалять. Для редактиро- вания связи ее нужно выделить щелчком мыши, затем нажать правую кнопку мыши и вы- брать нужную команду контекстного меню (рис.3.28).

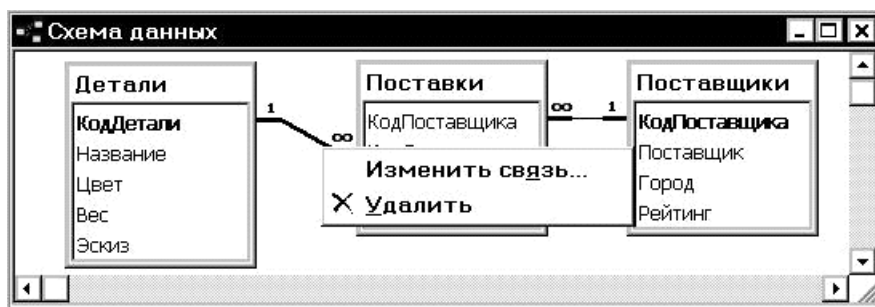


Рис. 3.28. Редактирование связей

Выделенная связь показывается жирной линией. Следует иметь в виду, что нельзя удалить связь, если какая-либо из связанных таблиц открыта в режиме таблицы или формы. Перед удалением связи соответствующие таблицы или формы нужно закрыть.

Перейти к редактированию существующей связи можно также двойным щелчком мыши по линии, представляющей связь.

В окне **Схема данных** одна из таблиц всегда активна. Активная таблица удаляется из данного окна (не из схемы данных!) командой контекстного меню **Скрыть таблицу** (рис.3.29),



при этом все связи скрытой таблицы сохраняются. Скрытие таблиц полезно, когда в схему данных входит много таблиц, и нужно разобраться в связях между ними.

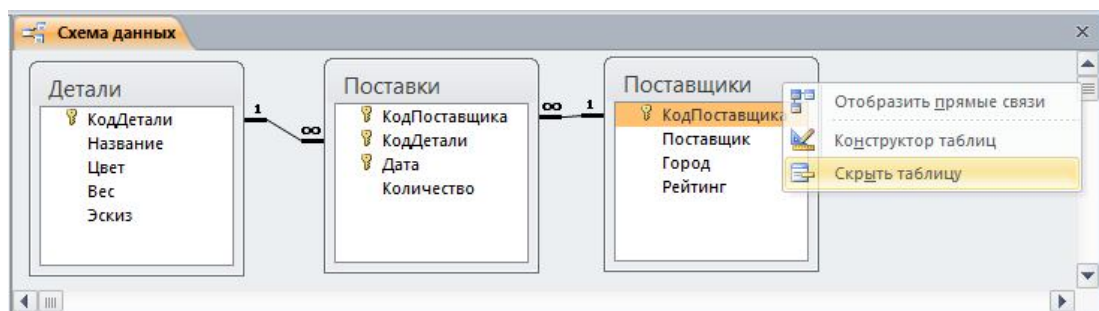


Рис. 3.29. Скрытие таблицы, входящей в схему данных

Если какая-либо таблица ошибочно повторно включена в схему данных, нужно удалить связи других таблиц с ней и выполнить команду **Скрыть таблицу**. После этого таблица будет удалена из схемы данных.

Одну и ту же таблицу можно повторно включить в схему данных с помощью окна **Добавление таблицы**, выводимого командой **Добавить таблицу** (рис.3.30). Чтобы увидеть все таблицы и связи, входящие в схему данных, нужно выполнить команду **Отобразить все**.

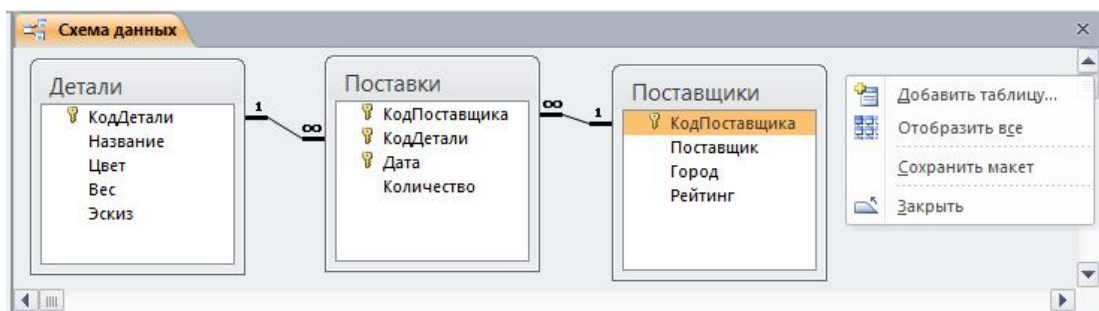


Рис. 3.30. Контекстное меню для работы со схемой данных


## 3.7. Ввод и просмотр данных в режиме таблицы

Таблица базы данных открывается в режиме таблицы, если ее выбрать в списке объектов базы данных и дважды щелкнуть мышью по названию таблицы.

### 3.7.1. Ввод и редактирование данных

В режиме таблицы информация расположена по строкам и столбцам. Информация в поля записи вводится путем занесения данных в ячейки таблицы. При этом можно использовать обычные приемы редактирования текста.

Новая запись добавляется в конец таблицы. Перемещаться между полями можно клавишей **Tab** (в прямом направлении) или **Shift+ Tab** (в обратном).

Если полю присвоен столбец подстановки, справа от поля отображается кнопка , раскрывающая список выбора (рис.3.12). После ввода данных в поле, для которого было задано значение свойства **Формат**, данные в поле форматируются.

Запись можно удалить, выделив её и выполнив, затем, команду **Правка, Удалить запись**. Удалённую запись восстановить *нельзя*.

В заголовках столбцов отображается значение свойства **Подпись**, если это свойство не является пустым. Если свойство **Подпись** не задано, в качестве заголовков столбцов используются имена полей.

В режиме таблицы имя поля можно изменить, дважды щёлкнув по названию столбца мышью. В этом случае свойство **Подпись** станет пустым. Другой способ – инструмент **Имя и подпись** с закладки **Поле** (рис.3.31)

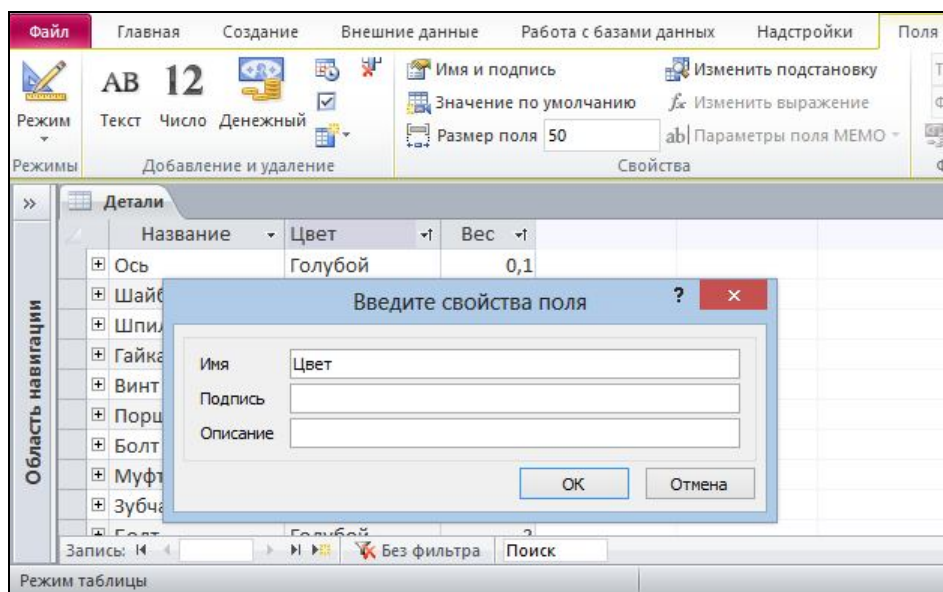


Рис. 3.31. Настройка имени и подписи поля

Чтобы добавить столбец, нужно выполнить команду контекстного меню **Вставить поле**, а для удаления выделенного столбца команду **Удалить поле**.

Access автоматически сохраняет все изменения, влияющие на структуру таблиц.

### 3.7.2. Ввод графической информации в поля таблицы

Если в полях таблицы предполагается хранить графические изображения, целесообразно подготовить их заранее. Это можно сделать сканированием фотографий, других изображений, можно создать изображение в каком-либо графическом редакторе или найти в Интернете. Для вставки готового изображения в поля таблицы, нужно открыть ее в режиме таблицы и перейти к полю для изображения. Откроем таблицу *Детали* и выберем поле *Эскиз* у какой-либо записи. Выполним команду **Вставить объект** (рис.3.32).

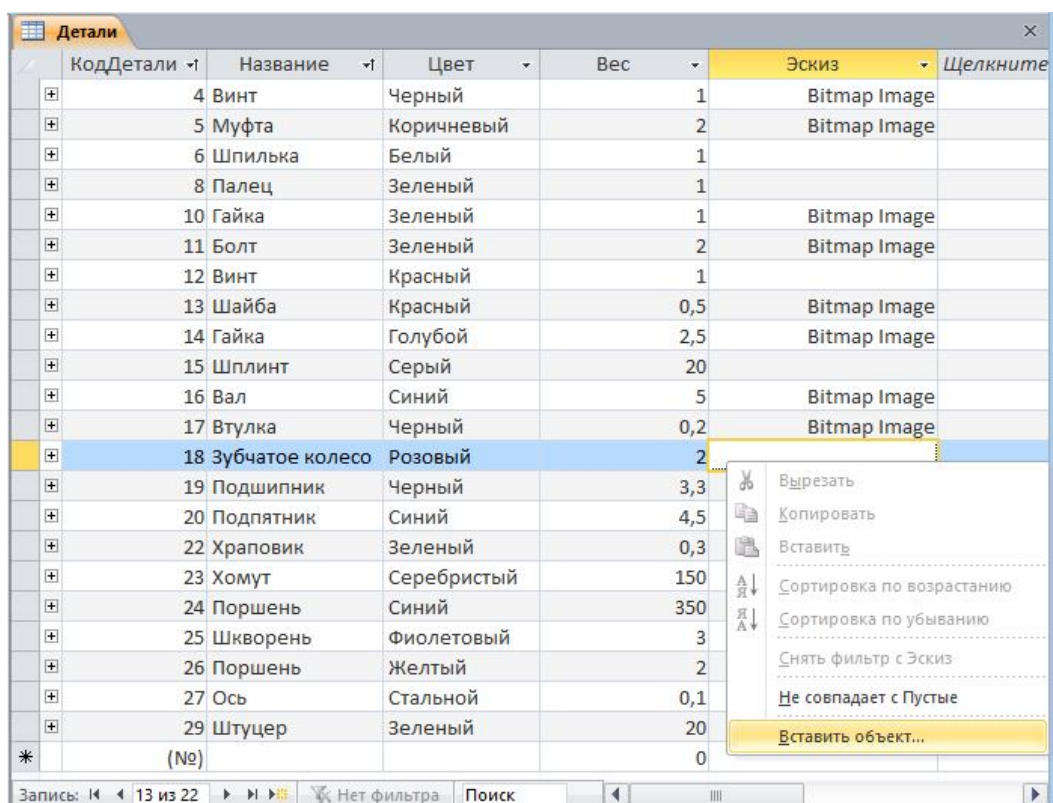


Рис. 3.32. Вставка рисунка в поле таблицы

На диалоге **Вставка объекта** выберем режим **Создать из файла** (рис.3.33).

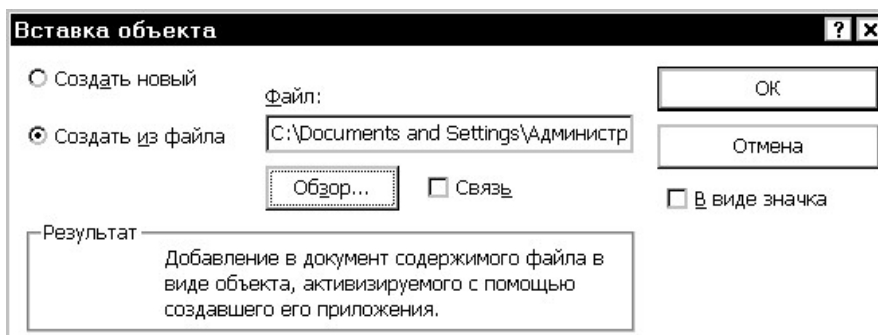


Рис. 3.33. Диалог вставки объекта в поле таблицы

Нажмем кнопку **Обзор** и перейдем на диалог открытия файла, с помощью которого отыщем файл с изображением (рис.3.34). На диалоге **Обзор** удобно установить режим **Промосмотр** с помощью кнопки **Вид**, чтобы видеть рисунки, содержащиеся в файлах. Когда рисунок найден, нажмем кнопку **ОК**, и рисунок будет помещен в таблицу базы данных. Следует иметь в виду, что файлы рисунков имеют, как правило, большой размер, поэтому файл базы данных, содержащей рисунки, может иметь большой объем.

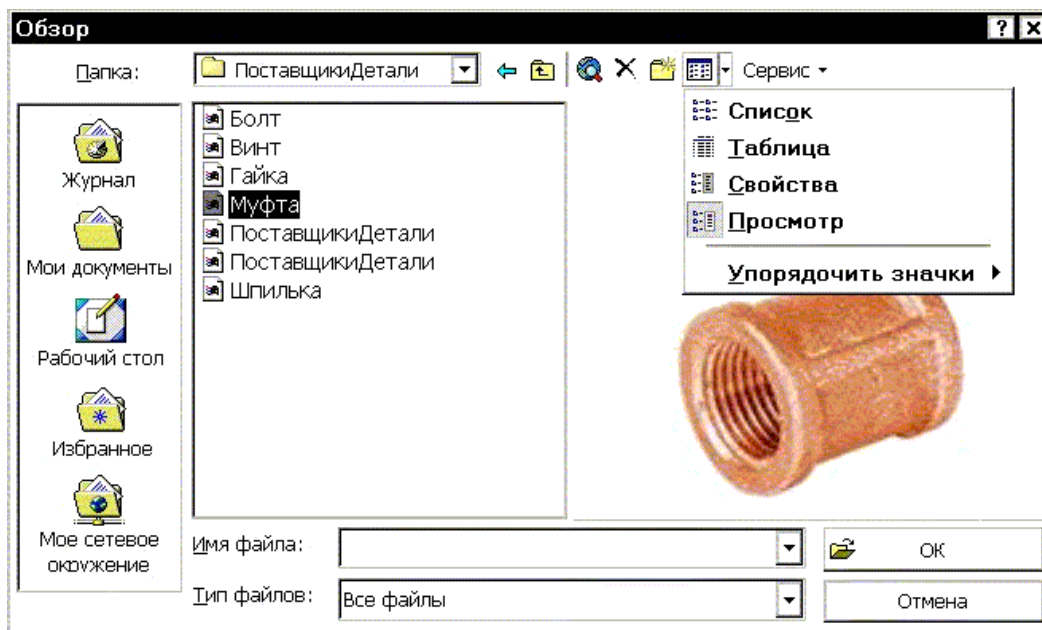


Рис. 3.34. Выбор рисунка для помещения в базу данных

В дальнейшем рисунки можно отображать на формах и в отчетах.

### 3.7.3. Подстановка данных

При заполнении таблицы *Поставки* в поля *КодПоставщика* и *КодДетали* нужно вводить числовые значения кодов, что неудобно, так как числовые коды трудно запоминать. Есть возможность отображать вместе с числовыми кодами соответствующие названия поставщиков и деталей.

Откроем таблицу *Поставки* в режиме конструктора, в нижней части конструктора перейдем на закладку **Подстановка** и зададим следующие свойства поля *КодПоставщика* (рис.3.35):

- |                         |                      |
|-------------------------|----------------------|
| Тип элемента управления | - Поле со списком    |
| Тип источника строк     | - Таблица или запрос |
| Источник строк          | - Поставщики         |
| Присоединенный столбец  | - 1                  |
| Число столбцов          | - 2                  |
| Ширина столбцов         | - 0                  |
| Ограничиться списком    | - Да                 |

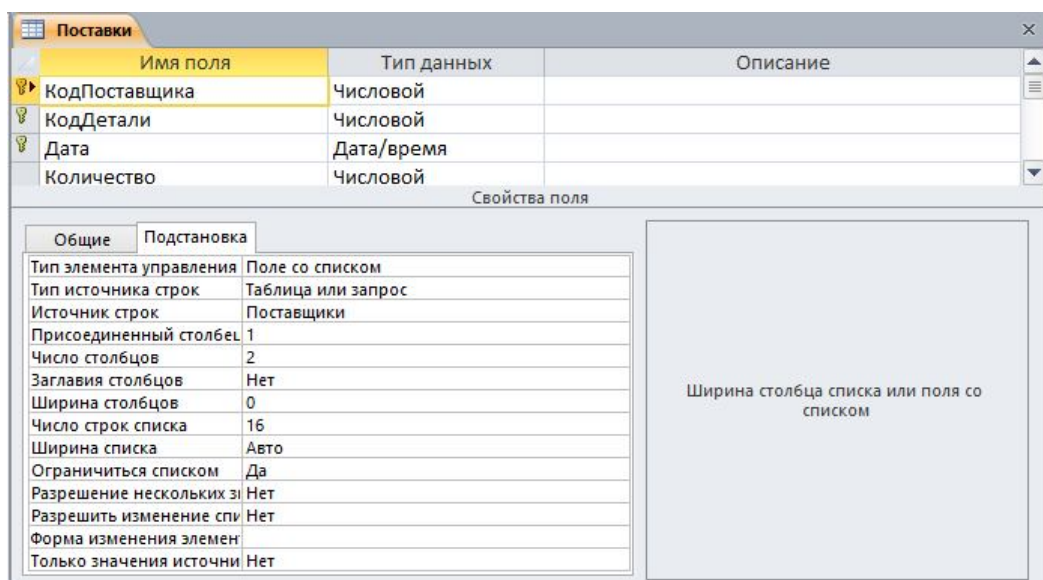


Рис. 3.35. Настройка подстановки

По умолчанию в качестве типа элемента управления устанавливается значение **Поле**. При выборе **Поля со списком** появляется возможность отображать целый список значений с возможностью выбора. Значение 2 параметра **Число столбцов** означает, что отбираются два столбца из таблицы *Поставщики*, указанной в качестве источника данных. Установка параметра **Ширина столбцов** в 0 скрывает ключевое поля *КодПоставщика*.

Аналогично настроим подстановку для поля *КодДетали*.

После настройки подстановок для полей *КодПоставщика* и *КодДетали* таблица *Поставки* в режиме таблица будет иметь вид, показанный на рис.3.36. Хотя в полях *КодПоставщика* и *КодДетали* хранятся числовые коды поставщика и поставленной им детали, благодаря подстановке отображаются названия поставщиков и деталей, что гораздо удобней.

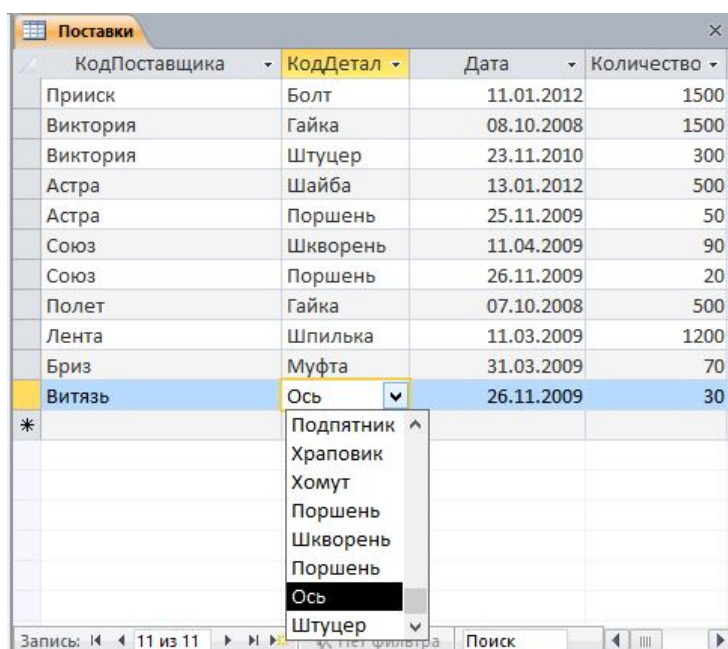


Рис. 3.36. Использование подстановки

### 3.7.4. Управление видом информации

Имеется возможность изменять ширину столбцов и высоту строк путём перетаскивания границы между ними курсором мыши, форма которого должна приобрести вид двуна-

правленной стрелки: ←, ↓. Нужная ширина столбца, соответствующая ширине данных, расположенных в столбце, устанавливается автоматически после двойного щелчка по его правой границе. Для управления шириной столбца можно также воспользоваться командой **Ширина поля** контекстного меню (рис.3.37).

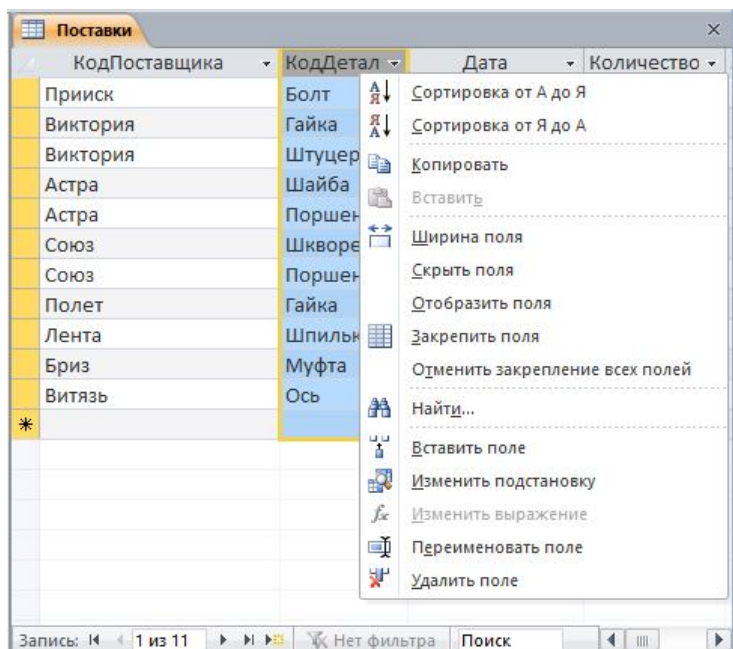


Рис. 3.37. Команды для работы со столбцами таблицы

Порядок столбцов можно изменять, для этого столбец нужно выделить щелчком мыши, затем подвести курсор к выделенному столбцу, нажать левую кнопку мыши и перетащить столбец в желаемое место. Порядок отображения столбцов в режиме таблицы не влияет на порядок хранения полей в таблицах.

В режиме таблицы можно отображать только часть столбцов таблицы. Для выбора отображаемых столбцов нужно выполнить команду **Отобразить поля** из контекстного меню (рис. 3.37) и отметить в появившемся окне нужные (рис.3.38).

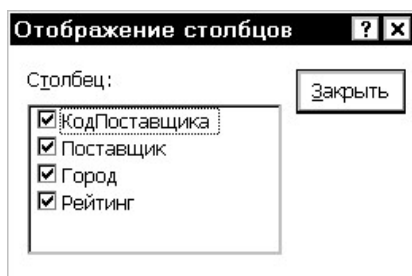


Рис. 3.38. Отбор столбцов для отображения в таблице

### 3.7.5. Подтаблицы

Если открыта главная таблица, которая имеет связь с подчинённой, то автоматически создаётся *подтаблица*, которую можно открыть, щелкнув по знаку «плюс» (+) (рис.3.39). При этом раскрывается часть подчинённой таблицы, содержащая записи, связанные с конкретной записью базовой таблицы.

КодПоставки	Поставщик	Город	Рейтинг	Щелкни												
1	Прииск	Н.Новгород	90													
2	Макрос	Н.Новгород	80													
3	Виктория	Кстово	40													
<table border="1"> <thead> <tr> <th>КодДетал</th> <th>Дата</th> <th>Количество</th> <th>Щелкните</th> </tr> </thead> <tbody> <tr> <td>Гайка</td> <td>08.10.2008</td> <td>1500</td> <td></td> </tr> <tr> <td>Штуцер</td> <td>23.11.2010</td> <td>300</td> <td></td> </tr> </tbody> </table>					КодДетал	Дата	Количество	Щелкните	Гайка	08.10.2008	1500		Штуцер	23.11.2010	300	
КодДетал	Дата	Количество	Щелкните													
Гайка	08.10.2008	1500														
Штуцер	23.11.2010	300														
4	Астра	Москва	30													
5	Глория	Казань	20													
6	Юнайт	Н.Новгород	45													
7	Союз	Кстово	80													
8	Полет	Н.Новгород	26													
9	Альбатрос	Н.Новгород	56													
10	Спутник	Дзержинск	30													
12	Коктейль	Самара	70													
13	Лента	Н.Новгород	80													
14	Эврика	Н.Новгород	50													
16	Возрождение	Н.Новгород	60													

Рис. 3.39. Подтаблица

Щелчок по знаку "минус" (-) слева от записи, сворачивает подтаблицу.

### 3.7.6. Сортировка информации

Для удобства просмотра информации в таблице ее содержимое можно отсортировать по значениям одного или нескольких столбцов. Для этого выделяется столбец и выполняется команда **Сортировка от А до Я** или **Сортировка от Я до А** (рис.3.37). Можно также использовать значки **По возрастанию** или **По убыванию** на закладке **Главная** (рис.3.40).

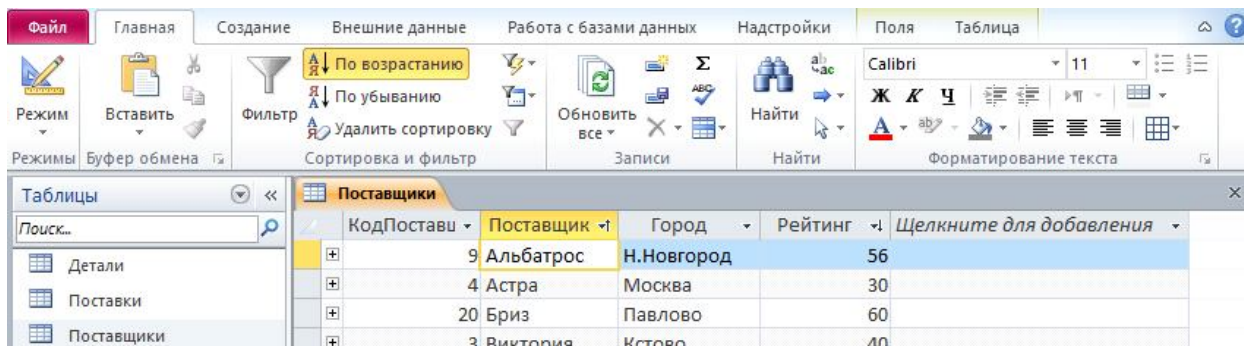


Рис. 3.40. Закладка Главная

### 3.7.7. Поиск и замена информации

Нужное значение в таблице можно найти, выбрав инструмент **Найти** на закладке **Главная** или нажав клавиши **Ctrl+F**. В диалоге **Поиск и замена** (рис.3.41), указывается образец для поиска, задаётся область поиска. Это может быть одно поле или таблица целиком. Задаётся также критерий совпадения, который может иметь значения: **С любой частью поля**, **Поле целиком**, **С началом поля**. Access найдёт запись, у которой первой произойдёт совпадение. Затем поиск можно продолжить кнопкой **Найти далее**.

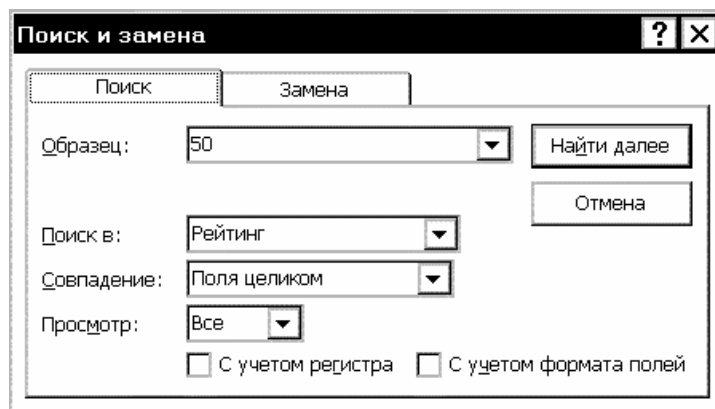


Рис. 3.41. Настройка параметров поиска

На вкладке **Замена** окна **Поиск и замена** можно задать параметры замены одних значений поля другими. Пример приведен на рис.3.42, где предполагается замена наименования города *Горький* на *Н.Новгород*.

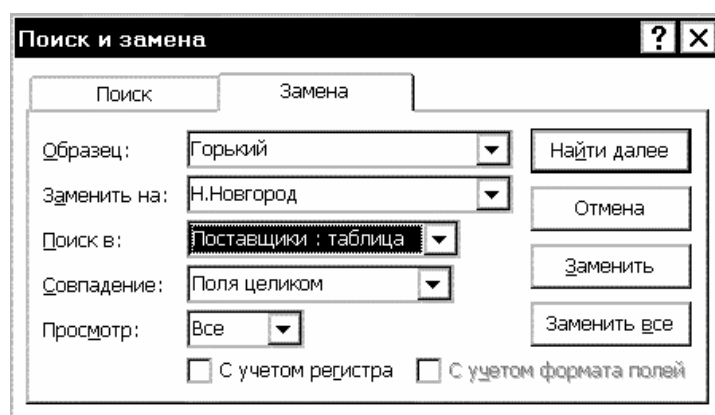


Рис. 3.42. Настройка параметров замены значений полей

## 3.8. Фильтрация записей

С помощью фильтра можно отобразить только часть записей таблицы, удовлетворяющих некоторому условию.

### 3.8.1. Фильтр по выделенному

**Фильтр по выделенному** позволяет отобразить для показа записи, у которых значения некоторого поля совпадают полностью или частично со значением поля в выбранной для фильтрации записи.

Для создания фильтра следует выделить содержимое всего поля таблицы или его части и выбрать одну из команд списка, показанного на рис.3.43, или выбрать аналогичную команду контекстного меню.



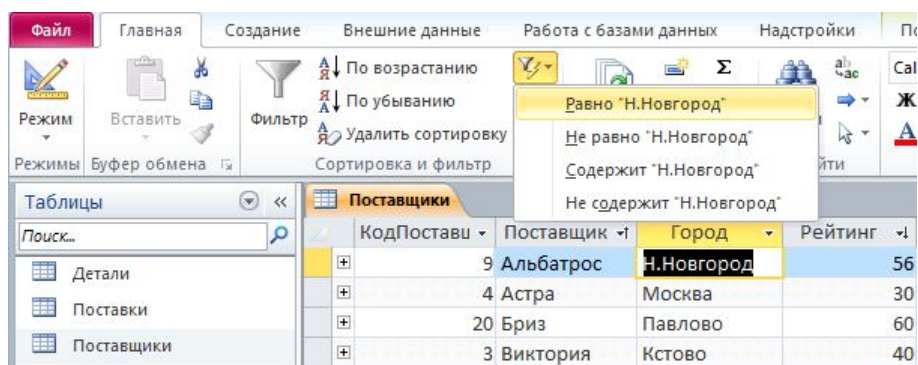


Рис. 3.43. Создание фильтра по выделенному

Access отобразит записи, удовлетворяющие критерию фильтрации. Например, если в таблице *Поставщики* в поле *Город* выделить значение *Н.Новгород*, будут предъявлены поставщики только из *Нижнего Новгорода*.

Чтобы вернуться к просмотру всех записей, нужно или щёлкнуть кнопку **Удалить фильтр** (рис.3.44).

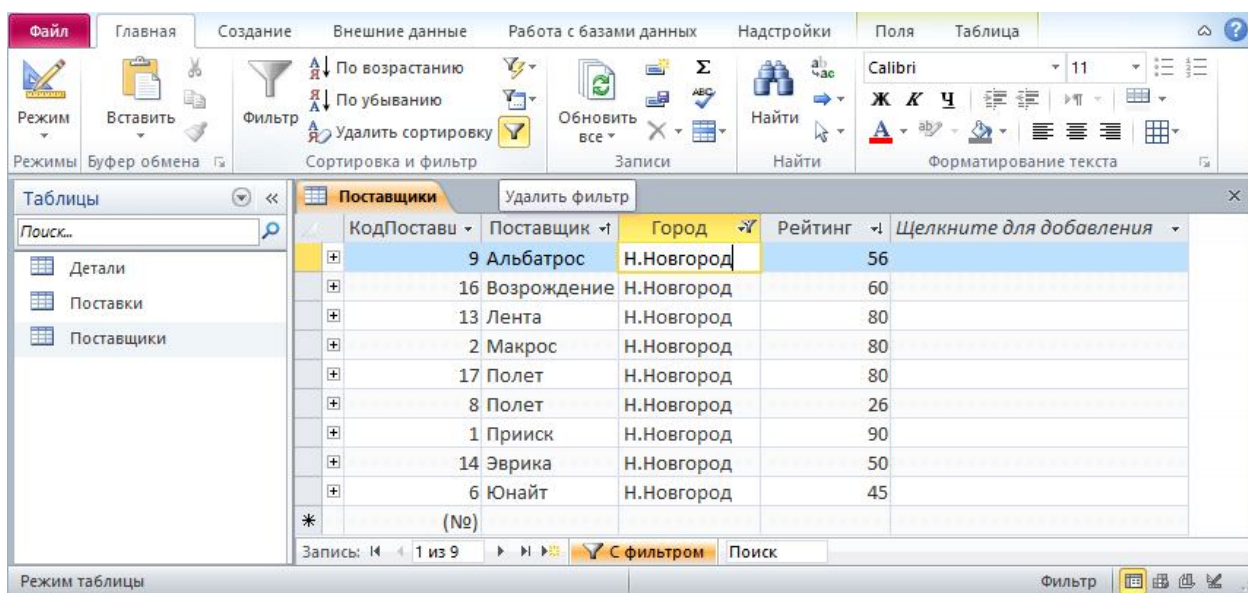


Рис. 3.44. Результат фильтрации и кнопка удаления фильтра

Позднее можно снова использовать последний назначенный фильтр нажав кнопку **Применить фильтр**.

Кнопка **Применить фильтр** может быть в двух состояниях: утопленном и отжатом. В утопленном состоянии, когда фильтр применён, нажатие данной кнопки удаляет фильтр, а если кнопка отжата, ее нажатие применяет фильтр.

Access всегда запоминает последние параметры фильтрации.

Можно перед фильтрацией выделить только часть поля, например, одну первую букву *К* в поле *Город* таблицы *Поставщики*, тогда фильтр выделит все записи с городами на букву *К* (рис.3.45).

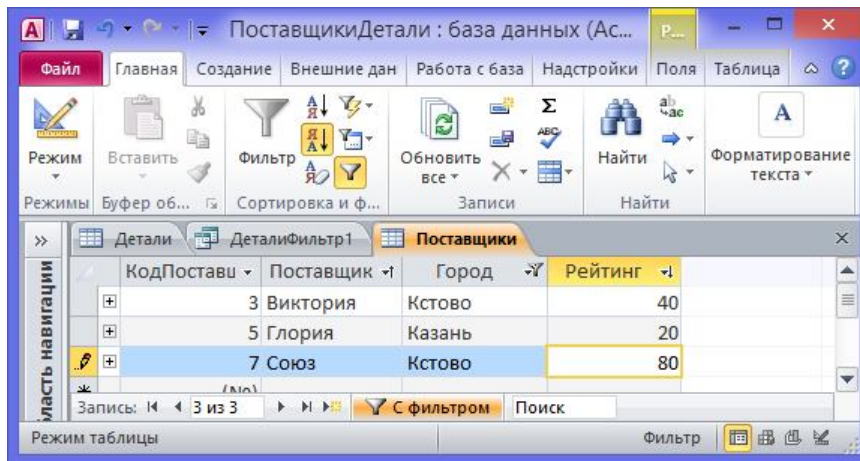



Рис. 3.45. Поставщики из городов на букву К

### 3.8.2. Использование обычного фильтра

Под обычным фильтром понимается способ отбора данных, использующий версию текущей формы или таблицы с пустыми полями. Пользователь вводит в эти поля значения, которые должны содержаться в отбираемых записях. Версия таблицы с пустыми полями выводится командой **Изменить фильтр** из списка команд кнопки  (рис.3.46).

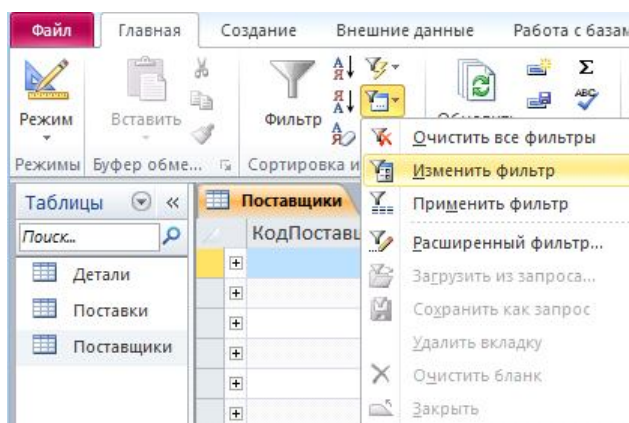


Рис. 3.46. Команды фильтрации

Пример окна для ввода условий фильтрации приведен на рис.3.47. Окно с таблицей исчезло, вместо него появилось окно **Фильтр**, содержащее единственную пустую строку.

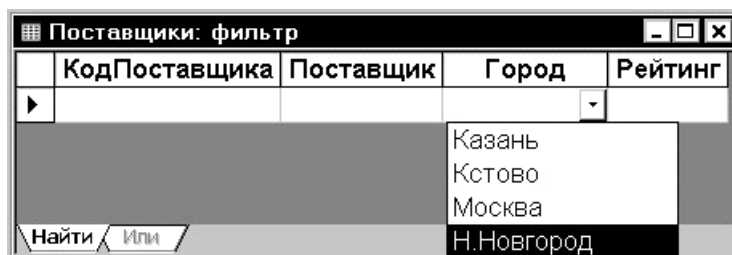



Рис. 3.47. Таблица с пустыми полями для задания условий фильтрации

Условия для фильтрации в окне **Фильтр** можно задавать на нескольких вкладках: на вкладке **Найти** и на одной или нескольких вкладках **Или**. В ячейках следует ввести выражение, с которым будут сравниваться значения полей. При вводе выражения можно воспользоваться списком, раскрываемым кнопкой , который содержит все значения соответствующего поля во всех записях таблицы. Например, на рис.3.47 видны названия всех городов, встречающиеся в таблице. Выберем, например, *Н.Новгород*. На вкладках **Или** задаются аль-

тернативные критерии фильтрации. Всякий раз, когда выбирается одно или несколько значений на вкладке **Или**, становится доступной новая вкладка **Или**. На рис.3.48 показана вкладка **Или**, на которой установлено условие отбора поставщиков с рейтингом больше 70.

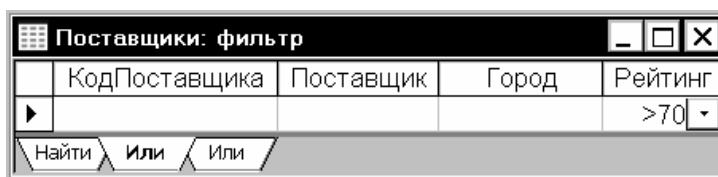


Рис. 3.48. Условие фильтрации на вкладке Или

После применения фильтра Access отобразит записи, которые совпадают или со значениями на вкладке **Найти** или со значениями на вкладке **Или**. На рис.3.49 приведен результат применения фильтра. Отобраны записи о поставщиках из *Нижнего Новгорода* и о поставщиках с рейтингом больше 70.

КодПоставщика	Поставщик	Город	Рейтинг
1	Прииск	Н.Новгород	100
2	Макрос	Н.Новгород	50
4	Астра	Москва	90
6	Юнайт	Н.Новгород	50
8	Полет	Н.Новгород	70
9	Альбатрос	Н.Новгород	20
* (Счетчик)			0

Рис. 3.49. Результат фильтрации по двум условиям

В выражениях, определяющих критерии фильтрации, можно использовать операторы сравнения  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $=$ ,  $<>$ .

Вкладку **Или** можно удалить командой **Удалить вкладку** контекстного меню (рис.3.50), всё содержимое фильтра очищается командой **Очистить бланк**.

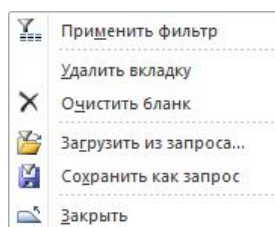


Рис. 3.50. Контекстное меню для работы с фильтром

### 3.8.3. Расширенный фильтр

Расширенный фильтр позволяет задать сложные условия отбора записей из активной таблицы.

Откроем таблицу *Детали* и с помощью команды контекстного меню **Отобразить поля** выберем три столбца *Название*, *Цвет*, *Вес* для показа, как наиболее информативные (рис.3.51).

Название	Цвет	Вес
Ось	Голубой	0,1
Шайба	Красный	0,5
Шпилька	Белый	1
Гайка	Зеленый	1
Винт	Черный	1
Поршень	Желтый	2
Болт	Зеленый	2
Муфта	Коричневый	2
Зубчатое колесо	Розовый	2
Болт	Голубой	3
Шкворень	Фиолетовый	3
Подшипник	Черный	3,3
Вал	Синий	5
Штуцер	Зеленый	20

Рис. 3.51. Таблица *Детали*

На закладке **Главная** из списка кнопки **Дополнительно** выберем команду **Расширенный фильтр**. Появится диалог с бланком запроса для ввода критериев отбора (рис.3.52).

Бланк состоит из двух частей. В верхней части представлена схема таблицы. Нижняя часть состоит из нескольких строк, в ячейках которых задаются условия отбора записей. Условие включает: поле, используемое для сравнения; оператор, описывающий тип сравнения; величину, с которой должно сравниваться значение поля. В строке **Поле** бланка запроса выбираем нужное поле из списка, который появляется после щелчка по кнопке . Обозначение \* в схеме таблицы означает все поля таблицы.

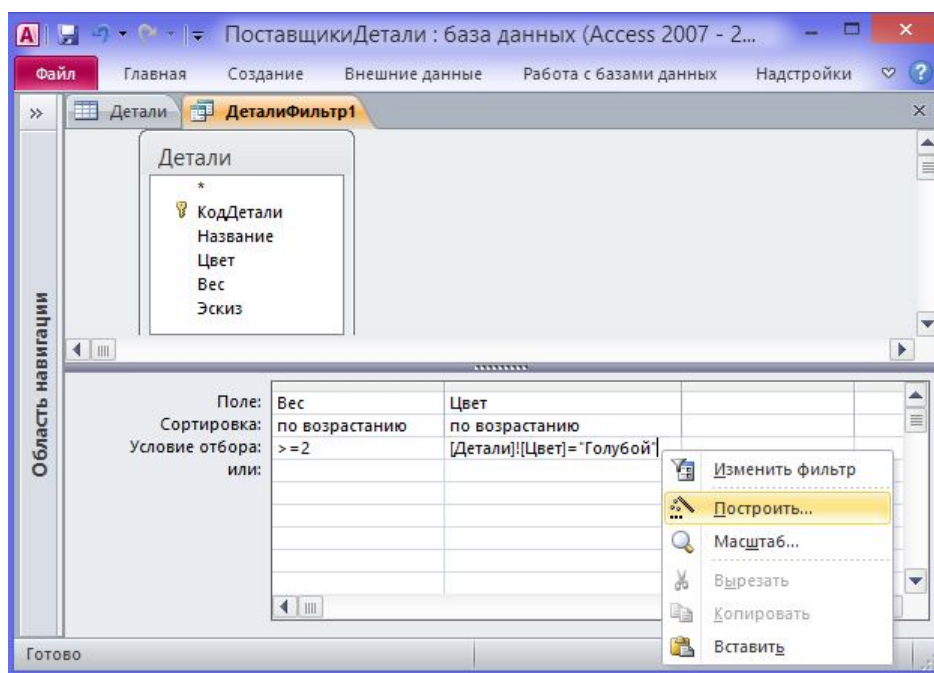


Рис. 3.52. Бланк расширенного фильтра

Выберем, например, поле *Вес*. В строке **Условие отбора** введём выражение  $\geq 2$ . Включим еще в фильтр поле *Цвет*. Для составления условия, которому должно удовлетворять это поле, вызовем **Построитель выражений** командой **Построить** контекстного меню. На рис.3.53 показан **Построитель выражений** в процессе ввода условия на цвет детали.

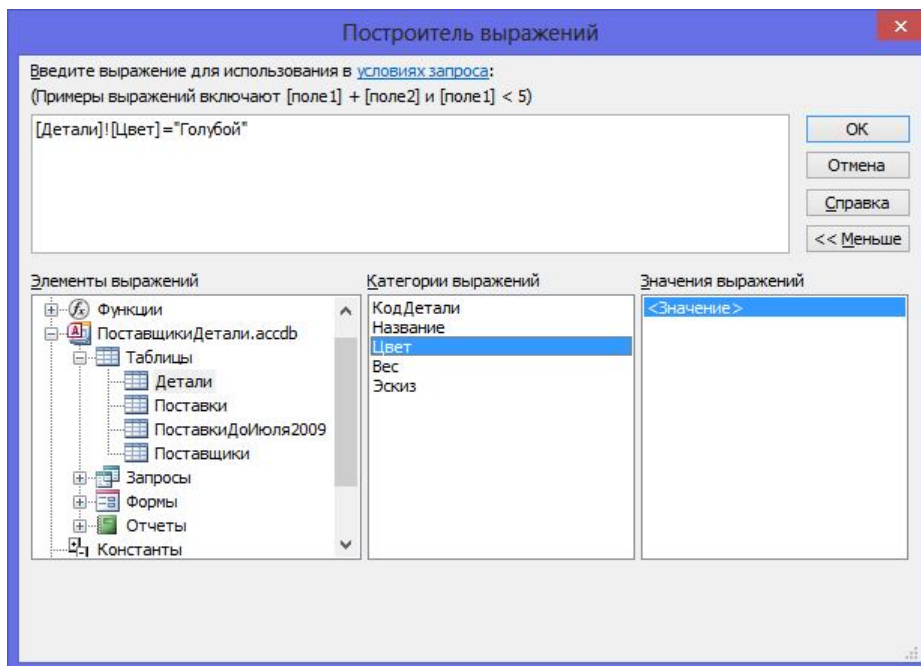


Рис. 3.53. Выбор поля таблицы в построителе выражений

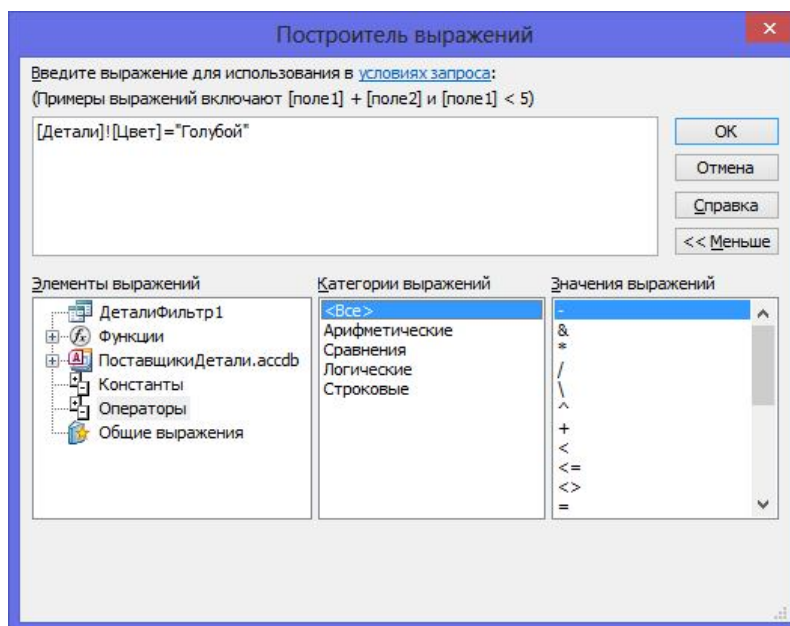



Рис. 3.54. Выбор операторов в построителе выражений

Слева внизу в окне **Построителя выражений** показаны доступные объекты. Выбираем таблицу *Детали*, ее поле *Цвет*. Двойным щелчком мыши (или перетаскиванием) переносим название поля в верхнюю часть **Построителя**. Для вставки в выражение оператора, находим его, выбирая **Операторы** в панели **Элементы выражений** (рис.3.54). Текстовые значения следует заключать в кавычки. Кнопкой **OK** возвращаемся в окно бланка расширенного фильтра (рис.3.52).

Для просмотра результатов фильтрации нажимаем кнопку  **Применить фильтр**. Результат фильтрации показан на рис.3.55. Видно, что отобраны только детали голубого цвета, вес которых не меньше 2.

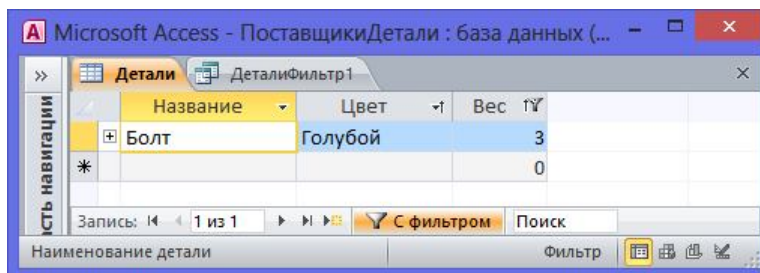


Рис. 3.55. Результаты фильтрации

Поместим условие на цвет в строку **Или**, а для веса в строке **Сортировка** зададим условие сортировки **по возрастанию** (рис.3.56). В фильтр отберутся записи о деталях с весом  $\geq 2$ , а также записи о деталях голубого цвета, причем записи будут расположены в порядке возрастания веса (рис.3.57).

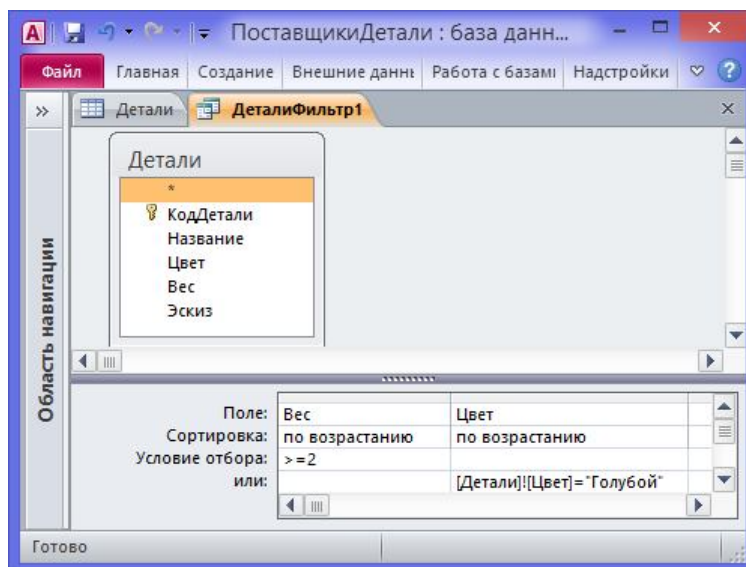


Рис. 3.56. Использование условия **или** в фильтре

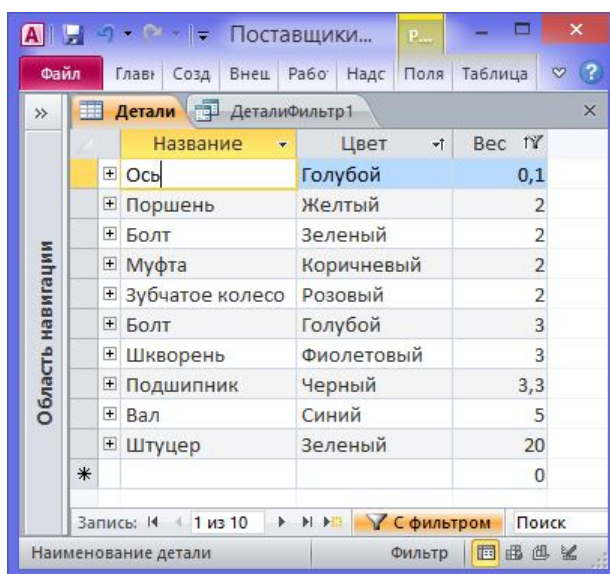


Рис. 3.57. Результаты фильтрации с условием **или**

## 4. Запросы

Запрос позволяет отображать множество записей из одной или нескольких таблиц, удовлетворяющих определенному условию. Запрос подобен фильтру, но фильтр действует только «здесь и сейчас» и не сохраняется в базе данных. Запрос же является объектом, хранимым в БД, поэтому его можно использовать по мере необходимости. Фактически в запросе хранится лишь описание способа отбора записей из различных таблиц, сами же данные хранятся в таблицах. Запрос может выступать источником данных для отображения в формах, для создания отчетов и построения других запросов наравне с таблицами.

Чтобы начать создание запроса, нужно перейти на вкладку **Создание** и выбрать или **Мастер запросов** или **Конструктор запросов**.

При выборе команды **Мастер запросов** открывается окно **Новый запрос** (рис.4.1), где можно выбрать вид создаваемого запроса. Рассмотрим эти виды.

**Простой запрос.** Запускается мастер, помогающий создать запрос.

**Перекрёстный запрос.** Специальный запрос для анализа данных по двум или более признакам. Например, может представлять интерес запрос в виде таблицы, где заголовками строк являются поставщики, заголовками столбцов – детали, а на пересечении строк и столбцов стоит суммарное количество деталей, поставленных определенным поставщиком.

**Повторяющиеся записи.** В этом режиме запускается мастер, который просматривает выделенную таблицу или запрос и отбирает все записи, содержащие повторяющиеся значения в заданном поле. Его можно использовать для проверки возможности установления связей между таблицами.

**Записи без подчинения.** Данный запрос используется для сравнения двух таблиц и поиска всех записей в первой таблице, для которых не имеется подчинённых записей во второй таблице.

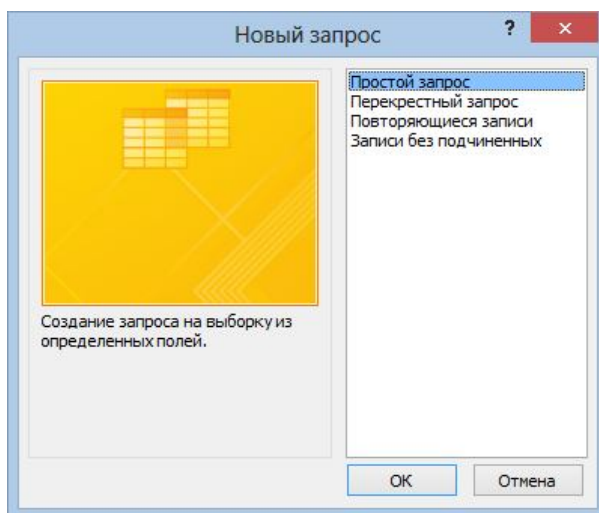


Рис. 4.1. Выбор способа создания запроса

### 4.1. Создание простого запроса

Предположим в БД *ПоставщикиДетали* нужно получить перечень всех деталей, поставляемых каждым поставщиком. В окне **Новый запрос** (рис.4.1), выберем **Простой запрос**. Начнет работать мастер создания простого запроса.

На первом диалоге мастера (рис.4.2) отбираем поля для включения в запрос: поле *Поставщик* из таблицы *Поставщики* и поле *Название* из таблицы *Детали*.

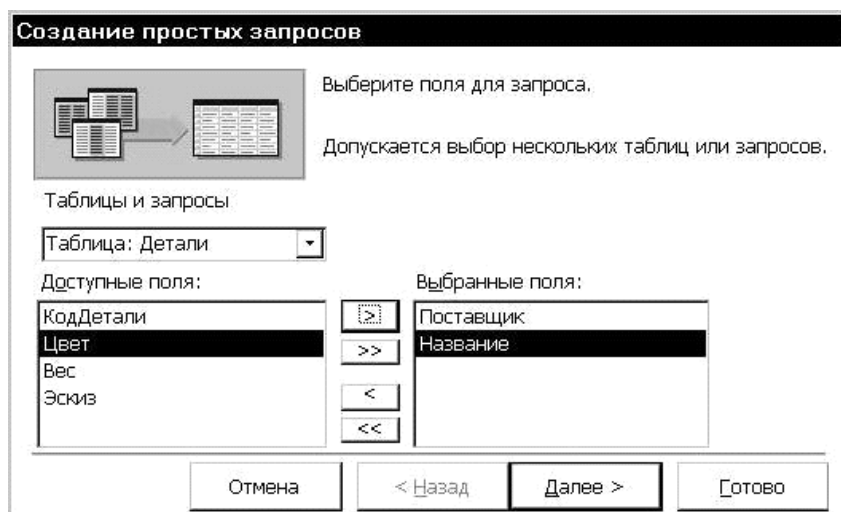


Рис. 4.2. Отбор полей для запроса

На втором диалоге (рис.4.3), можно выбрать тип запроса:

- **подробный** – выводятся все записи из различных таблиц, удовлетворяющие заданным условиям;
- **итоговый** – отображает итоговая информация, а не все подходящие записи.

Выберем тип запроса **подробный**.

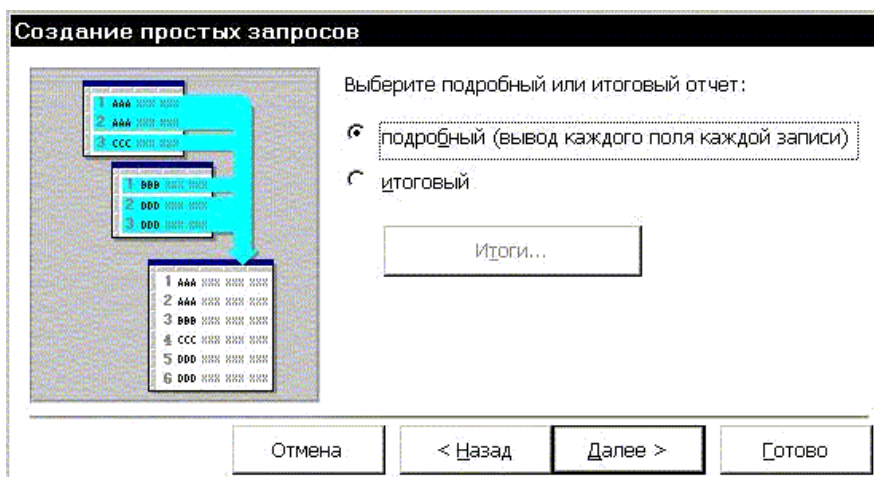


Рис. 4.3. Выбор типа запроса

На последнем третьем диалоге (рис.4.4), задается имя запроса. Согласимся с предложенным по умолчанию именем *Поставщики Запрос*. Кроме того, оставим переключатель **Открыть запрос для просмотра данных**. В результате будет показана таблица с результатами запроса (рис.4.5).



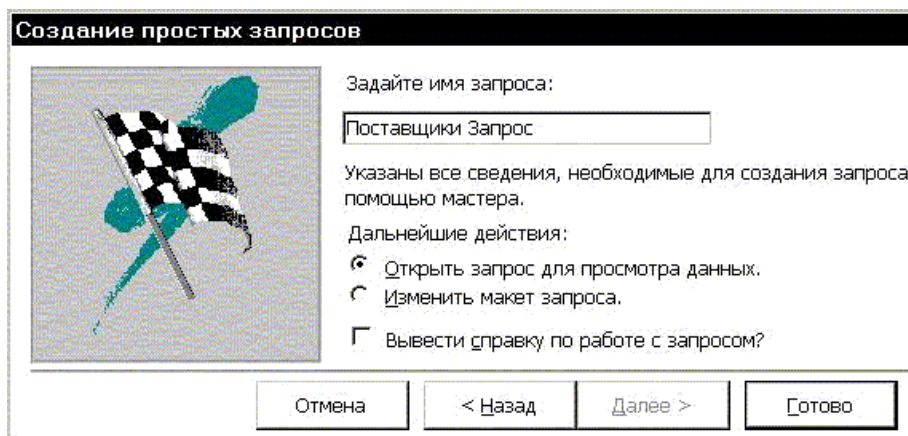


Рис. 4.4. Ввод имени запроса

Поставщик	Название
Лента	Шпилька
Бриз	Муфта
Союз	Поршень
Астра	Поршень
Витязь	Ось
Союз	Шкворень
Виктория	Гайка
Полет	Гайка
Виктория	Штуцер
Прииск	Болт
Астра	Шайба
*	

Рис. 4.5. Результат запроса *Поставщики Запрос*

#### 4.1.1. Кнопка Режим

На вкладке **Главная** слева расположена кнопка **Режим**, с помощью которой можно переключаться между разными режимами представления активного в данный момент объекта. Кнопка имеет список, в котором перечислены доступные режимы. Вид кнопки меняется в зависимости от текущего режима. Перейти к альтернативному режиму можно щелчком по кнопке или выбором нужного режима из списка. На рис.4.6 показан список кнопки **Вид** при работе с запросами.

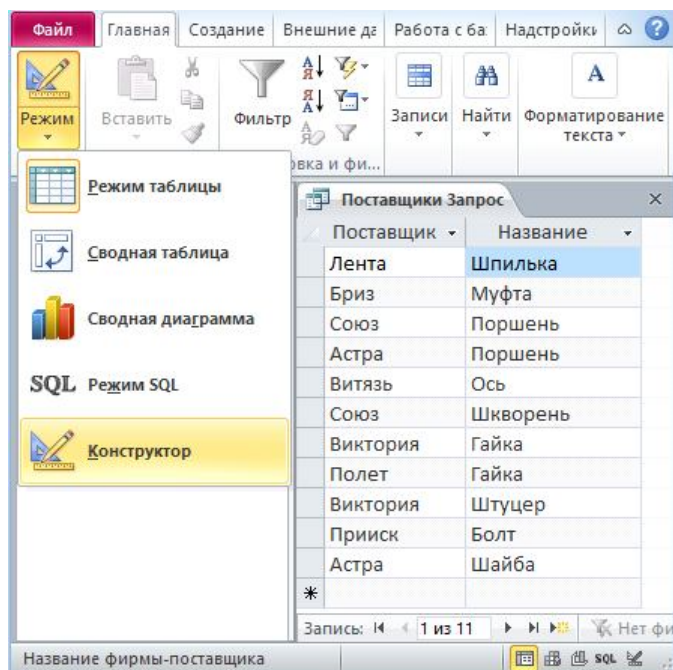


Рис. 4.6. Список команд кнопки Вид для запросов

Пример представления запроса в **Режиме таблицы** приведен на рис.4.5.

Режим **Конструктор** для запросов подробно обсудим ниже.

**Режим SQL** обеспечивает просмотр запроса в виде инструкции языка SQL. На рис.4.7 показан запрос *Поставщики Запрос* в режиме SQL. Именно эта краткая инструкция языка SQL и сохраняется в базе данных как запрос.

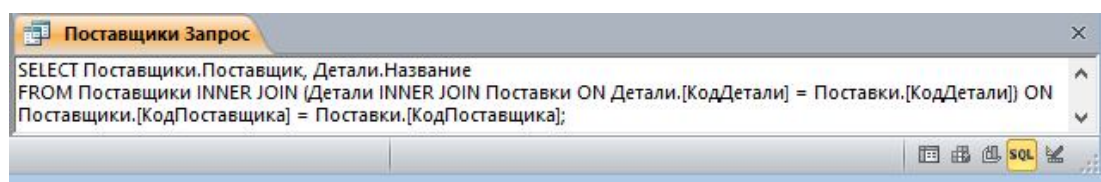


Рис. 4.7. Инструкция языка SQL, реализующая запрос *Поставщики Запрос*

## 4.2. Конструктор запросов

Перейдем в режим **Конструктор** для запроса *Поставщики Запрос*. Откроется бланк конструктора запросов, показанный на рис.4.8.

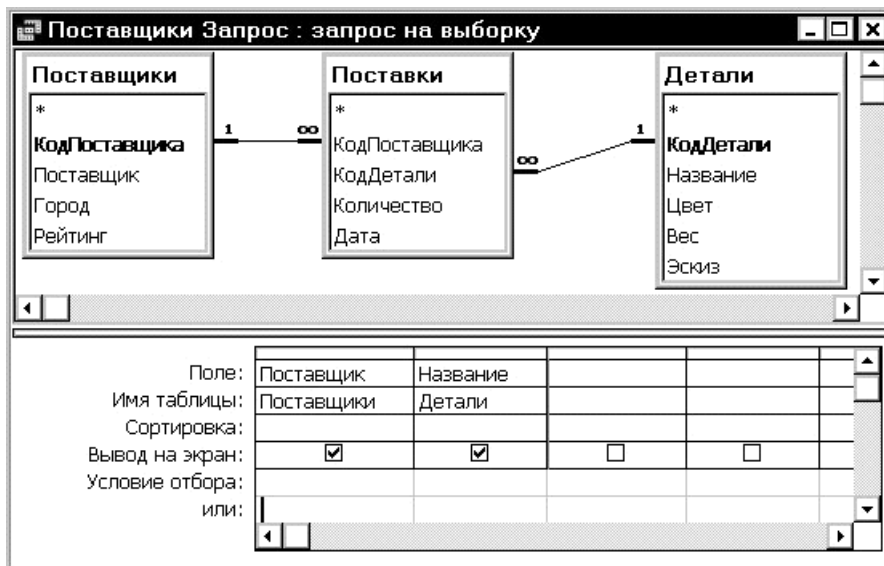


Рис. 4.8. Бланк конструктора запросов

В верхней части бланка помещены списки полей таблиц, из которых выбраны поля в запрос. Кроме того, добавляются поля таблиц, необходимых для установления связи между выбранными в запрос полями. В рассматриваемый запрос были отобраны поля из таблиц *Поставщики* и *Детали*, мастер добавил необходимый для связи список полей таблицы *Поставки*. При работе с бланком запроса можно самостоятельно добавлять или изменять связи между таблицами. Эти изменения будут касаться только данного запроса и не изменят связей, показываемых в окне **Схема данных**.

Работа с бланком запроса аналогична работе с подобным бланком для расширенного фильтра. В бланке на рис.4.8 не заданы условия отбора для полей *Поставщик* и *Название*, включенных в запрос. Это значит, что в запрос включаются записи о всех поставщиках, которые поставляют какие-либо детали. Зададим, например, в строке **Условие отбора** для поля *Название* значение «Болт» (рис.4.9), тогда в запрос будут включены только поставщики болтов. Результат такого запроса показан на рис.4.10.

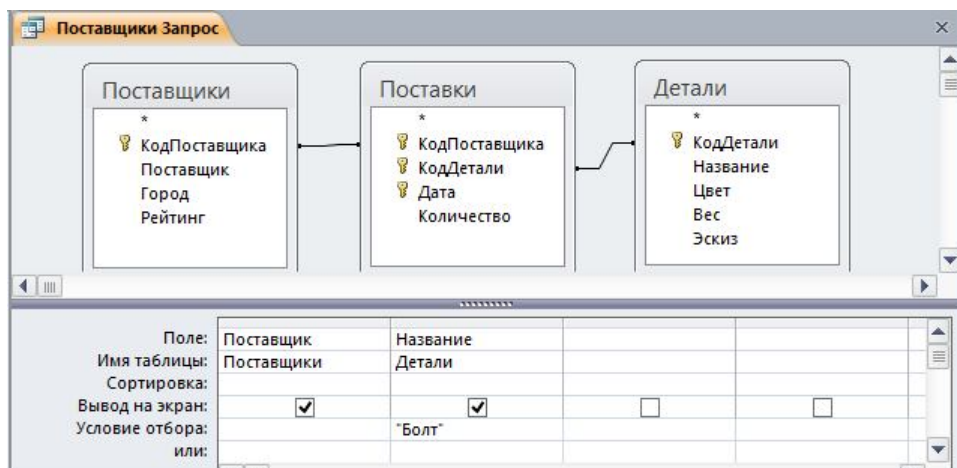


Рис. 4.9. Ввод условия отбора в запрос

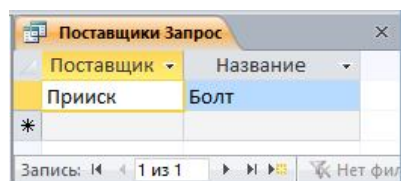


Рис. 4.10. Запрос о поставщиках болтов

Установка флажка **Вывод на экран** в строке конструктора обеспечивает показ соответствующего поля, снятие флажка отменяет вывод поля на экран.

### 4.3. Запросы с выводом наборов значений

Имеется возможность сортировать записи по некоторому полю и выводить несколько максимальных или минимальных значений из упорядоченного списка записей. Выберем, для примера, три самые тяжелые детали из таблицы *Детали*. В окне **Новый запрос** (рис.4.1), выберем **Конструктор**. При загрузке конструктора появится окно выбора таблицы (рис.4.11), в котором выберем таблицу *Детали*. Кстати, от выбора таблицы можно отказаться и сделать это позже, вызвав окно **Добавление таблицы** командой **Добавить таблицу** контекстного меню для запроса.

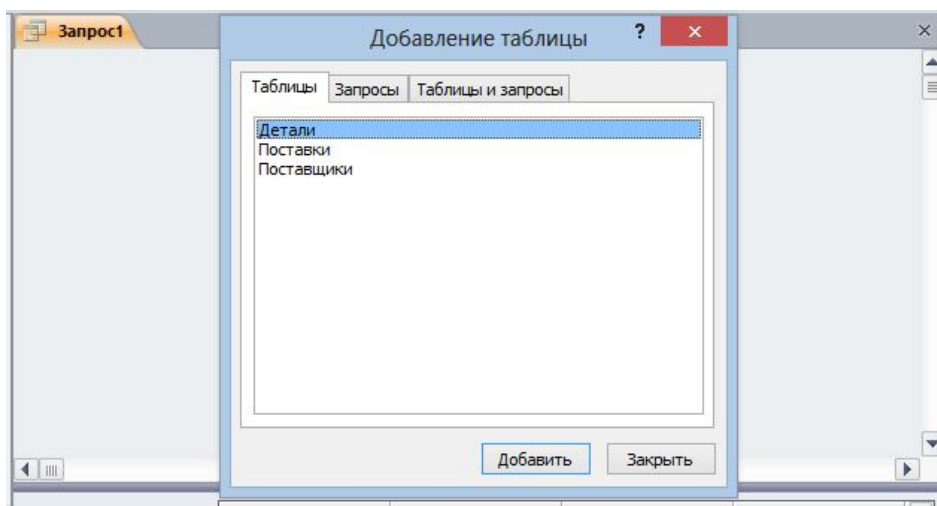


Рис. 4.11. Выбор таблицы для включения в бланк конструктора

Работая в конструкторе, включим в запрос поля *Название* и *Вес* и проведем сортировку по убыванию значений поля *Вес* (рис.4.12). В группе команд **Настройка запроса** в поле **Возврат** введем значение 3.

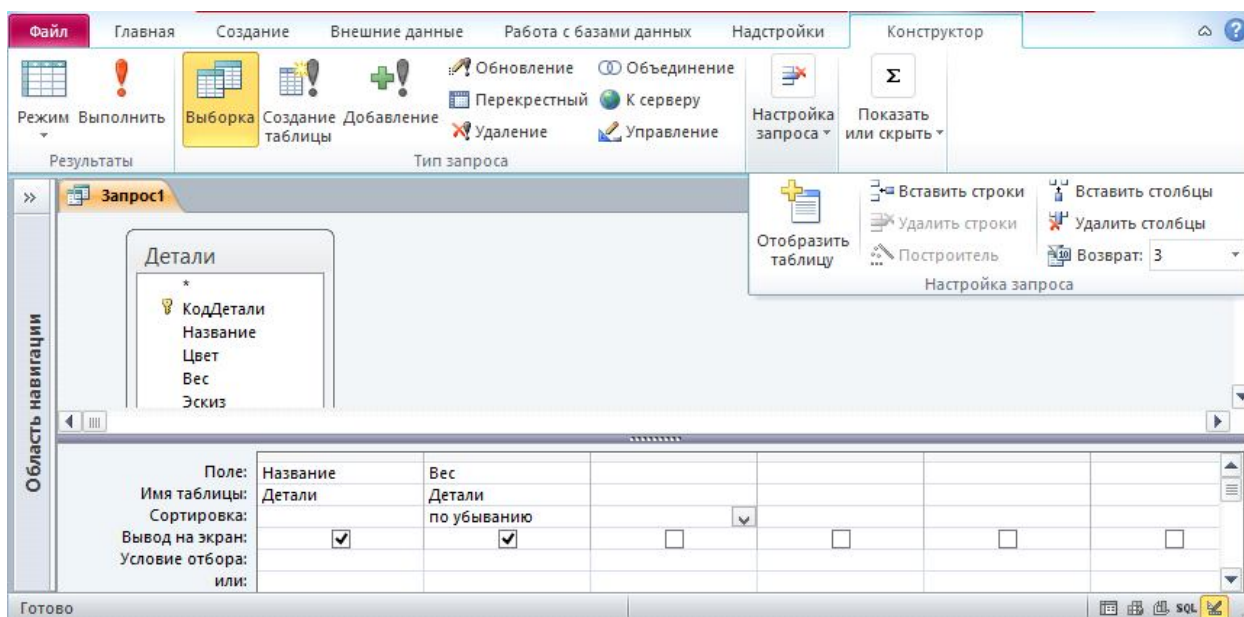


Рис. 4.12. Создание запроса для выборки трех самых тяжелых деталей

Записи, включенные в запрос, показаны на рис.4.13.

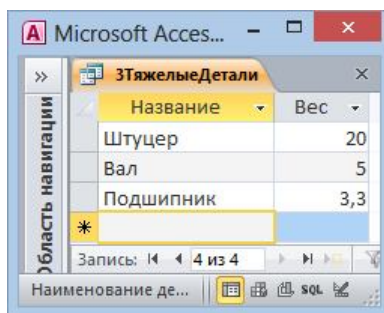


Рис. 4.13. Запрос о трех самых тяжелых деталях

## 4.4. Подведение итогов по записям

В запросе можно делать вычисления над полями, включенными в запрос. Создадим, например, запрос, в котором определим общее количество деталей каждого вида, полученных от различных поставщиков.

На вкладке **Создание** выберем инструмент **Мастер запросов**, в диалоге **Новый запрос** (рис.4.1) выберем **Простой запрос**. Далее (рис.4.2) выбираем из таблицы *Детали* поле *Название* и поле *Количество* из таблицы *Поставки*. На следующем диалоге (рис.4.3) выбираем **Итоговый**, при этом становится активной кнопка **Итоги**, нажимаем ее и переходим в окно настройки запроса (рис.4.14).

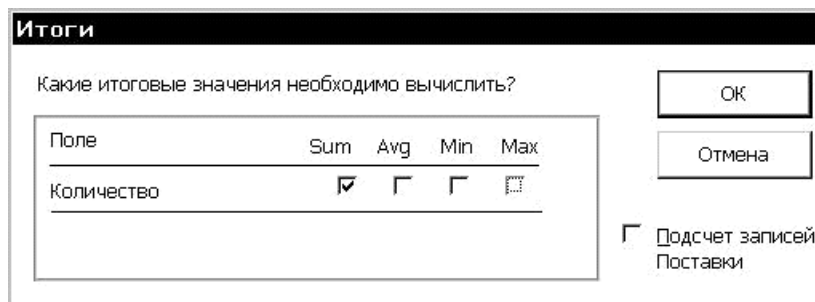


Рис. 4.14. Выбор групповой операции

Выбираем в окне **Итоги** групповую операцию **Sum**, которую следует выполнить над числовым полем *Количество*, включенным в запрос. После нажатия кнопки **OK** возвращаемся к предыдущему диалогу (рис.4.3) с которого по кнопке **Далее** переходим к диалогу ввода имени запроса (рис.4.4). Назовем запрос, например, *ОбщееЧислоДеталей*. Результат запроса показан на рис.4.15.

Название	Sum - Колич
Болт	1500
Гайка	2000
Муфта	70
Ось	30
Поршень	70
Шайба	500
Шкворень	90
Шпилька	1200
Штуцер	300

Рис. 4.15. Результат запроса об общем числе деталей

В режиме конструктора данный запрос имеет вид, показанный на рис.4.16. Из списка строки **Групповая операция** можно выбрать нужную функцию для обработки значений полей.

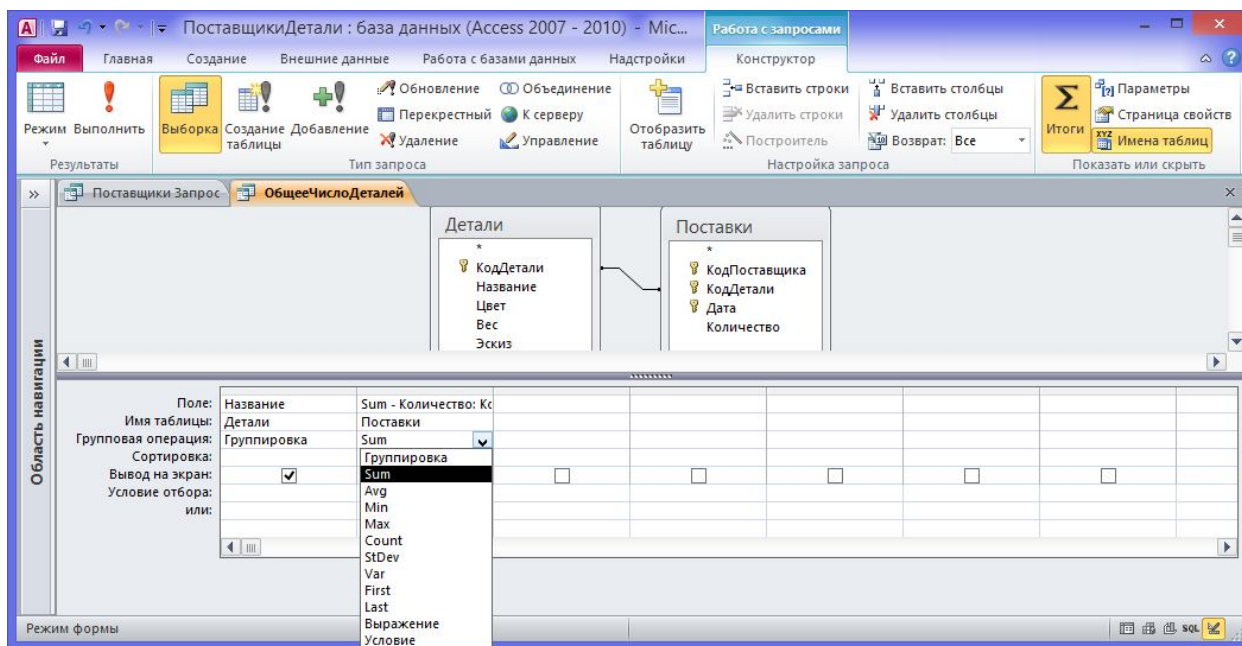


Рис. 4.16. Возможные функции для обработки группы записей

Строка **Групповая операция** включается в бланк запроса, если нажата кнопка **Итоги** на закладке **Конструктор**. Кнопка **Имена таблиц** включает соответствующую строку в бланк конструктора или убирает ее.

В табл.4.1 приведено описание доступных функций. В формулах предполагается, что  $x_1, x_2, \dots, x_n$ , – значения поля для записей, включенных в группу.

Таблица 4.1. Функции для групповых операций

Функция	Описание	Формула
Sum	Суммирование значений группы	$Sum = \sum_{i=1}^n x_i$
Avg	Среднее значение	$Avg = \sum_{i=1}^n x_i / n = Sum/n$
Var	Дисперсия для выборки, состоящей из значений группы	$Var = \sum_{i=1}^n (x_i - Avg)^2 / (n - 1)$
StDev	Стандартное отклонение для выборки, состоящей из значений группы	$StDev = \sqrt{Var}$
Min	Минимальное значение в группе	
Max	Максимальное значение в группе	
Count	Количество значений в группе	
First	Первое значение в группе	
Last	Последнее значение в группе	

## 4.5. Вычисляемые поля

Вычисляемое поле может быть создано в запросе, форме или отчете. Вычисляемое поле используется лишь для проведения подсчетов и не является обычным полем таблицы.

Для создания вычисляемого поля в строке **Поле** бланка запроса вводится некоторое выражение. Подсчитаем, например, общий вес деталей, поставленных поставщиками. Создадим запрос *ДеталиПоставщиков*, в который включим поле *Название* из таблицы *Детали*,

поле *Поставщик* из таблицы *Поставщики* и поле *Количество* из таблицы *Поставки*. Откроем данный запрос в режиме конструктора и в свободной ячейке строки **Поле** введем выражение  $Вес * Количество$ . После завершения ввода вычисляемого выражения оно автоматически преобразуется в виду **Выражение1: [Вес]\*[Количество]**, которое видно на рис.4.17.

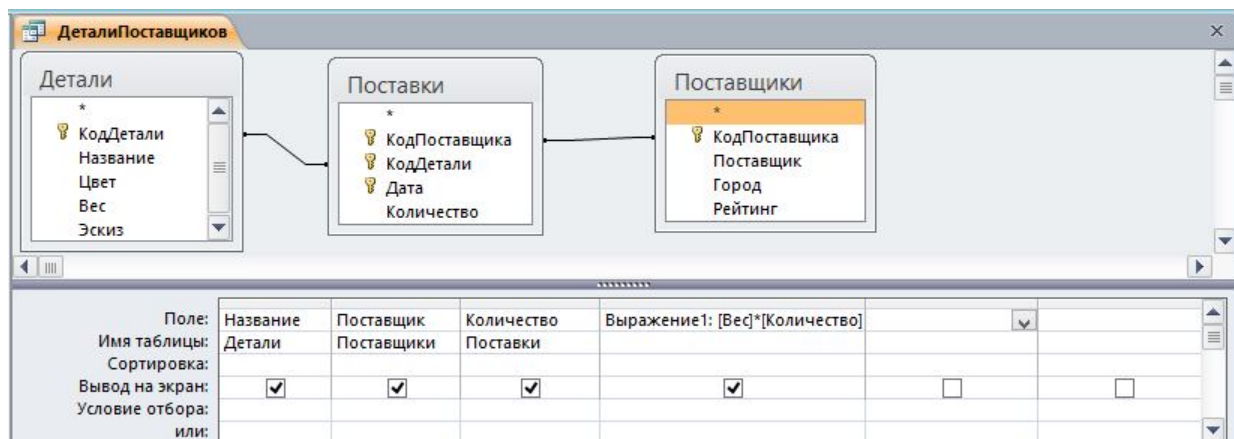


Рис. 4.17. Создание вычисляемого поля в запросе

На рис.4.18 показан результат этого запроса.

The screenshot shows the results of the query in a table view. The table has four columns: 'Название', 'Поставщик', 'Количество', and 'Выражение1'. The data is as follows:

Название	Поставщик	Количество	Выражение1
Шпилька	Лента	1200	1200,00
Муфта	Бриз	70	140,00
Поршень	Союз	20	40,00
Поршень	Астра	50	100,00
Ось	Витязь	30	3,00
Шкворень	Союз	90	270,00
Гайка	Виктория	1500	1500,00
Гайка	Полет	500	500,00
Штуцер	Виктория	300	6000,00
Болт	Прииск	1500	3000,00
Шайба	Астра	500	250,00
*			

Рис. 4.18. Запрос с вычислимым полем

Имя вычисляемого поля состоит из слова **Выражение** и номера, показывающего, каким по счету оно было создано. Это имя можно изменить в конструкторе запросов, записав нужное имя в ячейке перед двоеточием. Для рассматриваемого примера назовем вычисляемое поле *Общий вес*. Запрос в режиме таблицы с новым названием вычисляемого поля приведен на рис.4.19.

Название	Поставщик	Количество	Общий вес
Шпилька	Лента	1200	1200,00
Муфта	Бриз	70	140,00
Поршень	Союз	20	40,00
Поршень	Астра	50	100,00
Ось	Витязь	30	3,00
Шкворень	Союз	90	270,00
Гайка	Виктория	1500	1500,00
Гайка	Полет	500	500,00
Штуцер	Виктория	300	6000,00
Болт	Прииск	1500	3000,00
Шайба	Астра	500	250,00
*			

Рис. 4.19. Новое название для вычисляемого поля

При вводе выражения в вычисляемое поле названия полей можно выбирать из выпадающего списка, имеющегося у ячейки. Для построения сложных выражений можно воспользоваться **Построителем выражений**, который вызывается командой контекстного меню **Построить** (рис.4.20).

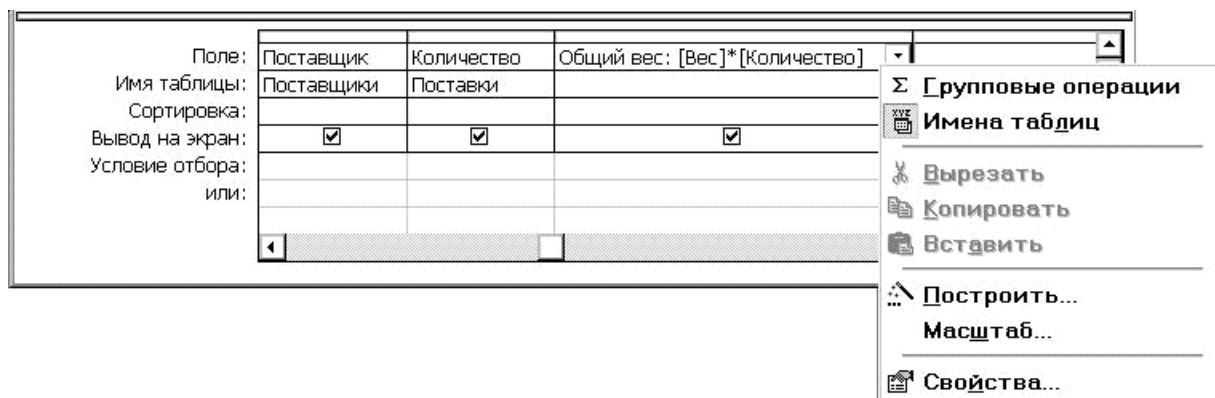


Рис. 4.20. Вызов построителя выражений командой контекстного меню

## 4.6. Запросы с параметрами

Запросы с параметрами позволяют запрашивать у пользователя некоторые значения, уточняющие запрос.

Предположим, нужно найти поставщика определенной детали из определенного города. Составим с помощью мастера запрос, в который включим поля: *Поставщики.Поставщик*, *Поставщики.Город*, *Детали.Название* (рис.4.21).



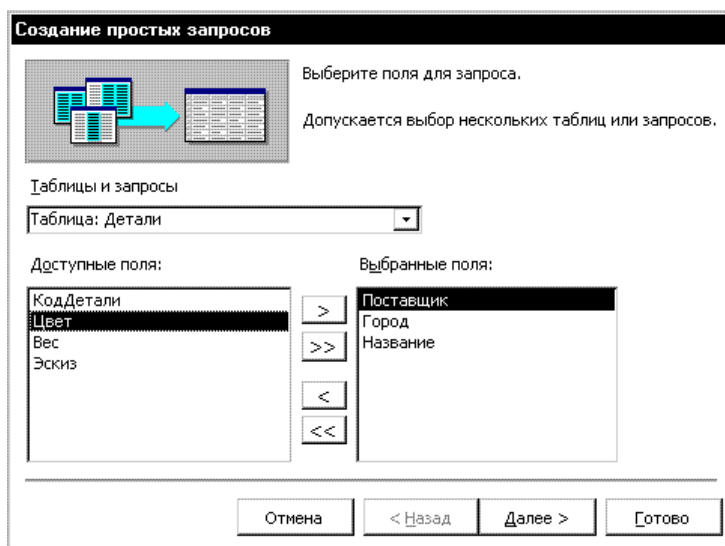


Рис. 4.21. Поля, отобранные в запрос

Сохраним запрос как *ПоставщикиГородаДетали*. В режиме таблицы этот запрос приведен на рис.4.22.

Поставщик	Город	Название
Лента	Н.Новгород	Шпилька
Бриз	Павлово	Муфта
Союз	Кстово	Поршень
Астра	Москва	Поршень
Витязь	Муром	Ось
Союз	Кстово	Шкворень
Виктория	Кстово	Гайка
Полет	Н.Новгород	Гайка
Виктория	Кстово	Штуцер
Прииск	Н.Новгород	Болт
Астра	Москва	Шайба
*		

Запись: 1 из 11 Нет фильтра Поиск

Рис. 4.22. Запрос *ПоставщикиГородаДетали*

Выполним для данного запроса команду **Файл, Сохранить объект как** и сохраним его дополнительно под именем *ПоставщикиГородаДеталиУсл* (рис.4.23).

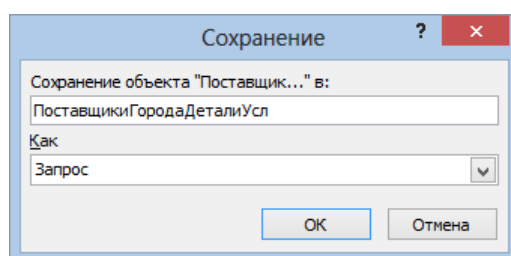


Рис. 4.23. Сохранение копии запроса

Откроем запрос *ПоставщикиГородаДеталиУсл* в режиме конструктора (рис.4.24).



Рис. 4.24. Запрос *ПоставщикиГородаДеталиУсл* в режиме конструктора

Для того чтобы при выполнении запроса выводилось окно для ввода параметра запроса, в конструкторе запросов в строке **Условие отбора** нужно написать текст, заключенный в квадратные скобки. Этот текст показывается как приглашение при вводе параметра запроса. В примере на рис. 4.24 текст в квадратных скобках задан для полей *Город* и *Название*.

Нажмем кнопку **Выполнить** на вкладке **Конструктор** или перейдем к режиму таблицы. При этом сначала появится окно для ввода значения поля *Город* (рис.4.25). Введем в это окно в качестве значения параметра *Н.Новгород*.

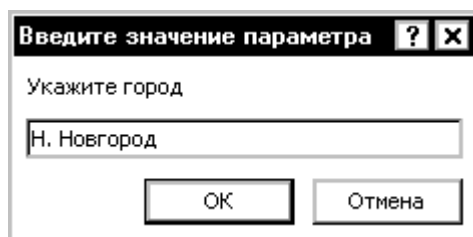


Рис. 4.25. Ввод значения для поля *Город*

Затем появится окно для ввода названия детали (рис.4.26).

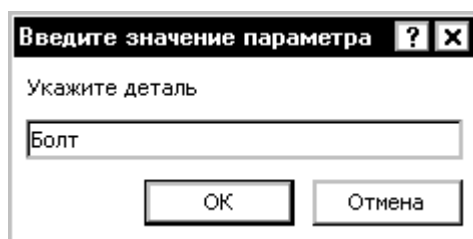


Рис. 4.26. Ввод значения для поля *Название*

Результат запроса приведен на рис.4.27.

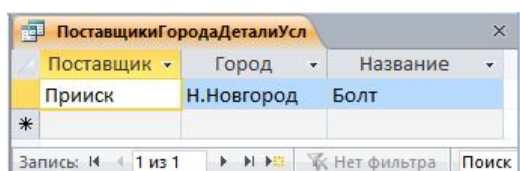


Рис. 4.27. Результат запроса с параметрами

## 4.7. Перекрестные запросы

Иногда бывает нужно упорядочить информацию из базы данных по двум или более параметрам. В этом случае используются перекрестные запросы.

### 4.7.1. Создание перекрестного запроса на основе таблицы

Пусть имеется таблица *Тестирование* со структурой, показанной на рис.4.28.

Имя поля	Тип данных	Описание
КодСтудента	Счетчик	Числовой код
ФИО	Текстовый	Фамилия, имя, отчество
Пол	Текстовый	
Возраст	Числовой	Возраст в годах
Балл	Числовой	Успеваемость

Свойства поля	
Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Список значений
Источник строк	Муж;Жен
Присоединенный столбец	1
Число столбцов	1
Заглавия столбцов	Нет
Ширина столбцов	
Число строк списка	8
Ширина списка	Авто
Ограничиться списком	Да

Тип данных определяет значения, которые можно сохранять в этом поле. Для справки по типам данных нажмите клавишу F1.

Рис. 4.28. Структура таблицы *Тестирование*

Заметим, что здесь для удобства ввода значений в поле *Пол* ему назначен элемент управления **Поле со списком**. В качестве **Типа источника строк** выбран **Список значений**, а возможные значения перечислены через точку с запятой в строке **Источник строк**. Содержание данной таблицы показано на рис.4.29.

КодСтудента	ФИО	Пол	Возраст	Балл
1	Иванов И.П.	Муж	18	3.5
2	Петров П.И.	Муж	19	4.4
3	Сидоров В.И.	Муж	20	4.1
4	Маркина М.В.	Жен	18	4.5
5	Меньшова Л.В.	Жен	19	4.3
6	Баканина У.С.	Жен	20	4.4
*	(Счетчик)	Муж	0	0

Запись: 6 из Жен

Рис. 4.29. Содержание таблицы *Тестирование*

Требуется вычислить средний балл в зависимости от возраста и пола. Для этого надо создать таблицу с двумя входами, у которой заголовками строк должен быть возраст участника теста, а заголовками столбцов – пол. Как она должна выглядеть, показано на рис.4.30. На пересечении строки с возрастом и столбца с полом стоит средний балл участников тестирования, относящихся к соответствующей группе.

	Возраст	Жен	Муж
▶	18	4.50	3.50
	19	4.30	4.40
	20	4.40	4.10

Запись: 1 из 3

Рис. 4.30. Перекрестная таблица с двумя входами

Создание перекрестного запроса с помощью мастера происходит следующим образом. На закладке **Создание** выберем инструмент **Мастер запросов**. В диалоговом окне **Новый запрос** (рис.4.1), выберем **Перекрестный запрос**, после чего начинает работать мастер создания перекрестного запроса.

На первом диалоге мастера указываем таблицу или запрос, поля которых будут включены в перекрестный запрос (рис.4.31).

**Создание перекрестных таблиц**

Выберите таблицу или запрос, поля которых необходимо вывести в перекрестном запросе.

Для включения полей из нескольких таблиц сначала создайте обычный запрос, содержащий все необходимые поля.

Показать:  Таблицы  Запросы  Все

Образец:

	Заголовок1	Заголовок2	Заголовок3
	ИТОГИ		

Отмена < Назад Далее > Готово

Рис. 4.31. Выбор источника данных для перекрестного запроса

На втором диалоге выбираем поле *Возраст*, значения которого будут заголовками строк (рис.4.32).

**Создание перекрестных таблиц**

Выберите поля, значения которых будут использованы в качестве заголовков строк. Допускается выбор не более трех полей. Выберите поля по порядку сортировки данных. Например, можно сначала выполнить сортировку значений по странам, а затем по городам.

Доступные поля:

- КодСтудента
- ФИО
- Пол
- Балл

Выбранные поля:

- Возраст

Образец:

Возраст	Заголовок1	Заголовок2	Заголовок3
Возраст1	ИТОГИ		
Возраст2			
Возраст3			

Отмена < Назад Далее > Готово

Рис. 4.32. Выбор поля для заголовков строк перекрестного запроса

На третьем диалоге выбираем поле *Пол*, значения которого будут использоваться как заголовки столбцов (рис.4.33).

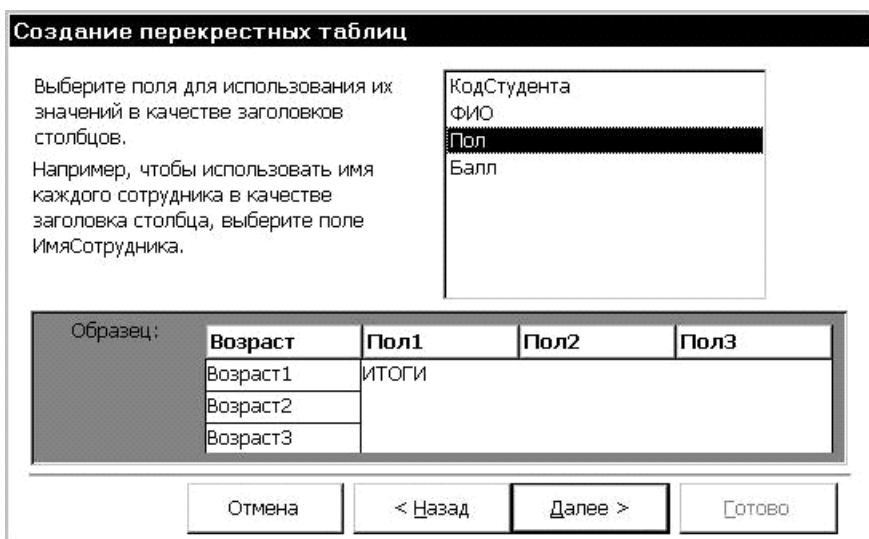


Рис. 4.33. Выбор поля для заголовков столбцов перекрестного запроса

На четвертом диалоге выбираем поле *Балл* и функцию **Среднее**, чтобы отобразить средний результат тестирования для каждого подмножества участников (рис.4.34).



Рис. 4.34. Выбор обобщаемого показателя для перекрестного запроса

Кроме этого можно установить флажок **Да**, для вычисления итогового значения для каждой строки. В этом случае будет создан еще один столбец, содержащий средний балл для каждой возрастной группы.

В пятом диалоговом окне задается имя запроса (рис.4.35).

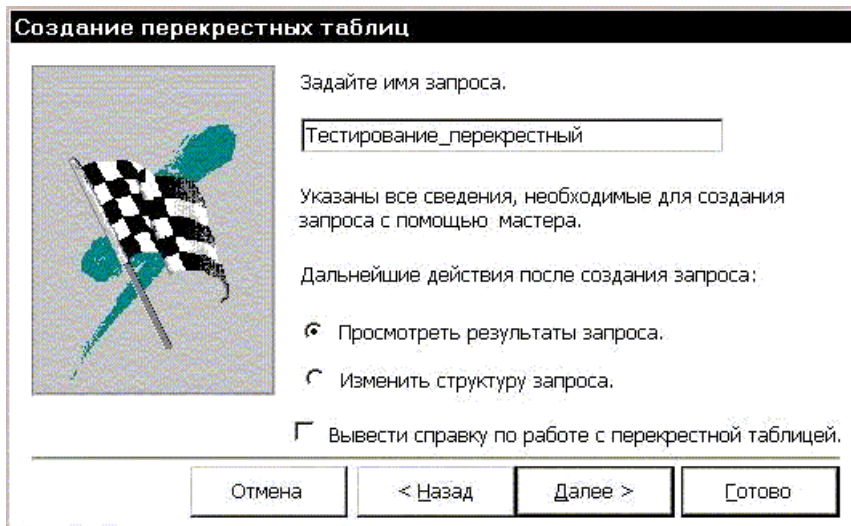


Рис. 4.35. Задание имени перекрестному запросу

Как видно на рис.4.30, числовые данные представлены в форме с фиксированной точкой и двумя знаками после десятичной точки. Для управления форматом представления данных нужно перейти в режим конструктора запросов и настроить формат поля. На рис.4.36 показан созданный перекрестный запрос в режиме конструктора и окно свойств для поля *Балл*, в котором показан выбор нужного числового формата. Окно свойств выводится нажатием клавиши **F4** или командой **Свойства** контекстного меню.

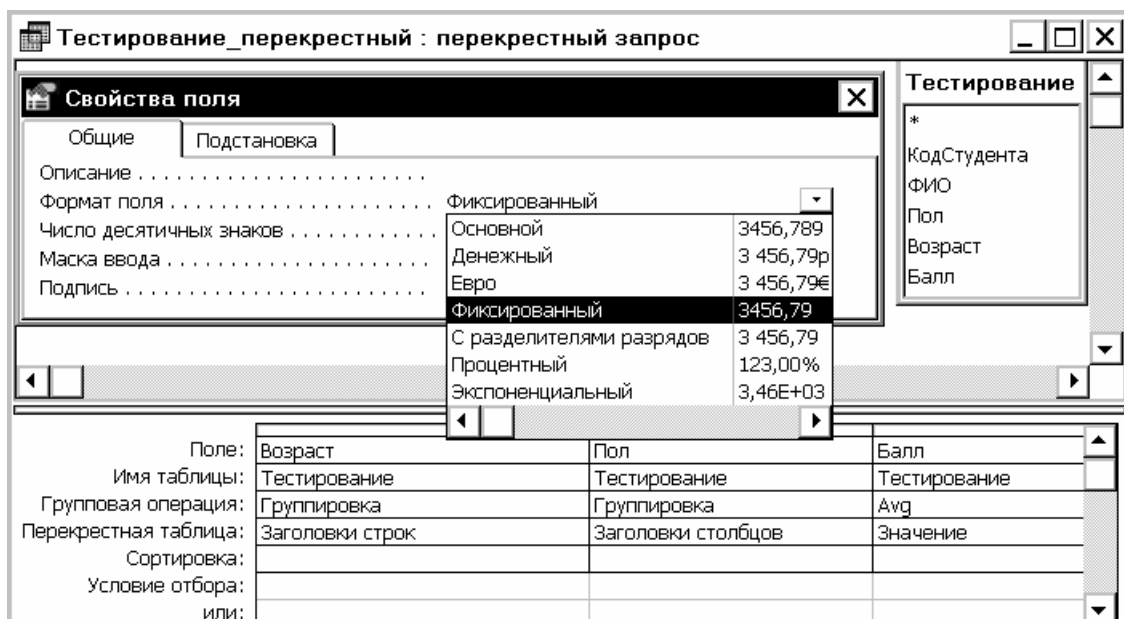


Рис. 4.36. Выбор формата для поля

#### 4.7.2. Создание перекрестного запроса на базе простого

Рассмотрим пример создания перекрестного запроса на базе другого запроса. Пусть в базе данных *ПоставщикиДетали* необходимо проанализировать поставки различных деталей от различных поставщиков. Создадим простой запрос, в который включим поле с названием поставщика *Поставщик* из таблицы *Поставщики*, поле *Название* из таблицы *Детали* и поле *Количество* из таблицы *Поставки*. Содержимое этого запроса показано на рис.4.37.

Название	Поставщик	Количество
Шпилька	Лента	1200
Муфта	Бриз	70
Поршень	Союз	20
Поршень	Астра	50
Ось	Витязь	30
Шкворень	Союз	90
Гайка	Виктория	1500
Гайка	Полет	500
Штуцер	Виктория	300
Болт	Прииск	1500
Шайба	Астра	500
*		

Рис. 4.37. Простой запрос *ДеталиПоставщиков*

Запустим мастер перекрестных запросов. На первом диалоге в качестве источника данных указываем **Запросы** и выбираем запрос *ДеталиПоставщиков* (рис.4.38).

**Создание перекрестных таблиц**

Выберите таблицу или запрос, поля которых необходимо вывести в перекрестном запросе.

Для включения полей из нескольких таблиц сначала создайте обычный запрос, содержащий все необходимые поля.

Показать:  Таблицы  Запросы  Все

Образец:

	Заголовок1	Заголовок2	Заголовок3
	ИТОГИ		

Отмена < Назад Далее > Готово

Рис. 4.38. Выбор источника данных для перекрестного запроса

На втором диалоге выбираем поле для заголовков строк (рис.4.39).

**Создание перекрестных таблиц**

Выберите поля, значения которых будут использованы в качестве заголовков строк.

Допускается выбор не более трех полей.

Выберите поля по порядку сортировки данных. Например, можно сначала выполнить сортировку значений по странам, а затем по городам.

Доступные поля:  Название  Количество

Выбранные поля:  Поставщик

Образец:

Поставщик	Заголовок1	Заголовок2	Заголовок3
Поставщик1	ИТОГИ		
Поставщик2			
Поставщик3			

Отмена < Назад Далее > Готово

Рис. 4.39. Выбор поля для заголовков строк

На третьем диалоге выбираем поле для заголовков столбцов запроса (рис.4.40).

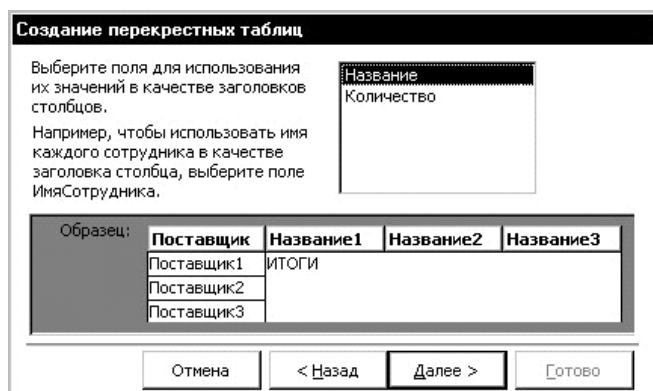


Рис. 4.40. Выбор поля для заголовков столбцов

На четвертом диалоге (рис.4.41), выбираем функцию **Сумма**, вычисляющую значения в ячейках запроса, отказываемся от вывода итоговых значений для строк, так как суммирование деталей разных типов не имеет смысла.

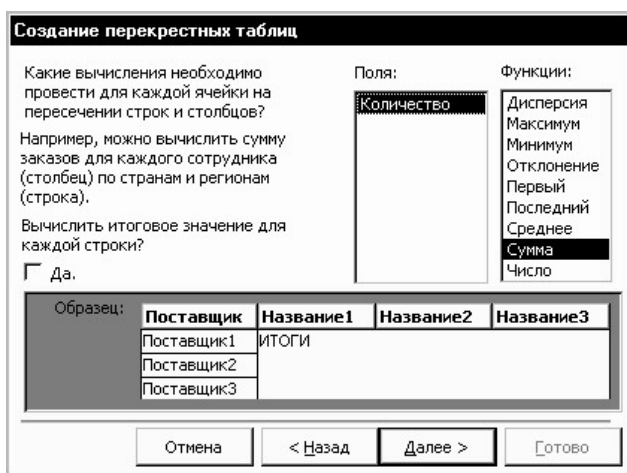


Рис. 4.41. Выбор функции для вычисления в перекрестном запросе

На последнем диалоге соглашаемся с предложенным именем запроса *ДеталиПоставщиков\_перекрестный*. Содержимое этого запроса показано на рис.4.42.

Поставщик	Болт	Гайка	Муфта	Ось	Поршень	Шайба	Шкворень	Шпилька	Штуцер
Астра					50	500			
Бриз			70						
Виктория		1500							300
Витязь				30					
Лента								1200	
Полет		500							
Прииск	1500								
Союз					20		90		

Запись: 1 из 8 Нет фильтра Поиск

Рис. 4.42. Перекрестный запрос

Перекрестный запрос представляет информацию в обобщенном и легко обозримом виде, позволяет провести анализ данных в различных разрезах.



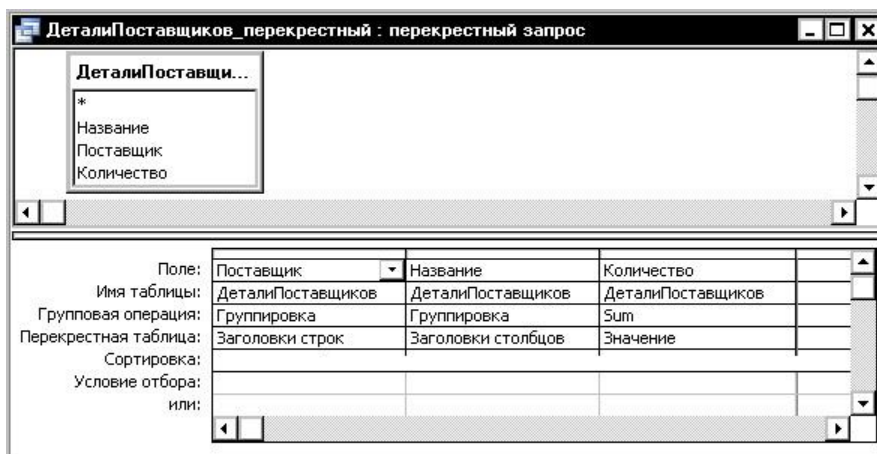


Рис. 4.43. Перекрестный запрос в режиме конструктора

На рис.4.43 показан созданный перекрестный запрос в режиме конструктора. Строка **Перекрестная таблица** появляется в бланке запроса после выбора соответствующей строки в списке кнопки **Тип Запроса** на вкладке конструктор (рис. 4.44).

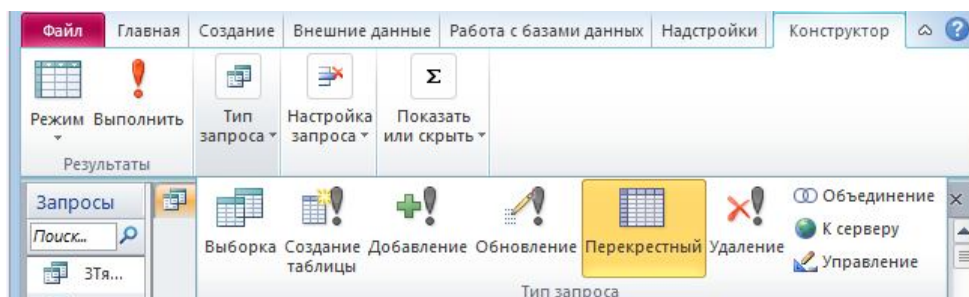


Рис. 4.44. Список кнопки **Тип запроса**

## 4.8. Запросы на изменение данных

При работе с запросом в режиме конструктора существующий запрос можно преобразовать к другому виду, воспользовавшись командами из группы **Тип запроса** (рис. 4.44). Рассмотрим примеры применения этих команд.

### 4.8.1. Создание таблицы из запроса

Когда активно используется база данных, количество записей в ней растет, что ведет к замедлению работы с ней, так как тратится много времени на переписывание данных с диска в оперативную память и обратно. Целесообразно периодически удалять неактуальные данные из базы, но делать это надо осторожно, так как удаленные из таблиц записи восстановить невозможно. Выходом может служить перенос части записей в архивную таблицу, чтобы при необходимости старые записи можно было восстановить.

В качестве примера перенесем из таблицы *Поставки* записи с датами более ранними, чем 1 июля 2009 г. в отдельную архивную таблицу.

Создадим простой запрос, в который включим все поля таблицы *Поставки*. В строке **Условие отбора** для поля *Дата* введем выражение  $<1.07.2009$  (рис.4.45). Значение даты будет автоматически окаймлено символами  $\#$ . Это условие означает, что в запрос следует отобразить записи с датами более ранними (меньшими), чем 1 июля 2009г. Созданный запрос можно предварительно просмотреть.

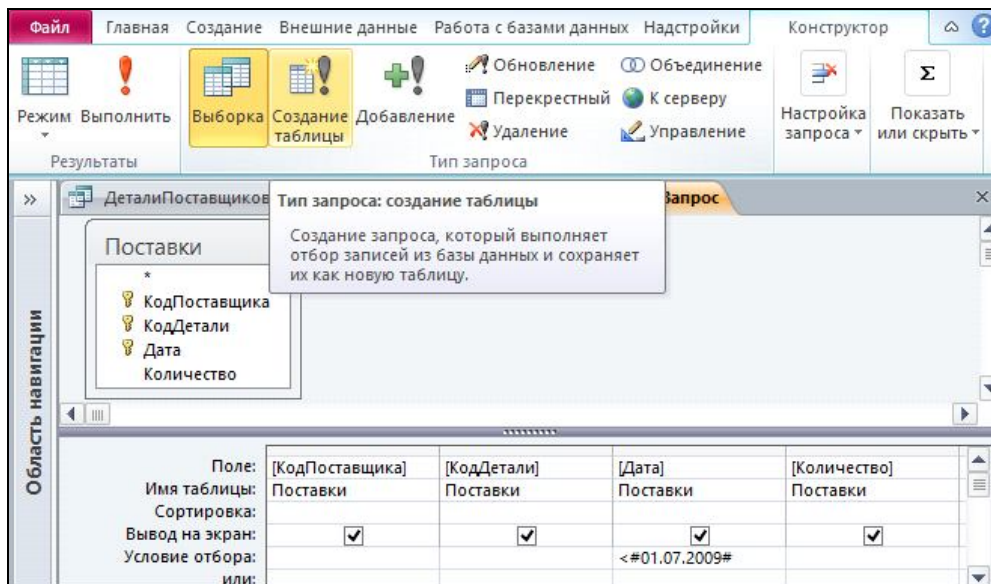


Рис. 4.45. Создание новой таблицы на основе запроса

Выберем, затем, в группе **Тип запроса** команду **Создание таблицы** и зададим имя создаваемой таблице (рис.4.46).

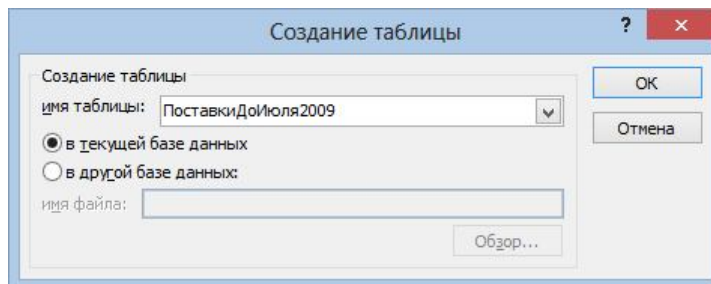


Рис. 4.46. Задание имени для новой таблицы

Таблица из запроса создается командой **Выполнить** из группы **Результаты**, на кнопке которой изображен восклицательный знак. В процессе создания таблицы выводится сообщение о числе помещенных в нее записей (рис.4.47).

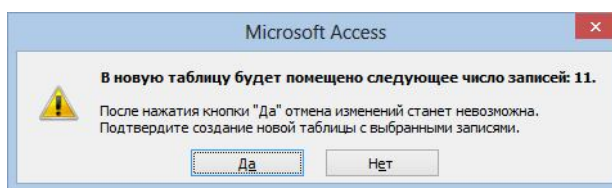


Рис. 4.47. Создание новой таблицы

Созданная таблица показана на рис.4.48

КодПоставки	КодДетали	Дата	Количество
13	6	11.03.2009	1200
20	5	31.03.2009	70
7	26	26.11.2009	20
4	26	25.11.2009	50
21	27	26.11.2009	30
7	25	11.04.2009	90
3	10	08.10.2008	1500
8	10	07.10.2008	500
3	29	23.11.2010	300
1	11	11.01.2012	1500
4	13	13.01.2012	500
*			

Рис. 4.48. Таблица, полученная из запроса

#### 4.8.2. Запрос на удаление данных

Удаленные записи восстановить нельзя, поэтому перед удалением их нужно тщательно просмотреть. Нами были сохранены данные о поставках до 1 июля 2009 г. в отдельной таблице, но они остались и в исходной таблице *Поставки*. Удалим их из нее.

Откроем запрос *ПоставкиЗапрос* в режиме конструктора (рис.4.49).

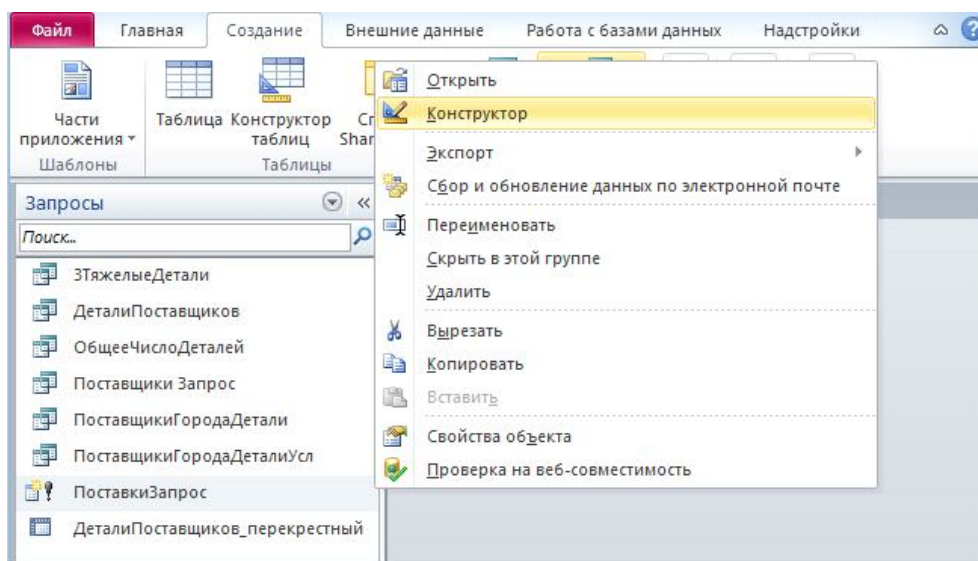


Рис. 4.49. Открытие существующего запроса в режиме конструктора

Оставим в нем то же ограничение на дату и выполним команду **Удаление** из группы **Тип запроса**. После этого в бланке запроса появится строка **Удаление**, в ячейках которой будет введено значение **Условие**. Это значит, что можно задать дополнительные условия отбора записей (рис.4.50).

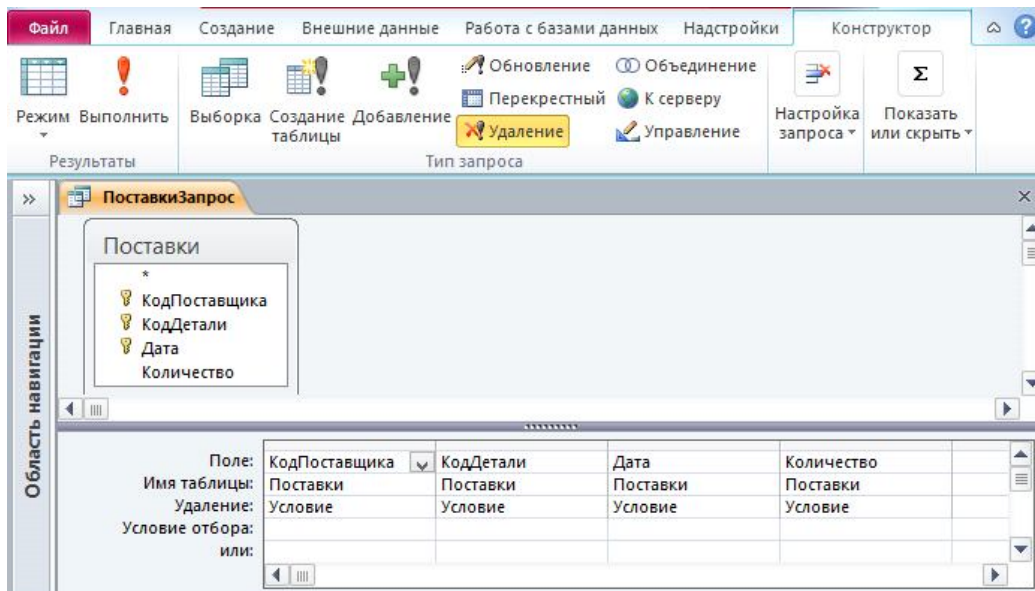


Рис. 4.50. Запрос на удаление в конструкторе

Перед удалением записей еще раз посмотрим запрос в режиме таблицы, он должен иметь вид как на рис.4.48. Выполним запрос на удаление командой **Выполнить**. Access выведет сообщение о числе удаляемых записей и потребует подтвердить удаление.

### 4.8.3. Запрос на добавление

С помощью запроса на добавление записи из одной таблицы можно поместить в конец другой таблицы. Поместим, например, записи из таблицы *ПоставкиДоИюля2009* с датами позже февраля 2009г. обратно в таблицу *Поставки*.

Создадим запрос, в который включим все поля таблицы *ПоставкиДоИюля2009* и зададим условие отбора по дате: `>=#01.03.09#`, которое позволит отобрать поставки, состоявшиеся позже 1 марта 2009г. Затем выполним команду **Добавление** (рис.4.51)

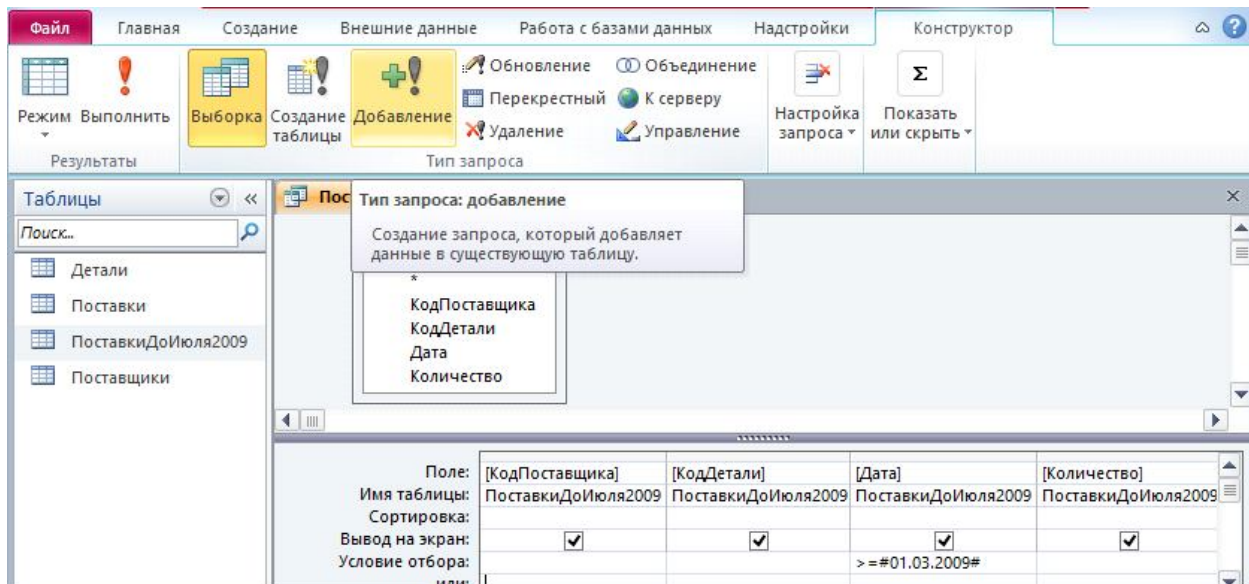


Рис. 4.51. Создание запроса на добавление

В окне **Добавление** (рис.4.52), выберем таблицу *Поставки*, в которую будут помещены отобранные записи. Запрос на добавления выполняется командой **Выполнить**, при этом выводится сообщение о количестве добавляемых записей и предлагается подтвердить запрос.

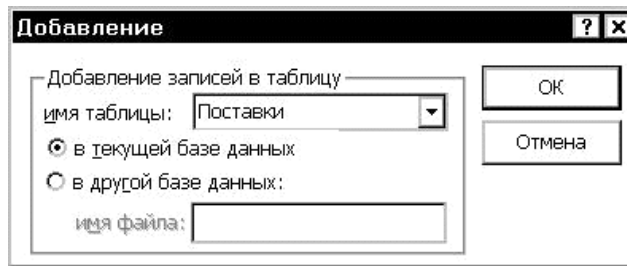


Рис. 4.52. Выбор таблицы для добавления

#### 4.8.4. Запрос на обновление

С помощью запроса на обновление можно изменить значения любого поля таблицы для записей, удовлетворяющих заданным критериям

Пусть принято решение увеличить диапазон значений поля *Рейтинг* таблицы *Поставщики* в 10 раз, чтобы более дифференцированно характеризовать поставщиков. Создадим простой запрос, в который включим поле *Рейтинг* таблицы *Поставщики*. Откроем этот запрос в режиме конструктора и выполним команду **Обновление** (рис.4.53).

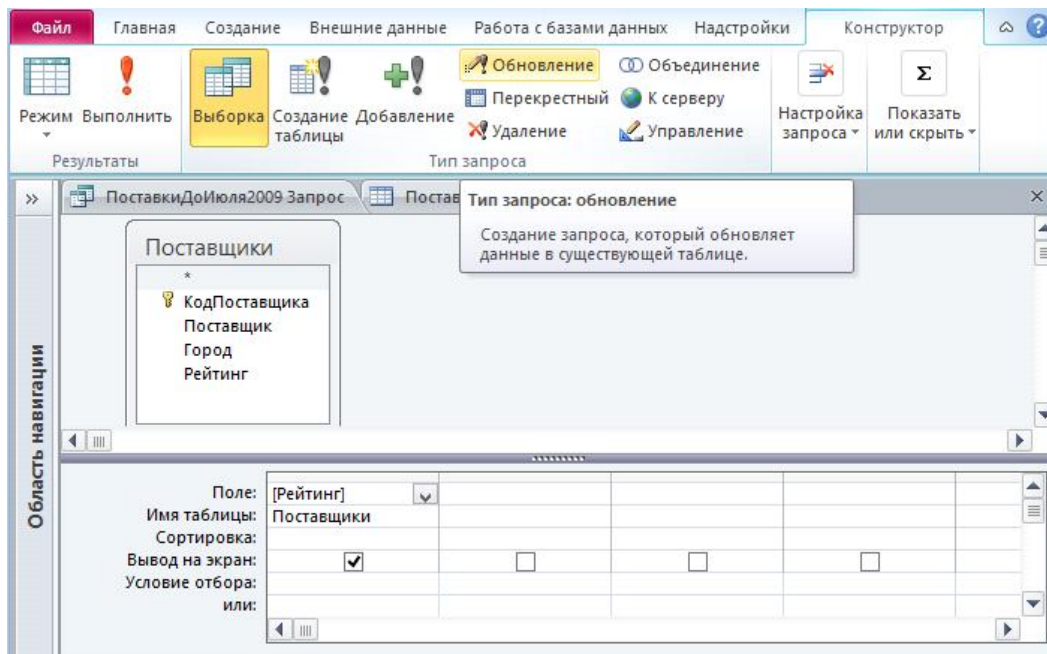


Рис. 4.53. Создание запроса на обновление

В бланк запроса добавляется строка **Обновление**, в которую вводятся значения или выражения, определяющие способ изменения обновляемых полей. В строке **Обновление** введем выражение  $[Рейтинг]*10$ , значение которого будет использовано как новое значение поля *Рейтинг* (рис.4.54).

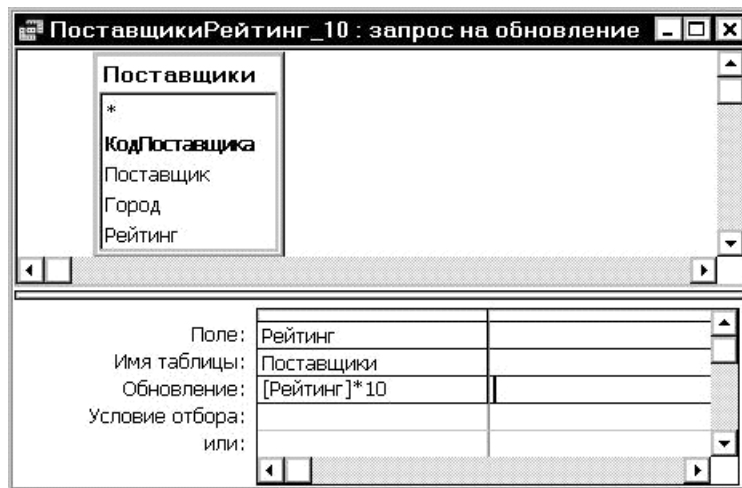


Рис. 4.54. Запрос на обновление в режиме конструктора

Если перевести данный запрос в режим таблицы, то запланированные изменения не будут показаны, а будут выведены прежние значения изменяемого поля. Чтобы в таблицу были внесены изменения, нужно выполнить запрос командой **Выполнить**. Access выдаст сообщение о количестве изменяемых записей и попросит подтвердить внесение изменений. Таблица *Поставщики* после выполнения запроса на изменение показана на рис.4.55.

КодПоставки	Поставщик	Город	Рейтинг
9	Альбатрос	Н.Новгород	560
4	Астра	Москва	300
20	Бриз	Павлово	600
3	Виктория	Кстово	400
21	Витязь	Муром	550
16	Возрождение	Н.Новгород	600
5	Глория	Казань	200
12	Коктейль	Самара	700
13	Лента	Н.Новгород	800
2	Макрос	Н.Новгород	800
17	Полет	Н.Новгород	800
8	Полет	Н.Новгород	260
1	Прииск	Н.Новгород	900
7	Союз	Кстово	800
10	Спутник	Дзержинск	300
14	Эврика	Н.Новгород	500
6	Юнайт	Н.Новгород	450
*	(№)		

Рис. 4.55. Таблица после выполнения запроса на изменение поля *Рейтинг*

## 5. Формы

*Форма* – это средство, упрощающее ввод, редактирование и отображение информации, хранящейся в таблицах базы данных или в запросах. Форма представляет собой окно с набором *элементов управления (ЭУ)*, которые используются для отображения информации, ввода и изменения данных, выполнения определённых действий или просто для украшения. Примерами элементов управления являются кнопки, надписи, текстовые поля, линии, рамки, списки и т.д.

Работа с формами выполняется инструментами группы **Формы** вкладки **Создание** (рис.5.1).

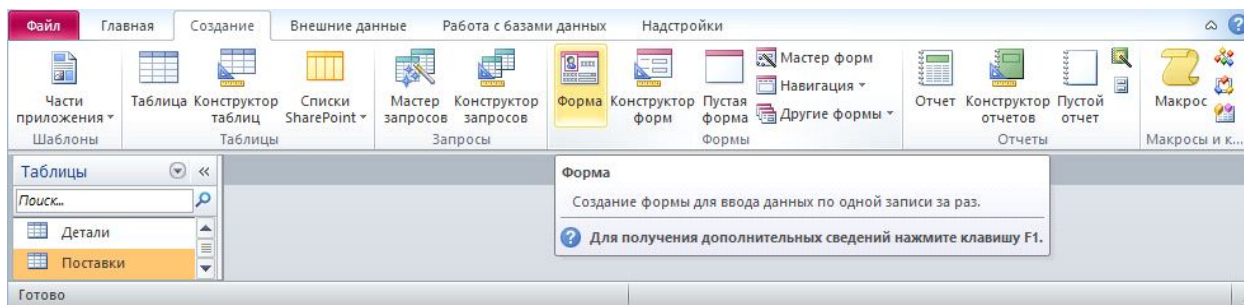


Рис. 5.1. Команды для работы с формами

### 5.1. Инструмент Форма

С помощью инструмента **Форма** создается форма для выбранного источника данных для отображения одной записи.

Выберем таблицу *Поставки* и нажмем кнопку **Форма**. Для выбранной таблицы автоматически создается форма (рис.5.2). Видно, что на форме размещены надписи с названиями полей таблицы и текстовые поля, отображающие значения полей. Хотя в полях *КодПоставщика* и *КодДетали* хранятся числовые значения соответствующих кодов, на форме отображаются названия поставщиков и деталей, доступные благодаря подстановке (см. п. 3.7.3).

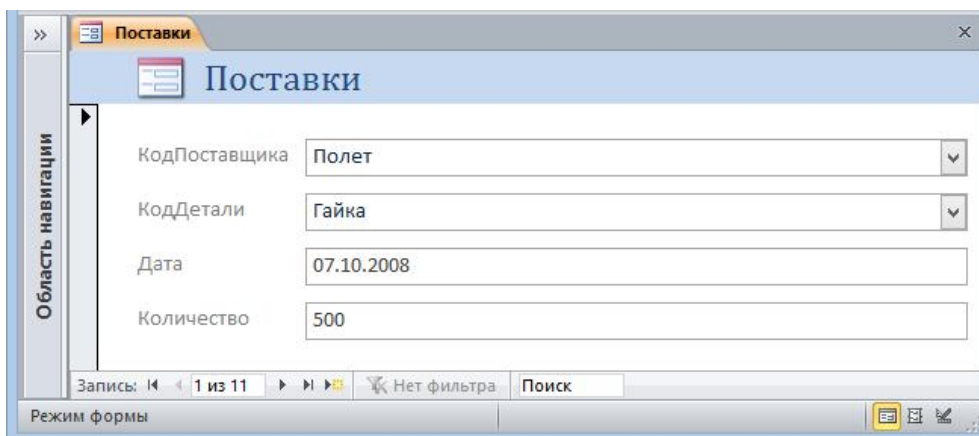


Рис. 5.2. Автоматически созданная форма для таблицы *Поставки*

Создадим теперь форму для таблицы *Поставщики*. Выберем эту таблицу и нажмем кнопку **Форма**. Результат показан на рис.5.3. Видно, что на форме отображается запись из таблицы *Поставщики* и записи из связанной таблицы *Поставки*.

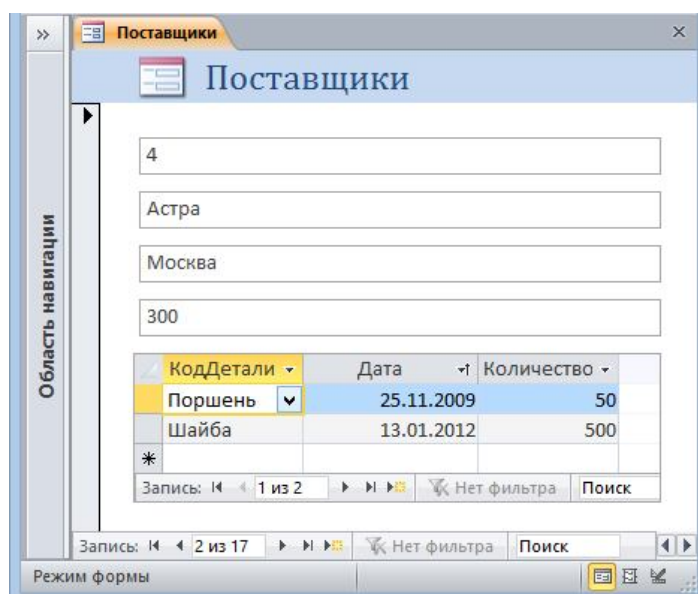


Рис. 5.3. Автоматически созданная форма для таблицы *Поставщики*

## 5.2. Использование формы

На форме, показанной рис. 5.2, имеются *надписи* с именами полей таблицы и *текстовые поля*, в которых отображаются значения полей таблицы. Надписи и текстовые поля являются *элементами управления (ЭУ)*. *Надпись* является пассивным ЭУ, так как она отображает неизменяемый текст. *Текстовые поля* являются активными элементами управления, так же как кнопки, списки и некоторые другие элементы, так как они способны реагировать на нажатия клавиш или щелчки мыши.

Между активными элементами управления, расположенными на форме, можно переходить, нажимая клавишу **Tab** (в одном направлении) или комбинацию клавиш **Shift+Tab** (в другом направлении). Клавиша **PgDown** обеспечивает переход к следующей записи, **PgUp** – к предыдущей.

Перемещаться между записями таблицы, отображаемой в форме, можно с помощью *навигатора*, расположенного в нижней части формы. Назначения кнопок навигатора показаны на рис.5.4.

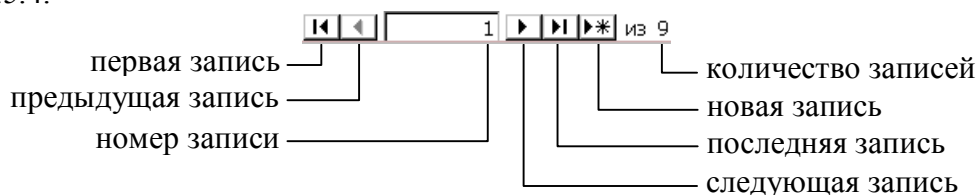


Рис. 5.4. Кнопки навигатора

Значения в текстовых полях можно редактировать, изменяя тем самым значения полей таблицы.

Щелчок на полосе слева на форме обеспечивает выделение целой записи. Работать с выделенной записью можно инструментами из группы **Буфер обмена** на вкладке **Главная**: **Вырезать (Ctrl+X)**, **Копировать (Ctrl+C)**, **Удалить (Del)**. Скопированную в буфер обмена запись можно вставить как значение новой записи командой **Вставить (Ctrl+V)**. Той же командой можно заменить существующую запись содержимым буфера обмена. Заменяемую запись нужно предварительно выделить.

Текст полей можно проверить на наличие грамматических ошибок с помощью инструмента **Орфография** из группы **Записи** на вкладке **Главная** или нажав клавишу **F7**. При обнаружении ошибки появится диалоговое окно **Орфография**, показанное на рис.5.5.



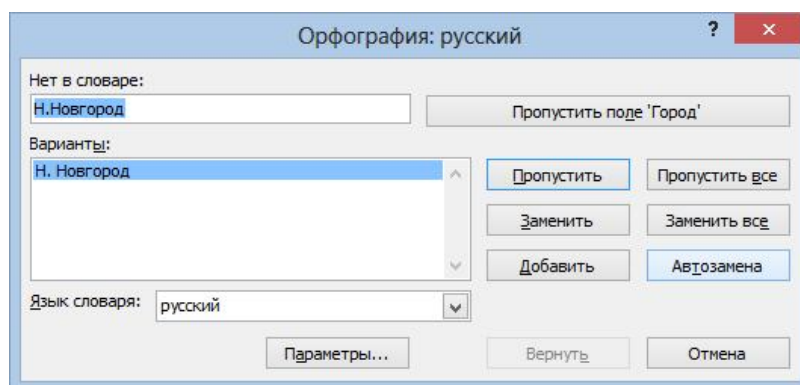


Рис. 5.5. Окно проверки орфографии

В этом окне выводятся слова, возможно содержащие грамматические ошибки, и варианты правильного написания этих слов, если они есть в словаре.

### 5.3. Мастер форм

**Мастер форм** предъявляет несколько диалогов для ввода данных, в соответствии с которыми форма будет создана автоматически.

Построим теперь форму для таблицы *Поставщики* с помощью мастера форм.

На первом диалоге мастера форм (рис.5.6) выбираем таблицу *Поставщики*, затем из списка доступных полей выбираем поля для включения в форму. Выберем все поля таблицы *Поставщики* кнопкой **>>**.

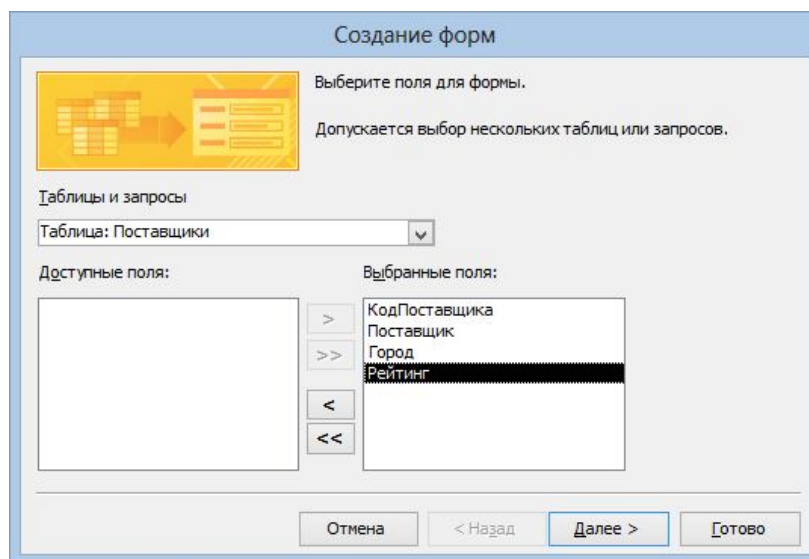


Рис. 5.6. Первый диалог мастера форм. Выбор полей для формы

На втором диалоге (рис.5.7) можно выбрать порядок расположения элементов управления на форме: **в один столбец**, **ленточный**, **табличный**, **выровненный**. Выберем, например, **в один столбец**.

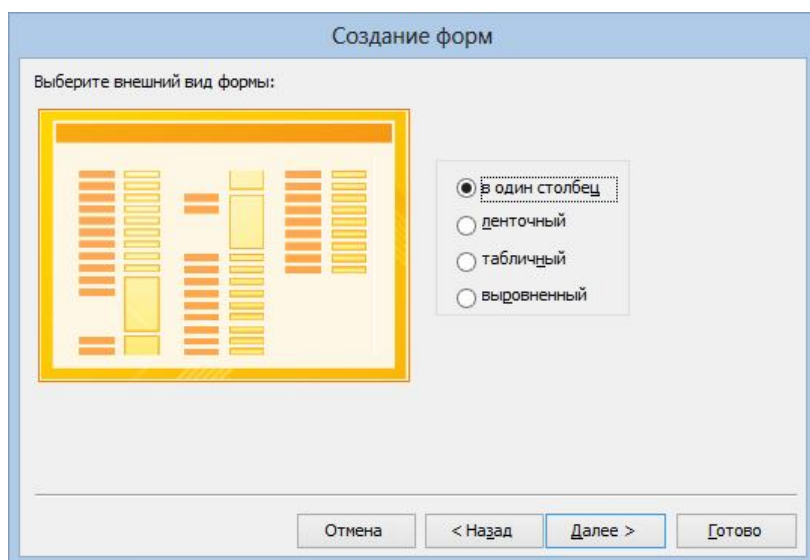


Рис. 5.7. Второй диалог мастера форм. Выбор внешнего вида формы

На третьем диалоге (рис.5.8), надо дать форме имя и выбрать дальнейшие действия. Назовём форму *Поставщики* и выберем переключатель **Открыть форму для просмотра и ввода данных**.

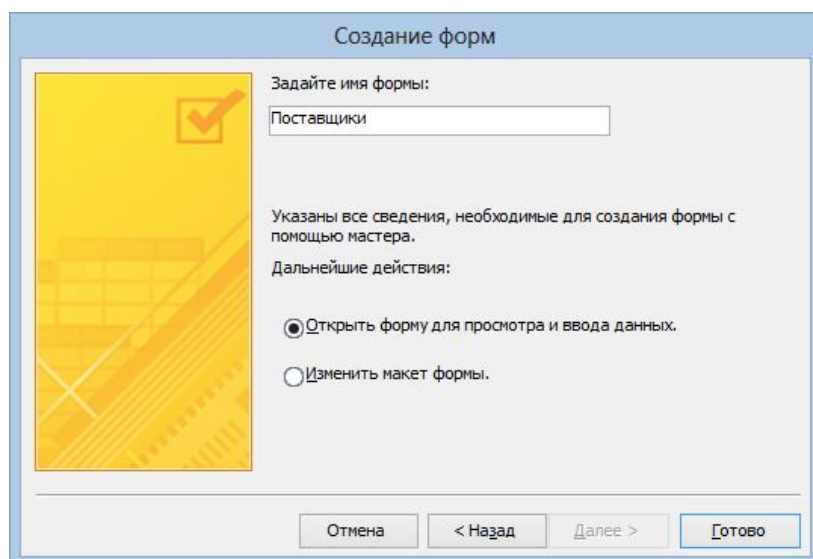


Рис. 5.8. Третий диалог Мастера форм. Задание имени формы

Поскольку в базе данных уже есть форма с именем *Поставщики*, выводится окно с предложением заменить ее новой (рис.5.9). Согласимся на замену.

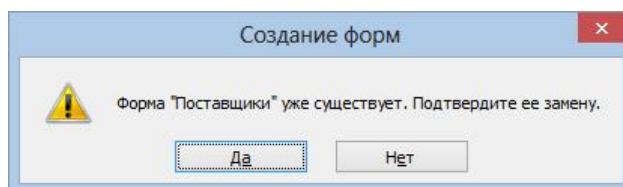


Рис. 5.9. Предложение заменить существующую форму

Созданная форма *Поставщики* показана на рис.5.10.

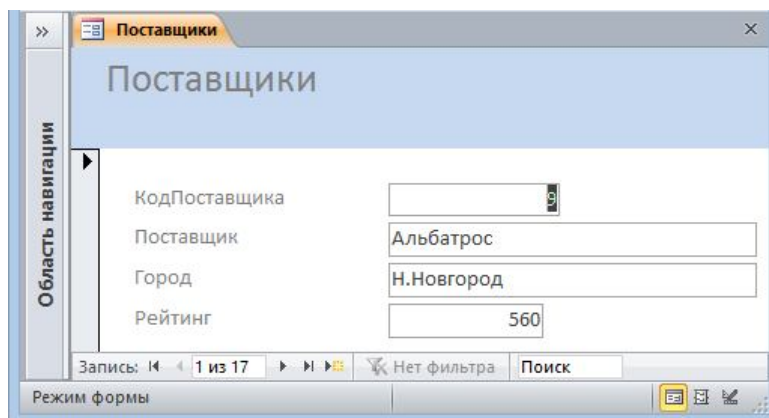


Рис. 5.10. Форма *Поставщики* в режиме формы

Создадим еще с помощью мастера форму для таблицы *Детали*. Эта форма показана на рис.5.11. Для поля, содержащего изображение, на форме размещен соответствующий элемент управления.

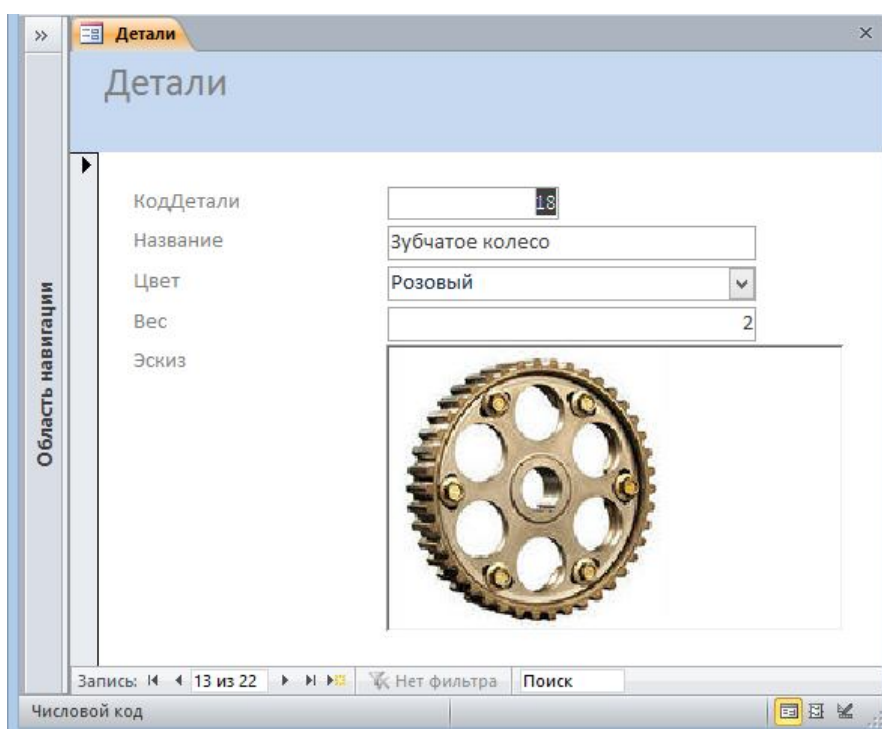


Рис. 5.11. Форма *Детали*

## 5.4. Модификация форм

Изменять форму можно, используя режим макета и режим конструктора. Переключаться между режимами работы с формой можно командами группы **Режим**, командами контекстного меню (рис.5.12), или кнопками в правом нижнем углу формы.

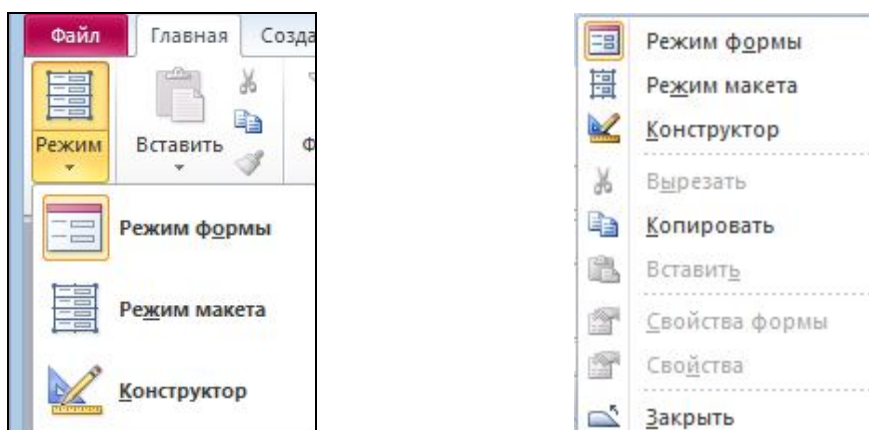


Рис. 5.12. Команды группы **Режим** и контекстного меню

### 5.4.1. Режим макета

В режиме макета на форме отображается информация из таблицы или запроса, и при этом можно редактировать элементы управления.

Откроем в режиме макета форму *Поставки* (рис.5.13).

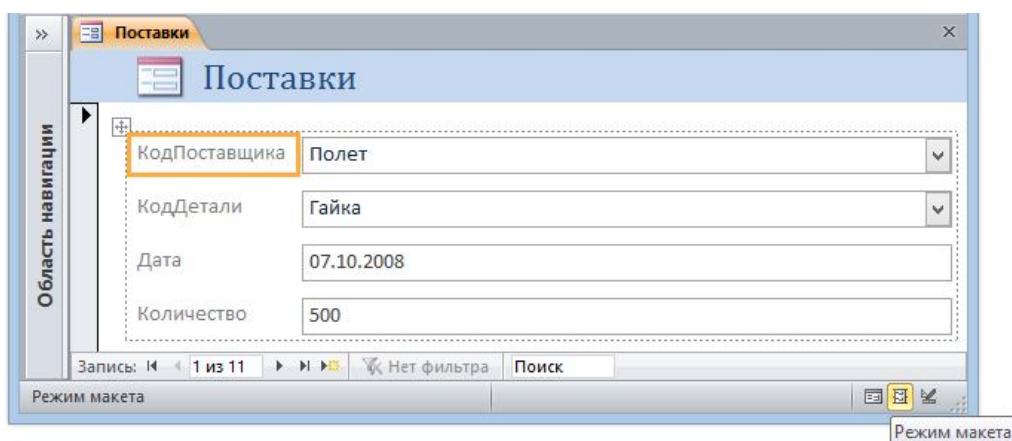


Рис. 5.13. Форма в режиме макета

Выбранный ЭУ выделяется рамкой, его свойства выводятся нажатием клавиши **F4** или инструментом **Страница свойства**. На рис.5.14 показано окно свойств надписи *Надпись0*.

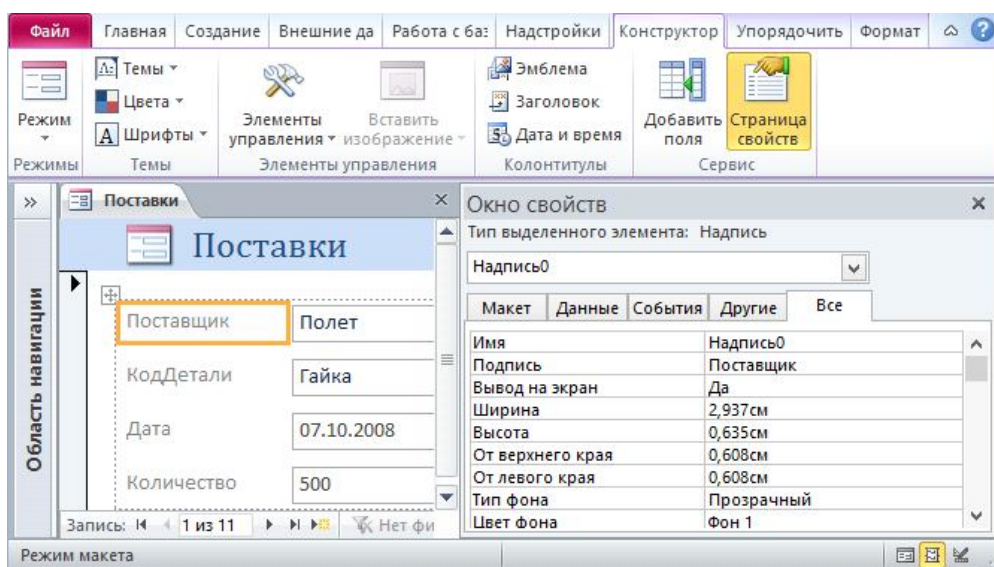


Рис. 5.14. Окно свойств элемента управления

Каждый элемент управления имеет свойство **Имя**, по которому к нему можно обратиться. Свойство надписи **Подпись** содержит отображаемый в надписи текст. Изменим подпись *КодПоставщика* на *Поставщик*, подпись *КодДетали* на *Деталь* и подпись *Дата* на *Дата поставки*. Изменим размеры текстовых полей так, чтобы они соответствовали размерам, отображаемым в них значений полей таблицы. В результате форма должна иметь вид, показанный на рис.5.15.

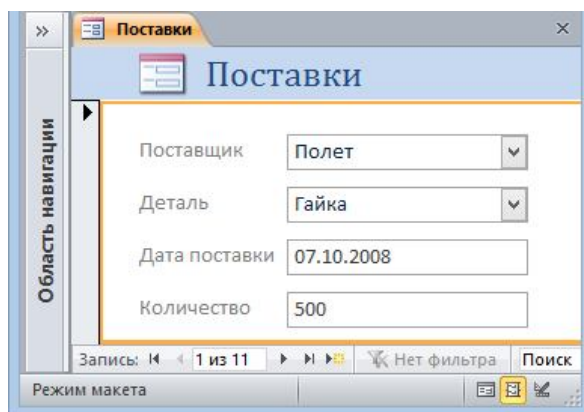


Рис. 5.15. Отредактированная форма *Поставки*

## 5.4.2. Режим конструктора

В режиме **Конструктор** форма создается путем размещения элементов управления на пустом бланке конструктора, рис.5.16.

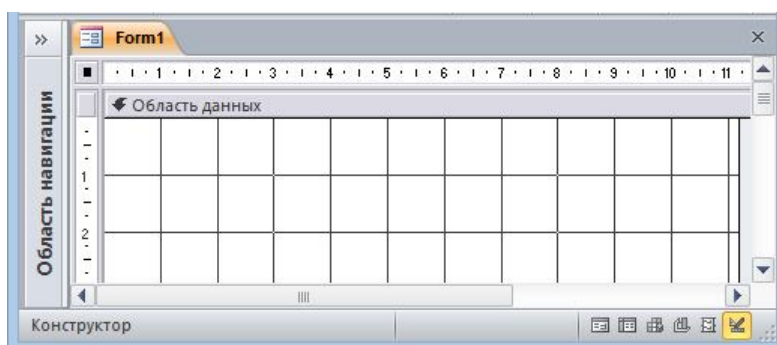



Рис. 5.16. Конструктор форм

Для внесения изменений в существующую форму нужно перейти в режим конструктора командой **Конструктор** контекстного меню или щелчком по соответствующей кнопке . Окно конструктора форм показано на рис.5.17.

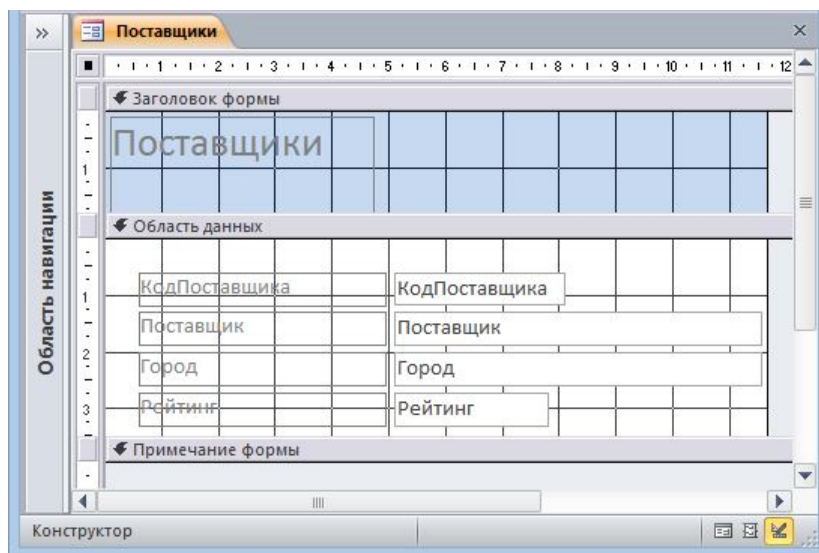


Рис. 5.17. Форма *Поставщики* в режиме конструктора

Бланк конструктора формы состоит из трех разделов. В разделе **Заголовок формы** находится название формы. Раздел **Примечание формы** в данном примере пуст. В этих разделах, помещаются элементы, которые не меняются при переходе от записи к записи. Данные разделы можно размещать на бланке или убирать с бланка конструктора командой контекстного меню **Заголовок/примечание формы** (рис.5.18).

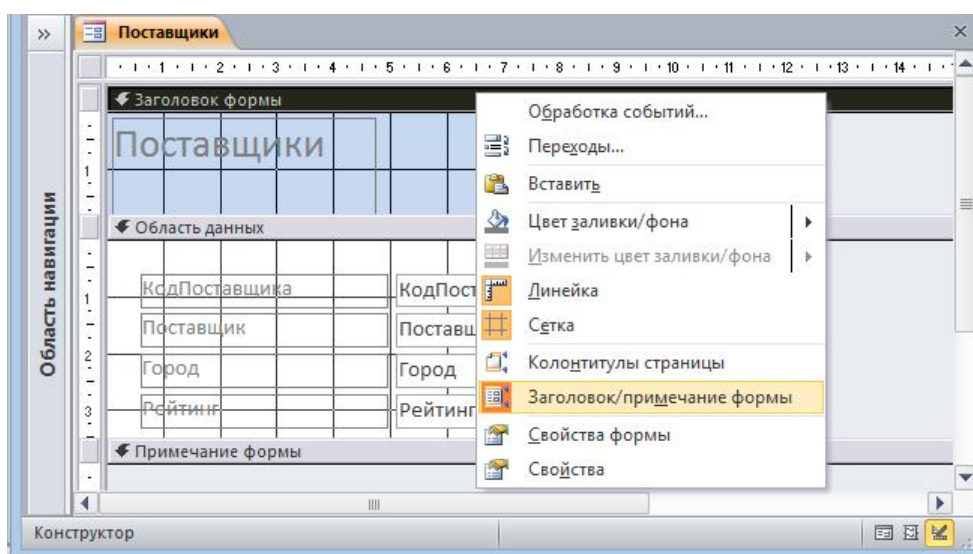


Рис. 5.18. Команда для создания или удаления заголовка и примечания формы

Размеры разделов можно изменять, перетаскивая границу между разделами на линейке, расположенной на бланке конструктора слева. Размеры самой формы также можно изменять, перетаскивая границу между формой и темным фоном справа и снизу от формы.

В разделе **Область данных** для каждого поля таблицы помещена надпись с названием поля таблицы и текстовое поле, в котором отображается содержимое поля соответствующей записи таблицы.

Новые элементы управления помещаются на форму с помощью инструментов, размещённых в группе **Элементы управления** (рис.5.19).

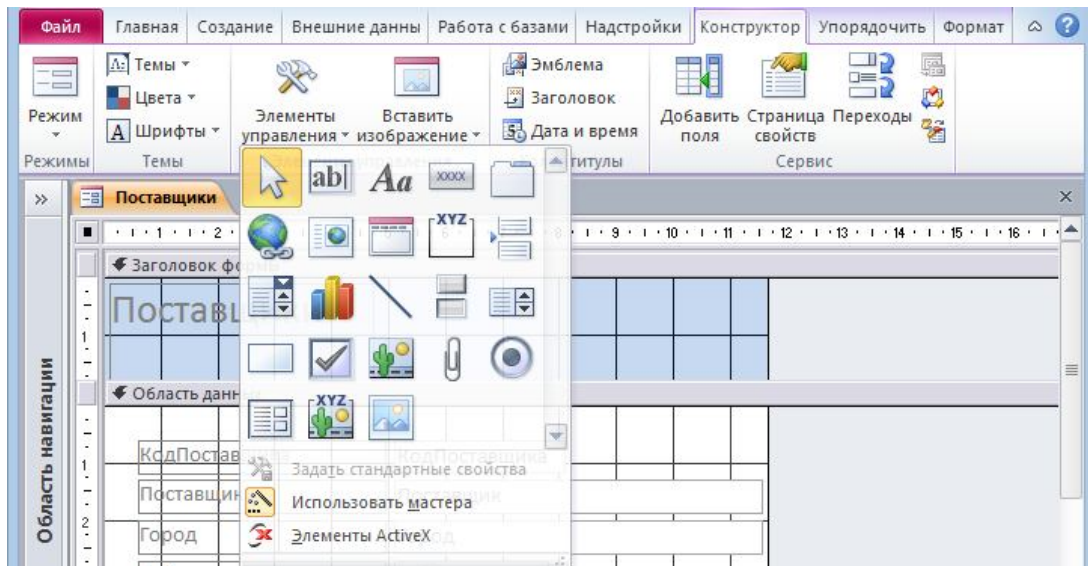



























Рис. 5.19. Элементы управления

Для размещения на форме нового ЭУ его нужно выбрать щелчком мыши на панели элементов, перейти на бланк конструктора, при этом курсор должен принять форму крестика +, и нарисовать прямоугольник, удерживая левую кнопку мыши. В этом прямоугольнике будет размещен созданный ЭУ.

Ниже перечислены кнопки на **Панели элементов** и указано их назначение.

-  – Выбор объектов. Позволяет выделить определённый элемент управления или группу элементов управления, если держать нажатой клавишу **Shift**.
-  – Текстовое поле для ввода и редактирования текста.
-  – Надпись. Описательный текст.
-  – Кнопка.
-  – Вкладка. Позволяет разделить форму на несколько вкладок.
-  – Гиперссылка на веб-страницу, рисунок, адрес электронной почты или программу
-  – Элемент управления типа "Веб-браузер"
-  – Элемент навигации
-  – Группа переключателей.
-  – Вставить разрыв страницы.
-  – Поле со списком. В поле можно ввести новое значение или выбрать существующее из списка.
-  – Диаграмма
-  – Линия.
-  – Выключатель. Может быть в двух состояниях: *включено* и *выключено*.
-  – Список, содержит значения из которых можно сделать выбор.
-  – Прямоугольник.
-  – Флажок. Используется для включения и отключения параметра.
-  – Свободная рамка объекта. Может отображать объект OLE.

-  – Вложение
-  – Переключатель. Несколько переключателей обычно объединяются в группу и позволяют выбрать одно из нескольких взаимоисключающих значений.
-  – Подчинённая форма/отчёт. В форму добавляется информация из дополнительной таблицы.
-  – Присоединённая рамка объекта. Отображает объекты OLE, хранящиеся в записях таблиц, например, рисунки, фотографии.
-  – Рисунок. Не меняется при переходе от одной записи к другой.
-  – Мастера. Если эта кнопка нажата, то при размещении на форме элемента управления запускается мастер, помогающий задать параметры элемента.
-  – Элементы ActiveX. Раскрывает список установленных на компьютере других элементов управления, которые также можно разместить на форме.

Элементы управления на бланке конструктора выделяются щелчком мыши. Если выделен какой-либо элемент, становится доступной закладка **Формат**, на которой расположены инструменты для выбора шрифта, способов выравнивания, цвета и т.д.

### 5.4.3. Добавление связанных элементов управления

Связанный элемент управления – это элемент, однозначно соответствующий полю таблицы или запроса, которые служат источником записей. Он позволяет просматривать или менять значение в связанном поле. Связанные элементы управления добавляются с помощью списка полей, который при работе в конструкторе выводится командой **Добавить поля** (рис.5.20).

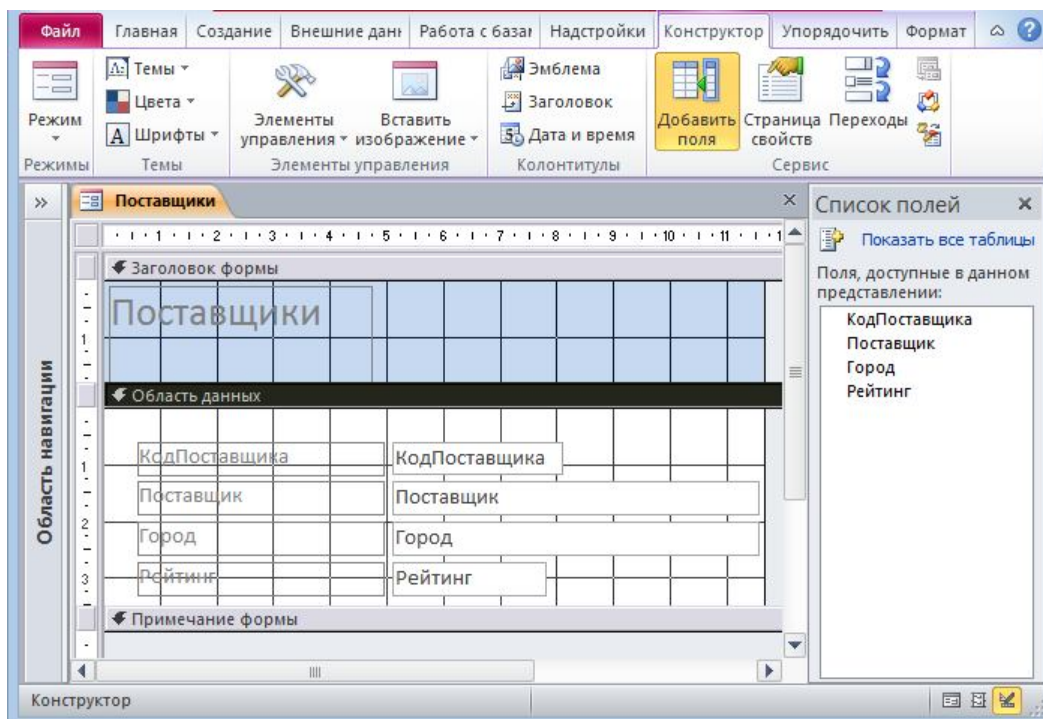


Рис. 5.20. Окно списка полей

Элемент управления, связанный с определённым полем, создаётся перетаскиванием поля на форму. Перед перетаскиванием поля из **Списка полей** предварительно можно выбрать какой-либо элемент управления на панели элементов. Тогда в результате перетаскивания будет создан соответствующий элемент управления, связанный с полем, например, *переключатель* или *список*. По умолчанию при перетаскивании поля на форму создаётся текстовое поле.



#### 5.4.4. Свойства элементов управления

В любой момент времени при работе в конструкторе выделен какой-либо один элемент управления, сама форма или ее раздел. Элемент управления или раздел выделяются щелчком мыши, для выделения формы в целом надо щелкнуть по серому фону незаполненной формой. Окно свойств выделенного элемента управления выводится клавишей **F4**, командой **Страница свойств** на вкладке **Конструктор** или командой **Свойства** контекстного меню (рис. 5.21).

Окно свойств имеет несколько вкладок. На вкладке **Данные** задается источник данных для элемента управления. Это может быть поле таблицы или запроса, например, на рис. 5.21 видно, что текстовое поле связано с полем *КодПоставщика* и отображает его значения. На вкладке **Другие** имеется свойство **Имя**, по которому к элементу управления можно обратиться из макроса или программы на VBA.

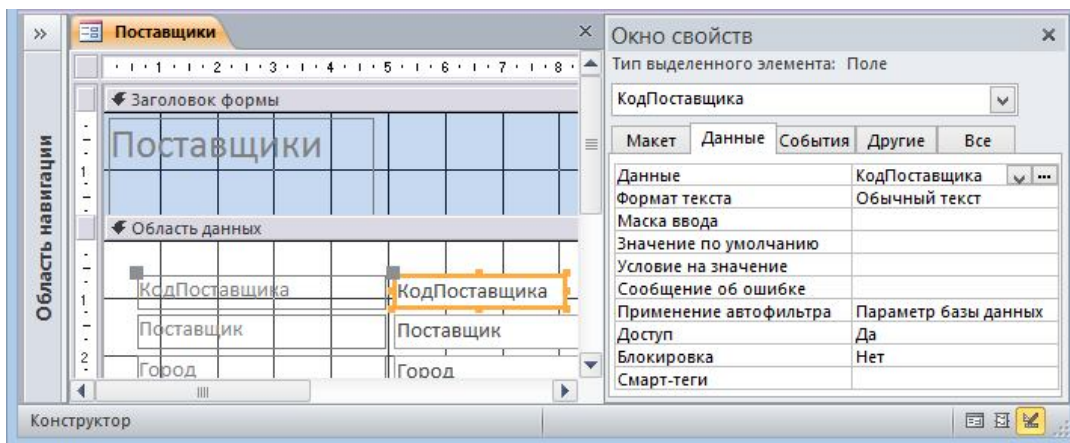


Рис. 5.21. Окно свойств элемента управления

На рис.5.22 показано окно свойств самой формы. Свойство **Подпись** содержит текст, отображаемый в заголовке окна формы.

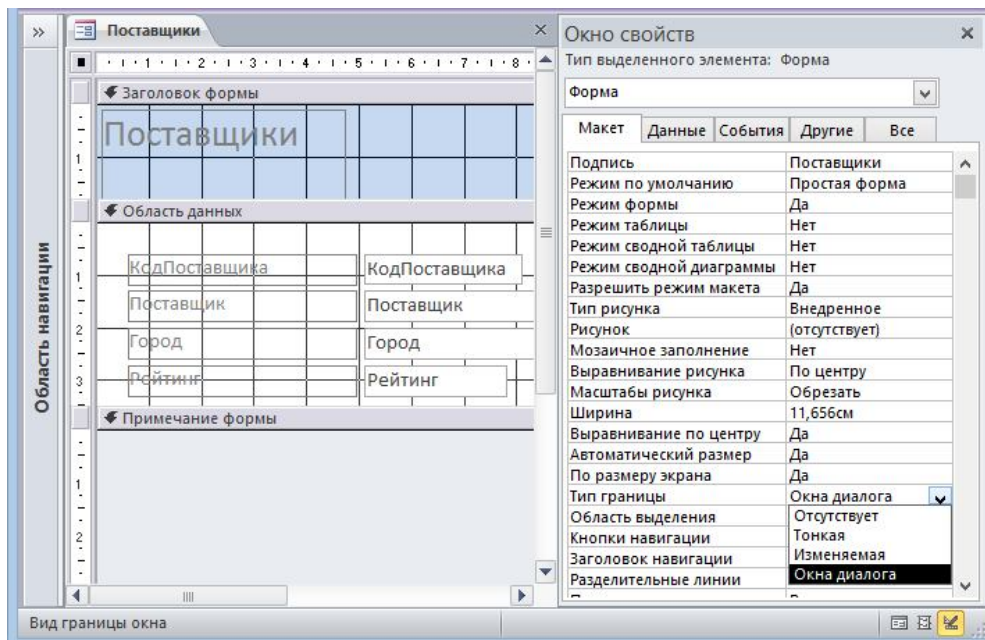


Рис. 5.22. Окно свойств формы

Если для свойства **Тип границы** установить значение **Окна диалога**, то у формы нельзя будет изменять размеры, и вместо трех кнопок в правом верхнем углу останется только кнопка закрытия (рис.5.23).

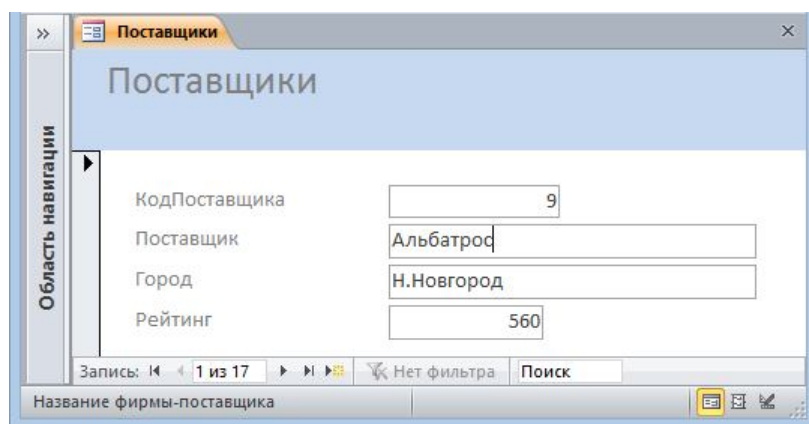


Рис. 5.23. Форма с типом границы **Окна диалога**

### 5.4.5. Форматирование элементов управления

Рассмотренные выше элементы управления можно располагать не только на формах, но и в отчетах. В обоих случаях работа с элементами управления ведется одинаковыми приемами. Настройка элементов ведется путем задания свойств, определяющих их положение, размеры, взаимное размещение, шрифт отображаемого текста, цвет, характер границы.

На макетах отчетов и форм имеется сетка, которую можно использовать для выравнивания ЭУ. Шаг сетки по горизонтали и вертикали устанавливается в окне свойств формы (рис.5.22) заданием свойства **Число делений по X** и **Число делений по Y**. По умолчанию устанавливается 10 делений на сантиметр по горизонтали и по вертикали. При использовании в качестве единиц измерения дюймов по умолчанию устанавливаются 24 деления на дюйм по горизонтали и вертикали.

### 5.4.6. Перемещение

Элементы управления выделяются щелчком мыши. Щелчком, например, по элементам управления для поля *Город* на рис.5.17. Выделенные ЭУ показаны на рис.5.24.



Рис. 5.24. Выделенный элемент управления

Выделенный ЭУ можно перемещать мышью. Часто два отдельных ЭУ образуют связанную группу и перетаскиваются вместе. По отдельности элементы управления из группы перемещаются за маркер перемещения, расположенный в левом верхнем углу.

Можно создать *группу* элементов управления, выделяя их щелчками мыши при нажатой клавише **Shift**. Тогда возможно перемещать одновременно все элементы группы. Ряд действий по выравниванию, рассмотренных ниже, применимы только к группам элементов.

### 5.4.7. Изменение размеров и выравнивание

Для доступа к командам управления размерами и взаимным расположением элементов управления нужно на вкладке **Упорядочить** раскрыть группу **Размер и порядок** и подгруппу **Размер или интервал** (рис.5.25).

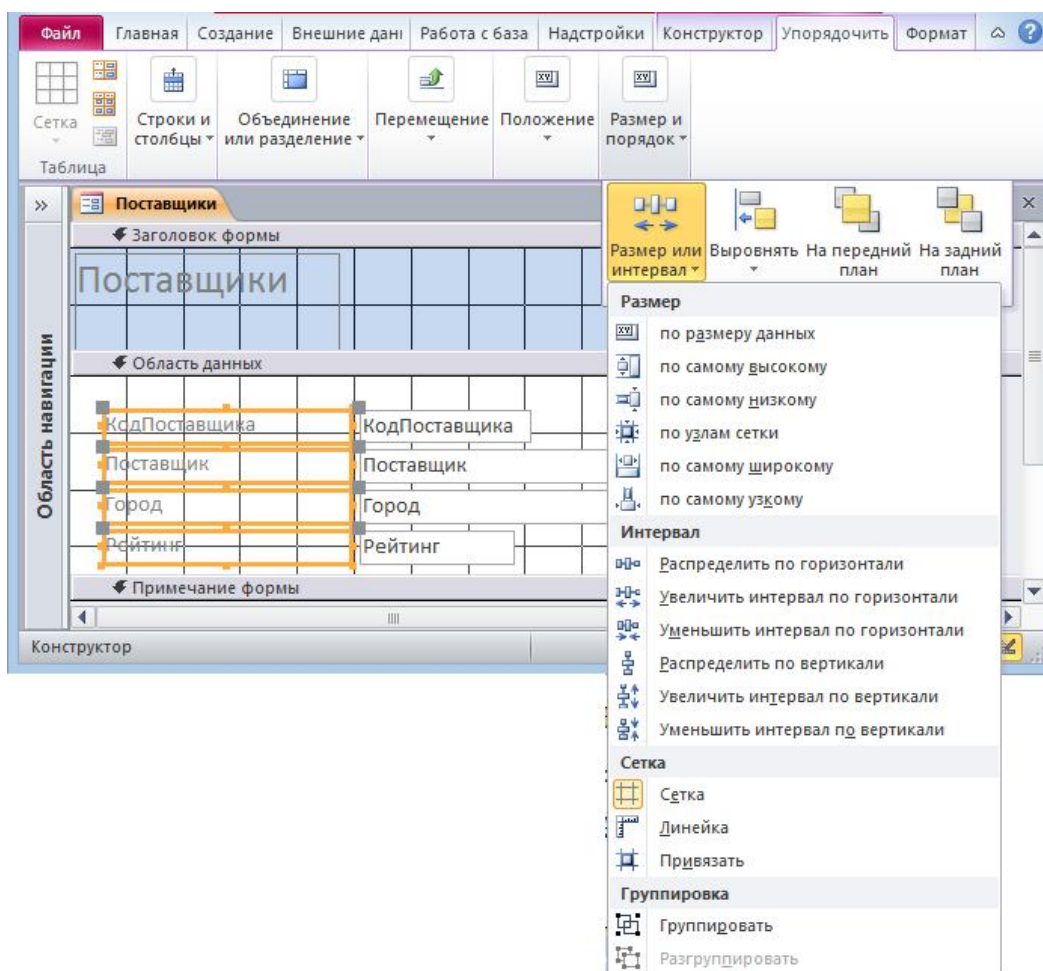


Рис. 5.25. Команды управления размерами и интервалами

Если выполнена команда **Привязать**, то положение и размеры ЭУ устанавливается по узлам сетки. В противном случае элемент может занимать произвольное положение.

Изменять размеры ЭУ можно, перетаскивая маркеры размеров (рис.5.24.), или используя команду **Размер**, которая предлагает выбрать способ задания размера: **по размеру данных** или **по узлам сетки**. Если выделена группа элементов управления, то становятся доступными дополнительные команды установки размеров: **по самому высокому**, **по самому низкому**, **по самому широкому**, **по самому узкому**.

Взаимное расположение элементов управления, включенных в группу, определяется интервалами по вертикали и горизонтали. Для изменения интервалов нужно выполнить команду из группы **Интервал** (рис.5.25).

В подгруппе **Выровнять** находятся команды для выравнивания элементов управления на форме (рис.5.26). Для одиночного элемента управления доступна одна команда - **Выровнять, по узлам сетки**, для группы ЭУ доступны дополнительные команды выравнивания: **Слева**, **Справа**, **Сверху**, **Снизу**.

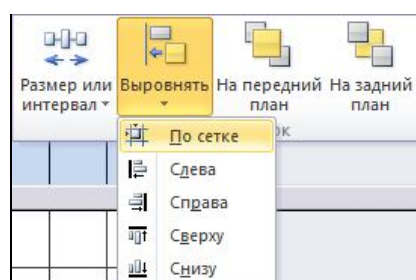


Рис. 5.26. Команды выравнивания элементов

## 5.4.8. Цвет и шрифт

Управление внешним видом выделенного ЭУ производится настройками в окне свойств, которое вызывается нажатием **F4** или командой **Свойства** контекстного меню. Цвет всех компонентов ЭУ можно установить на вкладке **Макет** окна свойств. Можно установить: **Цвет фона**, **Цвет текста**, **Цвет границы** и т.д.

Шрифт текста в элементе управления устанавливается инструментами, расположенными на вкладке **Формат** или кнопками на панели инструментов **Формат**, или на вкладке **Макет** окна свойств, где можно выбрать вид шрифта, его размер и другие свойства.

## 5.5. Подчиненные формы

Если имеются связанные таблицы, то информация из них может отображаться с помощью подчиненных форм. Примером подчиненной формы является форма, созданная автоматически с помощью инструмента **Форма** для таблицы *Поставщики* (рис.5.3). Созданием подчиненных форм можно управлять с помощью мастера форм.

Создадим форму, в которой для каждой детали будут указаны ее поставщики и количество деталей, поставленных каждым поставщиком.

На вкладке **Создание** выберем инструмент **Мастер форм**.

В первом диалоговом окне мастера форм (рис.5.27) выбираем поля *Название*, *Вес*, *Цвет* из таблицы *Детали*; поля *Поставщик*, *Город* из таблицы *Поставщики* и поля *Количество* и *Дата* из таблицы *Поставки*.

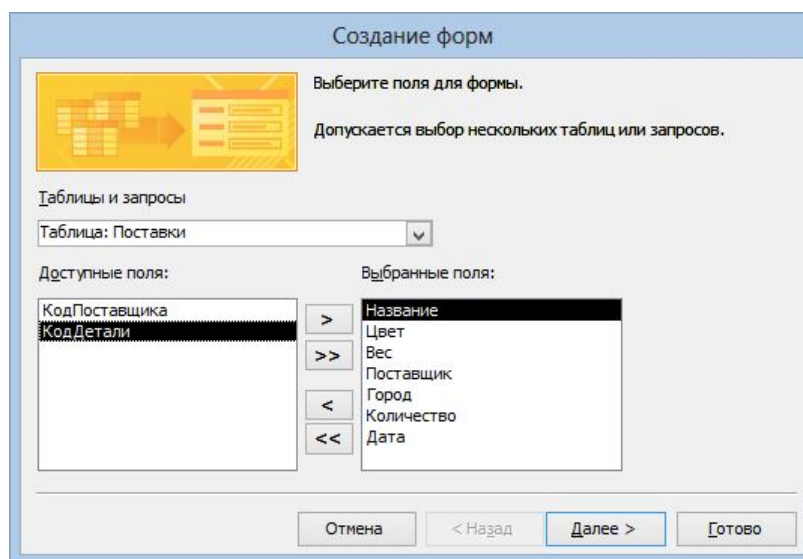


Рис. 5.27. Первый диалог мастера форм. Выбор полей для формы

На втором диалоге можно выбрать главную таблицу (рис.5.28). По умолчанию устанавливается первая выбранная таблица. Выберем также режим **Подчиненные формы**.

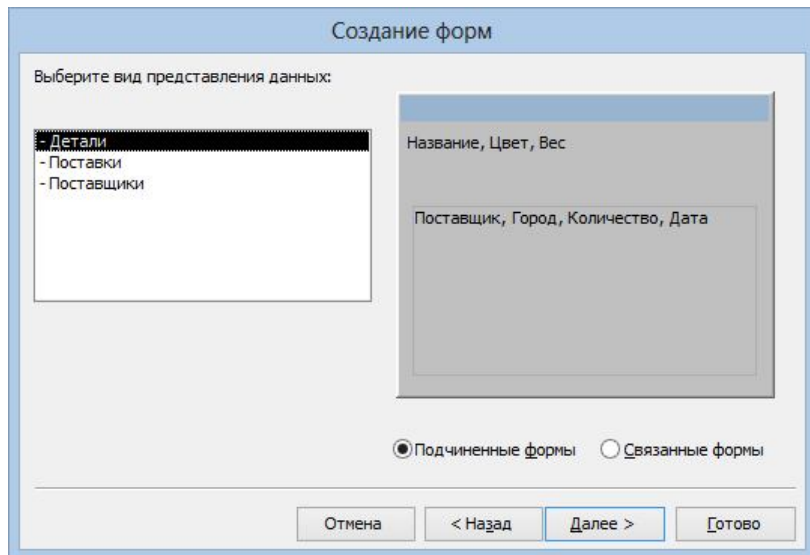


Рис. 5.28. Второй диалог мастера форм. Выбор главной таблицы

На третьем диалоге нужно выбрать вид формы (**ленточный** или **табличный**), выберем **ленточный**.

На четвертом диалоге нужно дать имена для главной и подчиненной форм (рис.5.29).

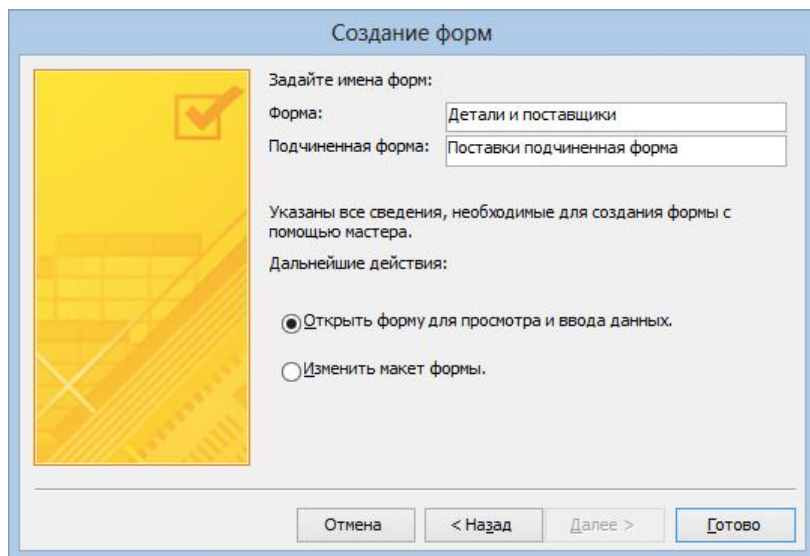


Рис. 5.29. Назначение имен главной и подчиненной формам

Окончательный вид формы представлен на рис.5.30.

Детали и поставщики

Название: Гайка

Цвет: Зеленый

Вес: 1

Поставщик	Город	Количество	Дата
Виктория	Кстово	1500	08.10.2008
Полет	Н.Новгород	500	07.10.2008
*			

Запись: 1 из 2   Нет фильтра   Поиск

Режим формы

Рис. 5.30 Подчиненная форма, созданная с помощью мастера

Отметим, что данная форма более информативна для просмотра данных, чем форма, созданная с помощью мастера автоформ, но не годится для ввода данных, так как содержит не все необходимые поля.

## 6. Отчеты

Отчеты предназначены для создания печатных копий информации из базы данных. К категории отчетов относятся также наклейки. Отчеты хранятся как объекты в базе данных, их можно просматривать на экране и, при необходимости, распечатывать на бумаге.

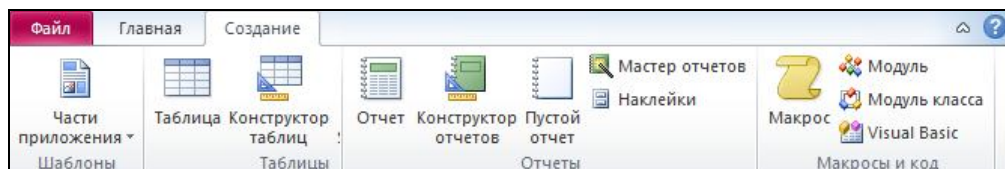


Рис. 6.1. Инструменты для создания отчетов

Для создания отчета нужно выбрать один из инструментов группы **Отчеты** на вкладке **Создание** (рис.6.1):

- **Отчёты** - создание простого отчета на основе данных из текущего запроса или таблицы;
- **Конструктор** - позволяет создавать отчет с нуля на бланке отчета;
- **Пустой отчет** - создание нового пустого отчета, в который можно вставить новые поля;
- **Мастер отчетов** - запуск мастера помогающего настроить отчет при его создании;
- **Наклейки** - предназначены для наклеивания на конверты при рассылке писем адресатам, информация о которых хранится в базе данных.

### 6.1. Инструмент Отчет

Выберем таблицу *Детали* и нажмем кнопку **Отчет** на вкладке **Создание**. На основе выбранной таблицы будет автоматически создан отчет, показанный на рис.6.2.

Название	Цвет	Вес
Винт	Черный	1
Муфта	Коричневый	2
Шпилька	Белый	1
Гайка	Зеленый	1
Болт	Зеленый	2
Шайба	Красный	0,5
Вал	Синий	5
Зубчатое колесо	Розовый	2
Подшипник	Черный	3,3
Шкворень	Фиолетовый	3
Поршень	Желтый	2
Ось	Стальной	0,1
Штуцер	Зеленый	20

Рис. 6.2. Автоотчет для таблицы *Детали*

Отчет может находиться в четырех режимах, выбор которых производится командами из списка кнопки **Режим** на вкладке **Главная** (рис.6.3), кнопками в правом нижнем углу окна отчета или командами контекстного меню.

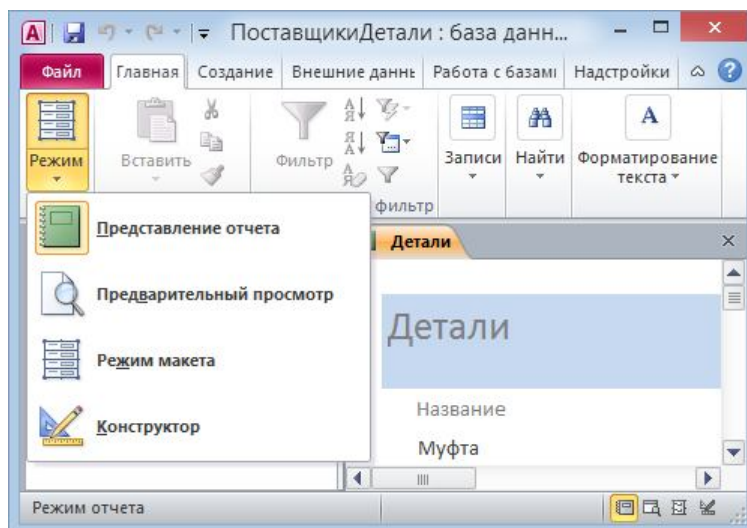


Рис. 6.3. Режимы работы с отчетом

Режим **Представление отчета** предназначен для просмотра отчета в среде Access. В режиме **Предварительный просмотр** дается вид отчета, который получится после его вывода на принтер. **Режим макета** позволяет изменять отчет при одновременном отображении данных. В режиме **Конструктор** можно модифицировать отчет.



## 6.2. Мастер создания отчетов

Пусть нужно создать отчет для базы данных *ПоставщикиДетали*, в котором требуется перечислить все детали, указать для них поставщика и количество поставленных им деталей. Выберем инструмент **Мастер отчетов** (рис. 6.1).

В первом диалоговом окне выбираются таблицы и запросы, на основе которых будет построен отчет, а также их поля. В состав выбранных полей включим поле *Название* из таблицы *Детали*, поле *Поставщик* из таблицы *Поставщики* и поле *Количество* из таблицы *Поставки* (рис.6.4).

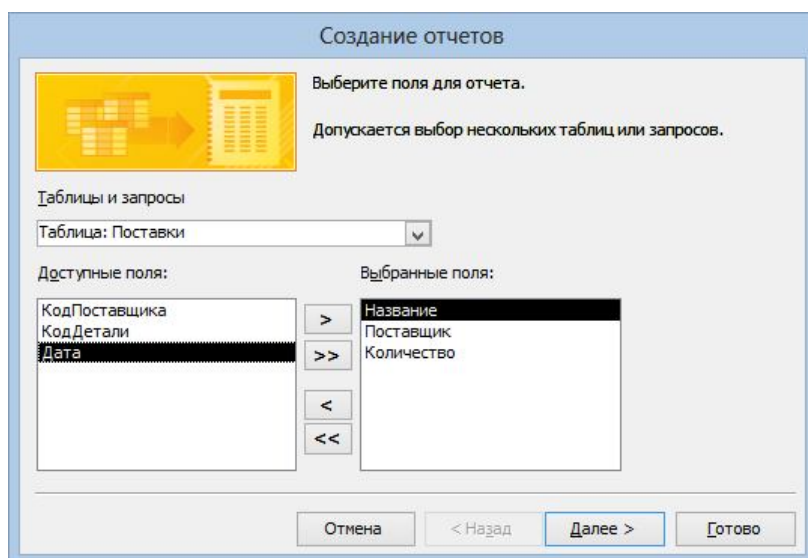


Рис. 6.4. Выбор таблиц и полей для отчета

На втором диалоге, который появляется только в случае, если для отчета выбрано несколько таблиц в предыдущем диалоге, можно установить таблицу для группировки записей в отчете. Если поля в рассматриваемом примере отбирались для включения в отчет в указанном выше порядке, то диалоговое окно будет иметь вид, приведенный на рис.6.5.

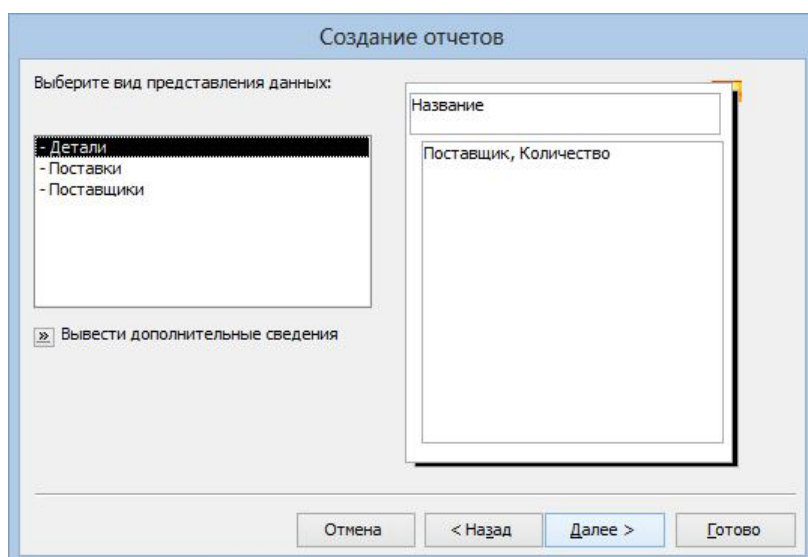


Рис. 6.5. Второй диалог мастера отчетов

По кнопке **Вывести дополнительные сведения** можно получить справочную информацию о создании отчётов (рис.6.6).

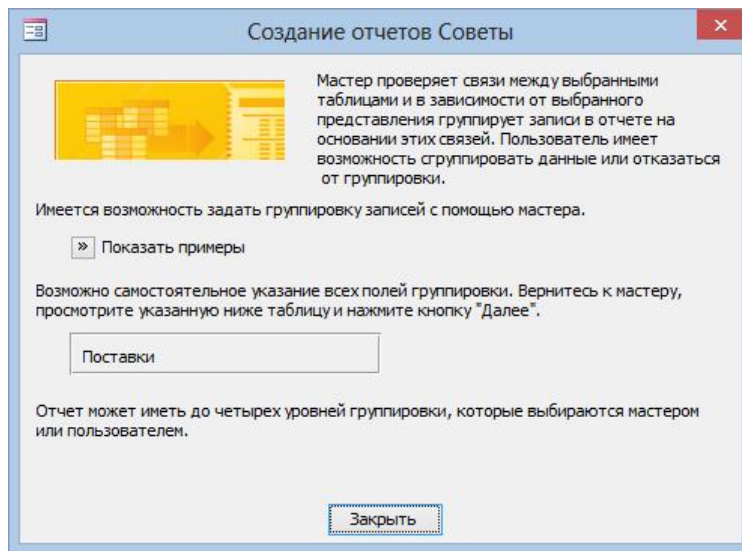


Рис. 6.6. Окно справочной информации по созданию отчетов

В данном примере выбранная первой таблица *Детали* используется для группирования записей отчета по полю *Название*. В отчете будут перечислены сначала все записи по *болтам*, затем по *винтам* и т.д.

Выбранная для группировки таблица выделяется в левом поле диалогового окна на рис.6.5. Здесь для группировки можно выбрать любую таблицу. Если выбрать таблицу *Поставщики*, то схема группировки данных в отчете будет такой, как показано на рис.6.7.

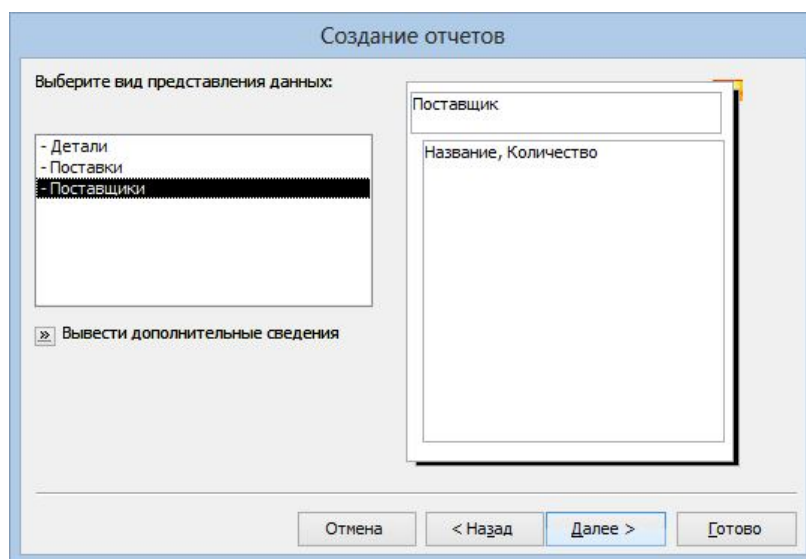


Рис. 6.7. Группировка отчета по полю *Поставщик*

В рассматриваемой базе данных таблицы *Поставщики* и *Детали* имеют связь «один-ко-многим» с таблицей *Поставки*, т.к. каждой записи в этих таблицах может соответствовать несколько записей в таблице *Поставки*, поэтому по ним возможна группировка. Если же для группировки выбрать таблицу *Поставки*, то группировка невозможна и схема отчета будет такой, как показано на рис.6.8.

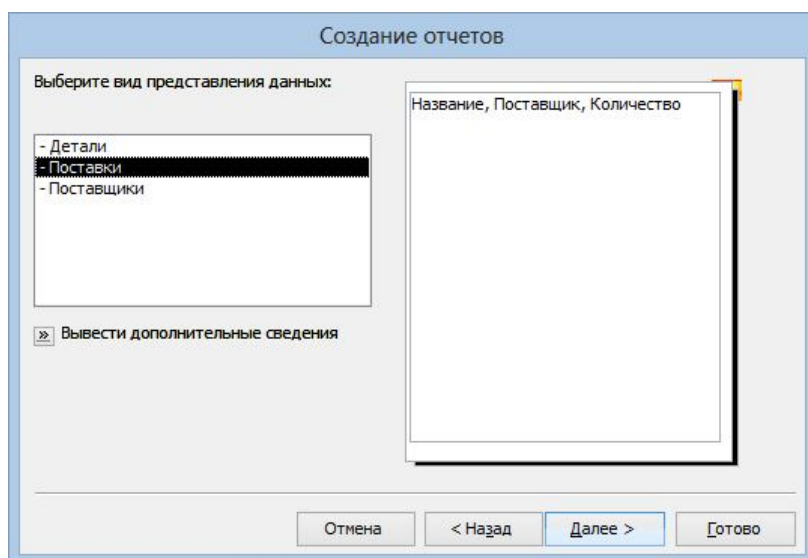


Рис. 6.8. Схема отчета без группировки

Вернемся ко второму диалоговому окну (рис.6.5), нажмем на нем кнопку **Далее**, перейдем к третьему диалогу мастера отчетов (рис. 6.9), на котором можно уточнить параметры группировки записей. Например, можно добавить для группировки поле *Поставщик*, и тогда схема отчета будет иметь вид, показанный на рис.6.9.

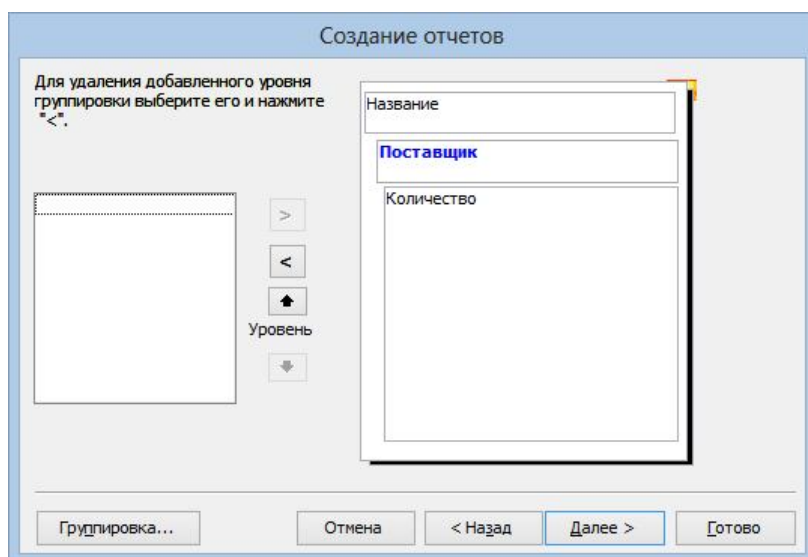


Рис. 6.9. Добавление второго поля для группировки

Добавленное поле *Поставщик* выделяется синим цветом. Кнопками со стрелками можно перемещать добавленное поле по уровням иерархии группировки.

В четвертом диалоговом окне (рис.6.10) можно указать порядок сортировки записей. Группы в отчетах автоматически сортируются по полям, используемым для группировки. На данном диалоге можно дополнительно выбрать для сортировки внутри группы одно или несколько полей, которые не используются для группировки.

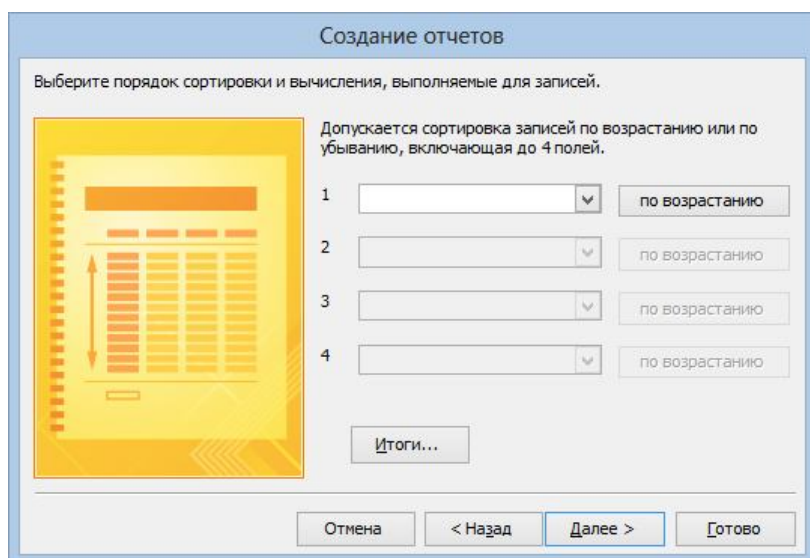


Рис. 6.10. Выбор порядка сортировки записей в отчете

По кнопке **Итоги** открывается диалог **Итоги** (рис.6.11), на котором перечислены все числовые и денежные поля, входящие в информационную часть отчета. В нашем примере это только поле *Количество*. Выберем, например, функцию **Sum**, чтобы узнать общее число деталей каждого вида.

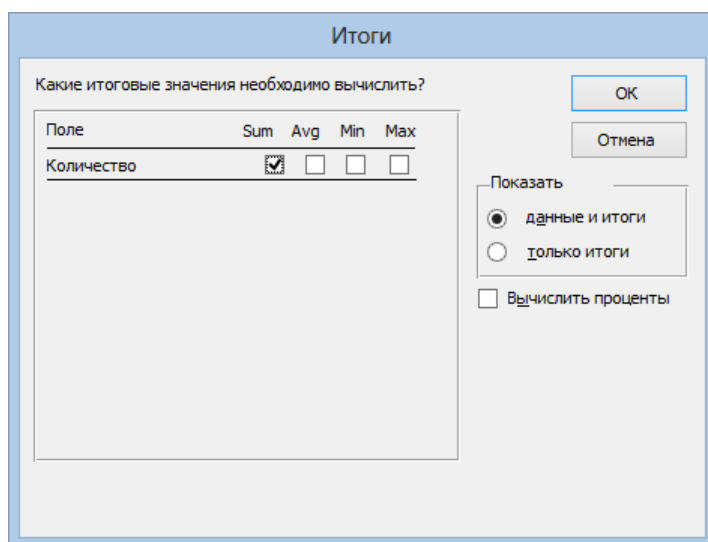


Рис. 6.11. Выбор функции для итоговых значений

Если выбрать переключатель **данные и итоги**, то в каждой группе будут перечислены все записи, подсчитаны и показаны итоги. При выборе переключателя **только итоги**, в отчет включаются только итоговые значения по группам. Если установить флажок **Вычислить проценты**, то Access подсчитает проценты от итоговой суммы для каждой группы.

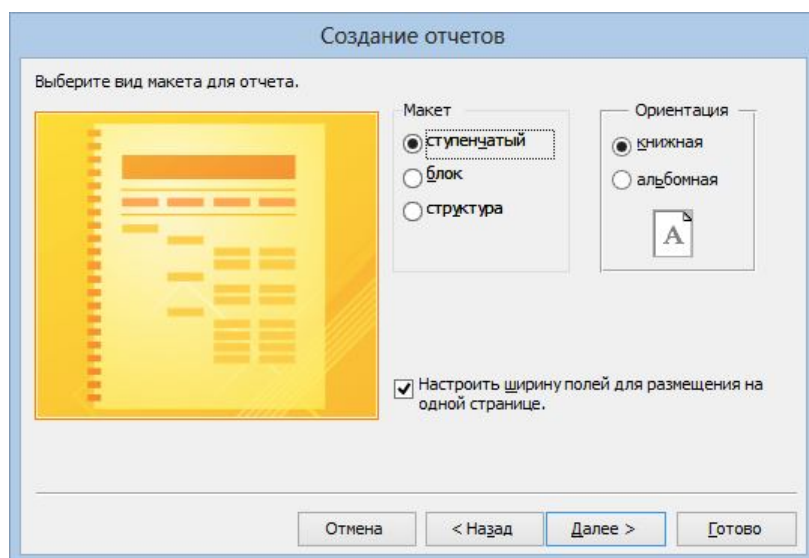


Рис. 6.12. Выбор макета и ориентации отчета

В пятом диалоговом окне (рис.6.12) выбирается макет и ориентация отчета. При выборе того или иного макета слева показывается его схема. Нужно выбрать макет, наиболее наглядно представляющий информацию и занимающий меньшее место.

В шестом диалоге (рис.6.13) отчету дается имя, например, *Детали* и завершается работа мастера.

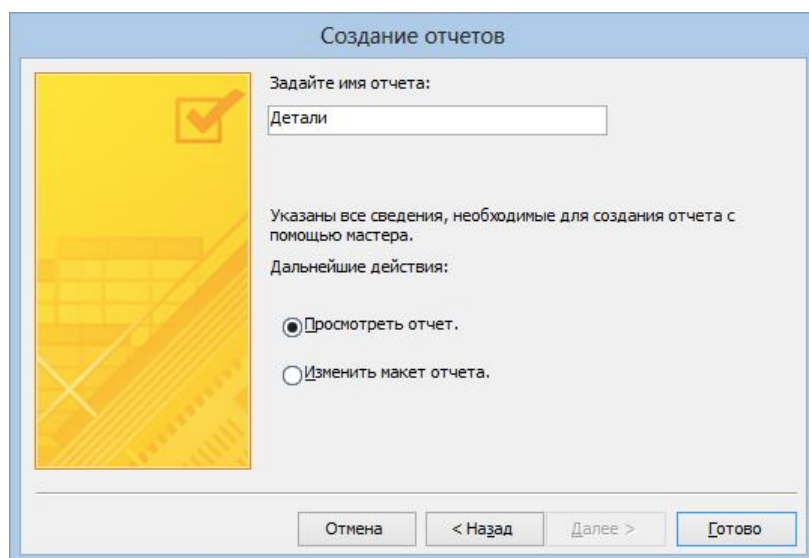


Рис. 6.13. Ввод имени отчета

По завершении создания отчет можно просмотреть. Первая страница созданного отчета показана на рис.6.14.

Детали		
Название	Поставщик	Количество
Муфта	Бриз	70
	Итоги для 'Поставщик' = Бриз (1 запись)	
	Sum	70
Итоги для 'КодДетали' = 5 (1 запись)		
	Sum	70
Шпилька	Лента	1200
	Итоги для 'Поставщик' = Лента (1 запись)	
	Sum	1200
Итоги для 'КодДетали' = 6 (1 запись)		
	Sum	1200
Гайка	Виктория	1500
	Итоги для 'Поставщик' = Виктория (1 запись)	
	Sum	1500
	Полег	500
	Итоги для 'Поставщик' = Полег (1 запись)	
	Sum	500
Итоги для 'КодДетали' = 10 (2 записей)		
	Sum	2000
Болт	Прииск	1500
	Итоги для 'Поставщик' = Прииск (1 запись)	
	Sum	1500
Итоги для 'КодДетали' = 11 (1 запись)		
	Sum	1500
Шайба	Астра	500
	Итоги для 'Поставщик' = Астра (1 запись)	
	Sum	500
Итоги для 'КодДетали' = 13 (1 запись)		
	Sum	500

19 октября 2013 г.

Стр. 1 из 2

Рис. 6.14. Первая страница отчета *Детали*

Если бы не задавался вывод итогов, то соответствующих строчек **Sum** в отчете бы не было.

### 6.3. Использование отчета

Основное назначение отчета – представить в обработанной и наглядной форме информацию из базы данных. Отчет можно просто просматривать на экране или распечатать на бумаге.

В режиме **Предварительный просмотр** на экран можно выводить разное число страниц отчета используя инструменты группы **Масштаб** (рис.6.15).

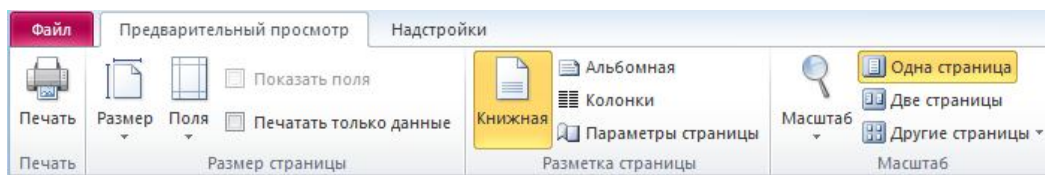


Рис. 6.15. Вкладка **Предварительный просмотр**

Внизу окна отчета расположен навигатор, облегчающий перемещение между страницами отчета. Назначение кнопок навигатора приведено на рис.6.16.

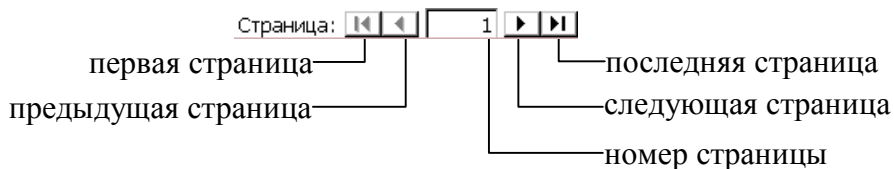


Рис. 6.16. Навигатор для отчета

На принтер отчет выводится командой кнопки **Печать** (рис.6.15).

Имеется возможность экспортировать отчет, созданный в Access, в документ формата RTF или HTML. Соответствующие команды находятся на вкладке **Внешние данные** в группе **Экспорт** в списке **Дополнительно** (рис. 6.17).

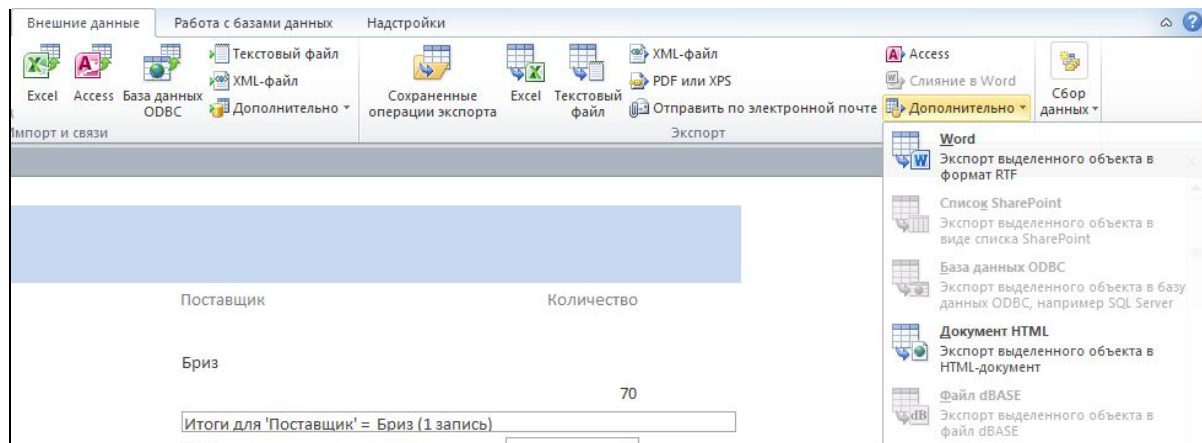


Рис. 6.17. Команды экспорта

Экспортируем отчет *Детали* в формат **RTF** командой **Экспорт выделенного объекта в формат RTF**. При экспорте надо выбрать папку для создаваемого файла и задать ему имя (рис.6.18).

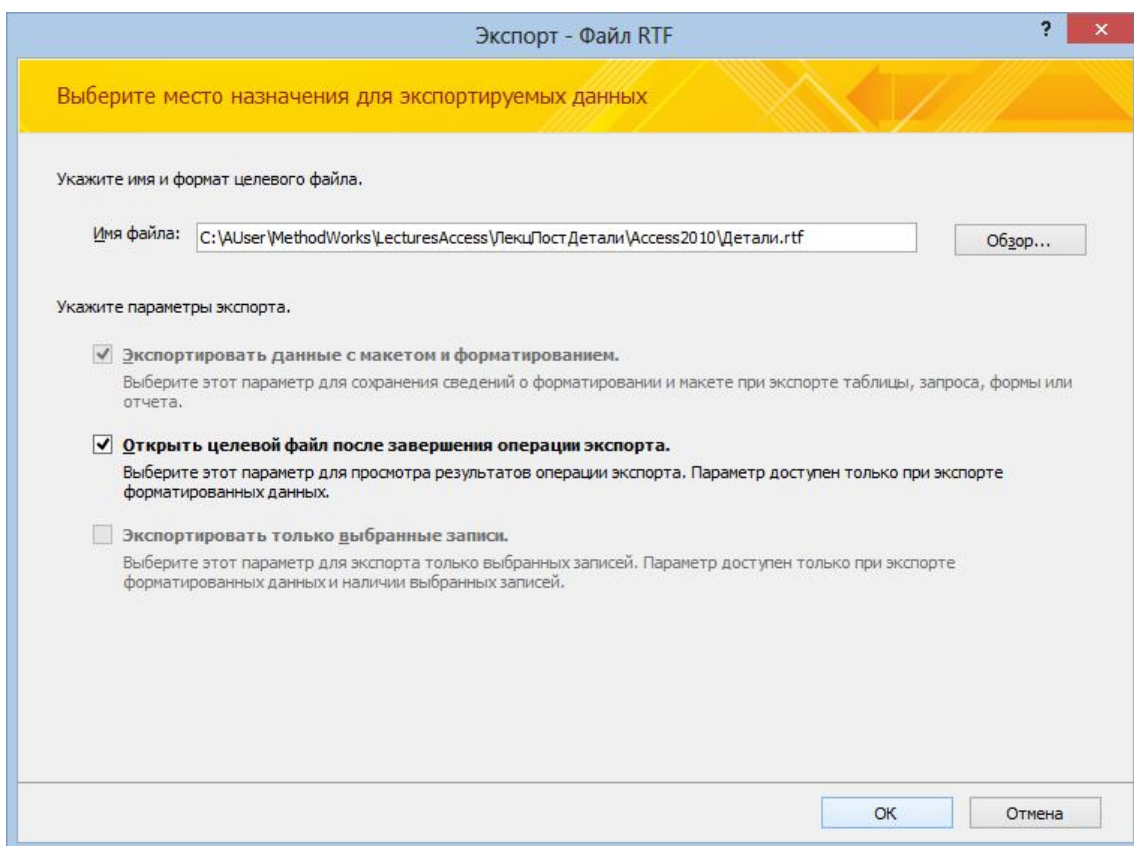


Рис. 6.18. Экспорт отчета

При экспорте в файл формата **RTF** (Rich Text Format – богатый текстовый формат) полностью сохраняется форматирование отчета. Формат **RTF** используется для обмена текстовыми документами. Он поддерживается многими приложениями, в том числе и текстовым редактором Word.

Наиболее популярным форматом для текстовых документов является формат PDF. Для экспорта отчета в формат PDF предназначен инструмент **PDF или XPS** из группы **Экспорт** на вкладке **Внешние данные** (рис. 6.17).

## 6.4. Модификация структуры отчета

### 6.4.1. Работа в конструкторе отчетов

Модификация структуры отчета выполняется в режиме **Конструктор**.

Воспользуемся мастером отчетов и создадим отчет со структурой, показанной на рис.6.19.



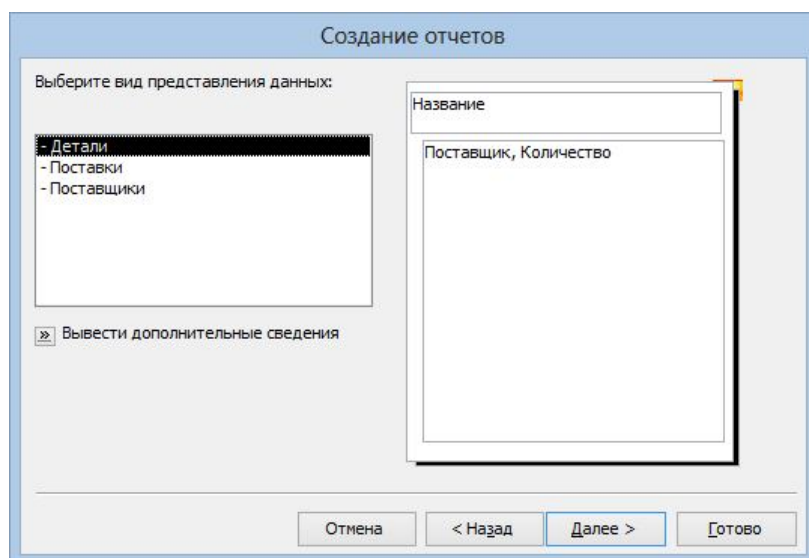


Рис. 6.19. Вариант группировки данных в отчете

Назначим на 4-м диалоге мастера отчетов **Итоги** (рис.6.11), суммарное значения по полю *Количество*. Назовем отчет *Детали\_Количество*.

Экспортируем этот отчет в формат PDF. Выполним команду **PDF или XPS** из группы **Экспорт** на вкладке **Внешние данные** (рис. 6.17). Выберем папку для размещения создаваемого файла и зададим имя файла (рис.6.20).

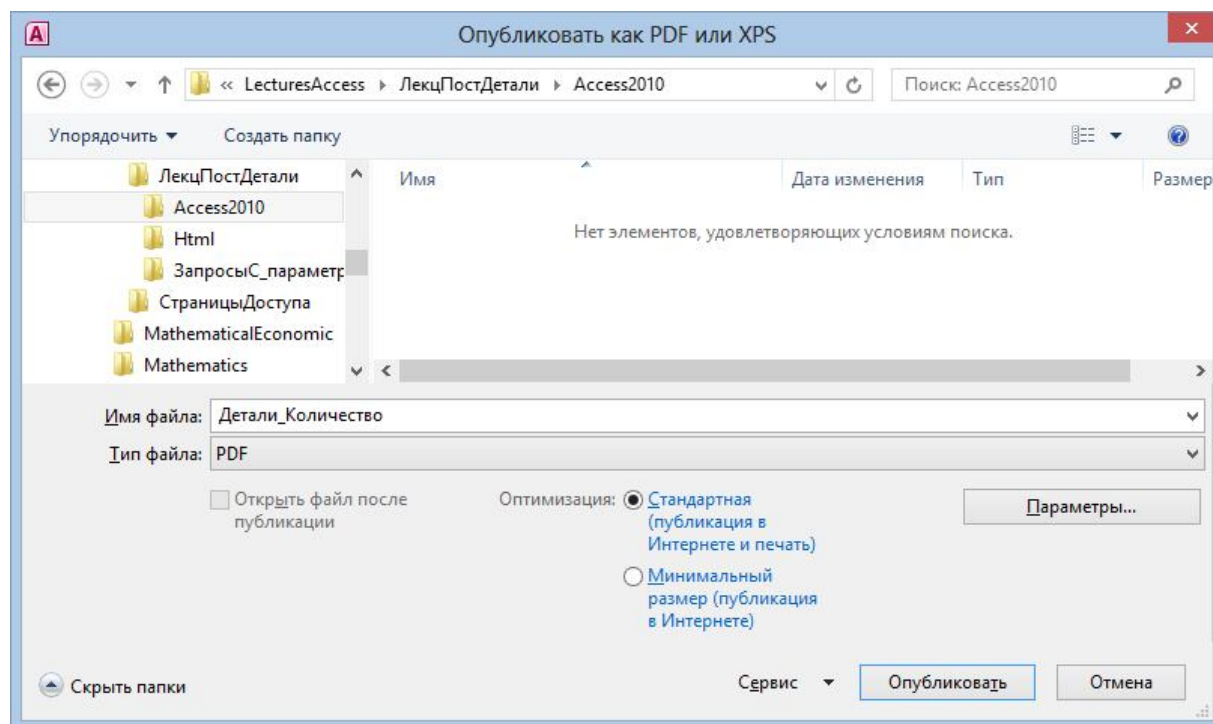


Рис. 6.20. Настройка экспорта отчета в файл PDF

Окно программы Acrobat Reader с экспортированным отчетом показано на рис.6.21.

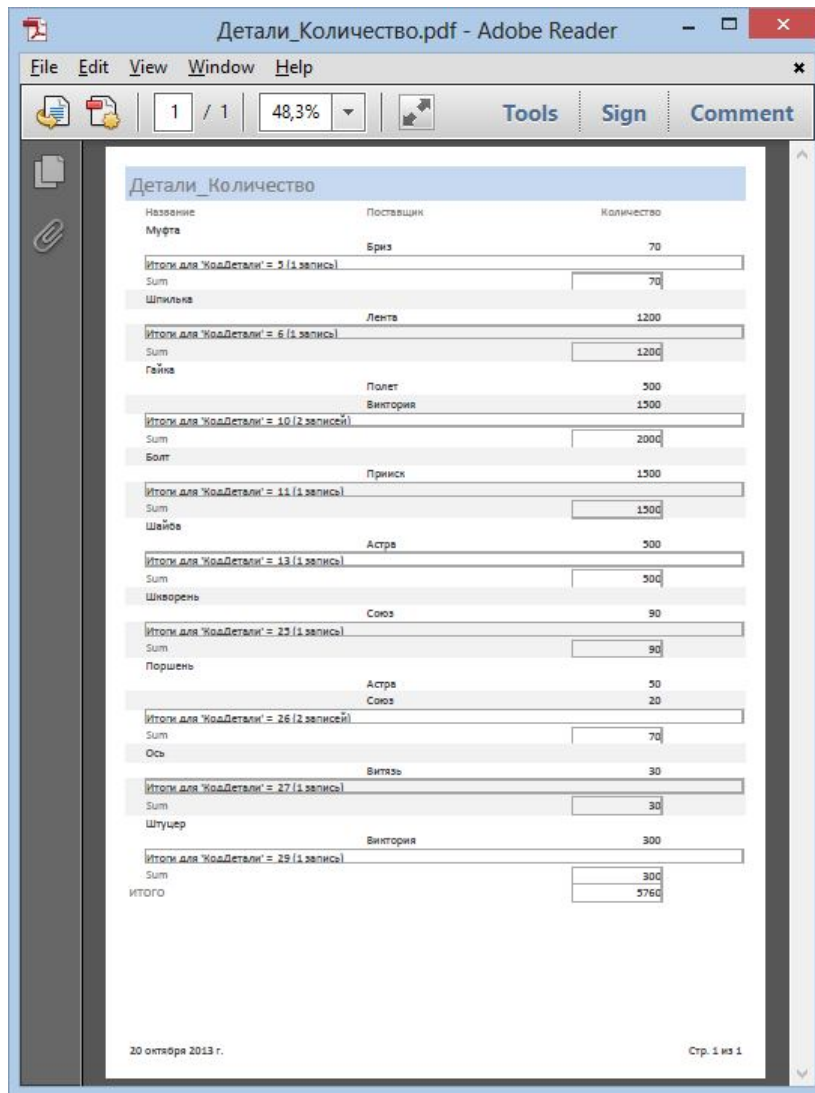


Рис. 6.21. Экспортированный отчет в окне программы Acrobat Reader

Переведем отчет *Детали\_Количество* в режим конструктора, нажав правую кнопку мыши и выбрав команду **Конструктор** (рис. 6.22).

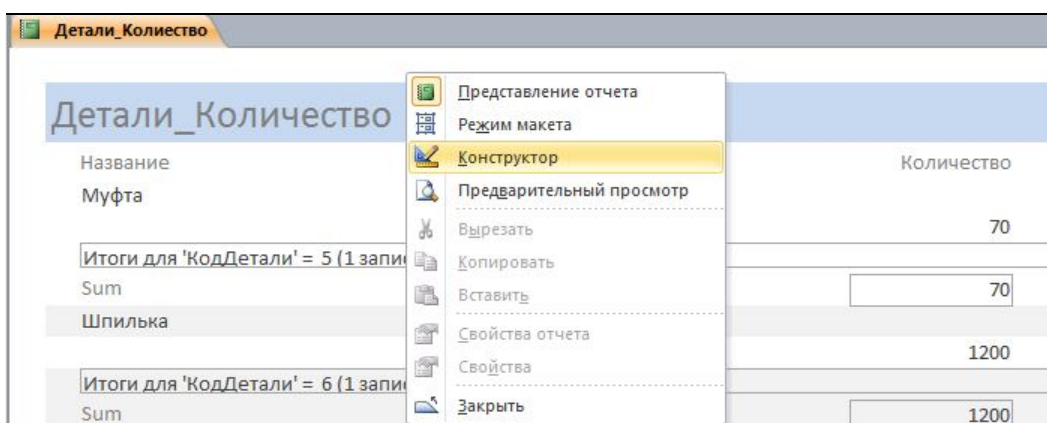


Рис. 6.22. Команды контекстного меню для отчета

Бланк конструктора отчетов состоит из нескольких разделов, названия которых показаны на горизонтальных полосах (рис.6.23). В разделах размещаются элементы управления.

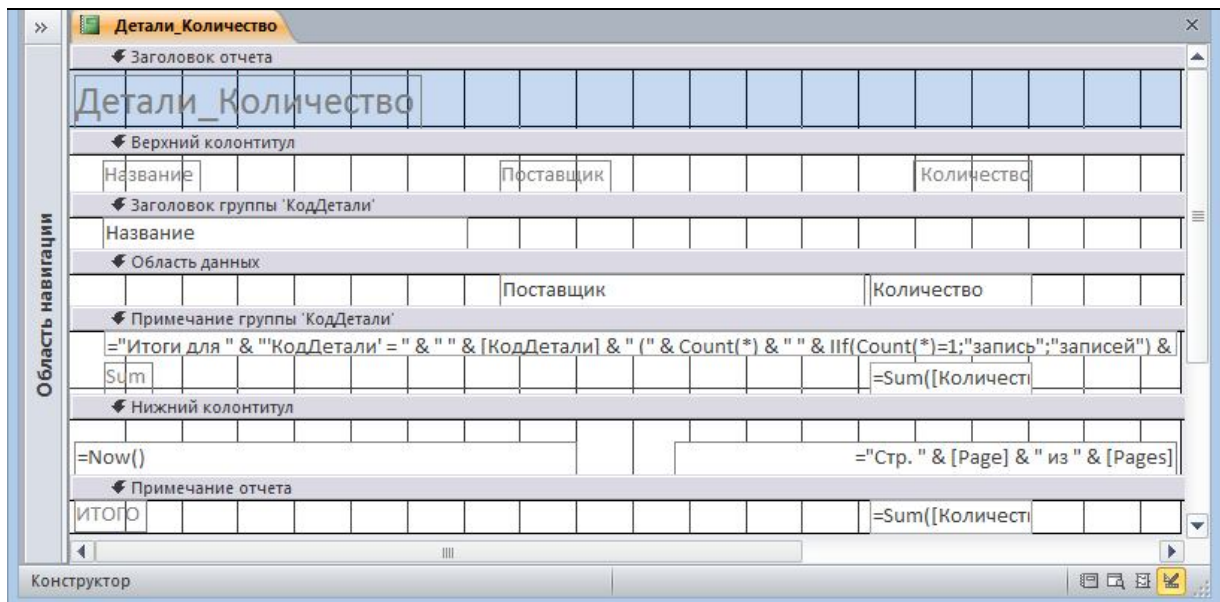


Рис. 6.23. Отчет *Детали\_Количество* в режиме конструктора

Раздел **Заголовок отчета** выводится только на первой странице отчета, раздел **Примечание отчета** – только на последней. Верхний и нижний колонтитулы печатаются на каждой странице отчета.

Если записи отчета группируются по одному или нескольким полям, для каждого из полей формируется отдельный заголовок. В приведенном примере это **Заголовок группы 'КодДетали'**.

В разделе **Область данных** выводится информация о каждой записи.

В разделе **Примечание группы** выводятся итоговые значения для группы записей.

Разделы отчета можно удалить командами контекстного меню.

В нашем примере в примечаниях к отчету выводится сумма значений поля *Количество* во всех записях, что излишне, т.к. не имеет особого смысла складывать болты и гайки, поэтому этот раздел можно удалить.

Выделим раздел **Примечание отчета** и нажмем правую кнопку мыши (рис.6.24). Видно, что команды **Сетка**, **Колонтитулы страницы**, **Заголовок/примечание отчета** выделены. Это значит, что соответствующие элементы присутствуют на бланке конструктора.

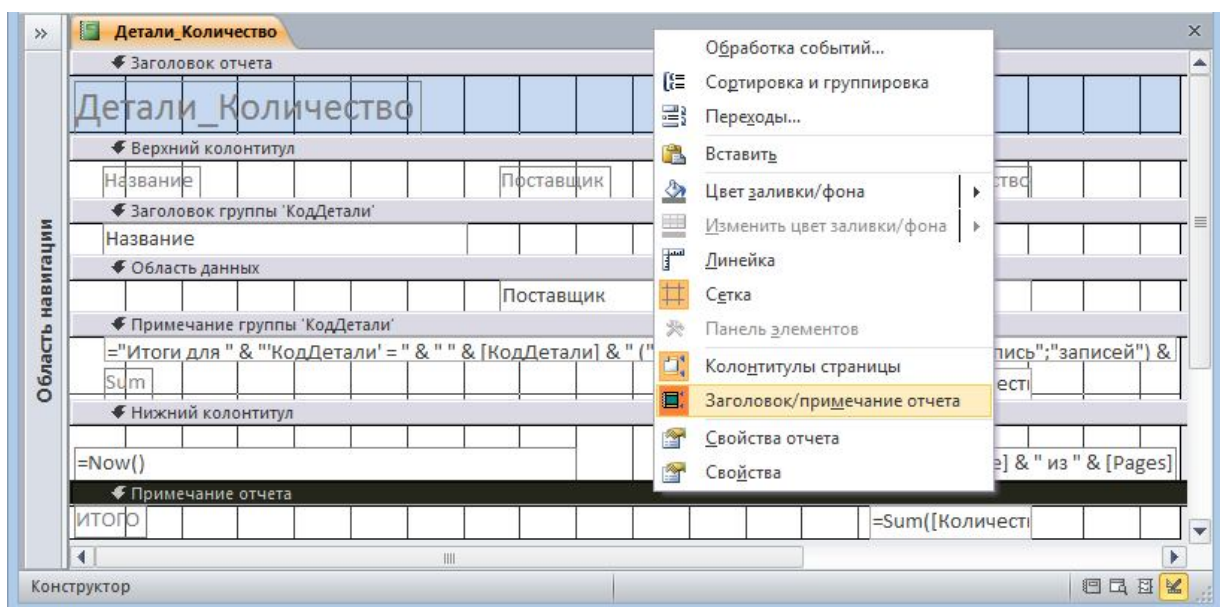


Рис. 6.24. Команды работы с разделами отчета

Выполним команду **Заголовок/примечание отчета**. Появится предупреждающее сообщение (рис.6.25). Нажимаем **Да**. Примечание и заголовок отчета будут удалены.

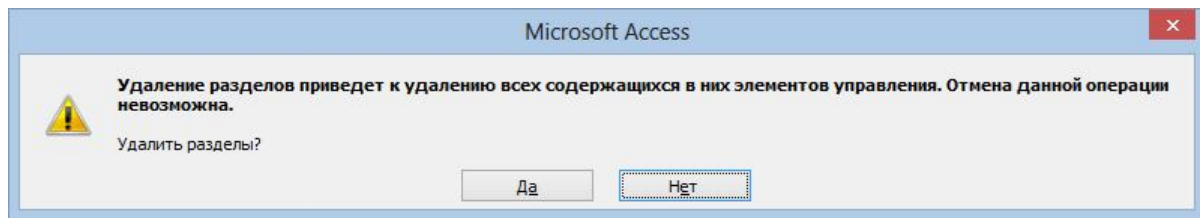


Рис. 6.25. Предупреждающее сообщение

Повторим команду **Заголовок/примечание отчета**. Примечание и заголовок отчета будут вставлены в отчет, но эти разделы будут пусты. Вставим в заголовок отчета надпись, введем в надпись текст *Поставленные детали*, установим шрифт (рис.6.26).

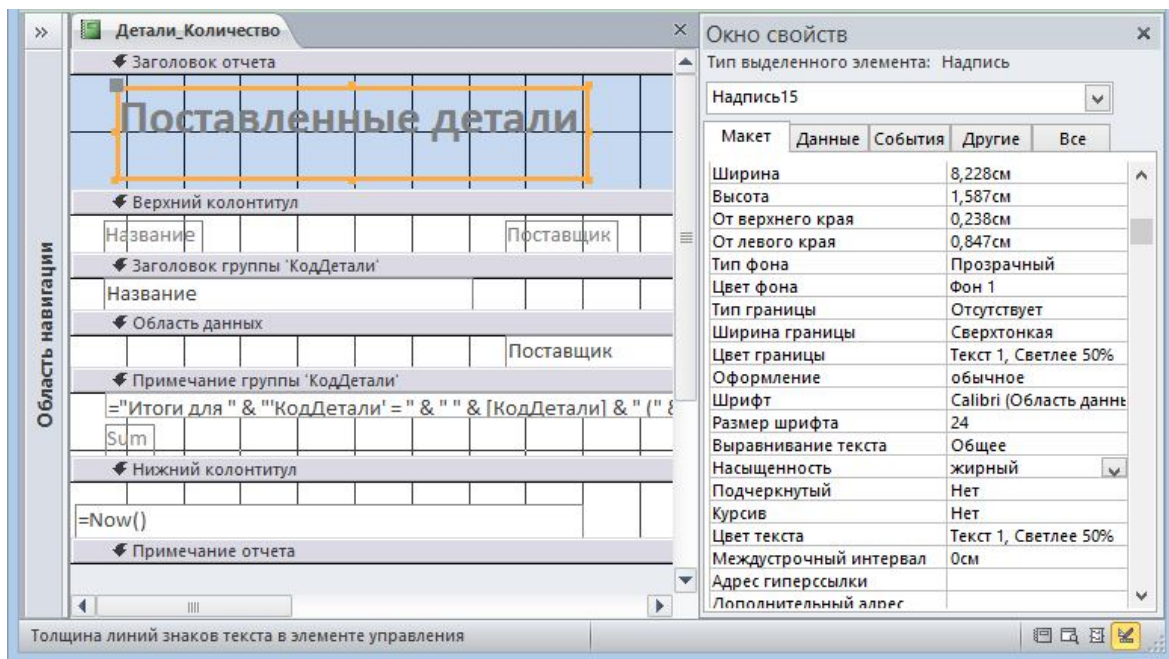


Рис. 6.26. Создание надписи в отчете

В разделе **Примечание группы 'КодДетали'** стоит автоматически созданное выражение, значением которого является или слово *запись* или слово *записей*. Это выражение не обеспечивает формирование правильных окончаний слова *записи* во всех возможных случаях. Отредактируем его. Откроем окно **Свойства** для ЭУ, расположенного в разделе **Примечание группы 'КодДетали'**, и рассмотрим выражение на вкладке **Данные** (рис.6.27).

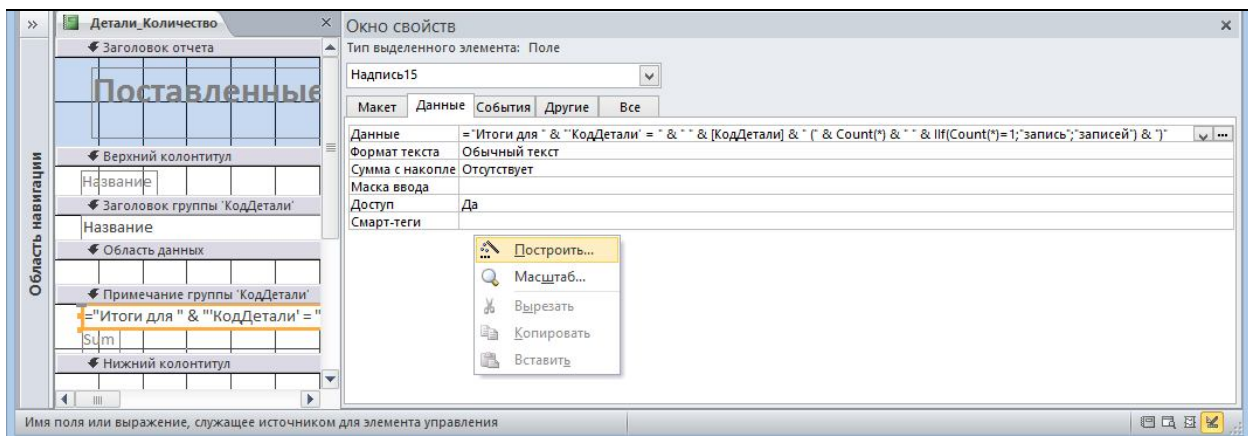


Рис. 6.27. Окно свойств раздела отчета **Примечание группы 'КодДетали'**

Полностью данное выражение имеет вид:

```
"Итоги для " & "'КодДетали'=" & " " & [КодДетали] & " (" & Count(*) & " " &
IIf(Count(*)=1; "запись"; "записей") & ")"
```

 (1)

В нем используется нотация VBA (Visual Basic for Application). Строки символов заключаются в двойные кавычки, имена полей выделяются квадратными скобками.

В окне свойств справа от свойства **Данные** находятся две кнопки:

 – список полей и

 – вызвать построитель выражений.

Вызовом для данного ЭУ окно построителя выражений, нажав указанную кнопку или командой **Построить** (рис.6.27). Окно построителя выражений показано на рис.6.28.

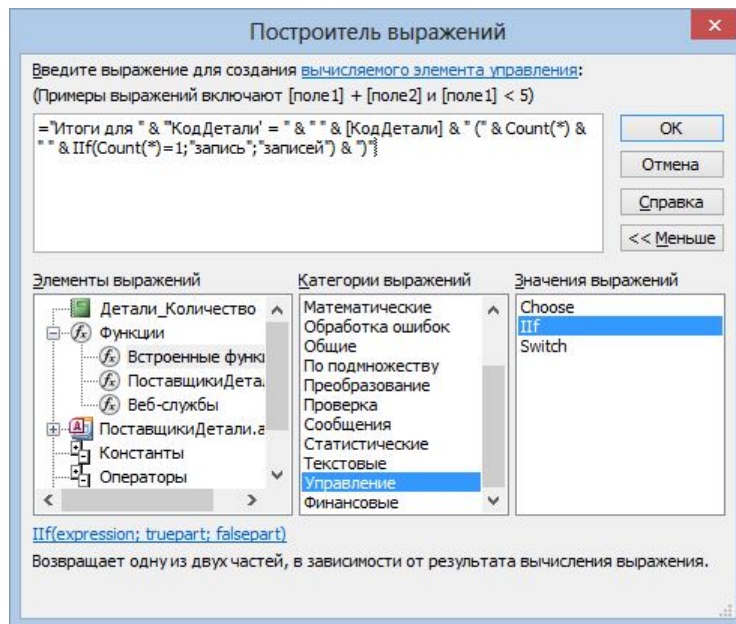


Рис. 6.28. Использование построителя выражений

Функция Count(\*) возвращает число записей в группе, которая включает записи с одинаковым значением поля *КодДетали*.

Функция IIf(«expr»; «truepart»; «falsepart») относится к категории **Управление** (рис.6.28). При истинности выражения «expr» функция возвращает значение выражения «truepart», а при ложности – «falsepart».

В окне **Построитель выражений** (рис.6.28) вместо простого выражения, сгенерированного автоматически мастером отчетов, введем более точное выражение:

```
"Итоги для " & "'КодДетали'=" & " " & [КодДетали] & " (" & Count(*) & " " &
IIf(Count(*)=1 Or (Count(*)>20 And Count(*) Mod 10=1); "запись";
IIf((Count(*)>1 And Count(*)<5) Or (Count(*)>21 And Count(*) Mod 10>1 And
Count(*) Mod 10<5);"записи"; "записей")) & ")"
```

Это выражение обеспечит создание правильного окончания для слова *записи* вплоть до количества записей равного 100. После ввода выражения в построителе выражений нажимаем **OK**, переходим в окно **Свойства**, куда в соответствующую строку будет перенесено составленное выражение. Здесь для завершения ввода нажимаем **Enter**, после чего новое выражение переносится на макет отчета, если в выражении нет синтаксических ошибок. При наличии ошибок выводятся диагностические сообщения.

Надпись со словом **Sum** удалим, поле, содержащее суммарное количество поставленных деталей переместим вверх и поставим напротив надписи *"Итоги для..."* (рис.6.29).

Изменим **Примечание отчета**. Вставим надпись, в которой напишем *Всего поставок*, вставим также поле для отображения общего числа поставок, у которого в качестве отображаемых данных зададим функцию =Count(\*) (рис.6.29).

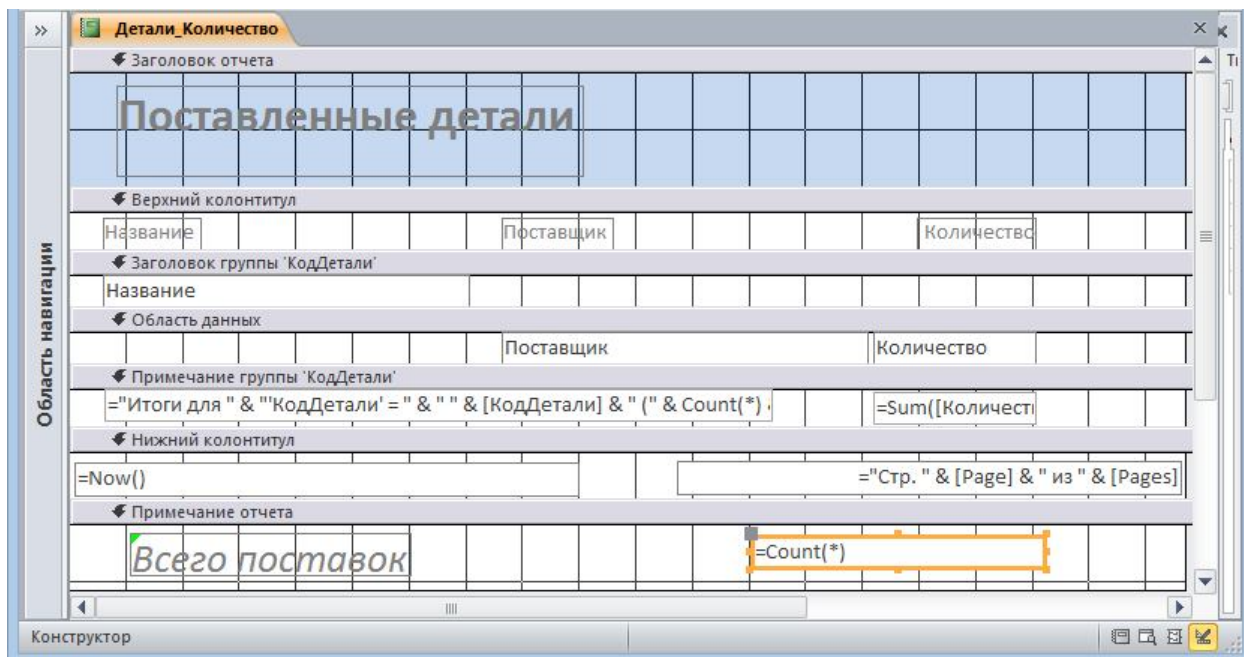


Рис. 6.29. Отредактированный отчет

Созданный отчет показан на рис.6.30.

<b>Поставленные детали</b>		
Название	Поставщик	Количество
Муфта	Бриз	70
Итоги для 'КодДетали' = 5 (1 запись)		70
Шпилька	Лента	1200
Итоги для 'КодДетали' = 6 (1 запись)		1200
Гайка	Полет	500
	Виктория	1500
Итоги для 'КодДетали' = 10 (2 записи)		2000
Болт	Прииск	1500
Итоги для 'КодДетали' = 11 (1 запись)		1500
Шайба	Астра	500
Итоги для 'КодДетали' = 13 (1 запись)		500
Шкворень	Союз	90
Итоги для 'КодДетали' = 25 (1 запись)		90
Поршень	Астра	50
	Союз	20
Итоги для 'КодДетали' = 26 (2 записи)		70
Ось	Витязь	30
Итоги для 'КодДетали' = 27 (1 запись)		30
Штуцер	Виктория	300
Итоги для 'КодДетали' = 29 (1 запись)		300
<b><i>Всего поставок</i></b>		<b>11</b>

Рис. 6.30. Отчет после коррекции структуры

### 6.4.2. Управление группировкой

В режиме конструктора отчетов можно управлять группировкой и сортировкой данных в разделе **Область данных**, используя инструмент **Группировка** из группы **Группировка и итоги** со вкладки **Конструктор** (рис.6.31).

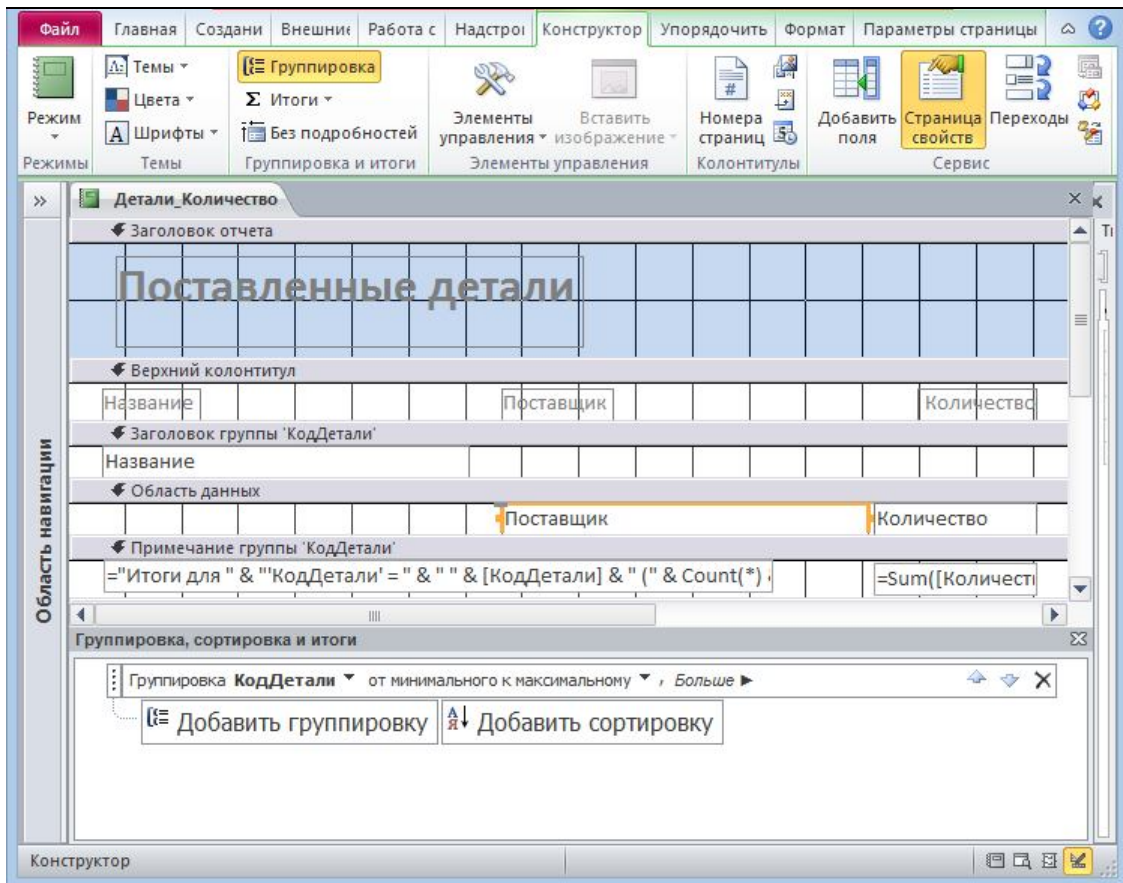


Рис. 6.31. Задание сортировки и группировки в отчете

Как видно на рис. 6.31 в отчете есть только группировка по полю *КодДетали*, то есть записи, отобранные в отчет, разбиты на группы, в которых поле *КодДетали* имеет одинаковое значение. Нажатие кнопки со стрелкой вниз раскрывает список полей, которые можно использовать для создания группировки (рис.6.32). В качестве критерия группировки можно использовать выражение, включающее несколько полей и функций.

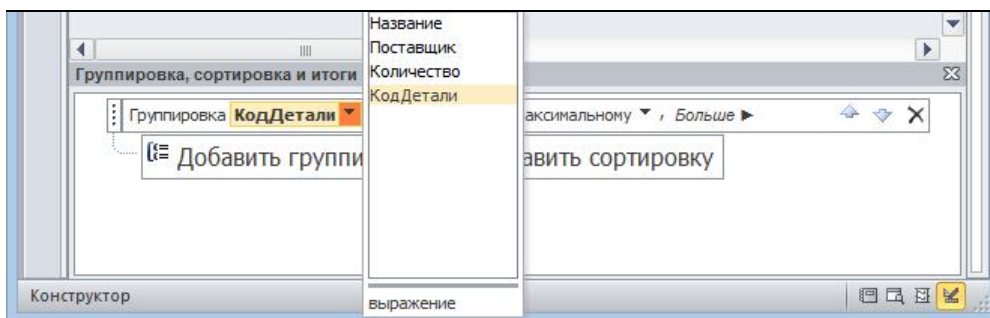


Рис. 6.32. Список полей для задания группировки

Кнопка **Больше** раскрывает список настроек группировки (рис.6.33)

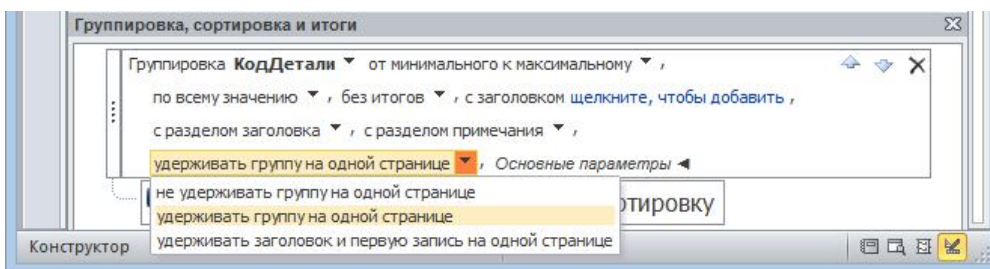


Рис. 6.33. Свойства группировки



Установка свойства **удерживать группу на одной странице** обеспечит печать группы без переноса на следующую страницу.

### 6.4.3. Форматирование элементов управления

Работа с элементами управления, размещенными в отчетах, ведется теми же приемами, что и в конструкторе форм. Для размещения новых элементов управления в отчете их нужно выбрать на **Панели элементов**. Элементы управления можно перемещать, изменять их размеры, выбирать для них цвет, шрифт текста, стиль границы и другие свойства. Их можно удалять, копировать, выравнивать поодиночке и в группе.

## 6.5. Создание наклеек

Наклейки используются, например, в качестве адресов получателей на почтовых конвертах и для других подобных целей.

Создадим наклейки с перечнем атрибутов деталей, чтобы использовать их, например, в качестве ярлыков на ящиках, в которых хранятся детали определенного вида. Для этого выберем таблицу *Детали*, переходим на вкладку **Создание** и выбираем инструмент **Наклейки** в группе **Отчеты** (рис.6.34).

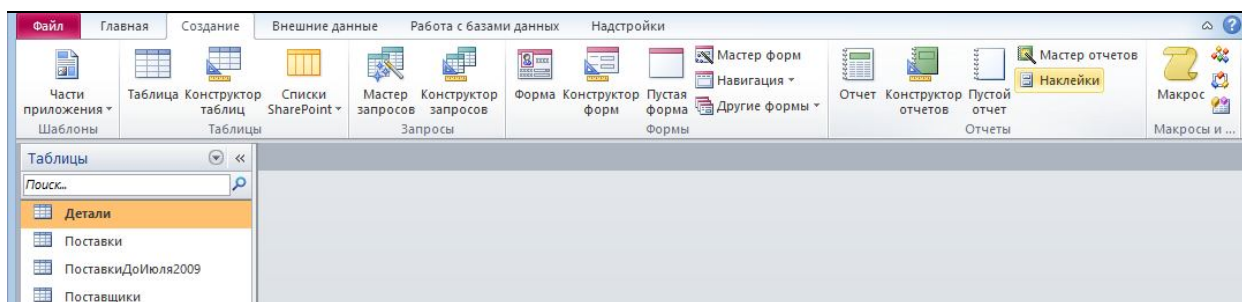


Рис. 6.34. Начало создания наклеек

В первом диалоговом окне выбирается система единиц измерения: **британская** или **метрическая** и тип бумаги: **на листах** или **рулонная** (рис.6.35).

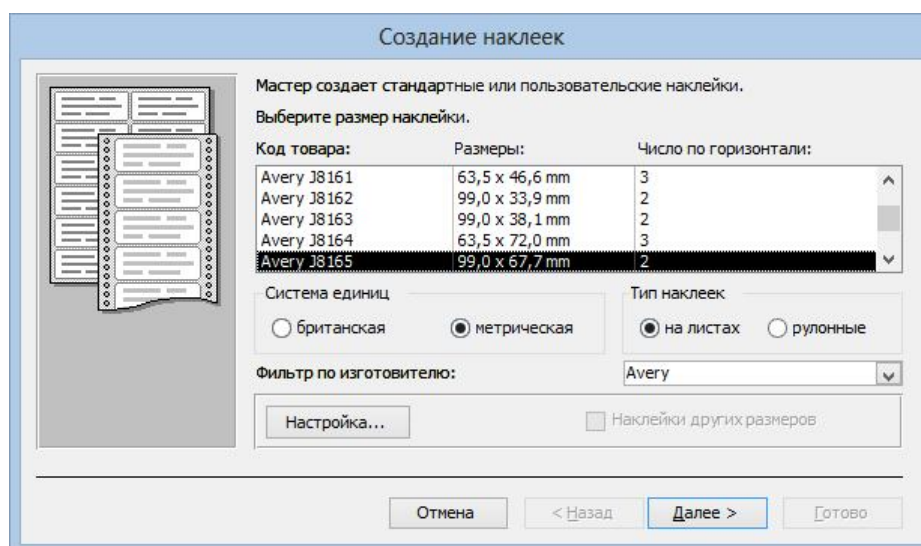


Рис. 6.35. Настройка наклеек

Во втором диалоговом окне (рис.6.36) задаются элементы форматирования: вид шрифта, его размер, начертание, цвет.

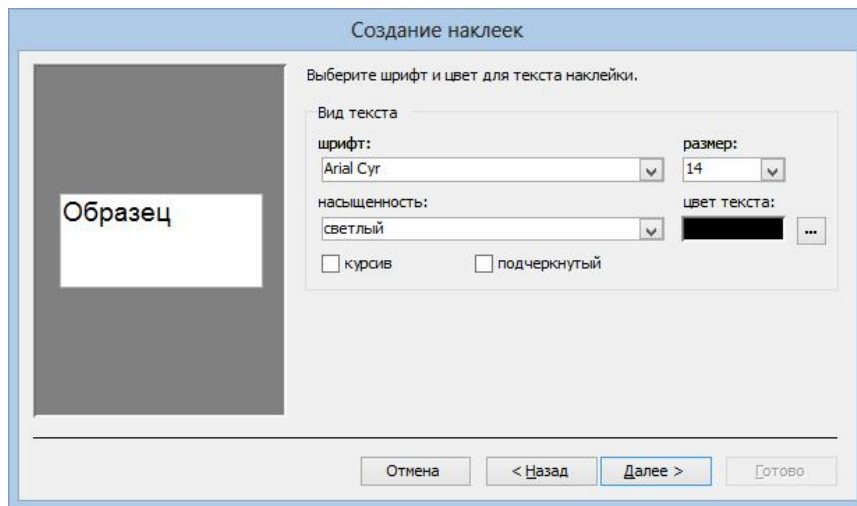


Рис. 6.36. Выбор шрифта для наклейки

На третьем диалоге отбираются поля для включения в наклейку (рис.6.37). Кроме того, здесь можно поместить на наклейку поясняющий текст, который будет повторяться на каждой наклейке. В приводимом примере это слова *Название, Цвет, Вес*.

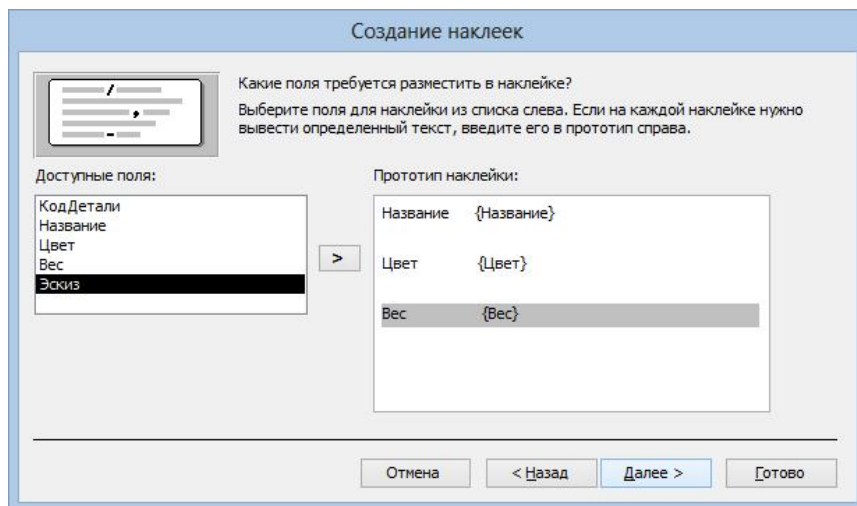


Рис. 6.37. Отбор полей для наклеек

На четвертом диалоге можно указать поле для сортировки наклеек (рис.6.38).

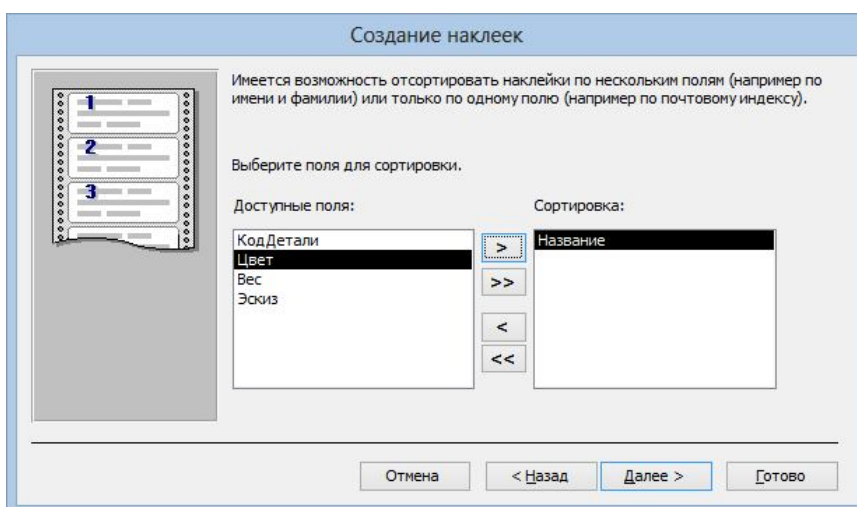


Рис. 6.38. Выбор поля для сортировки наклеек

На последнем диалоге (рис.6.39) вводится имя наклеек.

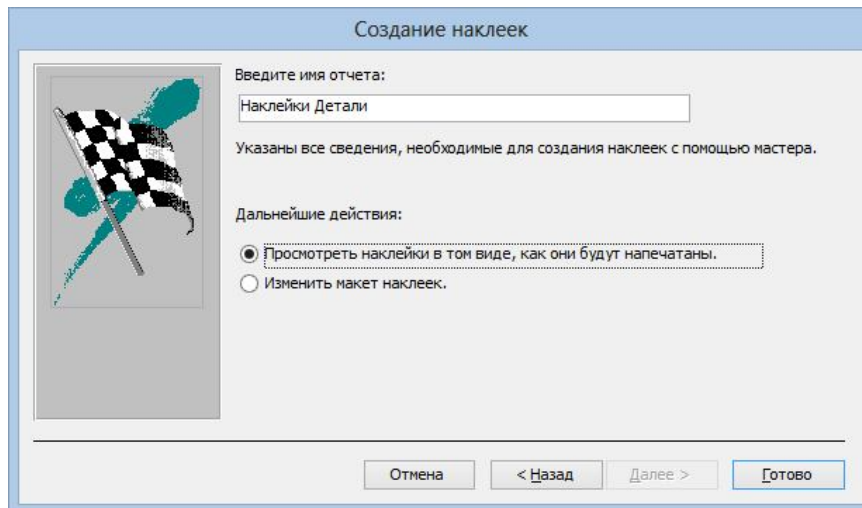


Рис. 6.39. Ввод имени наклеек

На рис.6.40 показано окно с созданными наклейками в режиме предварительного просмотра. На одном листе бумаги размещается равномерно несколько наклеек.

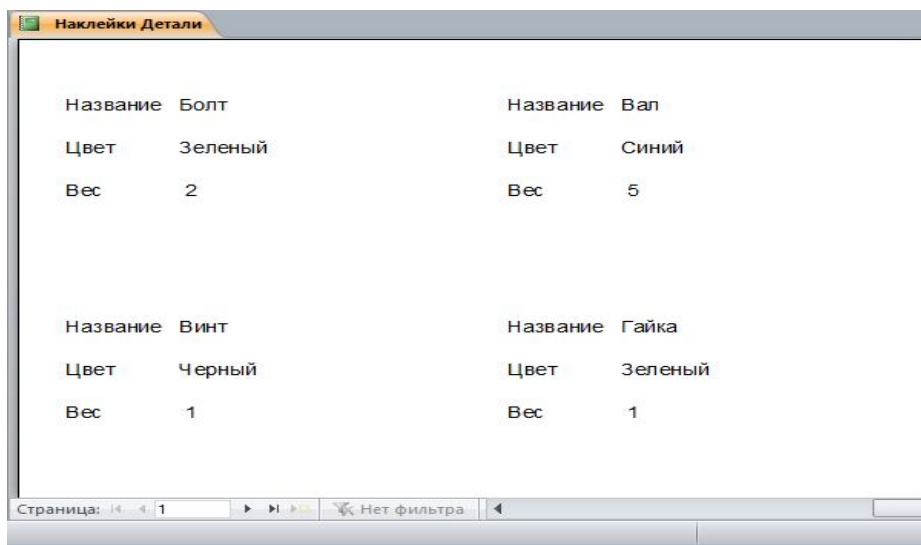


Рис. 6.40. Окно с наклейками по данным таблицы *Детали*

Открыв наклейки в режиме конструктора (рис.6.41), можно подправить их формат обычными средствами форматирования отчетов, например, можно добавить надписи для поясняющего текста и установить для них свой шрифт и свой формат. В наклейках, созданных мастером, поясняющий текст и значение поля объединяются в одном текстовом поле.

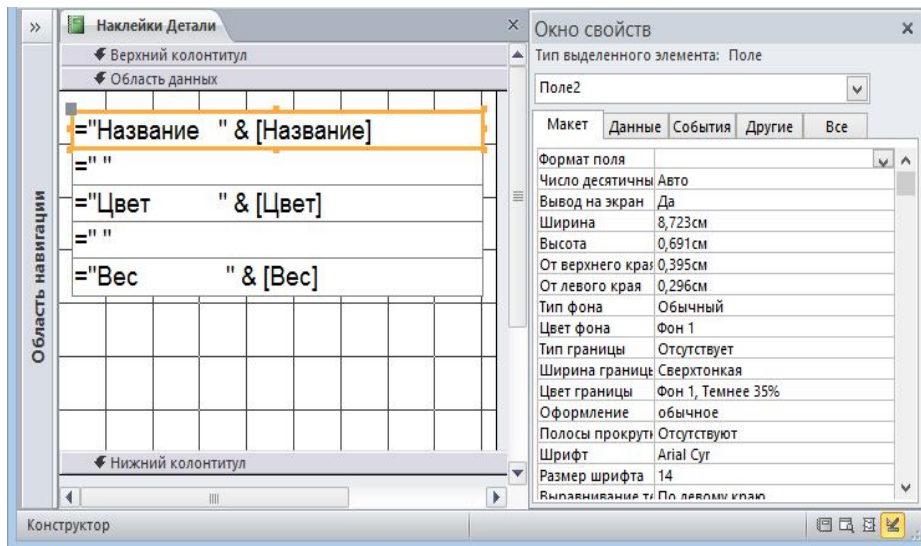


Рис. 6.41. Наклейки в режиме конструктора

## 6.6. Связь с другими компонентами Office

### 6.6.1. Создание документа слияния Word

Мощным средством создания большого количества однородных документов, например, типовых писем, поздравлений, напоминаний, конвертов, наклеек с использованием информации, хранящейся в базе данных, являются *документы слияния* текстового редактора Word. Их можно создавать, работая в Word и обращаясь к базе данных Access за данными, а можно начинать их создание в Access, переходя затем в Word.

Рассмотрим создание документов слияния на примере создания наклеек.

Выберем в окне базы данных Access таблицу *Детали* и выберем инструмент **Слияние в Word** из группы **Экспорт** на вкладке **Внешние данные** (рис.6.42).

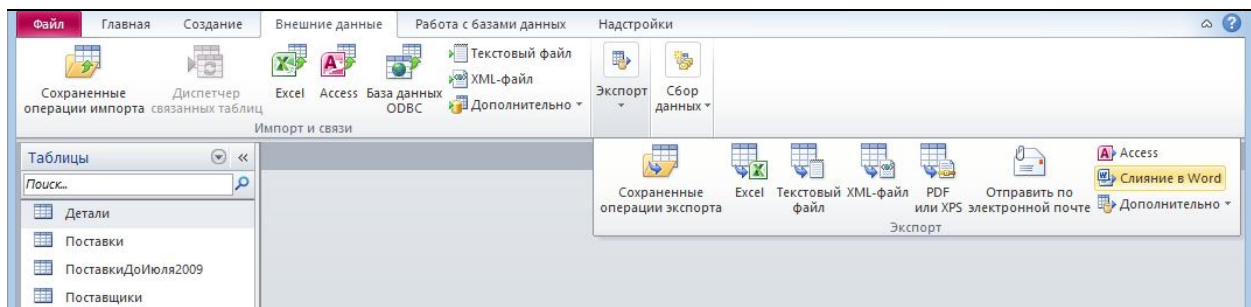


Рис. 6.42. Команда создания документа слияния

Возможно появление сообщения о том, что база данных открыта в монопольном режиме (рис.6.43). Монопольный режим означает возможность одновременной работы с базой только одного пользователя, а создание документа слияния требует доступа к базе данных еще одного приложения - программы Word.

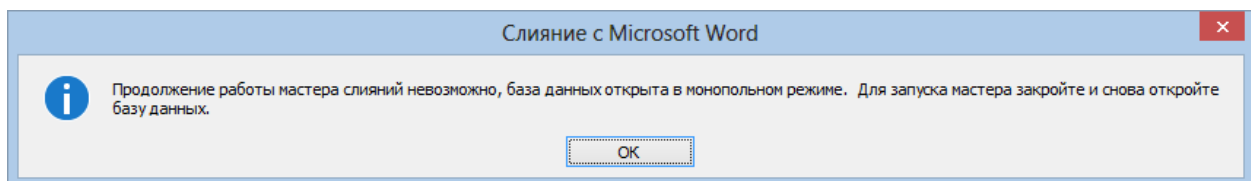


Рис. 6.43. Сообщение о монопольном режиме работы с базой данных

Для открытия базы данных в немонопольном режиме нужно сначала ее закрыть командой **Файл, Закрыть базу данных**. Затем выполнить команду **Файл, Открыть**, найти файл базы данных, выбрать в списке кнопки **Открыть** режим **Открыть**, который позволит открыть базу данных в немонопольном режиме (рис.6.44).

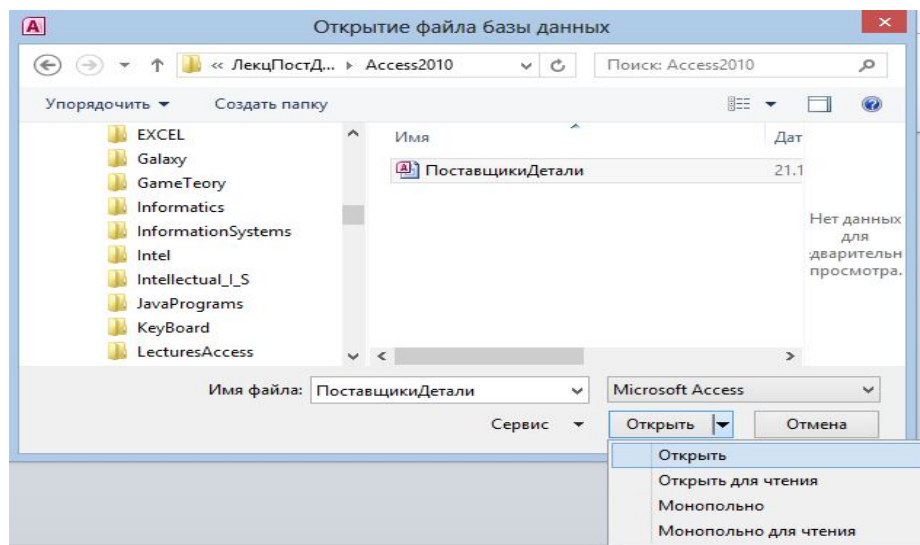


Рис. 6.44. Выбор режима открытия базы данных

Затем надо повторить команду создания документа слияния (рис.6.42). Появится диалог **Слияние с документами Microsoft Word** (рис.6.45), в котором выберем переключатель **Создать новый документ и установить связь с ним**.

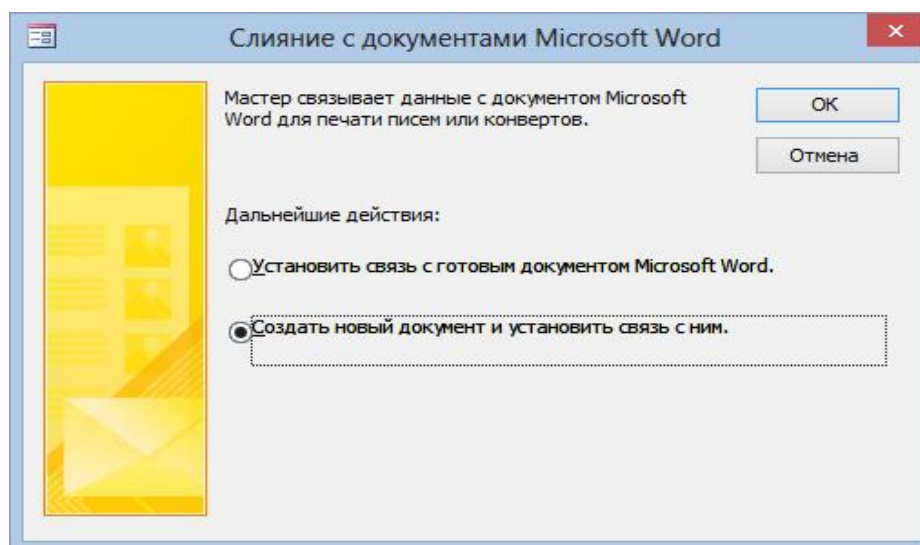


Рис. 6.45. Выбор способа создания документа слияния

При создании документа слияния в Word надо пройти шесть этапов. На первом из них (рис.6.46) выбирается тип создаваемого документа слияния. В рассматриваемом примере это наклейки.

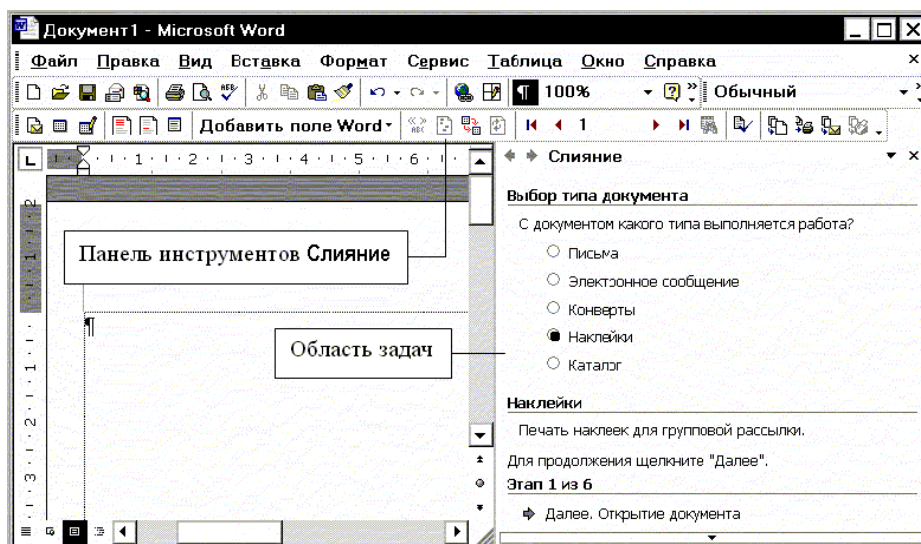


Рис. 6.46. Начало создания документа слияния

Создавать документ слияния можно с помощью кнопок на панели инструментов **Слияние**, но проще воспользоваться мастером из **Области задач**, который предлагает выполнить 6 этапов по созданию документа слияния.

Итак, на первом этапе выбирается тип документа слияния. Выберем **Наклейки** и, нажав кнопку **Далее**, перейдем ко второму этапу (рис.6.47).

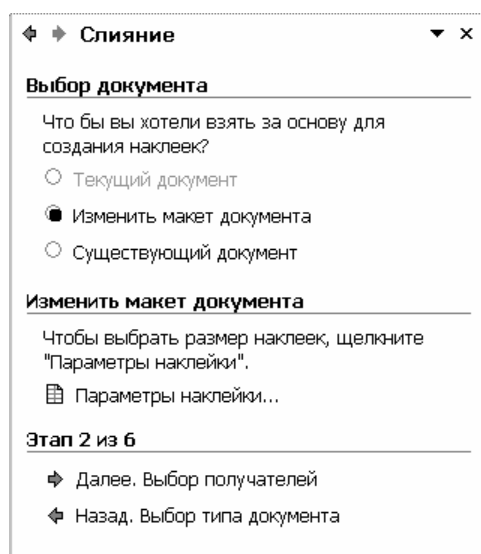


Рис. 6.47. Второй этап создания документа слияния

Здесь следует нажать кнопку **Параметры наклейки**, после чего появится диалог **Параметры наклейки** (рис.6.48), в котором выбирается один из шаблонов наклеек.

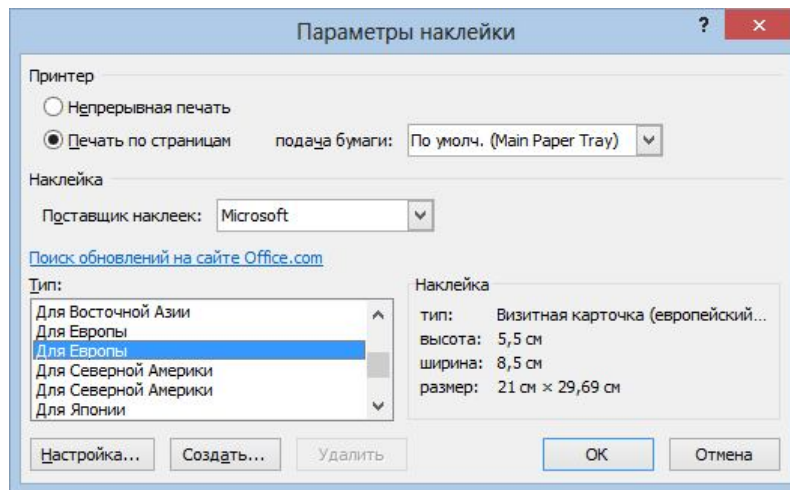


Рис. 6.48. Выбор параметров наклеек

После выбора шаблона возвращаемся в окно второго этапа (рис. 6.47) и переходим к третьему этапу (рис.6.49), где нужно выбрать «получателей» создаваемых типовых документов. В нашем примере список получателей уже имеется – это детали из таблицы *Детали* базы данных *ПоставщикиДетали*, поэтому просто переходим к следующему этапу.

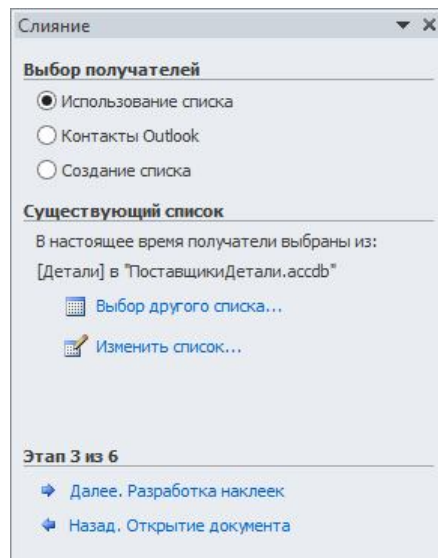


Рис. 6.49. Выбор получателей типовых документов

Заметим, что при создании документов слияния используется терминология (например, «получатели»), относящаяся к созданию типовых писем, хотя в нашем случае цель создания наклеек другая.

На четвертом этапе происходит разработка первой наклейки (рис.6.50). Здесь вводится текст, который будет повторяться во всех наклейках, а для того чтобы отображались данные из источника данных, нужно нажимать кнопку **Другие элементы в Области задач** и выбирать в окне **Добавление полей слияния** нужные поля таблицы базы данных для вставки в наклейку.

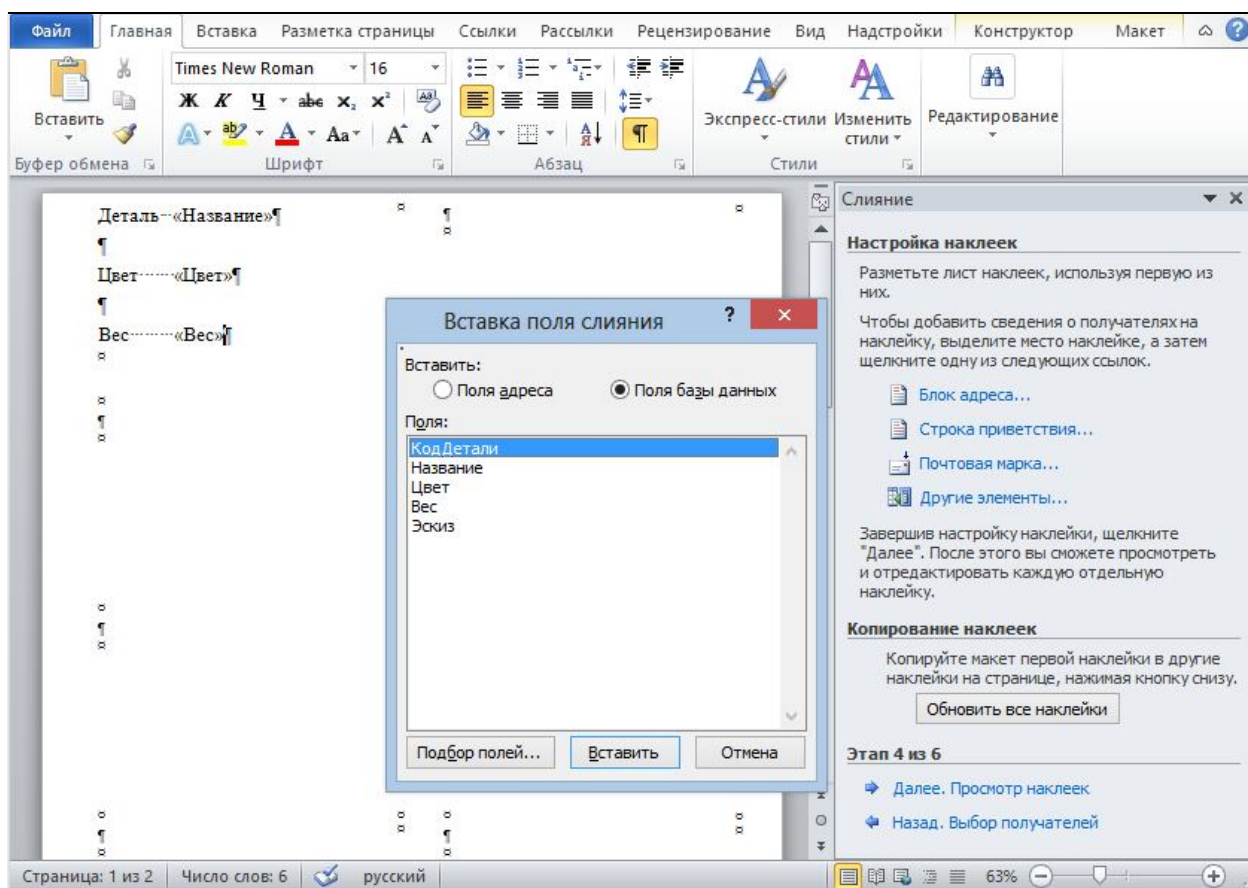


Рис. 6.50. Создание первой наклейки

После создания первой наклейки следует нажать кнопку **Обновить все наклейки**. Произойдет автоматическое создание наклеек для всех записей из таблицы **Детали** (рис.6.51). Наклейки в Word располагаются в ячейках таблицы. По команде **Обновить все наклейки** в ячейки таблицы заносятся наклейки для всех записей таблицы *Детали*.

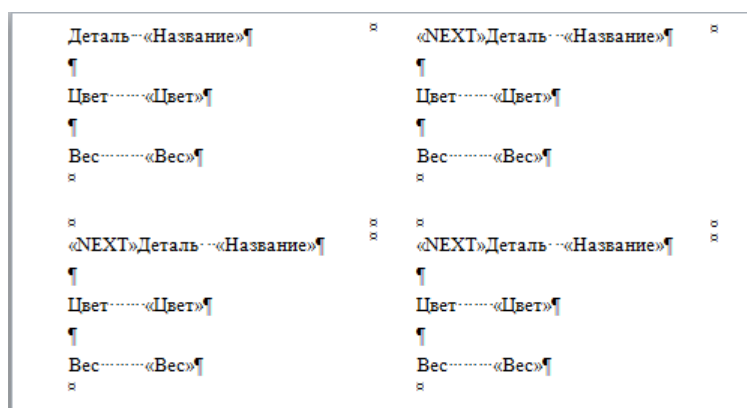


Рис. 6.51. Создание всех наклеек

На пятом этапе можно просмотреть отдельные наклейки (рис.6.52).



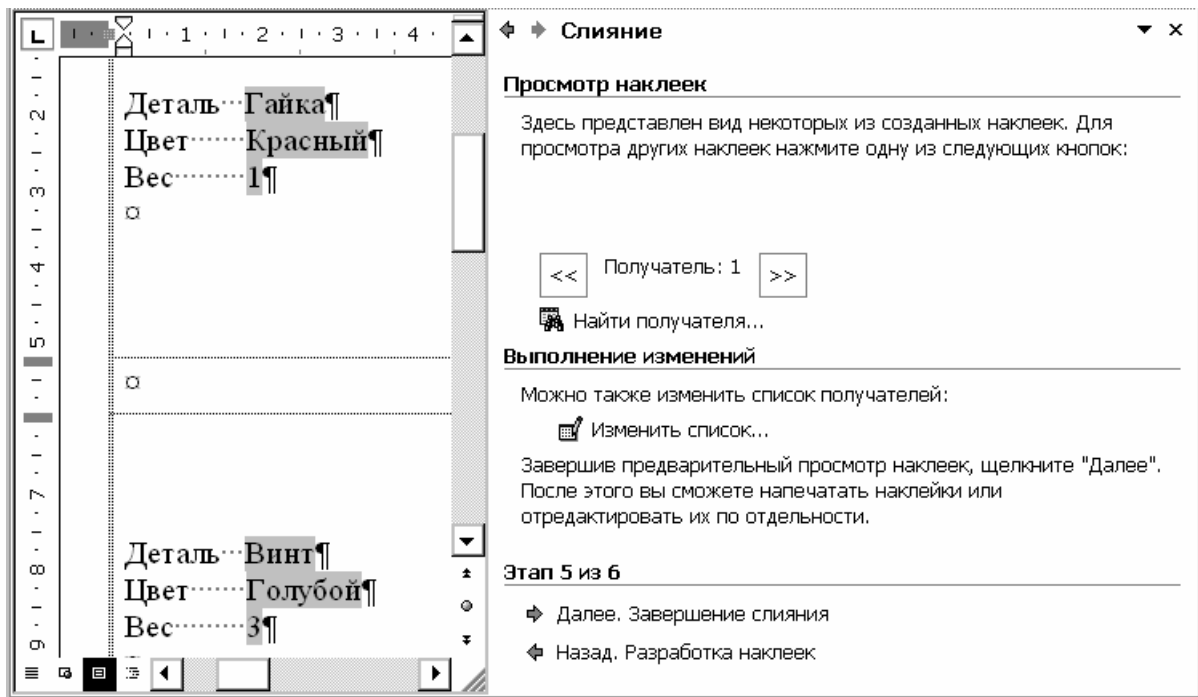


Рис. 6.52. Просмотр отдельных наклеек

На шестом этапе можно распечатать наклейки или перейти к просмотру отдельных наклеек (рис.6.53).

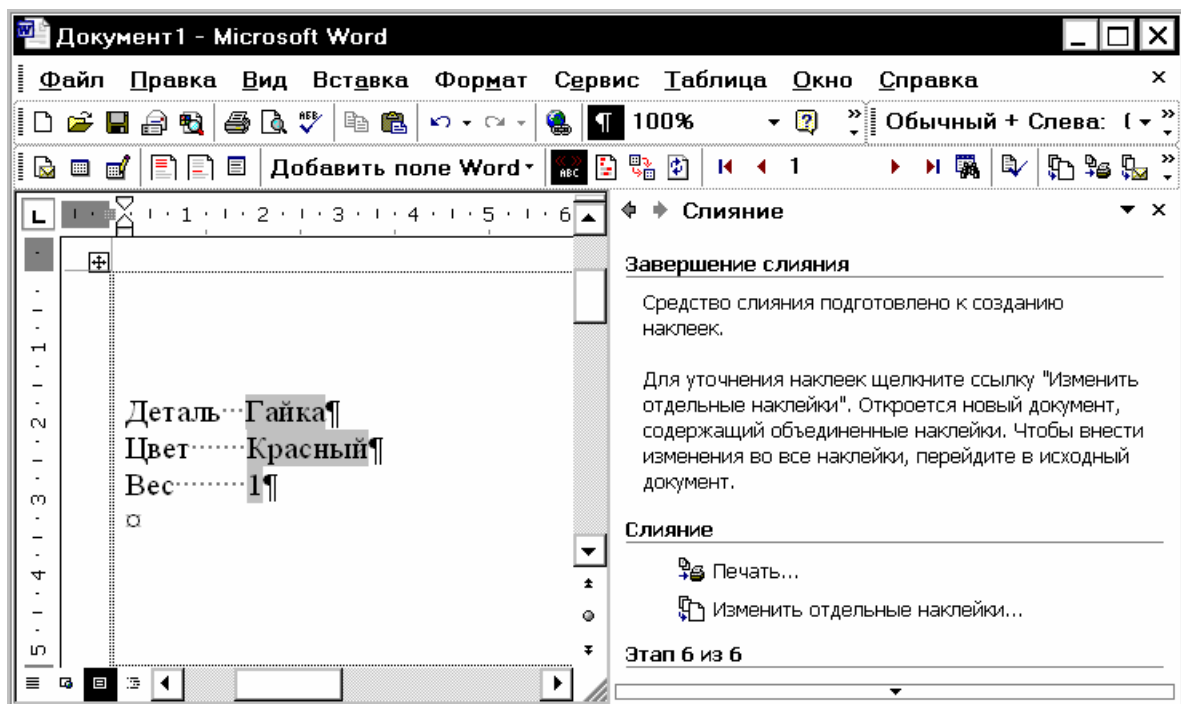


Рис. 6.53. Завершение создания документа слияния

Если нажать кнопку **Печать**, будет выведено окно, рис. 6.54, в котором можно выбрать диапазон для печати наклеек. Далее выполняются обычные действия по печати документа.

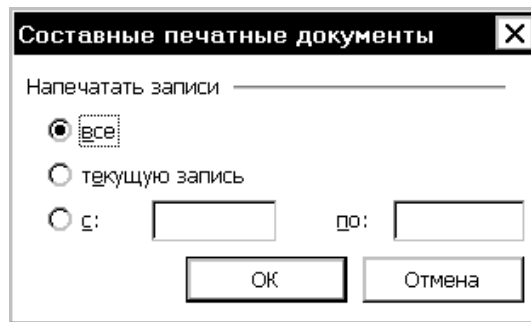


Рис. 6.54. Выбор диапазона для печати наклеек

Если нажать кнопку **Изменить отдельные наклейки**, выводится окно (рис.6.55), с помощью которого можно отобрать записи для включения в составной новый документ. После нажатия кнопки **ОК** будет создан *обычный* документ с названием Наклейки1.doc, содержащий отобранные наклейки (рис. 6.56).

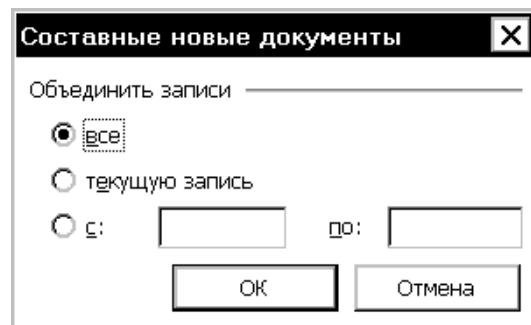


Рис. 6.55. Выбор записей для документа

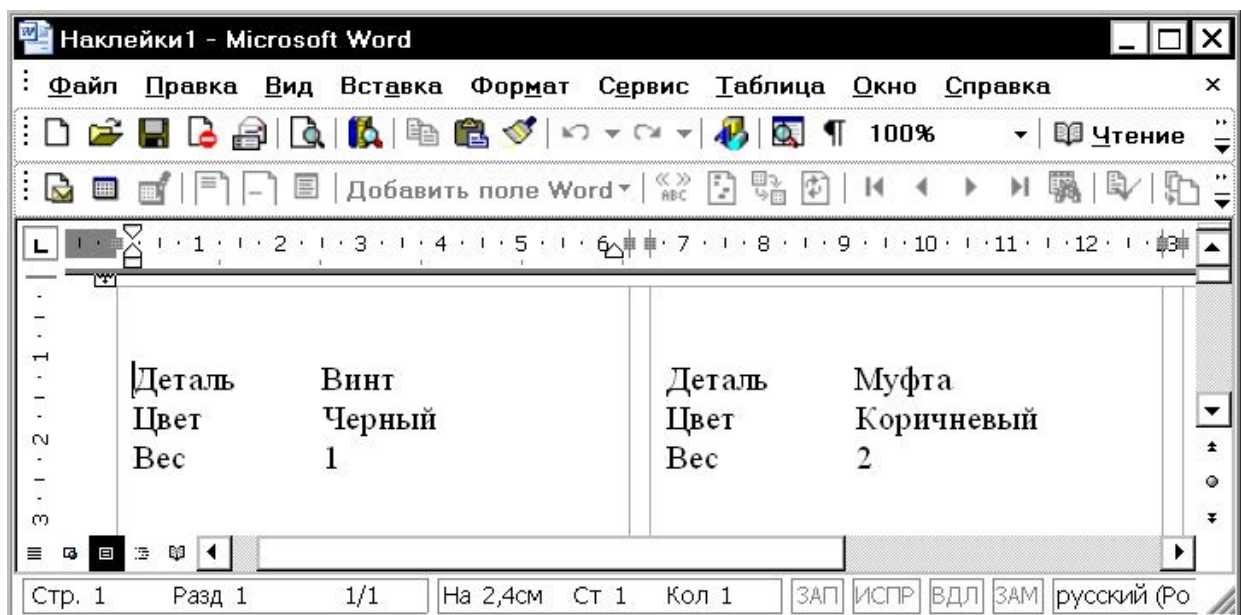


Рис. 6.56. Документ с текстами наклеек

Документ Наклейки1.doc с наклейками сохраним под именем НаклейкиДетали.doc. Перейдем к исходному документу слияния Документ1.doc, используя меню **Окно** (рис. 6.57).

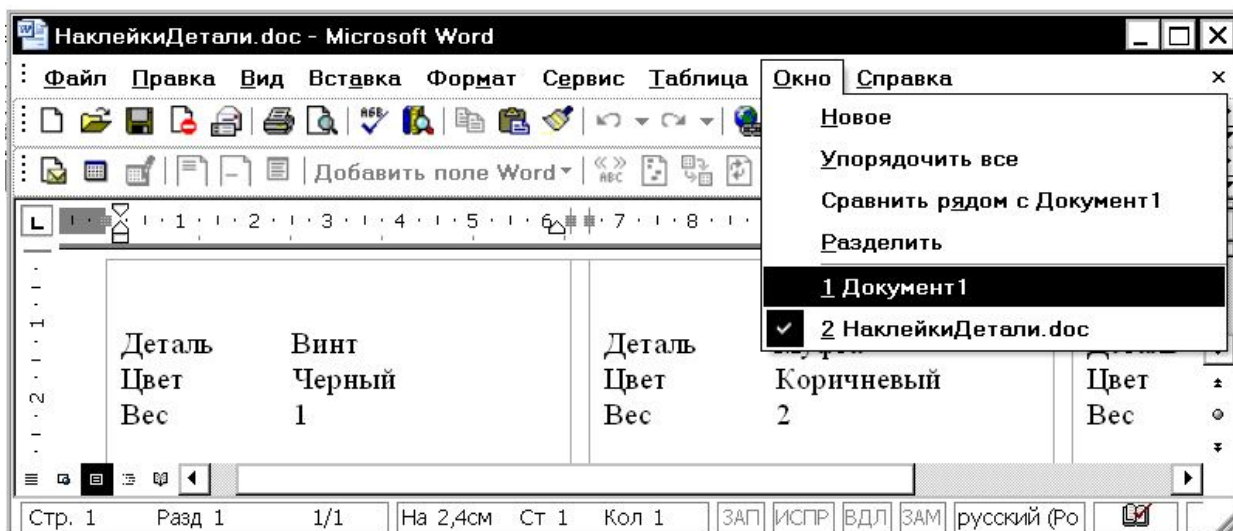


Рис. 6.57. Переход к исходному документу слияния

Сохраним документ слияния *Документ1.doc* под именем *СлияниеДетали.doc*. При повторном открытии этого документа будет выдаваться предупреждение об открытии соответствующего источника данных для документа (рис.6.58).

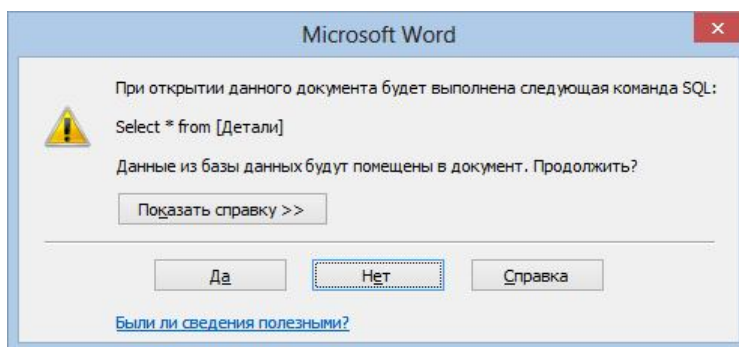


Рис. 6.58. Сообщение о помещении данных в документ слияния

Если данные в таблице *Детали* изменятся, в документе слияния *СлияниеДетали.doc* будут использованы обновленные данные, то есть в документе слияния всегда используются актуальные данные.

В обычном документе *НаклейкиДетали.doc* будут содержаться наклейки для записей таблицы *Детали* на момент создания этого документа.

## 6.6.2. Публикация в Word

Выберем в окне базы данных таблицу *Поставщики* и выполним команду **Word (экспорт выделенного объекта в формат RTF)** из группы **Экспорт, Дополнительно** на вкладке **Внешние данные** (рис.6.59).

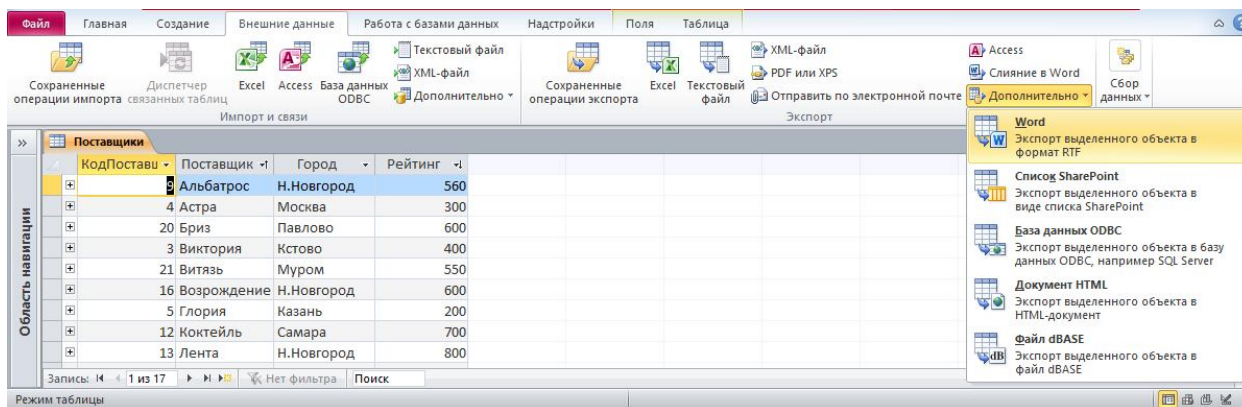


Рис. 6.59. Команда экспорта в формат RTF

В папке, выбранной при выполнении данной команды, будет автоматически создан файл *Поставщики.rtf* в формате RTF, в котором будет размещена информация из выбранной таблицы. Документ, созданный по данным таблицы *Поставщики*, показан на рис.6.60. Видно, что содержимое таблицы Access преобразовано в таблицу Word.

КодПоставщика	Поставщик	Город	Рейтинг
9	Альбатрос	Н.Новгород	560
4	Астра	Москва	300
20	Бриз	Павлово	600
3	Виктория	Кстово	400
21	Витязь	Муром	550
16	Возрождение	Н.Новгород	600
5	Глория	Казань	200
12	Коктейль	Самара	700
13	Лента	Н.Новгород	800
2	Макрос	Н.Новгород	800
17	Полет	Н.Новгород	800
8	Полет	Н.Новгород	260
1	Прииск	Н.Новгород	900
7	Союз	Кстово	800
10	Спутник	Дзержинск	300
14	Эврика	Н.Новгород	500
6	Юнайт	Н.Новгород	450

Рис. 6.60. Публикация таблицы *Поставщики* в Word

### 6.6.3. Анализ в Excel

Для обработки данных можно использовать мощные средства численного анализа электронных таблиц Excel.

Проведем анализ данных из запроса *ДеталиПоставщиков*, в котором определен общий вес каждой детали, поставленной поставщиком (рис.4.19). Откроем этот запрос и выполним команду **Excel** из группы **Экспорт** на вкладке **Внешние данные** (рис.6.61).

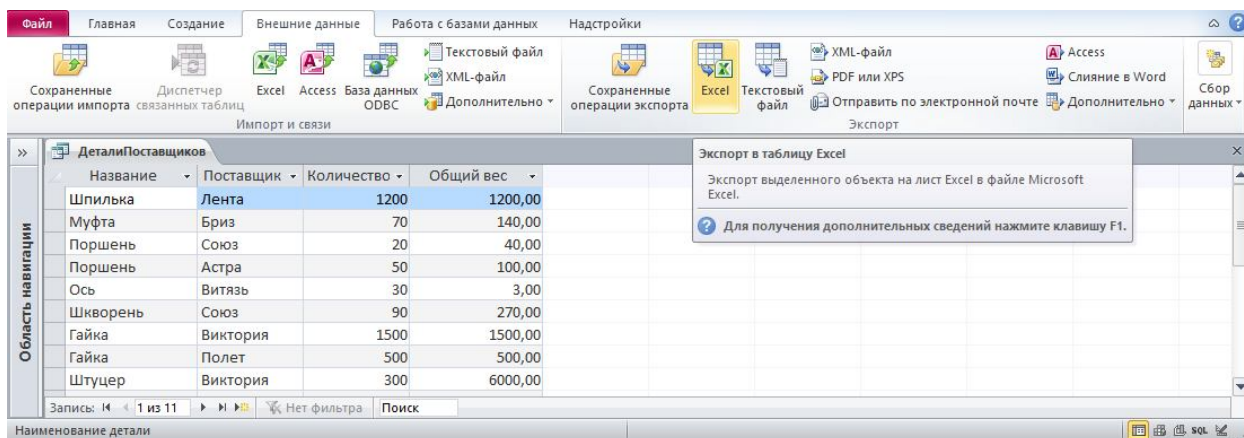


Рис. 6.61. Команда экспорта в Excel

При выполнении данной команды открывается диалог, в котором можно выбрать папку, формат и имя файла, в который будут экспортированы данные из запроса (рис.6.62).

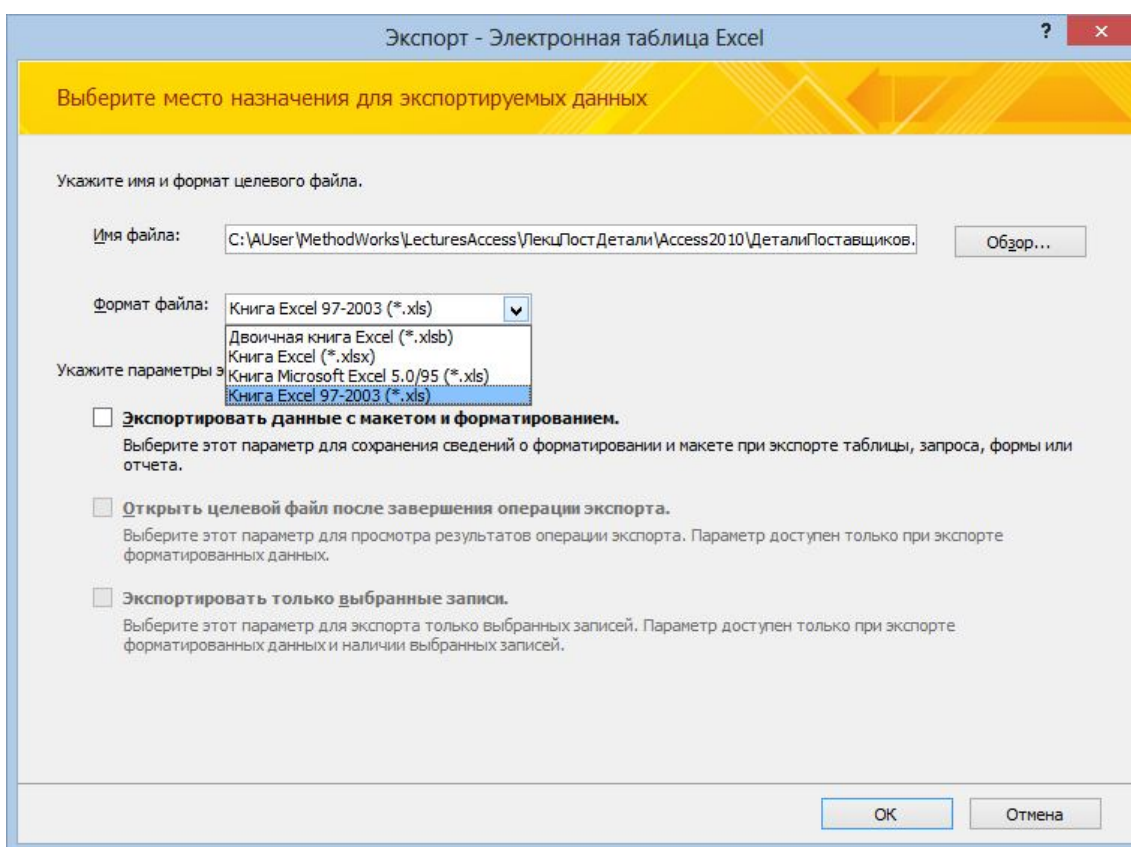


Рис. 6.62. Настройки экспорта в Excel

По данным запроса в папке **Мои документы** будет создана рабочая книга Excel, имя которой совпадает с именем таблицы или запроса. На рис.6.63 показан лист Excel с этим запросом.

В Excel можно проводить любую доступную обработку данных, например, на рис. 6.63 показано суммирование веса всех поставленных деталей, для чего была выбрана ячейка **D13** и нажата кнопка  $\Sigma$  на панели инструментов.

Microsoft Excel - ДеталиПоставщиков

Файл Правка Вид Вставка Формат Сервис Данные Окно Справка

MS Sans Serif 10 Ж К Ч

D13 =СУММ(D2:D12)

	A	B	C	D	E	F	G	H	I
2	Шпилька	Лента	1200	1200					
3	Муфта	Бриз	70	140					
4	Поршень	Союз	20	40					
5	Поршень	Астра	50	100					
6	Ось	Витязь	30	3					
7	Шкворень	Союз	90	270					
8	Гайка	Виктория	1500	1500					
9	Гайка	Полет	500	500					
10	Штуцер	Виктория	300	6000					
11	Болт	Прииск	1500	3000					
12	Шайба	Астра	500	250					
13				13003					

ДеталиПоставщиков

Действия Автофигуры

Готово

Рис. 6.63. Запрос *ДеталиПоставщиков*, экспортированный в Excel

## 7. Макросы

Access имеет в своем составе набор *макрокоманд*, которые позволяют выполнять некоторые типовые действия, например, имеется макрокоманда **Открыть** (таблицу, форму, отчет).

*Макрос* это набор макрокоманд, используемый для автоматического выполнения последовательности некоторых действий. Макросы полезны для автоматизации часто выполняемых последовательностей действий.

Макросы имеются также в Word, Excel, но там они представляют собой подпрограммы на VBA. Кроме того, в указанных программах имеется возможность запоминать выполняемую последовательность действий пользователя в виде макроса. Такой возможности в Access нет.

### 7.1. Создание макроса

Для создания макроса нужно выбрать инструмент **Макрос** из группы **Макросы и код** на вкладке **Создание**. Для создания макроса нужен монопольный доступ к базе данных, поэтому, если нет монопольного доступа, появится окно (рис.7.1)

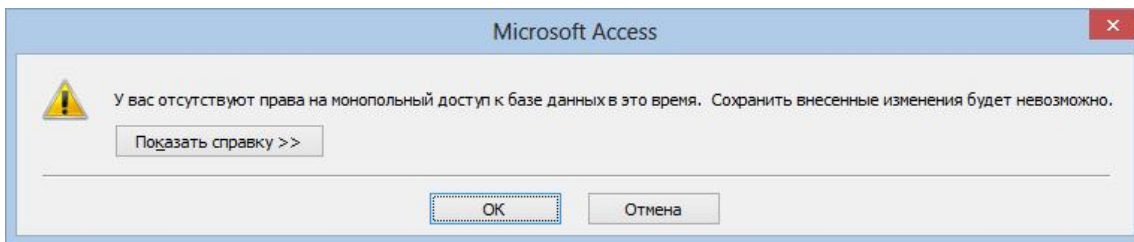


Рис. 7.1. Информация о необходимости монопольного доступа к базе данных

Для получения монопольного доступа к базе данных нужно ее закрыть и снова открыть, выбрав **Монопольно** в списке кнопки **Открыть** (рис.6.44). При открытии базы данных в монопольном режиме возможно появление сообщения о невозможности это сделать (рис.7.2). Так бывает, если, например, на момент открытия базы данных с ней работает Word в режиме создания документа слияния. В таком случае нужно закрыть Word и повторить открытие базы данных.

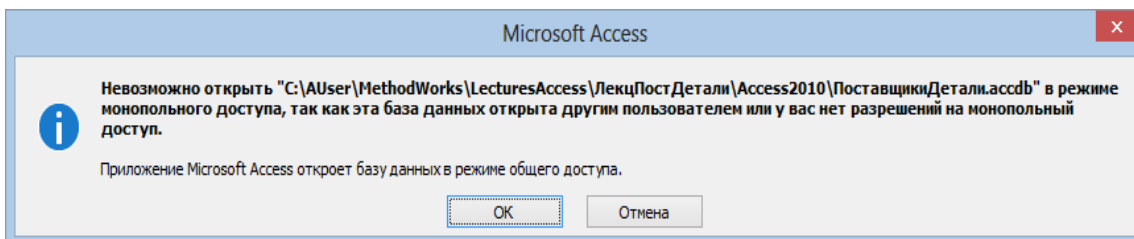


Рис. 7.2. Сообщение о невозможности открыть базу данных в монопольном режиме

Итак, выполним команду **Макрос**. Появится бланк конструктора макросов (рис. 7.3), с помощью которого в макрос вставляются макрокоманды.

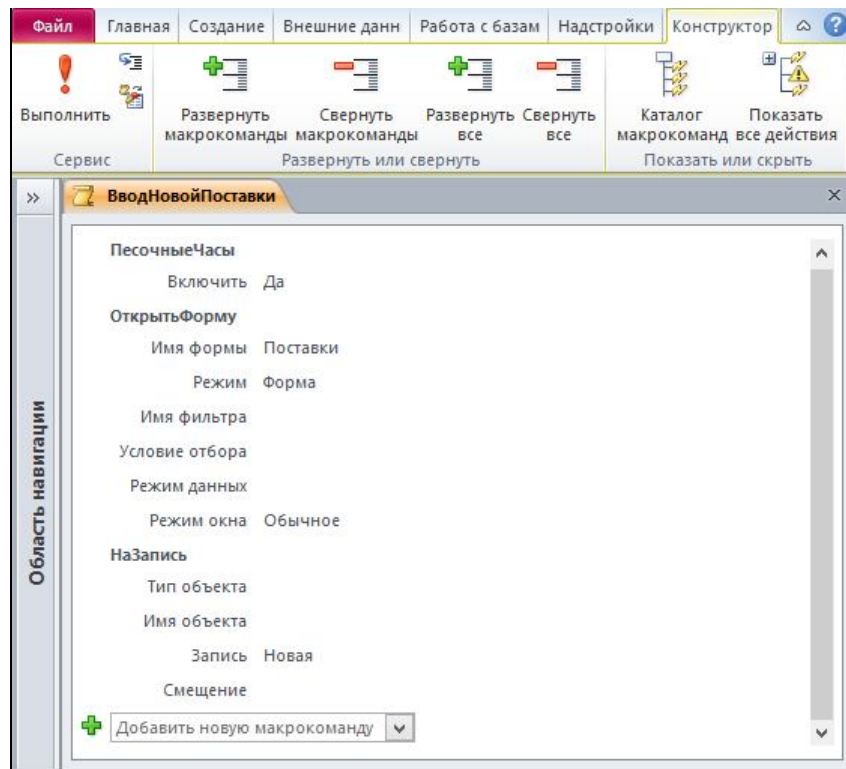


Рис. 7.3. Бланк конструктора макросов

Созданный макрос можно сохранить командой **Файл, Сохранить**.

В макрос *ВводНовойПоставки* включены три команды:

**ПесочныеЧасы** - задает форму курсора на время выполнения команды;

**ОткрытьФорму** - открывает форму, заданную параметром **Имя формы**;

**НаЗапись** - выполняет переход к записи, указанной в параметре **Запись**.

Макрокоманда выбирается из списка **Добавить новую макрокоманду**.

На рис.7.4 показан режим задания параметров макрокоманды **ОткрытьФорму**.

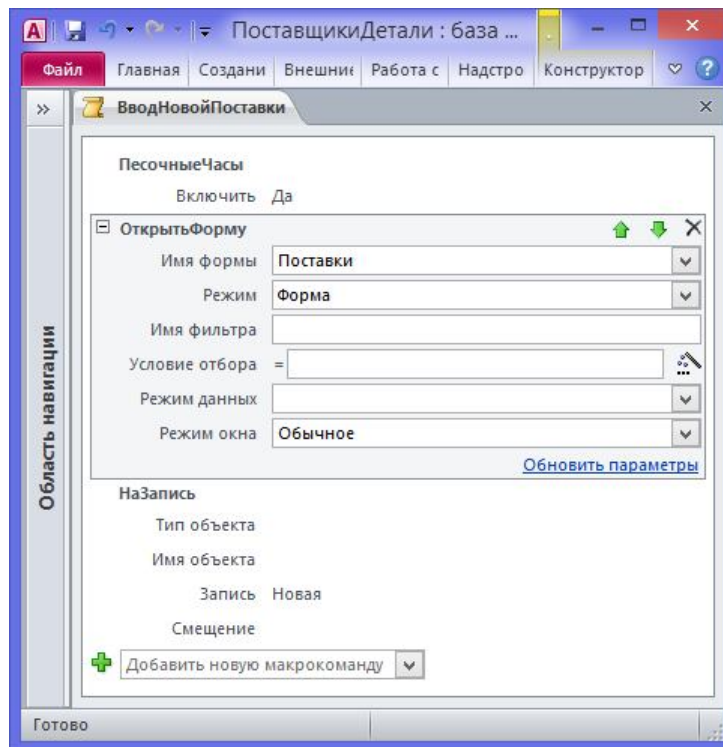


Рис. 7.4. Задание параметров макрокоманды



Макрос **ВводНовойПоставки** предназначен для ввода данных о новой поставке, для чего в него включена команда **ОткрытьФорму** с аргументом *Поставки*, и команда **НаЗапись** с аргументом **Новая**, что обеспечит переход в форме *Поставки* к вводу новой записи.

Макрос запускается на выполнение кнопкой **Выполнить**.

## 7.2. Вложенные макросы

Работая в режиме конструктора макросов, нажмем кнопку **Каталог макрокоманд** из группы **Показать или скрыть** (рис. 7.3). В окне **Каталог макрокоманд** имеется инструмент **Вложенный макрос**, с помощью которого можно создавать в составе макроса наборы макрокоманд, имеющие собственное имя. Создадим два набора макрокоманд - *Детали* и *Поставщики* (рис.7.5). Сам макрос назовем **МакросДеталиПоставщики**.

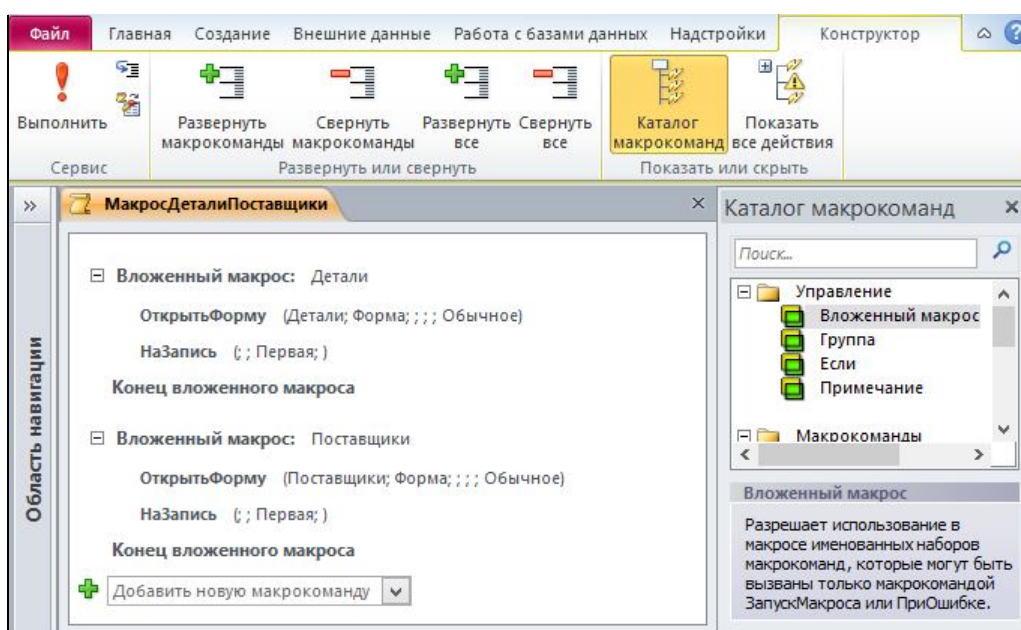


Рис. 7.5. Группы макросов

При запуске макроса командой **Выполнить** будет выполнен только первый набор макрокоманд, названный в рассматриваемом примере *Детали*.

Для запуска вложенного макроса нужно перейти на вкладку **Работа с базами данных**, выбрать инструмент **Выполнить макрос**, затем из выпавшего списка выбрать нужный макрос (рис.7.6). Имя вложенного макроса является составным и записывается в виде *МакросДеталиПоставщики.Поставщики* – через точку.

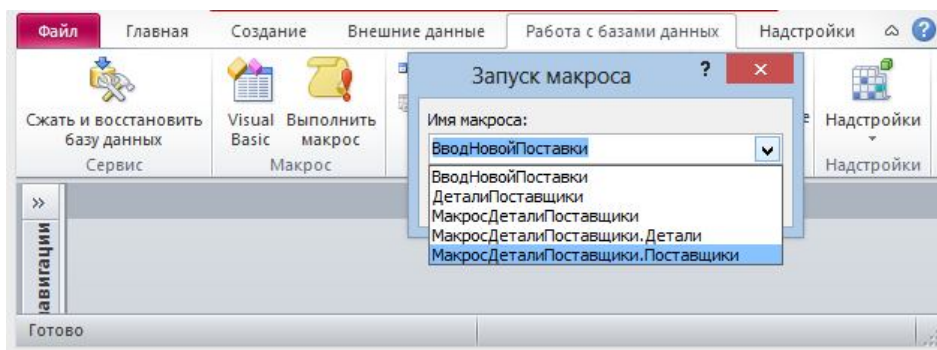


Рис. 7.6. Диалоговое окно **Запуск макроса**

## 7.3. Условные макрокоманды

При работе с конструктором макросов можно инструментом **Если** вставить в состав макроса условную макрокоманду, которая будет выполняться при истинности заданного условия (рис. 7.7).

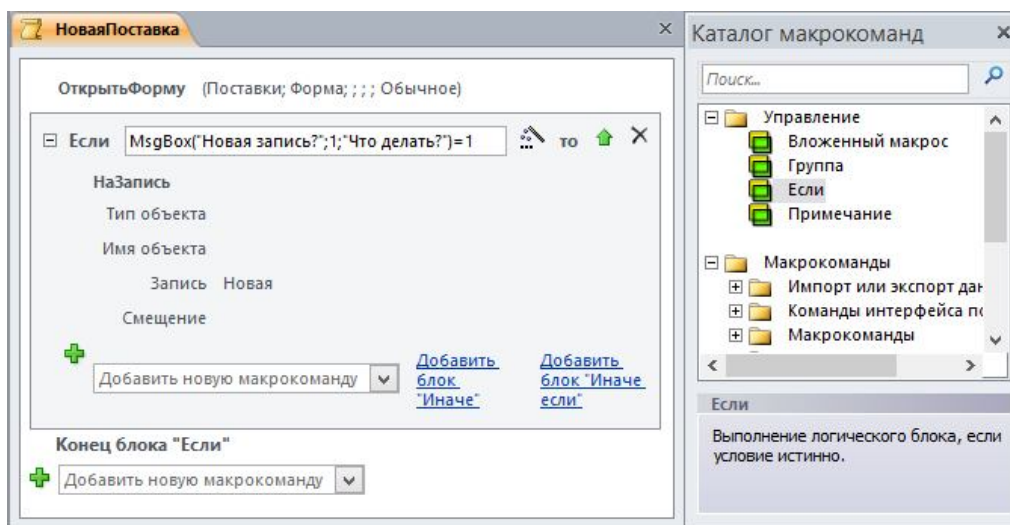



Рис. 7.7. Конструктор макросов с условной макрокомандой

Выражение, истинность которого будет проверяться, можно вводить в поле **Если** или создавать с помощью построителя выражений, который запускается кнопкой , расположенной справа. Окно построителя выражений приведено на рис.7.8.

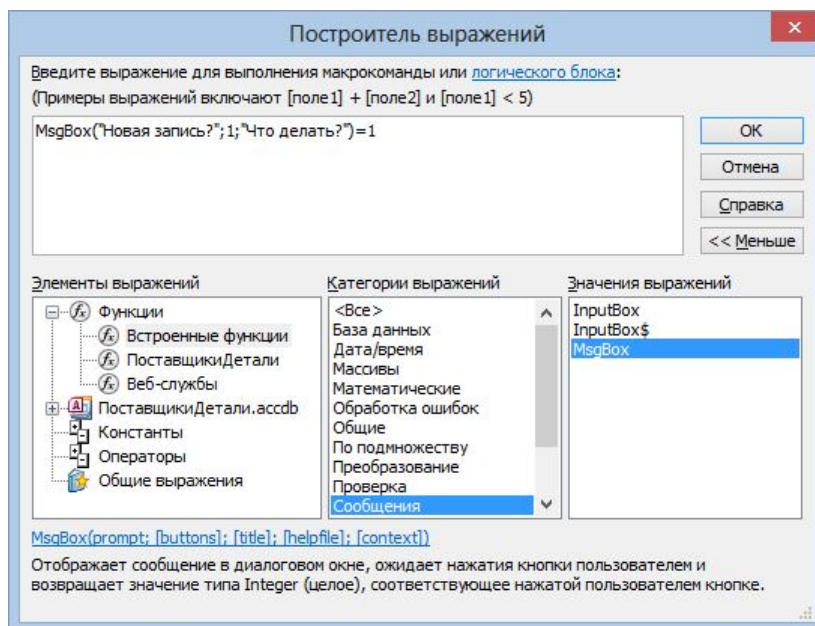


Рис. 7.8. Построение условного выражения

Условие, показанное на рис.7.7, состоит в том, что выводится окно сообщения, на котором пользователь должен нажать кнопку **ОК** для перехода к вводу новой записи. Если кнопка **ОК** нажата не будет, макрокоманда не выполнится (рис.7.9).

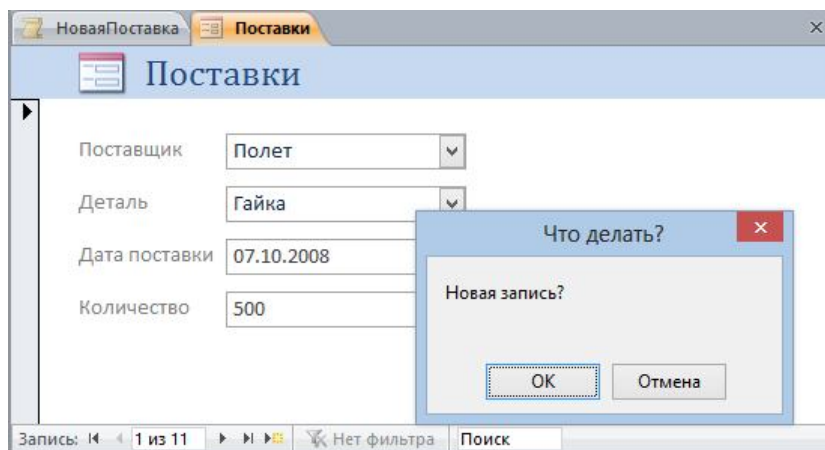


Рис. 7.9. Окно сообщения с вопросом

## 7.4. Запуск макроса

При запуске макроса выполнение макрокоманд начинается с первой строки макроса и продолжается до конца макроса или до начала следующего макроса, если макрос входит в группу.

Макроса может выполняться по команде пользователя, при вызове из другого макроса или процедуры обработки события, а также в ответ на событие в форме, отчете или элементе управления. Например, можно связать макрос с кнопкой на форме или отчете, в результате чего макрос будет выполняться при нажатии этой кнопки. Допускается также создание специального командного меню или кнопки панели инструментов, запускающей макрос, определение сочетания клавиш, которое запускает макрос, а также автоматический запуск макроса при открытии базы данных.

Создадим макрос **ПерваяЗапись**. Для этого выберем макрокоманду **НаЗапись**, в качестве значения аргумента **Запись** выберем из списка значение **Первая** (рис.7.10).

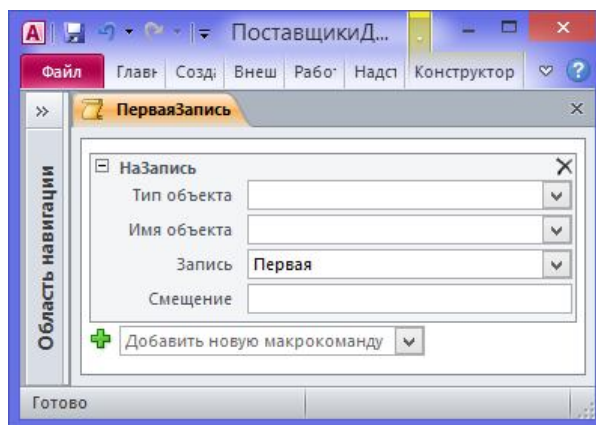


Рис. 7.10. Макрос без конкретного аргумента

Обратим внимание на то, что в макросе не указана конкретная таблица или запрос, в которых следует перейти к первой записи.

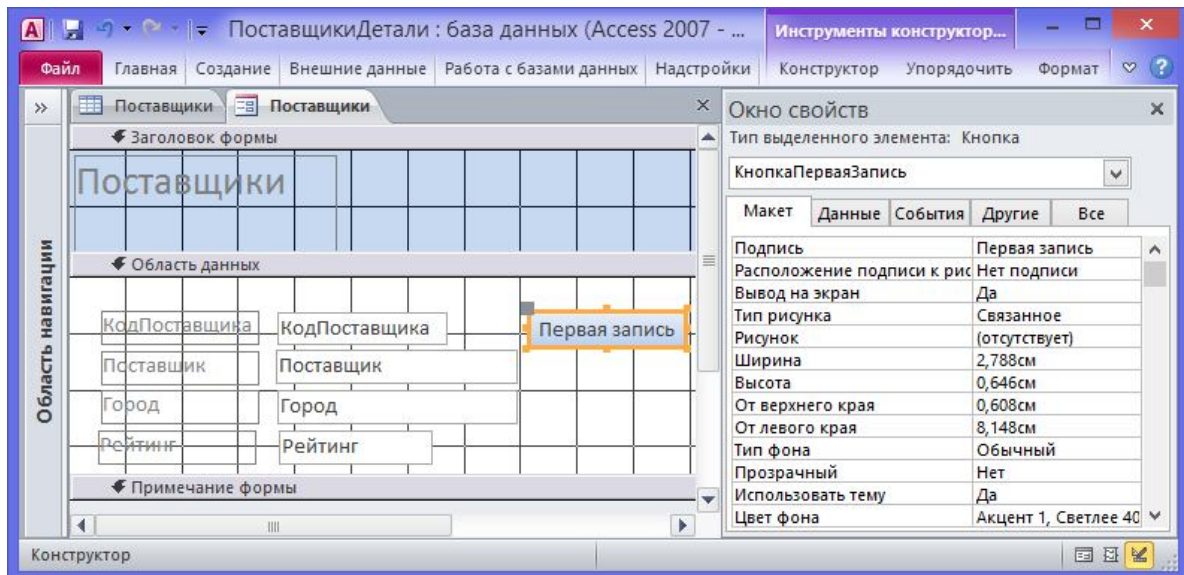


Рис. 7.11. Форма *Поставщики* в режиме конструктор и окно свойств кнопки

На форме **Поставщики** для работы с записями из таблицы *Поставщики* поместим кнопку (рис.7.11), и выведем для нее окно свойств. Свойству **Подпись** (вкладка **Макет**), которое содержит надпись на кнопке, дадим значение *Первая запись*, откажемся от показа рисунка на кнопке, удалив его из свойства **Рисунок**. На вкладке **Другие** введем для свойства **Имя** значение *КнопкаПерваяЗапись*, а событию **Нажатие кнопки** на вкладке **События** выбором из списка присвоим значение макроса *ПерваяЗапись* (рис.7.12).

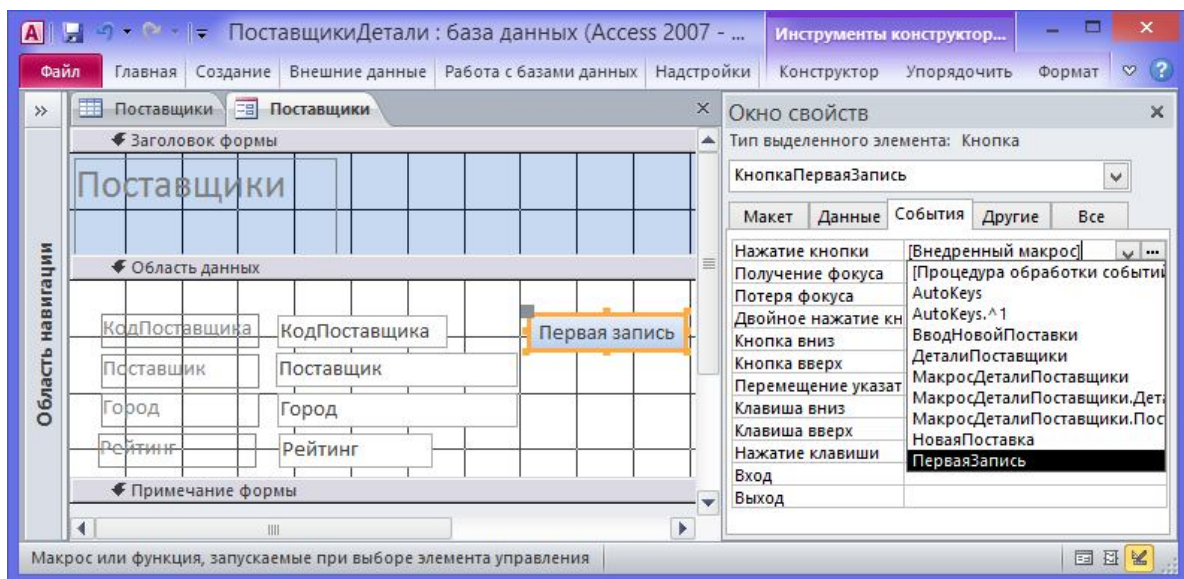


Рис. 7.12. Выбор макроса для обработки события

В результате при работе с данной формой в любой момент времени нажатие созданной кнопки выведет в форму первую запись.

## 7.5. Назначение клавиш для запуска макросов

Для того, чтобы макрокоманда или набор макрокоманд выполнялись по нажатию какой-либо клавиши или комбинации клавиш, нужно создать макрос с именем **AutoKeys** и включить в состав этого макроса отдельные макрокоманды или наборы макрокоманд со специальными именами, обозначающими клавиши или сочетания клавиш. Описания клавиш имеют вид:

<b>^ A</b> или <b>^4</b>	<b>Ctrl+</b> любая буква или любая цифра;
<b>{F1}</b>	любая функциональная клавиша;
<b>+{F1}</b>	<b>Shift+</b> любая функциональная клавиша;
<b>^{F1}</b>	<b>Ctrl+</b> любая функциональная клавиша;
<b>{INSERT}</b>	клавиша <b>Ins</b> ;
<b>^{INSERT}</b>	комбинация <b>Ctrl+Insert</b> ;
<b>+{INSERT}</b>	комбинация <b>Shift+Insert</b> ;
<b>{DELETE}</b> или <b>{DEL}</b>	клавиша <b>Delete</b> ;
<b>^{DELETE}</b> или <b>^{DEL}</b>	комбинация <b>Ctrl+Del</b> ;
<b>+{DELETE}</b> или <b>+{DEL}</b>	комбинация <b>Shift+Del</b> .

Сочетания клавиш вступят в действие сразу после сохранения макроса **AutoKeys** и будут доступны при каждом открытии базы данных. Если макрокоманда или набор макрокоманд связывается с сочетанием клавиш, которое уже используется в Access (например, **CTRL+C** – сочетание клавиш для команды **Копировать**), то новое назначение макрокоманд на это сочетание клавиш заменит стандартное назначение.

На рис.7.13 показан макрос **AutoKeys** в режиме конструктора, содержащий одну команду **НаЗапись**, которая будет выполняться при нажатии комбинации клавиш **Ctrl+1**, что обозначено как **^1**. Чтобы эта макрокоманда успешно выполнялась, должна быть открыта таблица, запрос или форма. Для быстрого перехода к первой записи в таблице, запросе или форме нужно нажать комбинацию клавиш **Ctrl+1**.

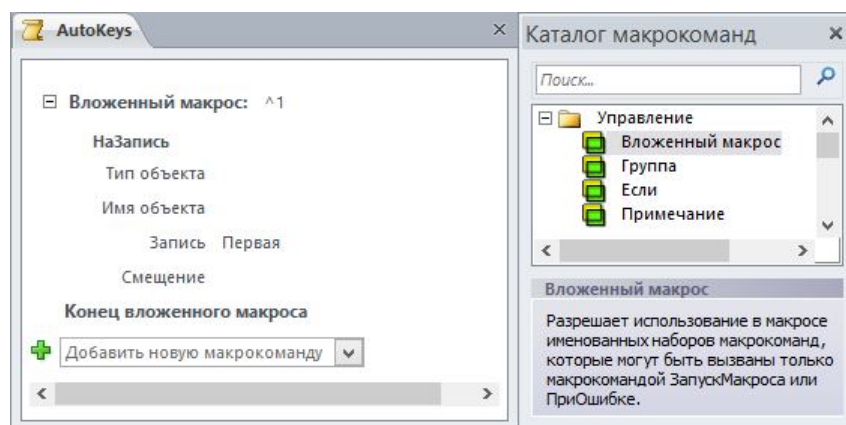


Рис. 7.13. Комбинация клавиш и имя для макроса

# Литература

## Основная

1. Швецов В.И., Визгунов А.Н., Мееров И.Б. Базы данных.- Н.Новгород: Изд-во ННГУ, 2004.-267с. (<http://www.unn.ru/rus/persons/shvetsov/>)
2. Дейт К. Введение в системы баз данных.– М.: Издательский дом «Вильямс», 2001.– 1072с.
3. Крёмке Д. Теория и практика построения баз данных.– СПб.: Питер, 2005.–859с.
4. Карпова Т.С. Базы данных: модели, разработка, реализация.– СПб.: Питер, 2002.– 304с.
5. Хомоненко А.Ф., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высш. уч. завед./ Под ред. проф. А.Д.Хомоненко.– СПб.: КОРОНА принт, 2002.– 672
6. Харрингтон Д.Л. Проектирование реляционных баз данных. Просто и доступно.– М.: Лори, 2000.– 230с.
7. Гурвиц Г. Microsoft Access 2010. Разработка приложений на реальном примере.- СПб.: БХВ-Петербург, 2010.-496 с.
8. Сеннов А. Access 2010. Учебный курс.- СПб.: Питер, 2010.- 288с.
9. Хелворсон М., Янг М. Эффективная работа с Microsoft Office 2000. – СПб.: Питер, 2000.– 1232 с.
10. Золотова С.И. Практикум по Access.– М.: Финансы и статистика, 2000.–144с.
11. Штайнер Г. Access 2000. М.: Лаборатория базовых знаний, 2000.– 480с.
12. Андерсен В. Базы данных Microsoft Access. Проблемы и решения.– М.: Эком, 2001.– 384с.
13. Карпов Б. Microsoft Access 2000. Справочник.– СПб.: Питер, 2001.– 416с.

## Дополнительная

14. Гурвиц Г.А. Microsoft Access 2007. Разработка приложений на реальном примере.— СПб.: БХВ-Петербург, 2007. — 672 с.
  15. Тимошок Т.В. Microsoft Access 2002. Краткое руководство. М.: Диалектика, 2004.– 272 с.
  16. Бекаревич. Ю., Пушкина Н. Самоучитель Microsoft Access 2003.– СПб.: БХВ-Петербург, 2004.– 738 с.
  17. Хобракен Д. Microsoft Access 2000. Шаг за шагом.– М.: АСТ; Астрель, 2004.– 350 с.
  18. Пасько В. Access 2000 (русифицированная версия).– К.: Издательская группа ВНУ, 1999.–384с.
- Харитоновна И.А., Михеева В.Д. Microsoft Access 2000.– СПб.: БХВ-Петербург, 2001.– 1088с.

Владимир Леонидович Тарасов

# **Работа с базами данных в Access 2010**

## **Часть 1**

Учебно-методическое пособие

Федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«Нижегородский государственный университет  
им. Н.И. Лобачевского».  
603950, Нижний Новгород, пр. Гагарина, 23.