

Министерство образования и науки Российской Федерации

Федеральное агентство по образованию

Саратовский государственный технический университет

**Практические задания по созданию баз данных в СУБД
Access.**

Методические указания
к проведению лабораторных работ по курсу «Информационные системы» для
студентов специальности «Прикладная информатика»

Электронное издание локального распространения

*Одобрено
редакционно-издательским советом
Саратовского государственного
технического университета*

Саратов 2006

Все права на размножение и распространение в любой форме остаются за разработчиком.

Нелегальное копирование и использование данного продукта запрещено.

Составитель: Сытник Ирина Федоровна
Под редакцией Сытник Ирины Федоровны

Рецензент: Горопова О.А.

410054, Саратов, ул. Политехническая, 77,
Научно-техническая библиотека СГТУ
Тел. 52-63-81, 52-56-01
<http://lib.sstu.ru>

Регистрационный номер

© Саратовский государственный
технический университет, 2006

Задание 1. Создание структуры таблиц БД ЗАКАЗ в режиме Конструктора

Для создания таблицы в Access следует описать ее структуру в режиме Конструктора. Создание структуры таблицы состоит из следующих действий:

1. Находясь в главном окне базы данных Access, выбрать тип объекта с надписью **Таблицы**;
2. Нажать кнопку **Создать** (в режиме Конструктора таблиц);
3. Описать все поля таблицы
4. Задать первичный ключ
5. Ввести имя таблицы и сохранить ее структуру

Создание структуры таблицы **Заказы** приведено на рисунке 1. Из рисунка видно, что описание каждого поля включает:

1. Имя поля
2. Тип данных поля
3. Описание (словесное)
4. Свойства поля (в частности, размер и формат поля, маску ввода для упорядочивания заполнения поля, начальное значение, условие на вводимые значения, сообщение об ошибке при нарушении наложенного условия, индексированно поле или нет, ключевое поле или нет и т.д.)

Придерживайтесь следующих рекомендаций при описании полей:

- Если имя состоит из двух или более слов, то запишите его слитно, сделав первые буквы заглавными, например, *ОбщаяСумма* и т.д.
- При установке типа данных, свойства поля устанавливаются автоматически; любое изменение свойств проводите осмысленно
- Не ленитесь писать комментарии в описаниях полей.



Рис.1

Первичный ключ, то есть *ключевое поле* определяют после задания имен и типов полей, не выходя из режима Конструктора. Для этого нужно выделить поле и нажать кнопку **Ключевое поле** на Панели инструментов.

Сохранить структуру таблицы и ввести ее имя можно при закрытии окна Конструктора таблиц либо нажатием кнопки **Сохранить** на Панели инструментов.

Аналогично создать структуры всех таблиц БД. Поскольку в последней таблице два поля являются составным ключом, то для его создания надо выделить эти два поля одновременно, а затем нажимать на кнопку с ключом.

Структура таблиц базы данных ЗАКАЗ

Заказы

Ключ	Имя поля	Тип поля	Примечание
<i>К</i>	<i>КодЗаказа</i>	<i>Счетчик</i>	<i>Номер заказа - первичный ключ</i>
	<i>КодКлиента</i>	<i>Числовой/Длинное целое</i>	<i>Поле связи с таблицей Клиенты</i>
	<i>ДатаЗаказа</i>	<i>Дата/Время</i>	<i>Дата выписки заказа</i>
	<i>КодМенеджера</i>	<i>Числовой/Длинное целое</i>	<i>Поле связи с таблицей Менеджеры</i>
	<i>ОбщаяСумма</i>	<i>Денежный</i>	<i>Стоимость заказа</i>

Клиенты

Ключ	Имя поля	Тип поля	Примечание
<i>К</i>	<i>КодКлиента</i>	<i>Счетчик</i>	<i>Это первичный ключ</i>
	<i>ФИО</i>	<i>Текстовый</i>	<i>Длина 40 символов</i>
	<i>Город</i>	<i>Текстовый</i>	<i>Длина 15 символов</i>
	<i>Улица</i>	<i>Текстовый</i>	<i>Длина 40 символов</i>
	<i>Телефон</i>	<i>Текстовый</i>	<i>Длина 25 символов</i>

Товары

Ключ	Имя поля	Тип поля	Примечание
<i>К</i>	<i>КодТовара</i>	<i>Счетчик</i>	<i>Это первичный ключ</i>
	<i>Название</i>	<i>Текстовый</i>	<i>Длина 100 символов</i>
	<i>Цена</i>	<i>Денежный</i>	

Менеджеры

Ключ	Имя поля	Тип поля	Примечание
<i>К</i>	<i>КодМенеджера</i>	<i>Счетчик</i>	<i>Это первичный ключ</i>
	<i>ФИОМенеджера</i>	<i>Текстовый</i>	<i>Длина 40 символов</i>
	<i>Телефон</i>	<i>Текстовый</i>	<i>Длина 25 символов</i>

ЗаказаноТовара

<i>Ключ</i>	<i>Имя поля</i>	<i>Тип поля</i>	<i>Примечание</i>
<i>К</i>	<i>КодЗаказа</i>	<i>Числовой/Длинное целое</i>	<i>Связывается с полем типа Счетчик</i>
<i>К</i>	<i>КодТовара</i>	<i>Числовой/Длинное целое</i>	<i>Связывается с полем типа Счетчик</i>
	<i>Количество</i>	<i>Числовой/Целое</i>	
	<i>Сумма</i>	<i>Денежный</i>	

Задание 2. Создание схемы данных

Схема данных в Access связывает таблицы в единую базу данных и обеспечивает ссылочную целостность базы. Процесс создания связей таблиц состоит из двух частей: добавления таблиц в схему данных и собственно установления связей.

Для добавления таблиц в схему данных выполните действия:

- Находясь в главном окне БД Access, нажмите кнопку **Схема данных** на панели инструментов – откроется соответствующее окно;
- Выполните команду меню **Связи/Добавить таблицу** (или нажмите кнопку на панели инструментов **Отобразить таблицу** и в появившемся окне выберите и добавьте все таблицы. Расположите их в окне с помощью мышки по вашему усмотрению.

Установка связи между полями двух таблиц предполагает следующие действия:

- Щелкните по связываемому полю одной таблицы и «перетащите» его на связываемое поле другой таблицы (рис.2);
- В появившемся окне (рис.3) установки параметров связи поставьте флажок «Обеспечение целостности данных»; после нажатия кнопки **Создать** связь будет создана и отображена на схеме данных.

При корректировке БД, если изменения производятся в полях связи (например, изменения имени или типа данных поля), сначала удаляется связь в схеме данных. Ее необходимо выделить, а затем нажать клавишу **Delete**. После корректировки нужно открыть схему данных и восстановить удаленную связь.

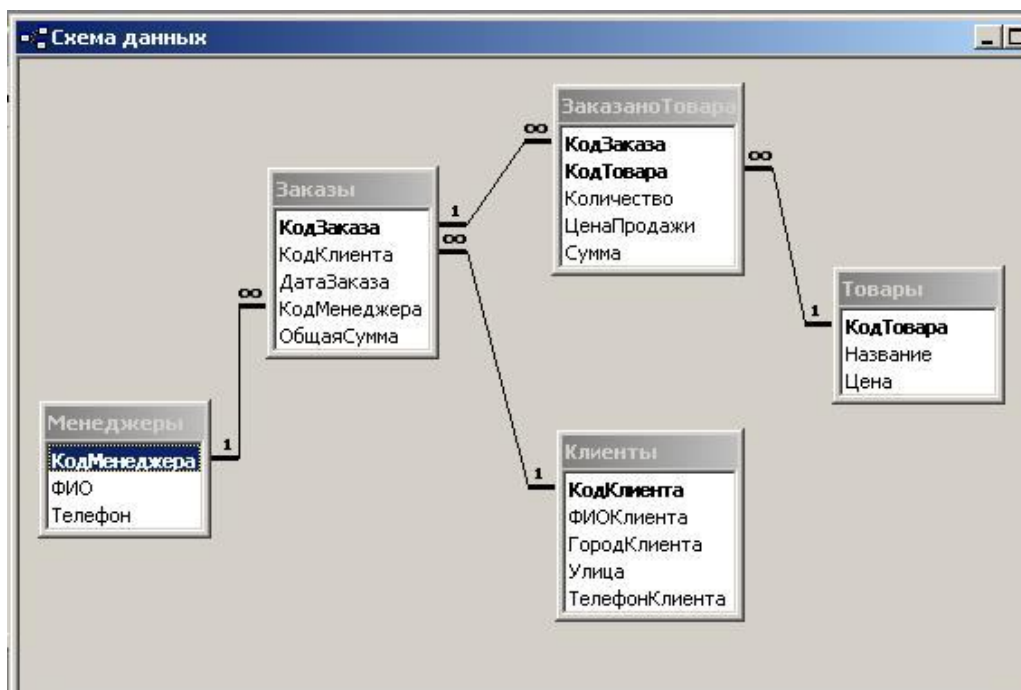


Рис.2. Итоговая схема данных

Рис. 3. Окно установки параметров связи.

После создания схемы данных заполните данные в таблицах о трех заказах. Ввод данных в таблицы следует выполнить, начиная с периферийных, чтобы соблюсти ссылочную согласованность данных. Например, сначала или таблицы **Клиенты**, **Мменеджеры**, **Товары**, затем таблицу **Заказы** и в самом конце таблицу **ЗаказаноТоваров**. Это означает, что при заполнении таблицы **Заказы** коды клиентов и менеджеров должны уже существовать в таблице **Клиенты**, а при заполнении таблицы **ЗаказаноТоваров** коды заказов и товаров также должны присутствовать соответственно в таблицах **Заказы** и **Товары**. В противном случае будет выдано сообщение об отсутствии связной записи, а добавление записи станет невозможным.

Задание 3. Простые формы

Форма — это объект Access, который *отображает* данные на экране в рамках некоторого окна.

Форма, отображающая одновременно одну запись в области данных, является простой. Форма, отображающая в области данных последовательность из нескольких записей, является *ленточной*.

Для формы, как правило, определен *источник данных* — это таблица или запрос. Данные из источника данных обычно размещают в области данных формы. Данные, изменяемые или вводимые в форме, сохраняются в таблицах.

Элементы управления

На форме можно размещать графические объекты, называемые в Access элементами управления. Это различные списки, переключатели, флажки, кнопки, рисунки и т.д., знакомые вам из других программ.

Элементы управления создаются на форме с помощью **Панели элементов управления**. **Панель элементов** включается/выключается нажатием одноименной кнопки на **Панели инструментов**.

Надписи используются для вывода текста в форме. *Поля ввода* — для ввода в форму некоторых данных, не привязанных к полям таблиц. *Переключатели и группы переключателей* - для выбора на экране одного из нескольких зависимых вариантов. *Флажки* — для выбора нескольких независимых вариантов. Выключатели обычно связаны с логическим полем и изменяют его значение. *Кнопки* — для инициирования некоторого действия и т.д.

Все элементы формы, в том числе и сама форма, характеризуются набором свойств и реакцией на определенные события (нажатие мыши, клавиши на клавиатуре и т.д.). Все эти характеристики можно просмотреть и изменить в окне **Свойства**, где они сгруппированы в 5 закладках. Окно **Свойства** вызывается в режиме Конструктора формы одноименной кнопкой на **Панели инструментов**.

В закладке **Макет** объединены свойства, касающиеся внешних атрибутов элемента: размер, цвета, обрамление, шрифты, формат, режим по умолчанию и пр.

В закладке **Данные** указываются источник данных элемента и другие свойства доступа к этим данным.

В закладке **События** перечислены события, на которые может реагировать элемент, такие как **Удаление**, **Нажатие кнопки**, **После обновления** и т.д.

Создание простых форм Клиенты, Товары, Менеджеры

Вид формы Клиенты показан на рис. 9. Это простая форма: клиентов можно просматривать по одному, листая записи кнопками в нижней части формы.

Для создания такой формы сделайте следующие действия.

1. Выберите объект **Формы** в окне базы данных Access.
2. Выберите режим **Создать**, далее **Конструктор**. Не пропустите выбор источника записей (данных) — таблица **Клиенты**. Откроется окно пустой формы.
3. Отобразите на экране область заголовка: из меню **Вид** выберите пункт **Заголовок/примечание формы**.
4. В появившуюся область заголовка добавьте надпись *Клиенты*. Для этого выберите на **Панели элементов** элемент **Надпись** (кнопку **Aa**), затем вставьте и растяните его в области Заголовка формы. Введите в элемент слово *Клиенты* и нажмите клавишу **Enter**. Для появившейся надписи *Клиенты* установите красный цвет и жирный шрифт размера 16.
5. Отобразите перечень полей таблицы **Клиенты**, нажав кнопку **Список полей** на Панели инструментов, если он не отображается автоматически.

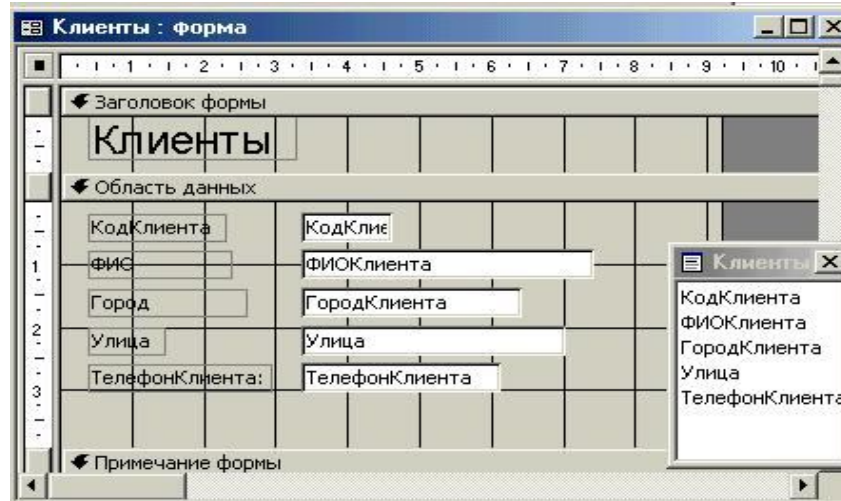


Рис. 9. Форма Клиенты в режиме Конструктора

6. Перенесите все поля из таблицы в форму. Для переноса поля из списка полей на форму щелкните по полю и, удерживая нажатой левую кнопку мыши, перетащите поле в область данных формы (рис. 9).
7. Разместите поля на форме по своему вкусу и скорректируйте размеры области данных. Местоположение и размеры полей и надписей можно изменять, для этого элемент выделяют, а затем его перетаскивают, растягивают или сужают. На этом этапе вам следует потренироваться

с изменением внешних свойств полей и надписей: попробуйте выделить любой элемент на форме, измените его размеры, цвет символов, цвет фона, размер символов и т.д. При необходимости можно выровнять расположение полей, надписей к ним и расстояние между ними с помощью команды меню **Формат** пункты **Выровнять**, **Размер**, **Интервал по вертикали**, **Интервал по горизонтали**.

8. Скорректируйте размер рабочей (серой) области до нужных размеров. До минимума сокращайте размеры неиспользуемого серого фона (пространства) между областями формы.

9. Закройте окно Конструктора формы и введите имя формы - **Клиенты**. Вы создали простую форму **Клиенты**. Откройте ее в режиме просмотра; на ней размещены отобранные вами поля и надписи к ним.

Аналогично создайте форму **Товары**, которая может иметь вид, показанный на рис. 10. Источник данных для этой формы – таблица **Товары**. Самостоятельно создайте форму **Менеджеры**

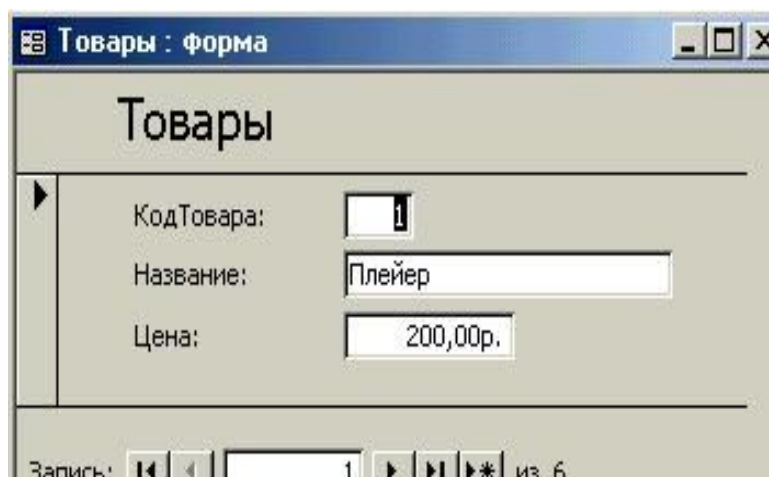


Рис.10. Вид формы Товары в режиме просмотра

Разработка формы Заказы. Создание поля со списком

Форма *Заказы* — это основная форма, через которую будет происходить диалог с пользователем.

В качестве источника данных формы следует использовать таблицу *Заказы*. Разработку формы *Заказы* начнем с создания полей *КодЗаказа*, *ДатаЗаказа*, *ОбщаяСумма* и надписи «Заказы».

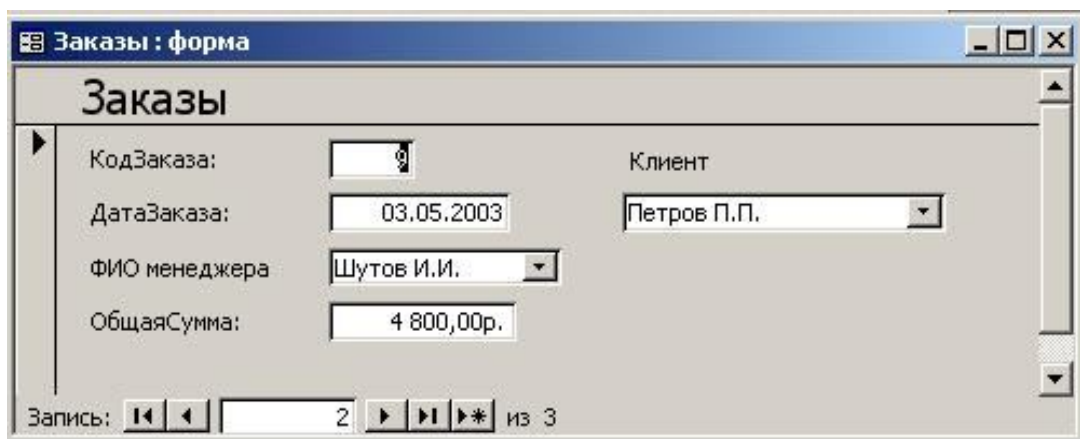





Рис.11. Предварительный вид формы Заказы

Для создания поля со списком Клиент воспользуемся **Мастером** создания элемента, который позволяет автоматизировать «настройку» элемента управления при его переносе на форму. Потребуется выполнить следующие действия.

1. Отобразите **Панель элементов**, нажав кнопку  на **Панели инструментов**, и на ней включите кнопку с всплывающей надписью **Мастера** . Отобразите также **Список полей**, нажав кнопку  на **Панели инструментов**.
2. Щелкните по элементу **Поле со списком** на **Панели элементов** и сразу же щелкните по полю КодКлиента в **Списке полей**. Установите указатель мыши в нужное место формы (он примет форму элемента) и, удерживая левую кнопку, растяните элемент до требуемых размеров.
3. В этом месте будет запущен **Мастер** создания поля со списком. Он последовательно выводит несколько диалоговых окон, в которых запрашивает у вас уточняющие данные. В первом окне уточняется источник данных для поля: выберите **пункт поле со списком использует значения из таблицы или запроса** и нажмите кнопку **Далее**. В следующем окне уточняется имя таблицы или запроса: выберите таблицу *Клиенты* и нажмите **Далее**. В следующем окне уточняются поля, которые требуется выводить в списке: выберите поля КодКлиента, ФИО и нажмите **Далее**. Следующее окно позволяет задать ширину (потянув за название поля) и видимость полей списка: задайте ширину полей (потянув за название поля) и нажмите **Далее**. Следующее окно уточняет, куда будет запоминаться выбираемое из списка значение: выберите опцию **Сохранить в поле** и поле Код Клиента и нажмите **Далее**. В

следующем окне запрашивается надпись к полю со списком: введите Клиент и нажмите **Готово**.

4. Полученное поле со списком измените до нужных размеров и сохраните форму (с именем Заказы).

Проверьте работу полученного списка. Для этого откройте форму Заказы, а в ней щелкните по созданному списку: в нем должны быть перечислены фамилии клиентов.

Аналогично создайте поле со списком **ФИО менеджера**

Задание 4. Создание запросов

Запросы — это объекты Access, которые позволяют выбирать необходимую информацию из таблиц, и представлять ее опять же в табличном виде.

Запрос 1.

Составим запрос, выдающий полную информацию о клиентах в заказах. Для этого сформируем запрос к двум таблицам — *Заказы* и *Клиенты*. Прделаите следующие действия.

1. Выберите категорию **Запросы**, находясь в окне базы данных Access, нажмите кнопку **Создать** и выберите **Конструктор**. Откроется окно построителя запросов.

2. Добавьте в запрос таблицы Заказы и Клиенты (в случае конструирования существующего запроса, окно добавления вызывается кнопкой **Добавление таблиц** на **Панели инструментов**). Отметим, что здесь базовой таблицей для запроса является таблица *Заказы*,

3. Последовательно перетащите вниз, в строку с надписью **Поле**, показанные на рис. 3, поля из этих таблиц, - эти поля и будут содержаться в итоговой таблице.

4. Можно установить сортировку поля (здесь поле *ОбщаяСумма*). Можно также установить условие отбора (здесь >600). После таких установок, в режиме просмотра, вы увидите только записи с *ОбщаяСумма* >600 , отсортированные по возрастанию.

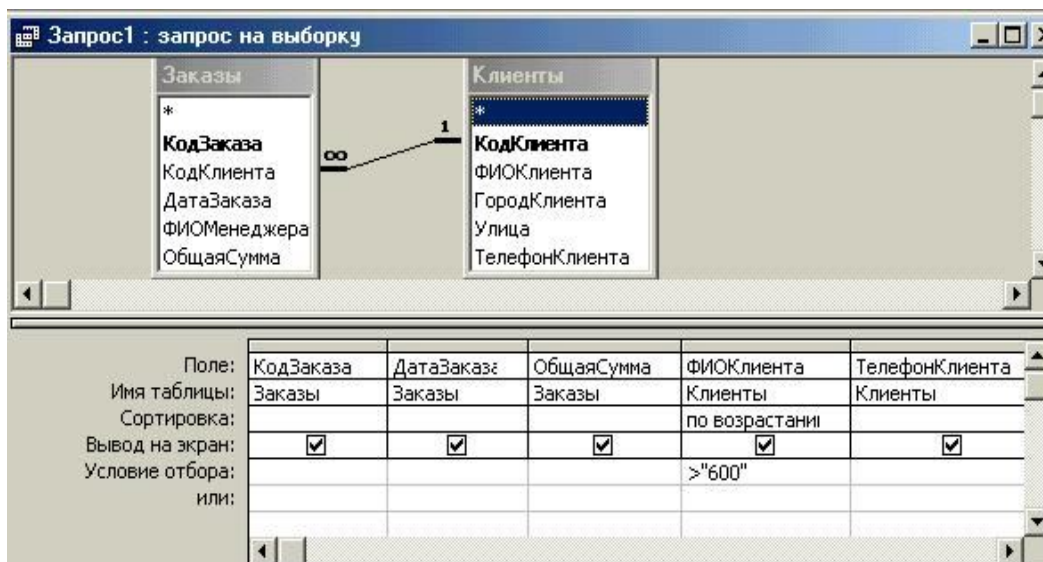


Рис.3. Окно построителя запросов

5. Закройте окно построителя запросов, сохраните запрос с именем ЗаказыКлиентов — в окне базы данных Access появится новый запрос. Откройте его — вы увидите данные из двух таблиц в виде единой таблицы с выбранными полями.

Статистические расчеты в запросах

Часто возникает необходимость поиска и получения информации по обобщенным данным из таблиц или расчета данных прямо в запросах. Для этих целей используют группировку записей по некоторому полю таблицы и суммирующие функции для группируемых записей. Эти функции перечислены ниже:

- Sum**.....вычисляет значения сумм в группах;
- Avg**.....среднее арифметическое;
- Max**.....максимальное значение;
- Min**.....минимальное значение;
- Count**.....число записей в группе;
- First**.....значение первой записи в группе;
- Last**.....значение последней записи в группе;
- StDev**.....среднеквадратичное отклонение (величина отклонения значений от среднего арифметического);
- Var**.....дисперсия (квадрат значения среднеквадратичного отклонения);
- Группировка**.....указывает, что для поля задана группировка по одинаковым значениям;
- Выражение**.....задает более сложное выражение для групп;
- Условие**.....используется для задания условия отбора записей в группе.

Запрос2:

Необходимо узнать суммы продаж по дням. Для этого создайте запрос из одной таблицы Заказы и отберите в строку Поле 2 поля — ДатаЗаказа и ОбщаяСумма.

Сгруппируйте записи, нажав на **Панели инструментов** кнопку с изображением суммы Σ , а для поля **ОбщаяСумма** параметр **Группировка** замените на функцию **Sum**. В результате при просмотре запроса суммы продаж будут вычислены по датам.

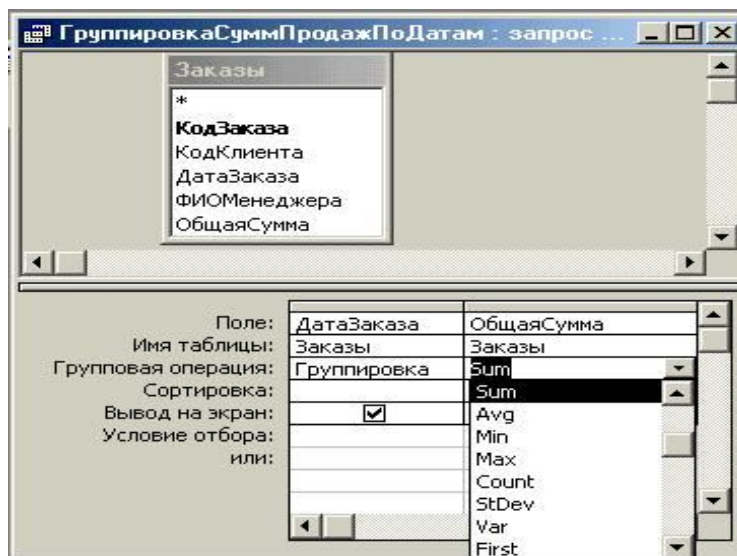


Рис.4. Расчет сумм по датам в запросе

Запросы с параметрами

В качестве значений в условии отбора запроса можно задавать параметры. При каждом исполнении запроса будет затребован ввод конкретного значения параметра и в зависимости от введенного значения результат выборки данных будет различным.

Запрос 3.

Требуется просмотреть заказы в некотором, заранее неизвестном диапазоне дат. Постройте запрос, показанный на рис. 5

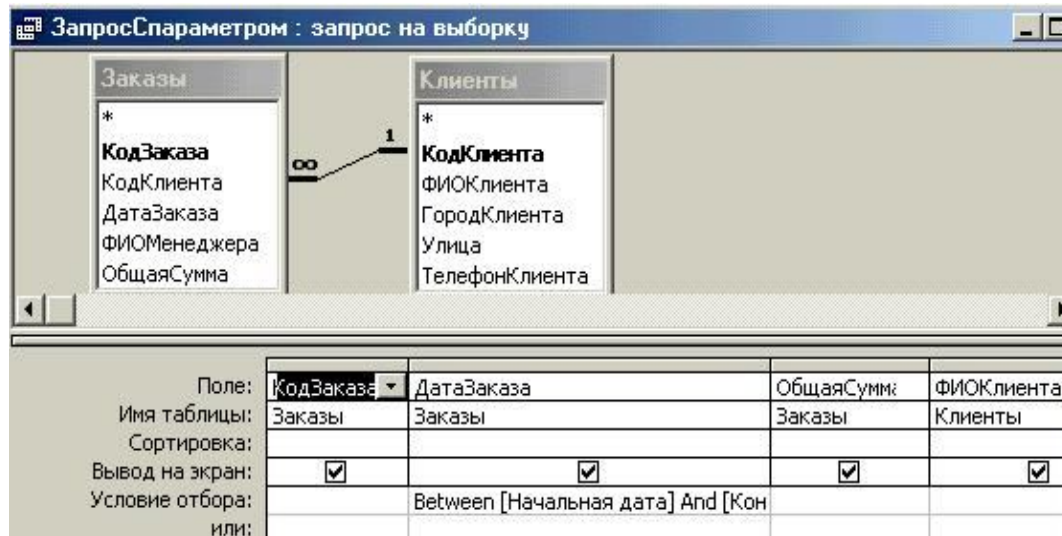


Рис.5. Запрос с двумя параметрами, задающими диапазон дат

В условии отбора поля **ДатаЗаказа** запишите оператор **Between [Начальная дата] And [Конечная дата]**. В переводе это означает: **Между [Начальная дата] и [Конечная дата]**, где [Начальная дата] и [Конечная дата] есть параметры. Поскольку запрос заранее «не знает» их значений, он попросит вас ввести значения этих параметров и выдаст записи в диапазоне введенных дат. Окно ввода значений параметров выглядит как на рис.6.

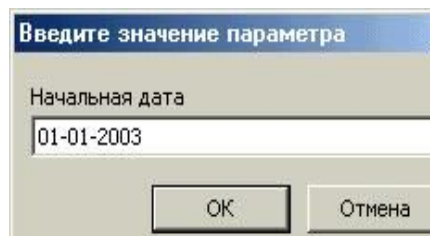


Рис.6.

Рассчитываемые поля

Запрос 4.

Проверить правильность расчета поля **Сумма** в таблице **ЗаказаноТовара**. Постройте запрос, как показано на рис. 7.

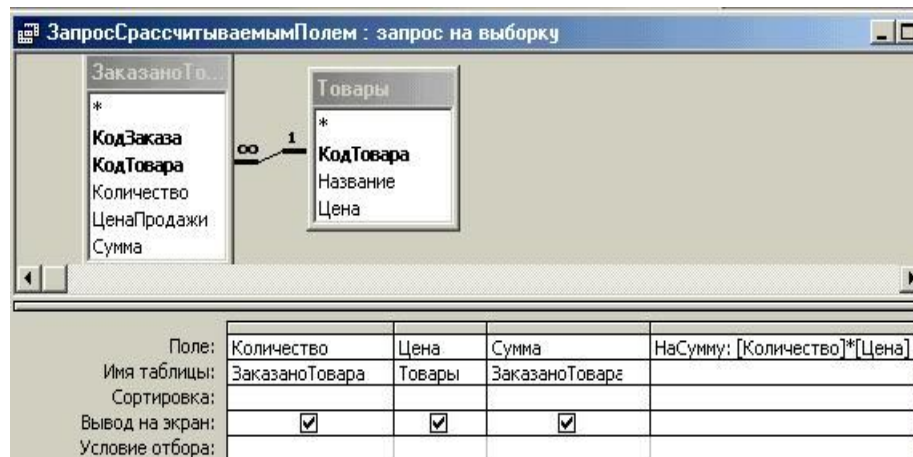


Рис.7. Запрос с рассчитываемым полем **НаСумму**

Здесь рассчитываемое поле называется **НаСумму** и содержит выражение **[Количество]*[Цена]**. При просмотре запроса значения, хранящиеся в поле **Сумма**, должны совпадать со значениями вычисленного поля **НаСумму**. Как видно, поле **Сумма** является избыточным в таблице **ЗаказаноТовара**, поскольку эти значения всегда могут быть рассчитаны с помощью запросов. Отметим, что здесь базовой для запроса является таблица **ЗаказаноТовара**.

Запрос 5.

Рассчитать в запросе суммы продаж по датам (другой способ). Постройте запрос согласно рис. 8.

Здесь записи группируются по дате и по каждой группе вычисляется сумма с помощью функции Sum. Базовой таблицей для запроса является таблица **ЗаказаноТовара**.

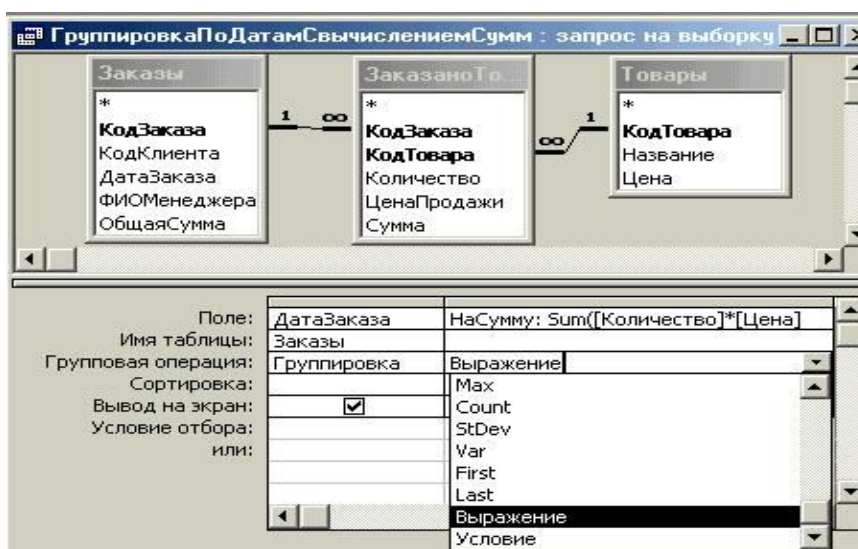


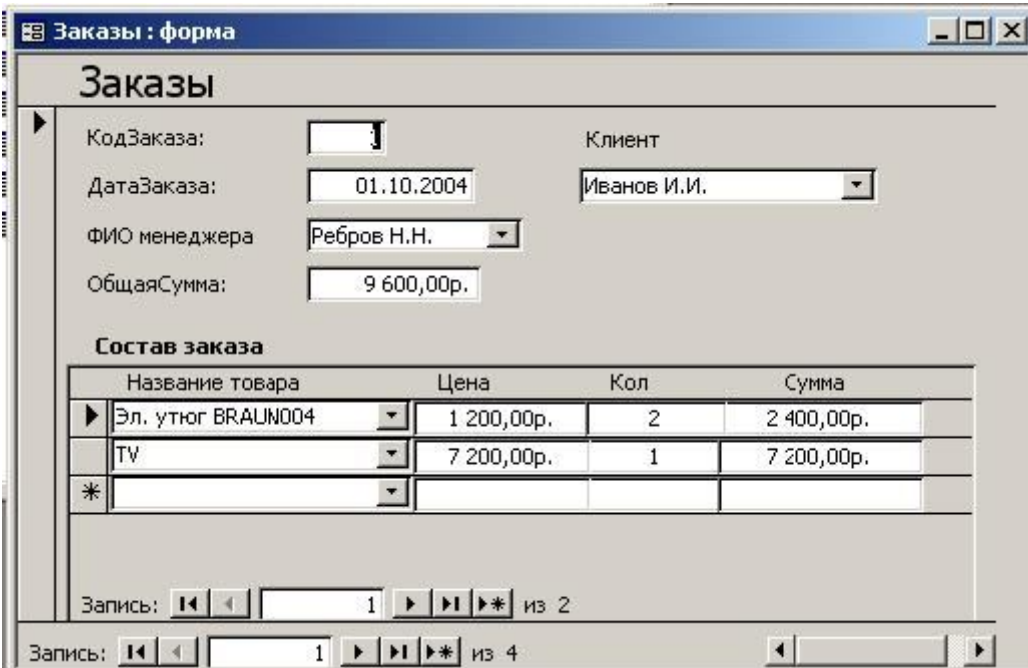
Рис.8. Запрос с группировкой и вычислением сумм и групп

Запрос 6

Рассчитать суммы продаж по клиентам. Для этого в запрос из предыдущего примера добавьте таблицу Клиенты, а в итоговой таблице поле ДатаЗаказа замените на поле ФИО из таблицы Клиенты.

Задание 5. Подчиненные формы. Доработка формы Заказы

Форма заказы должна содержать состав заказа. То есть она должна иметь вид, представленный на рис.12. Поле «Состав заказа» представляет собой так называемую *подчиненную форму*, здесь перечисляются заказанные товары, должна быть возможность ввода/просмотра заказываемых товаров.



Название товара	Цена	Кол	Сумма
Эл. утюг BRAUN004	1 200,00р.	2	2 400,00р.
TV	7 200,00р.	1	7 200,00р.
*			

Рис.12. Вид формы Заказы с составом заказа

Элемент **СоставЗаказа** является не полем, а *подчиненной формой*, или *подформой*. Подчиненная форма - это форма, вложенная в другую форму. Обычно подформу синхронизируют с формой по некоторому полю так, чтобы в подформе отбирались лишь записи, соответствующие текущей записи в форме. Поле для синхронизации в форме именуют *основным полем*, а в подформе — *подчиненным полем*.

В нашем примере основным полем следует выбрать поле **КодЗаказа** формы **Заказы**, а подчиненным - поле **КодЗаказа** подформы. Таким

образом, в подформе будут отбираться строки, соответствующие текущему заказу.

Подчиненную форму с перечнем товаров для формы **Заказы** назовем *ЗаказаноТовараПодформа*. Порядок создания подформы такой же, как и для формы, поскольку подформа есть полноценная форма, но используемая в другой форме.

В качестве источника данных для *ЗаказаноТовараПодформа* следует использовать запрос с именем *ЗаказаноТовараЗапрос*. Его необходимо создать на основе таблиц *ЗаказаноТовара* и *Товары* (рис. 13). Обратите внимание, что поле **КодТовара** выбрано из таблицы *ЗаказаноТовара*, а не из таблицы *Товары*

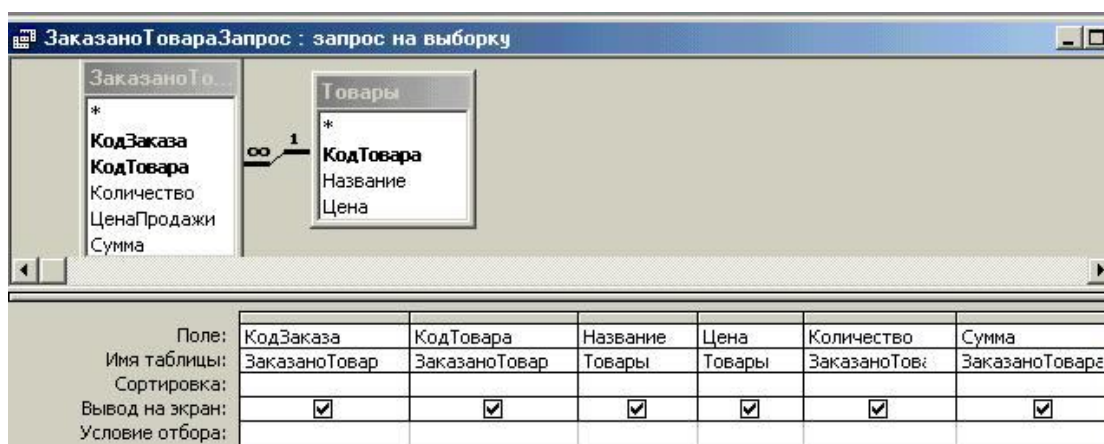


Рис.13. Запрос для подчиненной формы *ЗаказаноТовараПодформа*

Теперь приступим к созданию подчиненной формы *ЗаказаноТовараПодформа*.

1. Создайте форму. В качестве источника данных используйте *ЗаказаноТовараЗапрос*.
2. Сделайте форму ленточной: вызовите окно **Свойства**, откройте в ней закладку **Макет**, в свойстве **Режим по умолчанию** значение *Простая* измените на *Ленточная*. (Рис.14)
3. Разместите поля и надписи, как показано на рис. 14, и закройте форму с сохранением. Отметим, что поля **КодЗаказа** и **КодТовара** переносить на подформу необязательно, но в запрашиваемых данных они присутствуют.
4. Проверьте работу подформы. Вы увидите, что выводится перечень товаров для всех заказов, поскольку не указано поле для синхронизации

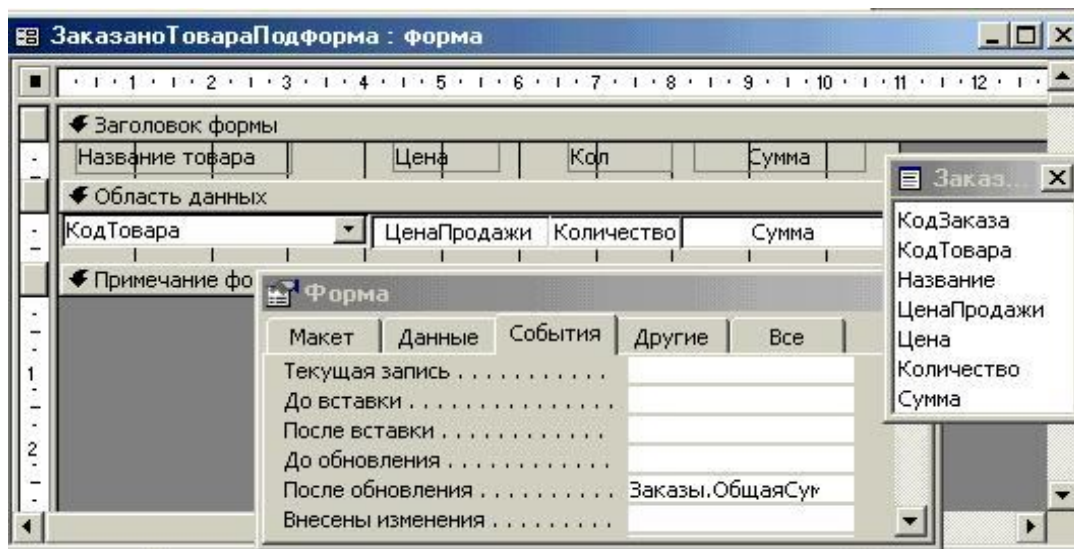


Рис.14. Подформа в режиме Конструктора, список полей запроса и окно Свойства

Внедрите созданную форму в качестве подчиненной в форму Заказы.

Включите **Мастер** на **Панели элементов**, выберите элемент **Подчиненная форма/отчет**, установите указатель мыши на форму (на область данных или примечания) и растяните его до нужных размеров (см. рис.12).

В этом месте будет запущен **Мастер** создания подчиненной формы. В первом окне уточняется источник данных для подформы: выберите пункт **Формы**, выберите из списка *ЗаказаноТовараПодформа*. Следующее окно уточняет имена основных и подчиненных полей: выберите **Выбор из списка**, строку «ЗаказаноТовараЗапрос для каждой записи Заказы». В следующем окне запрашивается надпись к подчиненной форме: введите *Состав заказа* и нажмите **Готово**.

Если Мастер не установлен, то для появившегося на форме элемента **Подчиненная форма/отчет** следует самостоятельно задать уточняющие свойства. В частности, в свойстве **Объект-источник** (закладка **Данные**) выберите из списка имя *ЗаказаноТовараПодформа*, в свойства **Подчиненные поля** и **Основные поля** запишите имя поля для синхронизации — **КодЗаказа**.

Доработка формы Заказы: устранение недостатков

С учетом вышесказанного, перечислим минимальные удобства, которые необходимо обеспечить при работе с формой *Заказы*.

1. Поле *ДатаЗаказа* должно заполняться автоматически текущей датой.
2. Товары должны отбираться из списка товаров (примерно так, как выбираются из списка клиенты).

Реализация этих возможностей требует применения **встроенных функций**.

Access содержит встроенные функции, которые объединены в следующие группы: *текстовые, статистические, управления, сообщения, математические, проверки, общие, работы с базами данных и массивами, DDE/OLE, ввода/вывода, финансовые, преобразования, обработки ошибок, даты/времени*.

Более подробную информацию по любым функциям и другим возможностям Access можно получить в справочной системе Access. Вызов справки — клавиша F1.

Автоматическое заполнение даты

Для автоматического заполнения поля **ДатаЗаказа** при вводе данных пользователем свойству *Значение по умолчанию* поля следует присвоить значение функции **Date()**.

1. Откройте форму *Заказы* в режиме Конструктора и выделите поле **ДатаЗаказа**, щелкнув по нему.
2. Откройте закладку **Данные** окна **Свойства** и в свойство *Значение по умолчанию* поля введите значение **Date()**

Эта функция будет автоматически присваивать полю **ДатаЗаказа** текущую дату для каждого нового заказа. Значение даты функция «читает» из системных часов (таймера) компьютера.

Отбор товаров в подформе с помощью поля со списком

Второй шаг к усовершенствованию формы *Заказы*: замена поля **Название** в подформе *ЗаказаноТовараПодформа* списком. Прделайте следующие действия.

Откройте форму *ЗаказаноТовараПодформа* в режиме Конструктора и удалите поле **Название**. Затем создайте поле со списком на основе поля **КодТовара** таблицы **Товары**. Используйте **Мастер**, аналогично проделанному ранее с полем **КодКлиента**.

Напомним, что следует щелкнуть по элементу **Поле со списком** на **Панели элементов** и сразу же щелкнуть по полю **КодТовара** в **Списке полей**, установить указатель мыши в нужное место и, удерживая левую кнопку, растянуть элемент до требуемых размеров. Обратим внимание, что, отвечая на вопросы **Мастера**, в качестве источника данных для списка надо выбрать таблицу *Товары*, отобразить в список поля **КодТовара** и **Название**, выбрать сохранение выбранного значения в поле **КодТовара**.

Если требуется выводить товары в списке в алфавитном порядке, то для поля Название укажите сортировку по возрастанию. Форма заказы теперь полностью доработана до вида, представленного на рис.12.

Проверьте работу формы Заказы: откройте форму, попытайтесь добавить новый заказ, т.е. введите фамилию менеджера, клиента, а затем отберите товары для заказа

Задание 6. Подключение макросов

Если попытаться заполнить новый заказ, то эта процедура окажется достаточно трудоемкой: поля Сумма, ОбщаяСумма, ДатаЗаказа придется рассчитывать и вводить вручную, некоторые поля, такие как КодТовара, Название, Цена, вообще невозможно заполнить, не имея под рукой таблицу Товары.

И еще один недостаток: изменение цены товара в таблице Товары влечет за собой изменение цены товара во всех заказах, хотя для ранее проданных товаров должна сохраняться прежняя (продажная, до изменения) цена. Этого можно избежать, если добавить в подчиненную форму *ЗаказаноТовара-Подформа* дополнительное поле ЦенаПродажи, которое будет хранить цену на момент выписки заказа. Т.о. необходимы следующие доработки:

Дополнительное поле ЦенаПродажи должно заполняться при вводе данных автоматически (с возможностью изменения вручную при необходимости). Поля Сумма и ОбщаяСумма должны рассчитываться автоматически.

Реализация этих возможностей требует применения **макросов**.

Макросы

Макрос — это объект Access, используемый для автоматизации действий пользователя. Макровывозы выполняются в ответ на события в форме. Например, событию *Нажатие клавиши* кнопки можно «приписать» макрокоманду **ОткрытьФорму**, событию *После обновления* некоторого поля «приписать» макрокоманду **ЗадатьЗначение** другому полю и т.д.


Access содержит более 50 макрокоманд. Перечень имен макрокоманд доступен в окне конструктора макроса — запоминать названия не надо.


Прежде, чем рассмотреть как создать макрос, проведем изменения в таблице *ЗаказаноТовара*. где наряду с количеством и суммой следует хранить цену на момент продажи.

Поле ЦенаПродажи необходимо добавить в таблицу *ЗаказаноТовара*, а не в таблицу *Товары*, так как оно характеризует конкретную строку заказа наряду с полями Количество и Сумма. Следует добавить поле ЦенаПродажи и

в соответствующий запрос: в запрос для подчиненной формы *ЗаказаноТовараПодформа* введите поле *ЦенаПродажи*. А также в форме *ЗаказаноТовараПодформа* удалите поле *Цена*, а вместо него поставьте поле *ЦенаПродажи*.

Создание простого макроса предполагает выполнение нескольких действий:

1. Выберите закладку **Макросы** в главном окне базы данных Access и нажмите кнопку **Создать**.
2. После открытия окна **Конструктора макросов** нажмите кнопку 

Имена макросов и кнопку **Условия**  на **Панели инструментов**, для того чтобы появились соответствующие колонки окна. В этом окне и строятся макросы (рис. 15).

Здесь каждая строка с именем представляет собой отдельный макрос. В первой колонке записывают *имя макроса*, во второй -- *условие* его выполнения (если необходимо), в третьей — *макрокоманду*, которую выбирают из списка, в последней, если требуется, - *примечание*. В нижней части окна задают значения аргументов макрокоманды, каждая макрокоманда имеет свой перечень аргументов.



Рис.15. Создание макроса ЦенаПродажи

На рис. 15 показан макрос с именем *ЦенаПродажи*, который выполняет макрокоманду *ЗадатьЗначение*. Аргументы указывают, что полю *[ЦенаПродажи]* присваивается значение поля *[Цена]* при вводе товара. Отметим, что имена полей заключаются в квадратные скобки.

Отметим также, что в одном окне **Конструктора макросов** можно записать несколько макросов (с разными именами), а затем сохранить их в закладке **Макросы** под общим именем, например, *Заказы*.

Откройте макрос *Заказы* и впишите второй макрос *Сумма*, который выполняет . присваивание $[Сумма]=[ЦенаПродажи]*[Количество]$ при любом обновлении полей *ЦенаПродажи* и *Количество* (рис 16).



Рис.16. Создание макроса Сумма

Таким образом, макросы созданы, но они ниоткуда не вызываются, макросы без событий - вещь бесполезная.

Поэтому теперь необходимо обеспечить вызов созданных макросов в подчиненной форме *ЗаказаноТовараПодформа* с помощью событий: первый макрос должен быть «приписан» событию **После обновления** поля КодТовара; второй — событию **После обновления** полей Количество и ЦенаПродажи. Прделайте следующие действия:

- Откройте форму *ЗаказаноТовараПодформа* в режиме Конструктора, выберите поле со списком КодТовара и откройте для него закладку **События** окна **Свойства**.
- В свойство **После обновления** поля введите имя макроса *Заказы.ЦенаПродажи* (выберите из списка макросов).
- Аналогично для полей Количество и ЦенаПродажи установите событие **После обновления**, но уже на макрос *Заказы.Сумма*.

Закройте форму *ЗаказаноТовараПодформа* и проверьте как все это работает. Откройте форму *Заказы*: при отборе новых товаров или замене товара цена продажи заполняется автоматически, а при вводе/изменении количества и цены продажи -сумма в строке пересчитывается.

Автоматический расчет поля *ОбщаяСумма*: суммирование полей

Нам осталось составить макрос, пересчитывающий поле *ОбщаяСумма* при изменении состава заказа, т.е. при наступлении в подформе *ЗаказаноТовараПодформа* следующих событий:

- * обновления (изменения) суммы в одной из строк или добавления строки — событие **После обновления** подформы;
- * удаления строки — событие **После подтверждения Del** подформы.

Сформируем макрос *ОбщаяСумма*, а затем поясним его (рис. 17).

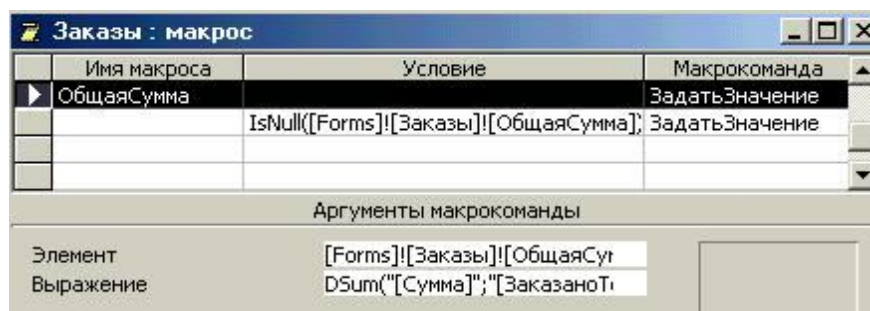


Рис.17. Сложный макрос ОбщаяСумма

Макрос *ОбщаяСумма* состоит из двух макрокоманд *Задать-Значение*. Первая макрокоманда непосредственно рассчитывает общую сумму с помощью функции *DSum*:

```
DSum("[Сумма]";"[ЗаказаноТовара]";"[КодЗаказа] = [Forms]![Заказы]![КодЗаказа] ")
```

Как видно, вычисленное функцией значение присваивается полю *ОбщаяСумма* в форме *Заказы*. При этом используется *полное имя поля* — *[Forms]![Заказы]![ОбщаяСумма]*, поскольку обращение к нему производится из подформы.

Функция *DSum* работает так, что если отсутствуют записи, удовлетворяющие условию в функции *DSum*, то она получает значение *Null*. Поэтому вторая макрокоманда задает значение *[ОбщаяСумма]=0* при условии, что в предыдущей макрокоманде был рассчитан *Null* (рис. 18).

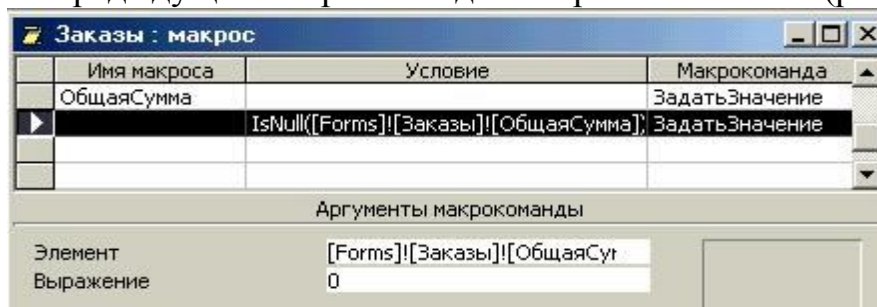


Рис.18. Аргументы второй макрокоманды макроса *ОбщаяСумма*.


Самую большую вычислительную работу в этом макросе выполняет функция *DSum*. Более конкретно, она суммирует поля *Сумма* в таблице *ЗаказаноТовара*, отбирая записи по условию *[КодЗаказа]=[Forms]![Заказы]![КодЗаказа]*, т.е. отбирает только те записи, которые относятся к текущему заказу.

Приведем синтаксис (правило записи) функции *DSum*:

```
DSum("Что суммировать"; "Где суммировать"; "Условие отбора суммируемых записей")
```

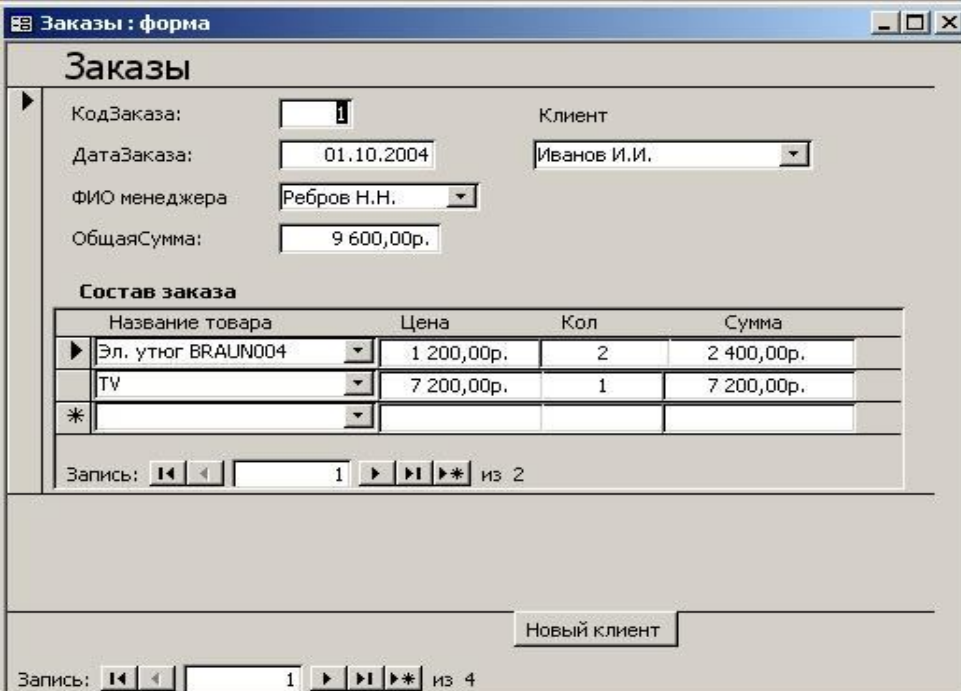
Условие отбора может отсутствовать, тогда суммируются все записи из указанной таблицы запроса.

Теперь обеспечим вызов построенного макроса, т.е. припишем его событиям подформы *ЗаказаноТовараПодформа*.

1. Откройте форму *ЗаказаноТовараПодформа* в режиме Конструктора, и выделите её кнопкой  и затем откройте ее свойства
 2. В закладке **События** найдите свойства **После обновления** и **После подтверждения Del** и присвойте им имя созданного макроса *Заказы.ОбщаяСумма* и закройте форму.
 3. Проверьте работу формы *Заказы*: при изменении строки заказа, добавлении или удалении строки общая сумма будет пересчитываться
- Таким образом, мы ввели самые необходимые усовершенствования для удобной работы с формой *Заказы*

Задание 7. Усовершенствование формы Заказы

Для быстрого ввода новых клиентов и товаров, которых еще нет в базе данных, удобно иметь на форме кнопку **Новый клиент**. Это даст возможность вызывать соответствующую форму и вводить данные о новых клиентах, не выходя из формы *Заказы* (рис. 19).



The screenshot shows a window titled "Заказы : форма" with a form containing the following elements:

- Fields for "КодЗаказа:", "ДатаЗаказа:" (01.10.2004), "ФИО менеджера:" (Ребров Н.Н.), "ОбщаяСумма:" (9 600,00р.), and "Клиент:" (Иванов И.И.).
- A section titled "Состав заказа" containing a table with columns: "Название товара", "Цена", "Кол", and "Сумма".
- The table contains two rows: "Эл. утюг BRAUN004" (Price: 1 200,00р., Qty: 2, Sum: 2 400,00р.) and "TV" (Price: 7 200,00р., Qty: 1, Sum: 7 200,00р.).
- A record navigation bar at the bottom shows "Запись: 1 из 2" and a "Новый клиент" button.

Рис. 19. Усовершенствованный вид формы Заказы

Важно отметить, что, при закрытии формы *Клиенты*, списки клиентов в форме *Заказы* не будут обновляться, если она в этот момент открыта. Поэтому при каждом включении формы *Заказы* следует делать ее обновление: Свойству Включение формы *Заказы* надо присвоить макрос ее обновления. Одновременно этот же макрос

должен закрывать форму *Клиенты*, если пользователь забыл ее закрыть и перешел в форму *Заказы*.

Выполните эти усовершенствования в следующей последовательности:

1. Вначале создайте макросы для этих трех кнопок (рис. 20).

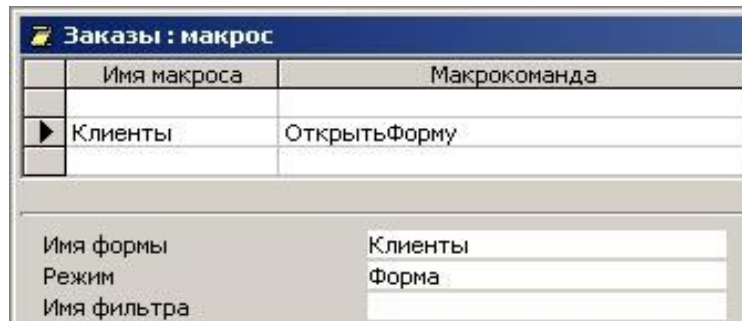


Рис. 20. Макросы вызова для кнопок

Макрос использует макрокоманду **Открыть-Форму**, который открывает форму *Клиенты*.

Вставьте кнопку в раздел **Примечания** формы *Заказы*, введите название кнопок (закладка **Макет** окна **Свойства**, свойство **Подпись**). Далее припишите кнопке (закладка **События** окна **Свойства**, свойство **Нажатие кнопки**) созданный макрос *Заказы.Клиенты*.

Закройте форму и проверьте работу кнопки. Кнопка **Новый Клиент** должна вызывать форму *Клиент*.

Смысл задуманного при вводе нового клиента пока не достигнут: при открытой форме *Заказы* после закрытия формы *Клиенты* новый клиент сразу не попадает в список клиентов формы *Заказы*. Убедитесь в этом, проделав следующую последовательность действий: в форме *Заказы* нажмите кнопку **НовыйКлиент**, в открывшейся форме *Клиенты* введите новую фамилию и закройте форму, в форме *Заказы* откройте список *Клиенты*, чтоб найти эту фамилию — ее нет в списке.

Этот недостаток устраните присвоением свойству **Включение** формы *Заказы* макроса обновления поля со списком, в котором вводится код клиента. Сначала на форме *Заказы* в режиме конструктора откройте окно свойств для этого поля со списком. На вкладке *Другие* измените свойство **Имя** на *КодКлиентаПС*. На рис. 21 показано создание соответствующего макроса *ОбновитьЗаказы*. Он состоит из макрокоманды **Обновление**. Параметром этой макрокоманды является имя поля со списком.

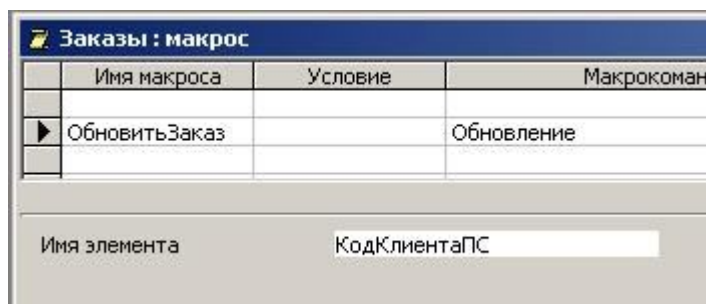


Рис.21. Макрос для события Включение формы Заказы

Проверьте совместную работу формы *Заказы* с формой *Клиенты*.
Создайте макрос, который позволит открывать .формы *Заказы* и *Клиенты* не на первой, а на новой, пустой записи.

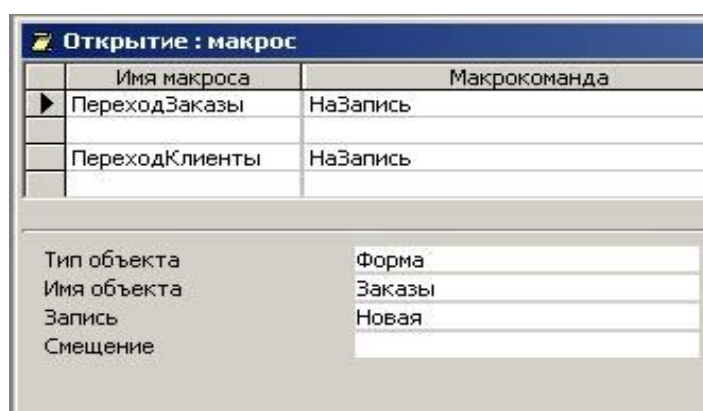


Рис.22. Макросы перехода на новую запись.

На Рис.22. показаны параметры для макрокоманды **НаЗапись** макроса для формы *Заказы*. Аналогично задайте параметры макрокоманды для формы *Клиенты* (изменится Имя объекта на **Клиенты**). Сохраните оба макроса под общим именем *Открытие*. Присвойте свойству **Открытие** первый макрос для формы *Заказы*, второй для формы *Клиенты*.

Задание 8. Фильтрация данных

Удобно подготовить и сохранить для таблицы или формы целый ряд фильтров, которые могут быть загружены в окно фильтра и применены. Это возможно в режиме Расширенного фильтра.

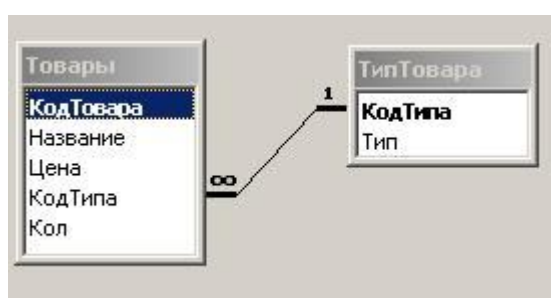
Расширенный фильтр вызывается командой **Записи/ Фильтр/ Расширенный фильтр** или соответствующей кнопкой панели инструментов. Окно расширенного фильтра похоже на окно конструктора запросов. В строку *Поле* перетаскиваются мышью поля списка, по которым задаются условия отбора. В строке *Сортировка* может быть выбран тип сортировки для любого из выбранных полей

Для сохранения подготовленного фильтра выполните команду **Файл/Сохранить как запрос**. Чтобы использовать сохраненный фильтр, можно вызвать его командой **Файл/Загрузить из запроса** также в режиме Расширенного фильтра.

Фильтрация данных с помощью макрокоманд

Для отбора записей в формах, отчетах, таблицах используется макрокоманда **ПрименитьФильтр (AppleFilter)**.

Введите поле КодТипа (длинное целое). Добавьте поле Кол в таблицу Товары – количество товара на складе. Названия типов введите в таблице ТипТовара.

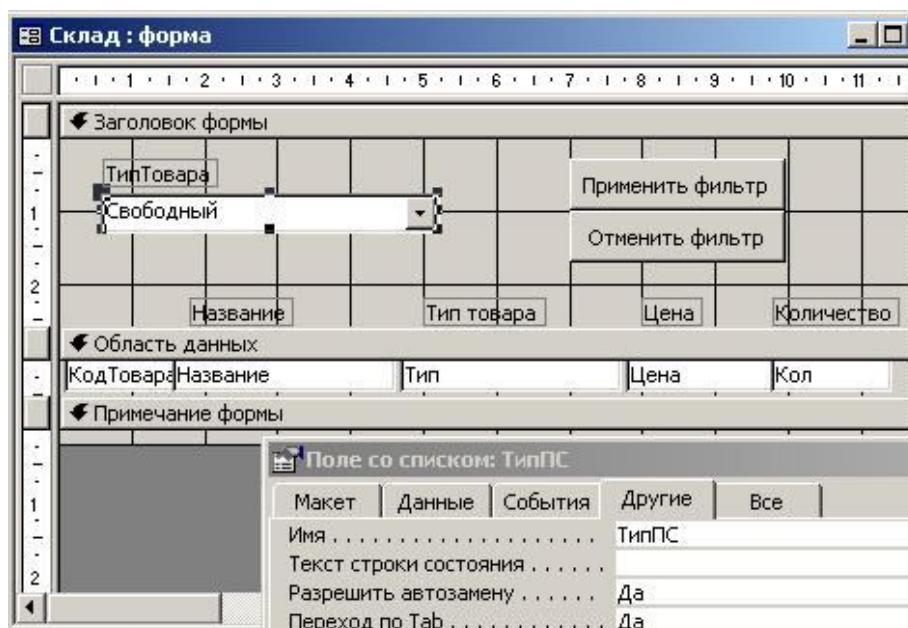


КодТипа	Тип
1	Холодильник
2	Телевизор
3	Пылесос
4	Кофемолка
5	Фен
6	Эл. утюг

Создайте запрос с именем СкладЗапрос на основе таблиц Товары и ТипТовара с полями КодТовара, Название, Цена, Кол из таблицы Товары и поля Тип из таблицы ТипТовара. На основе созданного запроса создайте форму Склад.

Название	Тип товара	Цена	Количество
1 SAMSUNG301	Телевизор	7 200,00р.	10
2 SAMSUNG501	Телевизор	3 500,00р.	5
3 BRAUN004	Эл. утюг	1 300,00р.	6
4 BRAUN003	Эл. утюг	1 100,00р.	17
5 LJ3402	Пылесос	3 400,00р.	27
6 BOSCH-02	Пылесос	2 900,00р.	54
9 Атлант	Холодильник	9 000,00р.	7
10 Самсунг	Холодильник	17 000,00р.	3
11 Indezit	Холодильник	15 000,00р.	7

Поле со списком КодТипа основано на таблице ТипТовара. Оно остается свободным. Переименуйте его: дайте ему имя ТипПС.



Для кнопки **Применить фильтр** с событием **Нажатие кнопки** свяжите макрос **УстановитьФильтр** с одной макрокомандой **Применить фильтр**. Значение аргумента макрокоманды **Условие отбора** задать следующим:

Forms![Склад]![ТипПС] Like [Товары]![КодТипа]

Для кнопки **Показать все записи** создайте макрос **ОтменитьФильтр**:

Имя макроса	Макрокоманда
ОтменитьФильтр	ПоказатьВсеЗаписи
	ЗадатьЗначение
Аргументы макрокоманды	
Элемент	[Forms]![Склад]![ТипПС]
Выражение	Null

Макрокоманда **ПоказатьВсеЗаписи** не имеет аргументов. Вторая макрокоманда очищает поле со списком ТипПС после отмены фильтра.

Задание 9. Разработка отчетов

Создание отчета Заказы

Отчет **Заказы** должен содержать данные конкретного заказа и печататься из формы **Заказы** при нажатии соответствующей кнопки. (рис. 27)

Как видно, отчет **Заказы** фактически повторяет форму **Заказы**, поэтому и содержание действий по его созданию во многом сходно с тем, что мы проделали, создавая форму **Заказы**. Здесь нам также потребуется создать отчет и подчиненный отчет, определив для них запросы, вставить подотчет в отчет.

Последовательно выполните следующие действия.

1. Создайте запрос для отчета **Заказы** с именем **ЗаказыДляОтчетаЗапрос** на основе таблиц **Заказы**, **Клиенты**, **Менеджеры**. Включите в него поля **КодЗаказа**, **ДатаЗаказа**, **ОбщаяСумма** – из таблицы **Заказы**, поле **ФИОМенеджера** – из таблицы **Менеджеры** и поля **ФИОКлиента**, **ГородКлиента**, **Улица**, **ТелефонКлиента** – из таблицы **Клиенты**.
2. Создайте отчет на базе запроса **ЗаказыДляОтчетаЗапрос** - рис. 23.

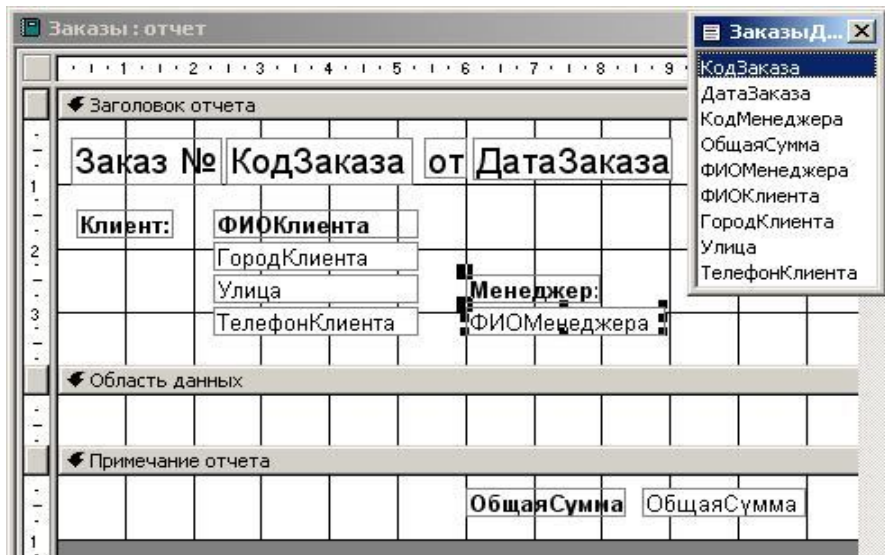


Рис.23. Отчет Заказы в режиме Конструктора

Создайте подчиненный отчет на базе имеющегося запроса **ЗаказаноТовараЗапрос** (он уже создан для формы **ЗаказаноТовараПодформа**) - рис. 24. Присвойте подчиненному отчету имя **ЗаказаноТовараПодотчет**.

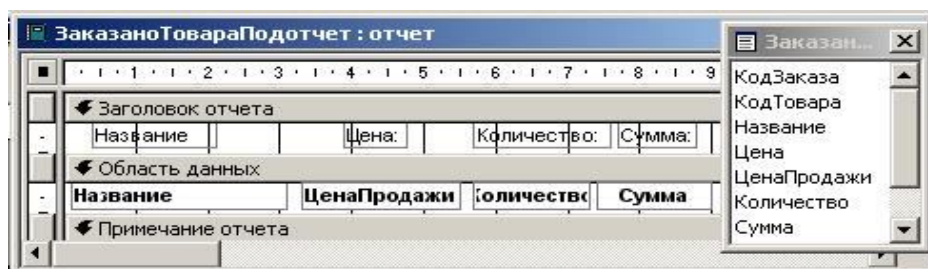


Рис.24.

3. Внедрите подотчет в область данных отчета **Заказы**. В режиме Конструктора включите **Мастер** на Панели элементов, выберите элемент **Подчиненная форма/отчет**, растяните его до нужных размеров в области данных (Рис.25). В первом окне Мастера выберите в качестве источника данных для подотчета из списка отчетов **ЗаказаноТовараПодотчет**. Следующее окно уточняет имена основных и подчиненных полей: выберите **Выбор из списка**, строку «ЗаказаноТовараЗапрос для каждой записи ЗаказыДляОтчетаЗапрос». В следующем окне запрашивается надпись к подчиненному отчету: введите **Состав заказа** и нажмите **Готово**.

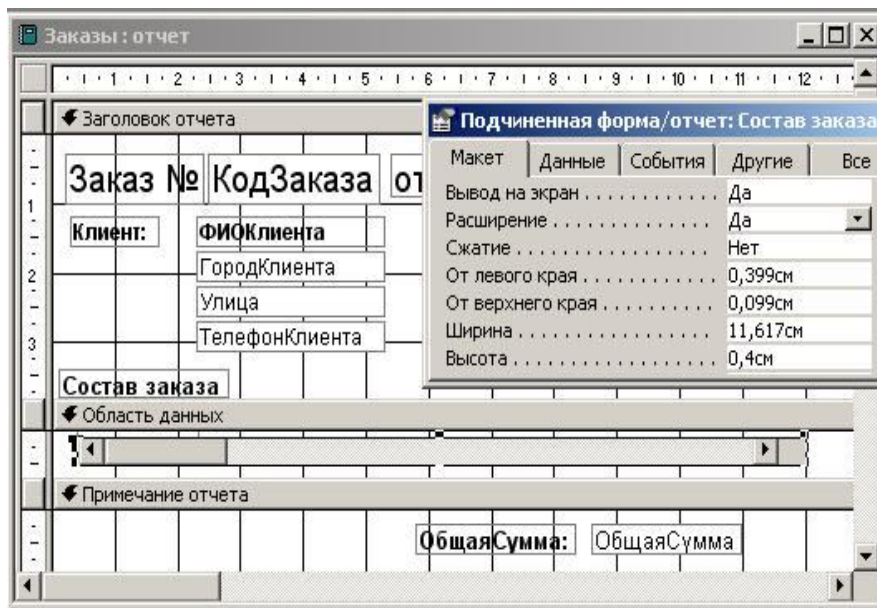


Рис.25

Высоту подотчета сделайте минимальной: 0,3.-0,4 см. Свойству **Расширение** (закладка **Макет**) подотчета присвоено значение **Да**. То есть его размер при необходимости будет увеличиваться, чтобы вместить все данные, содержащиеся в подотчете.

При просмотре отчета можно увидеть, что в перечне товаров присутствуют записи из всех заказов. Условие, отбирающее записи только текущего заказа, присутствует в макросе **ПечатьЗаказа**.

В форме **Заказы** в разделе **Примечания** создайте кнопку с надписью **Печать заказа**. С событием **Нажатие кнопки** свяжите макрос **ПечатьЗаказа** - рис. 26

Условие отбора `[КодЗаказа]=[Forms]![Заказы]![КодЗаказа]` предписывает распечатать текущий заказ.

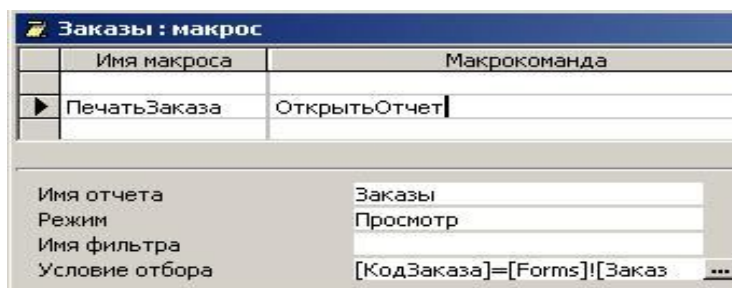


Рис.26

Для того, чтобы иметь возможность просматривать на экране отчет перед выводом на принтер, в макросе установлен режим Просмотр, а не Печать.

При вызове отчета из формы **Заказы** (кнопка **Печать заказа**): в отчет попадут строки только текущего заказа – рис. 27.

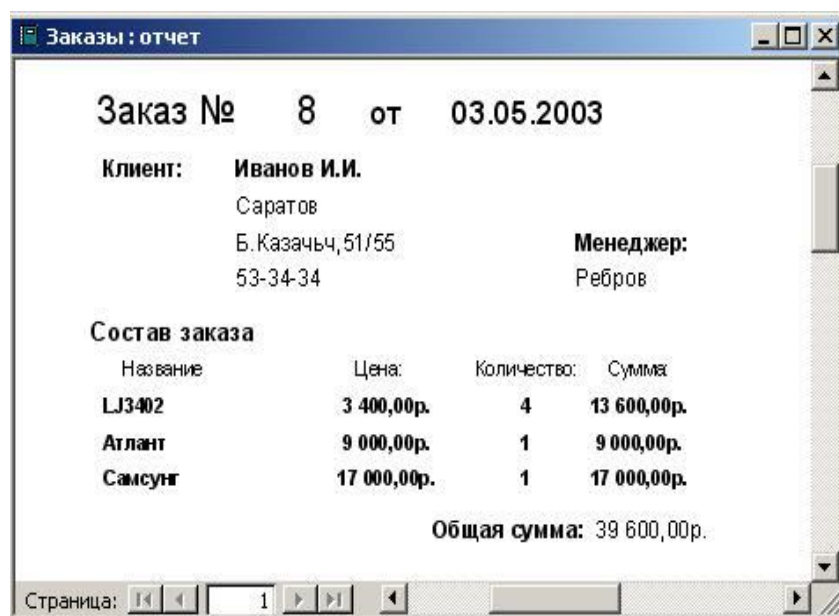


Рис.27. Отчет Заказы в режиме предварительного просмотра на экране

Создание итогового отчета Продажи

В отчете Продажи должен выводиться перечень всех заказов в указанном диапазоне дат с итоговой суммой (рис. 30).

Создайте запрос с параметрами на базе таблиц **Заказы** и **Клиенты**.

В условие отбора поля ДатаЗаказа впишите:

Between [Начальная дата] And [Конечная дата].

Такой запрос отберет заказы в этом диапазоне дат. Сохраните запрос с именем **ДляОтчетаПродажи** и проверьте его работу.

28. Создайте отчет **Продажи** на базе запроса **ДляОтчетаПродажи** - рис.

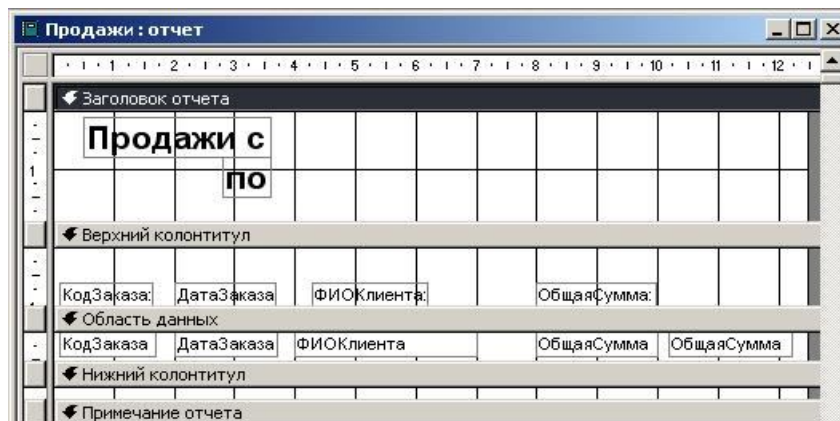


Рис. 28. Отчет Продажи в режиме Конструктора.

Сделаем дальнейшие усовершенствования отчета (рис.30). Значения полей [Начальная дата] и [Конечная дата] выбираются в отчет не из таблиц базы данных, а передаются из запроса при вводе параметров. Поэтому создайте свободные поля вручную с использованием элемента **Поле** Панели элементов. В свойство **Данные** этих полей впишите имена параметров: **Начальная дата** и **Конечная дата**.

Пронумеруйте страницы отчета. Для этого из меню **Вставка** выберите пункт **Номера страниц...** В появившемся окне выберите значения *Страница N из M*, *Верхний колонтитул* и выравнивание *По правому краю*. Соответствующие поля вставятся в отчет, как показано на рис. 29.

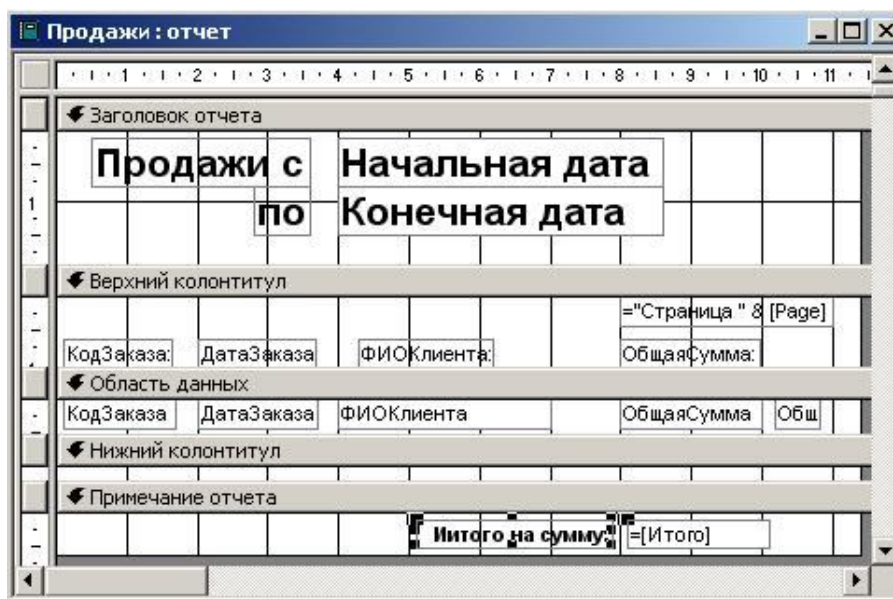


Рис. 29. Доработка отчета Продажи

В отчете поле **ОбщаяСумма** вставлено 2 раза. Крайнее (правое) из этих полей будет использоваться для накопления итоговой суммы, поэтому измените свойства этого правого поля **ОбщаяСумма** так:

- а) в свойство **Имя** (закладка **Другие**) введите имя, например, **Итого**;
- б) в свойстве **Сумма с накоплением** (закладка **Данные**) значение **Отсутствует** замените на значение **Для всего**. Так как в этом поле будет накапливаться сумма продаж;
- в) уменьшите ширину поля до 5... 10 мм и сделайте поле невидимым: свойство **Вывод на экран** - **Нет**.

Для вывода накопленной суммы создадим свободное поле в области примечаний отчета. В свойство **Данные** этого поля запишите =[Итого]. Таким образом, этому полю присвоится значение поля **Итого**, т.е. значение накопленной суммы. Надпись измените на **Итого на сумму**.

Сохраните отчет и просмотрите его (рис. 30).




КодЗаказа:	ДатаЗаказа:	ФИО Клиента:	ОбщаяСумма:
8	03.09.2004	Иванов И.И.	39 600,00р.
9	03.09.2004	Петров П.П.	24 400,00р.
10	24.09.2004	Николаев Н.Н.	77 000,00р.
11	24.09.2004	Иванов Н.С.	31 500,00р.
12	04.10.2004	Иванов Н.С.	22 300,00р.
13	07.10.2004	Иванов И.И.	24 000,00р.
Итого на сумму:			218800

Рис.30. Отчет Продажи в режиме просмотра.

Создание отчета Продажи: группировка

Отчеты позволяют группировать записи при выводе по значению поля или нескольких полей. Внутри группы записи можно сортировать, добавлять заголовок и примечание. Пример группировки по дате заказа приведен на рис. 33.

Группировка записей в отчете устанавливается в специальном окне, которое можно вывести двумя способами: нажать кнопку  на **Панели инструментов** или в меню **Вид** выбрать пункт **Сортировка и группировка**.

Скопируйте отчет **Продажи** в отчет с именем **ПродажиГруппировка**. В режиме Конструктора откройте окно **Сортировка и группировка** (рис.31).

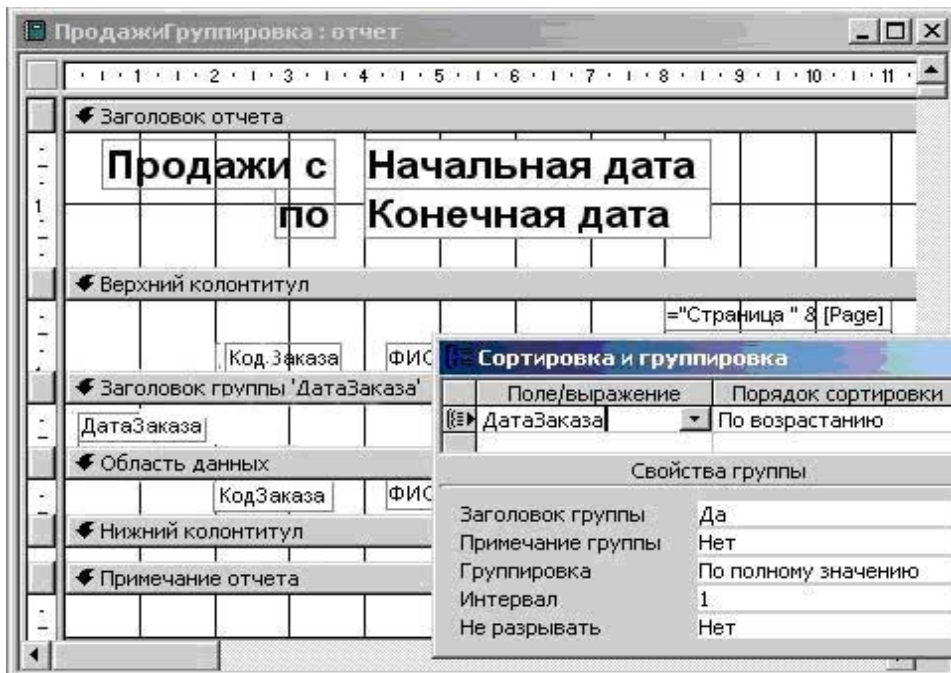


Рис.31. Установка группировки по полю ДатаЗаказа.

Из списка выберите поле **ДатаЗаказа**, укажите порядок сортировки, В параметре **Заголовок группы** (то есть необходимость вывода заголовка группы) укажите **Да**. В появившемся разделе **Заголовке группы** добавьте поле **ДатаЗаказа**. Сохраните отчет и проверьте его работу (рис.32).

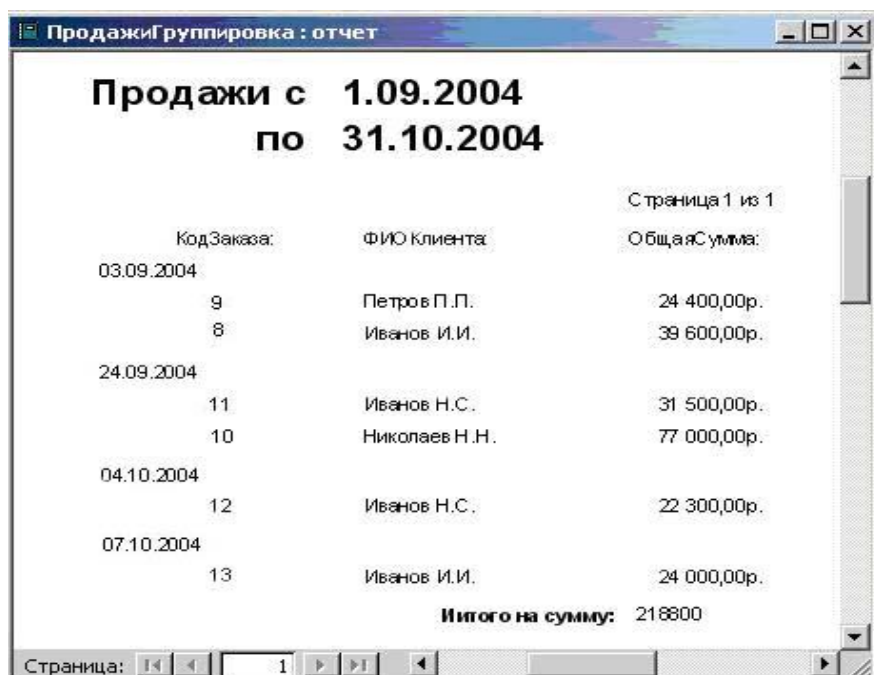


Рис.32 Просмотр отчета с группировкой по дате

Задание 10. Настройка ИС

Для того, чтобы объединить все формы и отчеты в единую систему, используем главную форму, из которой будут вызываться все остальные.

Создание главной формы системы

Наиболее очевидный вид главной формы нашей ИС показан на рис. 33. Как видно, на форме размещено 5 кнопок для вызова форм и отчетов, кнопка выхода из системы. Кроме того, в форме нет полос прокрутки, области выделения и поля номера записи, кнопок управления окном.



Рис.33

Создайте форму (назовите Главная) без источника данных, разместите на ней кнопки и надписи согласно вышеприведенному рисунку. Кнопки создайте с помощью мастера.

Установите свойства формы **Главная** следующим образом:

Подпись Информационная система
«ЗАКАЗЫ»

Полосы прокрутки Отсутствуют

Область выделения Нет

Поле номера записи Нет

Выравнивание по центру Да

Кнопка оконного меню Нет

Кнопка размеров окна Отсутствуют

Кнопка закрытия Нет.

По желанию можно вставить рисунок. Вы можете вставить любой имеющийся у вас рисунок. На **Панели элементов** выберите кнопку **Свободная рамка объекта**, установите и растяните элемент на форме.

В появившемся окне выберите пункт **Создать из файла** и нажмите **Обзор**, чтоб отыскать рисунок среди файлов. Флажок **Связь** в данном случае не устанавливайте, поскольку нет необходимости отображать на форме возможные изменения в файле рисунка.

После вставки рисунка скорректируйте размеры (закладка **Макет**, свойства **Установка размеров, Ширина, Высота**). Напомним, что рисунки могут также храниться в базе данных в полях таблиц типа Поле объекта OLE. Для отображения таких полей используется элемент Присоединенная рамка объекта **Панели элементов**. Это позволяет создавать различные электронные картотеки: фотоальбомы, кадровые системы, каталоги изображений товаров и т.д.

Сохраните форму и проверьте ее работу.

Настройка параметров ИС

Общие настройки среды Access осуществляются в специальном окне **Параметры**, которое вызывается через меню **Сервис**, пункт **Параметры...** Откройте это окно, вы увидите 8... 10 закладок. Например, в закладке **Общие** можно установить рабочий каталог по умолчанию, в закладке **Другие** — порядок блокировки записей, режим открытия базы данных и т.д.

Часть параметров можно инициализировать при запуске ИС в среде Access. Сделать это крайне просто. Выберите в меню **Сервис**, пункт **Параметры запуска...** Открывшееся окно позволяет настроить основные параметры запуска. Сделайте настройки, как показано на рис. 34

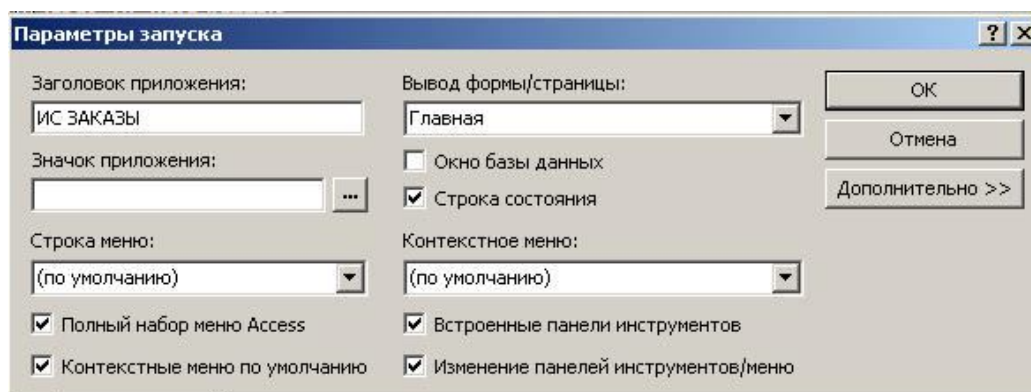


Рис.34

Теперь при запуске ИС автоматически будет открываться форма **Главная**, а окно базы данных Access будет скрыто. Если вы ошиблись и у вас при запуске не открылась ни Главная-Форма, ни окно базы данных, то не отчаивайтесь: меню **Окно**, пункт **Отобразить...** поможет отобразить скрытое окно базы данных, подумайте и исправьте настройки запуска.

Хорошим тоном является создание на рабочем столе ярлыка для запуска ИС в Windows. Для этого щелкните левой кнопкой мыши на рабочем столе Windows, в появившемся меню выберите пункты **Создать**, затем **Ярлык**. В первом появившемся окне найдите и установите путь к файлу приложения, во втором окне введите подпись к ярлыку. Двойной щелчок по ярлыку будет запускать ИС, сам Access будет запущен автоматически.