

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего профес-  
сионального образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

---

В.А.Галанина, Л.А.Решетов

БАЗЫ ДАННЫХ  
Лабораторный практикум

Приведены краткие теоретические сведения, методические указания и справочные материалы, раскрывающие основы проектирования, разработки и использования баз данных в СУБД MS Access(версий 2010 и выше) на практических примерах пошагового выполнения лабораторных работ. Предназначены для студентов, обучающихся по направлениям 09.03.03 09.

## **ВВЕДЕНИЕ**

Данное пособие является методическим обеспечением комплекса лабораторных работ по курсу «Базы данных». В результате выполнения этого комплекса работ студенты получают практические навыки реализации технологий проектирования реляционной модели данных; принципов разработки структуры представления данных в реляционных базах данных; методов и средств обработки данных, хранящихся в базе данных (БД);

Для реализации БД в лабораторных работах используется СУБД Access, входящая в пакет MS Office и являющийся на данный момент самым распространенным пакетом автоматизации работы в офисе. Данная СУБД крайне полезна в малом бизнесе для организации оперативного управления и, кроме того, автоматизации документооборота, так как позволяет исключить дублирование и ускорить обработку документов клиентов и, в конечном итоге, время его обслуживания, и, следовательно, увеличить оборот и доход фирмы при сравнительно небольших затратах на разработку БД.

С учебно-методической точки зрения данная СУБД позволяет эффективно освоить основные принципы работы с реляционными базами данных в первую очередь за счет наличия средств автоматизации создания основных элементов БД (таблиц, форм, отчетов, запросов), что позволяет студентам научиться создавать полноценные базы данных за достаточно короткое время.

### **1. Схема данных в СУБД Access.**

В СУБД Access процесс создания реляционной базы данных включает создание схемы данных. Схема данных наглядно отображает таблицы и связи между ними, а также обеспечивает использование связей при обработке данных.

Основное отличие схемы данных от ER-диаграммы [4] состоит в том, что в схеме данных устанавливаются параметры обеспечения целостности связей в базе данных.

Таким образом, осуществляется неразрывная связь вне машинного проектирования базы данных с этапом ее создания с помощью конкретной СУБД. В схеме данных, построенной по нормализованной модели данных [3] (например, с использованием семантического моделирования с помощью ER-диаграмм), могут быть установлены одно-однозначные и одно-многочленные связи по конкретным полям связываемых таблиц (а не связи между сущностями, как в ER-диаграммах). Для таких связей обеспечивается поддержание целостности взаимосвязанных данных, при которой не допускается наличия в базе данных подчиненной записи без связанной с ней главной, при первоначальной загрузке базы данных и ее корректировках. Связи, определенные в схеме данных, используются автоматически при разработке многотабличных форм, запросов, отчетов, существенно упрощая процесс их конструирования.

При создании в Access схемы данных в ней определяются и запоминаются связи между таблицами. Это позволяет системе автоматически использовать связи, один раз определенные в схеме данных, при создании форм, запросов, отчетов на основе взаимосвязанных таблиц, а пользователь освобождается от необходимости указывать эти связи при конструировании этих объектов. Схема данных базы графически отображается в своем окне, где таблицы представлены списками полей, а связи - линиями между полями разных таблиц.

При создании схемы данных пользователь включает в неё таблицы и устанавливает связи между ними. Для связей типа 1 : 1 и 1 : М можно задать параметр обеспечения связной целостности данных, а также автоматическое каскадное обновление и удаление связанных записей.

*Замечание.* Связь типа 1 : 1 обычно удаляется путем объединения двух таблиц по ключевому полю.

Обеспечение связной целостности данных означает, что Access при корректировке базы данных обеспечивает для связанных таблиц контроль за соблюдением следующих условий:

- В подчиненную таблицу не может быть добавлена запись с несуществующим в главной таблице значением ключа связи;
- В главной таблице нельзя удалить запись, если не удалены связанные с ней записи в подчиненной таблице;
- Изменение значений ключа связи в записи главной таблицы невозможно, если в подчиненной таблице имеются связанные с ней записи.

При попытке пользователя нарушить эти условия в операциях добавления и удаления записей или обновления ключевых данных в связанных таблицах Access выводит соответствующее сообщение и не допускает выполнения операции.

Установление между двумя таблицами связи типа 1 : М или 1 : 1 и задание для нее параметров целостности данных возможно только при следующих условиях:

- Связываемые поля имеют одинаковый тип данных, причем имена полей могут быть различными;
- Обе таблицы сохраняются в одной базе данных Access;
- Главная таблица связывается с подчиненной по первичному простому или составному ключу (уникальному индексу) главной таблицы.

Access автоматически отслеживает целостность связей при добавлении и удалении записей, и изменении значений ключевых полей, если между таблицами в схеме данных установлена связь с параметрами обеспечения целостности. При действиях, нарушающих целостность связей таблиц, выводится сообщение. Access не позволяет установить параметр целостности для связи таблиц, если ранее введенные в таблицы, данные не отвечают требованиям целостности.

Если для выбранной связи обеспечивается поддержание целостности, можно задать режим каскадного обновления связанных полей и режим каскадного удаления связанных записей. В режиме каскадного обновления связанных полей при изменении значения поля связи в записи главной таблицы Access автоматически изменит значения в соответствующем поле в подчиненных записях.

В режиме каскадного удаления связанных записей при удалении записи из главной таблицы будут автоматически удаляться все связанные записи в подчиненных таблицах. При удалении записи из главной таблицы выполняется каскадное удаление подчиненных записей на всех уровнях, если этот режим задан на каждом уровне.

При удалении записей непосредственно в таблице или через форму выводится предупреждение о возможности удаления связанных записей.

## 2. Цикл лабораторных работ с использованием СУБД Access

Для выполнения лабораторных работ используется пакет MS Office 2013, в состав которого входит СУБД Access 2013. При описании лабораторных работ все примеры ориентированы на эту версию Access, но могут быть использованы и для более ранних версий (до Access 2007).

### 2.1. Лабораторная работа 1. Работа с таблицами в Access.

В цикле лабораторных работ используется база данных, которая содержит четыре таблицы: *Клиенты*, *Поставщики*, *Товары*, *Заказы* и *Поставки*.

Таблица *Клиенты*

Поле	Тип поля	Свойство поля	Ключевое поле
КодКлиента	Счетчик		Да
ИмяКлиента	Текстовое		-
Адрес	Текстовое		-
Телефон	Текстовое	Маска ввода	-
Рейтинг	Числовое		-

Таблица *Поставщики*

Поле	Тип поля	Свойство поля	Ключевое поле
КодПоставщика	Счетчик		Да
ИмяПоставщика	Текстовое		-
Адрес	Текстовое		-
Телефон	Текстовое	Маска ввода	-
Рейтинг	Числовое		-

Таблица Товары

Поле	Тип поля	Свойство поля	Ключевое поле
КодТовара	Счетчик		Да
Наименование	Текстовое		-
Описание	Гиперссылка		-
Цена	Денежное		-
Остаток	Числовое		-
СтраховойЗапас	Числовое		-
СтоимостьОстатка	Числовое		-

Таблица Продажи

Поле	Тип поля	Свойство поля	Ключевое поле
КодПродажи	Счетчик		Да
Наименование	Подстановка		-
ИмяКлиента	Подстановка		-
Дата	Дата		-
Количество	Числовое		-

Таблица Поставки

Поле	Тип поля	Свойство поля	Ключевое поле
КодПоставки	Счетчик		Да
Наименование	Подстановка		-
ИмяПоставщика	Подстановка		-
Дата	Дата		-
Количество	Числовое		-

Рис.2.1. Структура таблиц базы данных

## 1. Запуск Access

Для запуска программы Microsoft Access 201X можно использовать любой из стандартных для Windows способов:

- Нажмите кнопку **Пуск**, в меню выберите последовательно пункты **Все программы** ▶ **Microsoft Office** ▶ **Microsoft Office Access 201X**.
- Дважды щелкните кнопкой мыши на ярлыке программы на рабочем столе или в панели быстрого запуска.
- Дважды щелкните кнопкой мыши на файле существующей базы данных.

Сразу после запуска эта база данных будет открыта для работы.

При запуске появляется начальное окно Access (рис. 2.2), которое позволяет быстро перейти к работе с конкретной базой данных или создать новую базу.

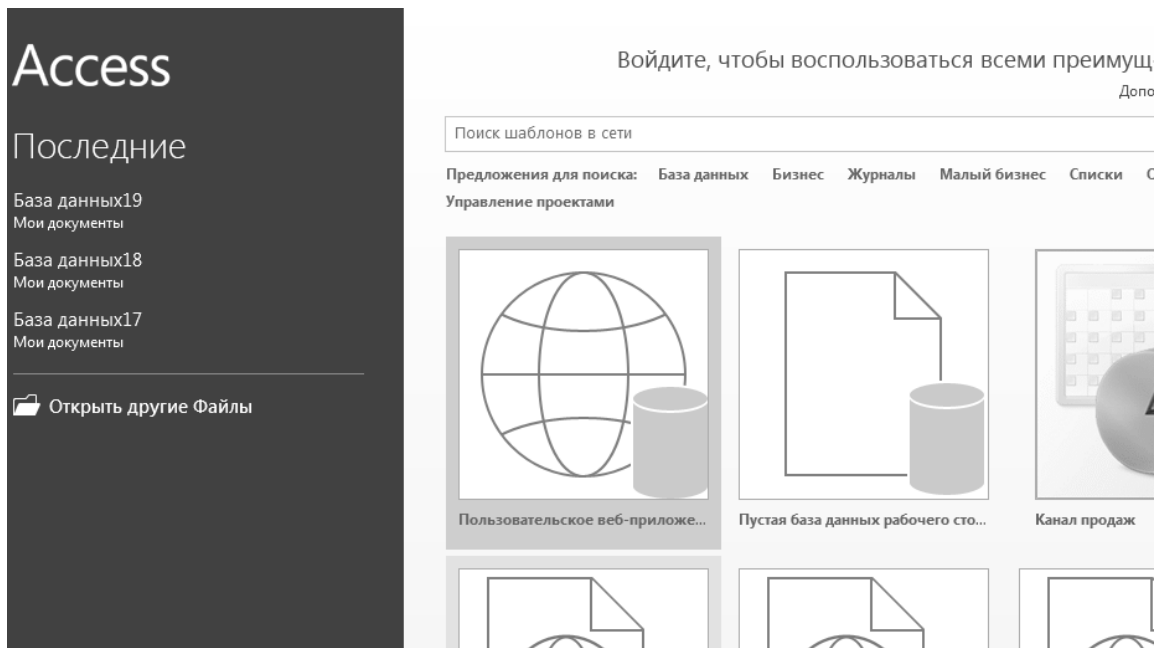


Рис.2.2 Начальное окно Access 13

Теперь необходимо определиться, с какой базой данных вы будете работать. Каждая база данных хранится в файле с расширением ACCDB. В этом файле также хранятся все отчеты, запросы, формы ввода данных и другие документы, созданные для этой базы.

## 2. Создание новой базы данных

Для создания новой базы данных выполните следующие действия.

1. Щелкните кнопкой мыши на значке **Пустая база данных рабочего стола** (см. рис. 2.3)

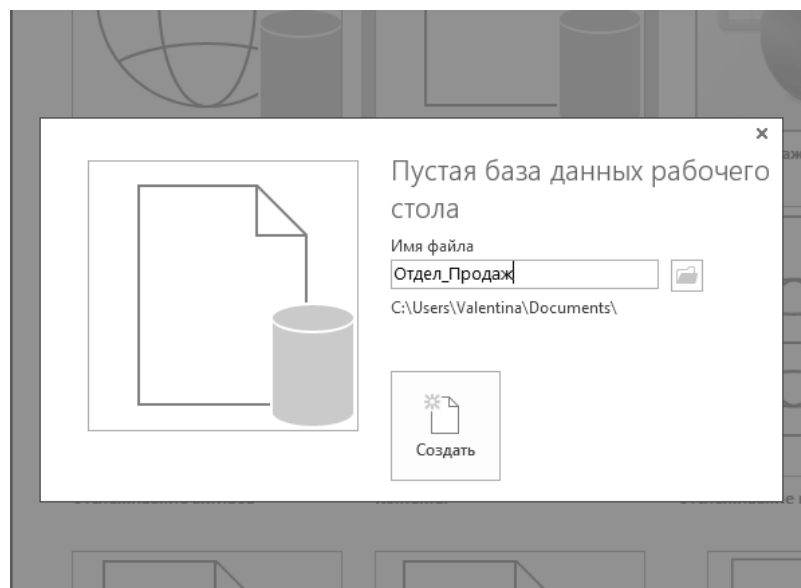


Рис.2.3. Создание новой базы данных

2. В открывшемся окне в поле **Имя файла** введите имя новой базы данных

(например, *Отдел\_Продаж*). Нажмите кнопку для выбора папки, в которой будет храниться файл базы данных (рис.2.3).

3. Нажмите кнопку **Создать**. Откроется новая база данных с единственной пустой таблицей (рис.2.4)

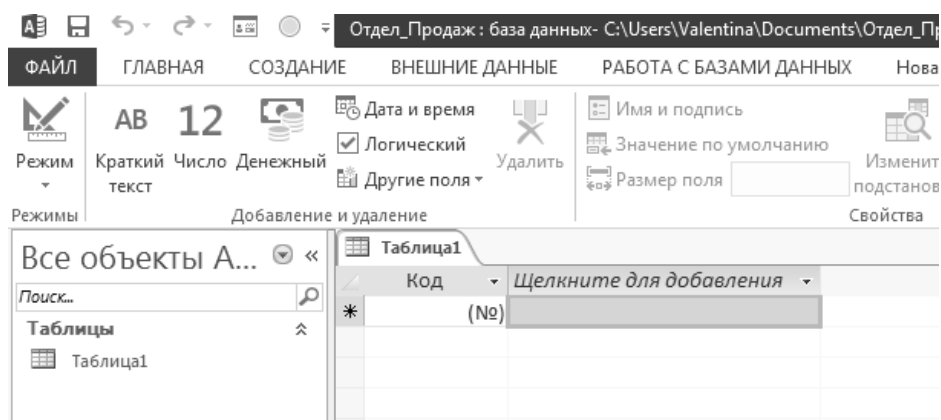


Рис.2.4.Начальное окно создания таблицы

4. Перейдите в режим **Конструктора**. Для этого выделите надпись *Таблица1*, правой клавишей мыши вызовите контекстное меню (рис. 2.5), выберите пункт **Конструктор**.

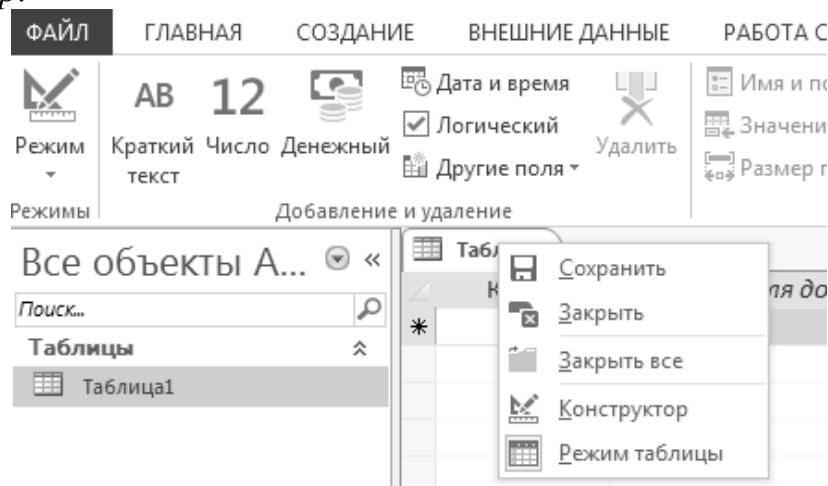


Рис.2.5.Создание таблицы в режиме Конструктора

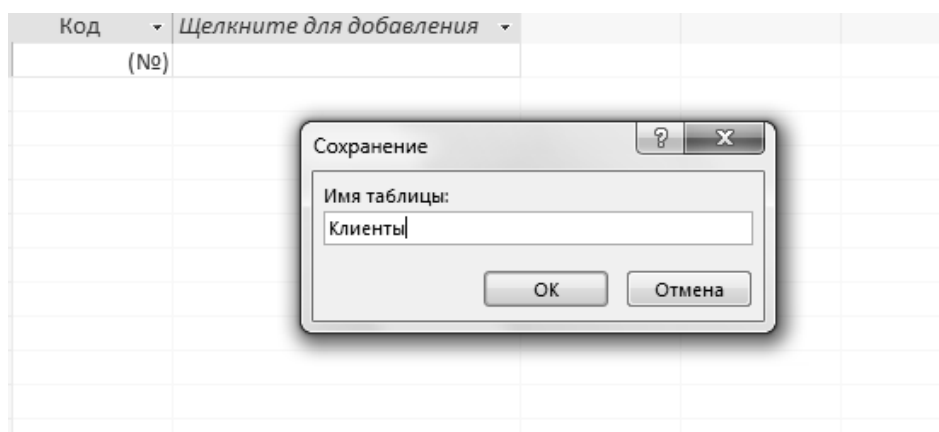



Рис.2.6 Задание имени таблицы



5. В появившемся окне задайте имя первой таблицы (*Клиенты*), нажмите ОК. (рис.2.6).

6. В появившемся окне справа находятся два поля: *Имя поля* и *Тип данных*. Первое поле заполняется автоматически с именем *Код* и типом *Счетчик*. Это поле предлагается использовать в качестве ключевого поля. Измените имя поля на *КодКлиента*. Тип данных оставьте без изменения.

7. Для добавления нового поля установите курсор в первую пустую запись, введите название поля и нажмите клавишу **Enter**. Курсор перейдет в следующую ячейку, и в правой части ячейки появится кнопка . С ее помощью выберите из списка (рис.2.7) тип данных поля (см.Приложение).

В нижней части *Конструктора* отображаются свойства выбранного типа данных.

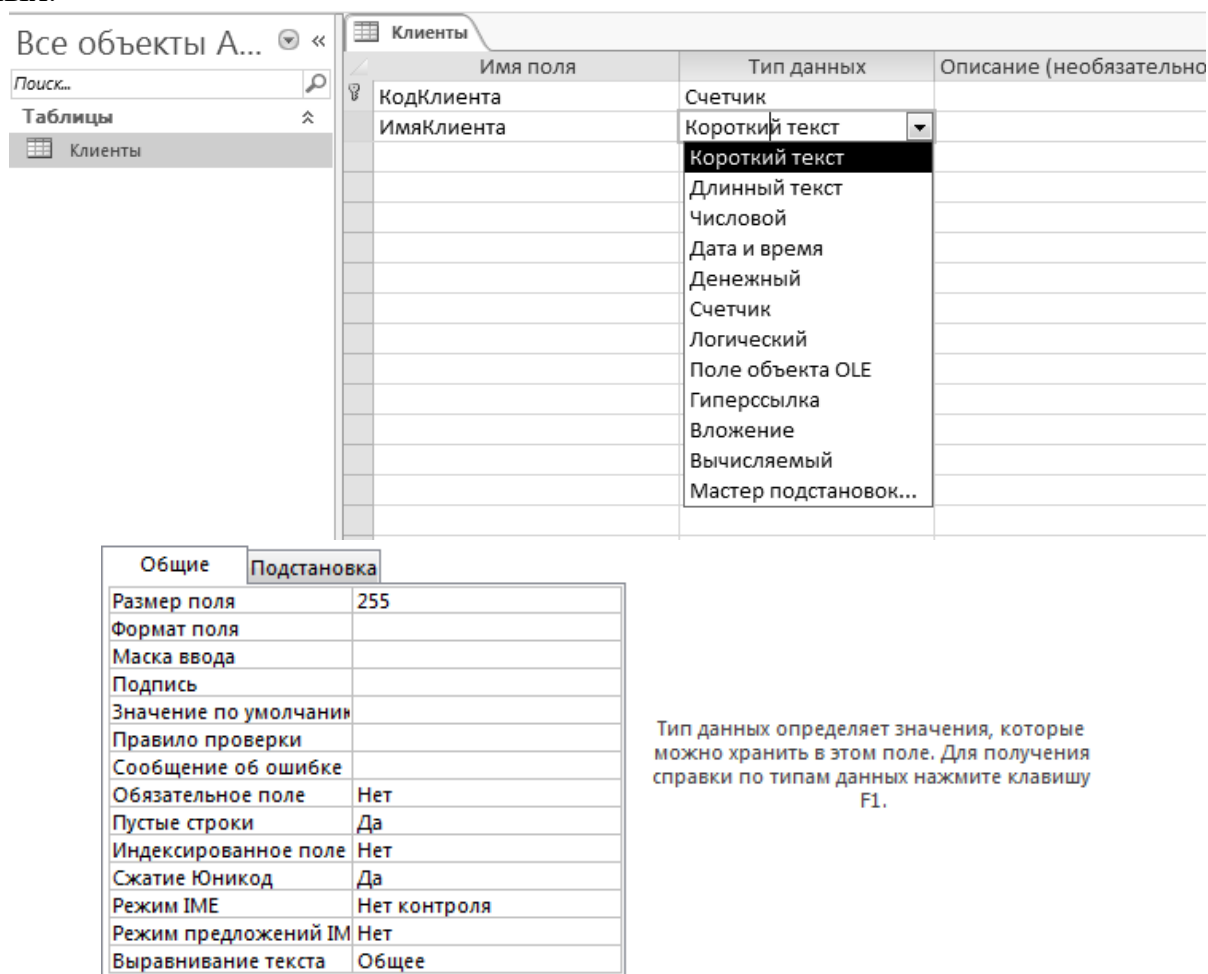


Рис.2.7. Создание полей таблицы *Клиенты*

7.Создайте остальные поля таблицы *Клиенты* в соответствии с рис.2.1.

Для поля *Телефон* используйте маску (999) 000-00-00. Эта маска означает, что в поле необходимо ввести семь цифр номера и до трех цифр кода. При сохранении таблицы программа Access автоматически исправит маску, она будет выглядеть так: \ (999) "000\ -00\ -00.

Если поле вставлено ошибочно, для его удаления нажмите кнопку слева от названия поля, чтобы выделить строку целиком, и нажмите клавишу **Delete**.

На появившейся панели подтвердите удаление, нажав кнопку *Да*. Если поле было удалено ошибочно, сразу после удаления нажмите сочетание клавиш **Ctrl+Z**.

8. Создайте таблицу *Поставщики*. Для этого на вкладке **Конструктор** выберите пункт меню **Создать**. Щелкните на иконке **Конструктор таблиц**.

9. Задайте имя первого поля – *КодПоставщика*. Выберите тип поля – *Счетчик*. Сделайте это поле ключевым. Для этого выделите строку, правой клавишей мыши вызовите контекстное меню (рис. 2.8) и выберите пункт *Ключевое поле*.

10. Создайте остальные поля в таблице *Поставщики* в соответствии с рис.2.1. Сохраните таблицу.

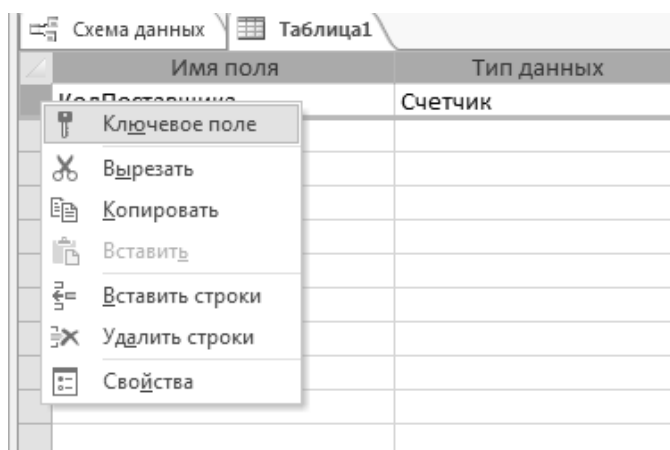


Рис.2.8. Создание ключевого поля в таблице

11. Аналогично создайте таблицу *Товары*.

12. Тип поля *СтоимостьОстатка* выберите *Вычисляемый* (рис.2.8. а)

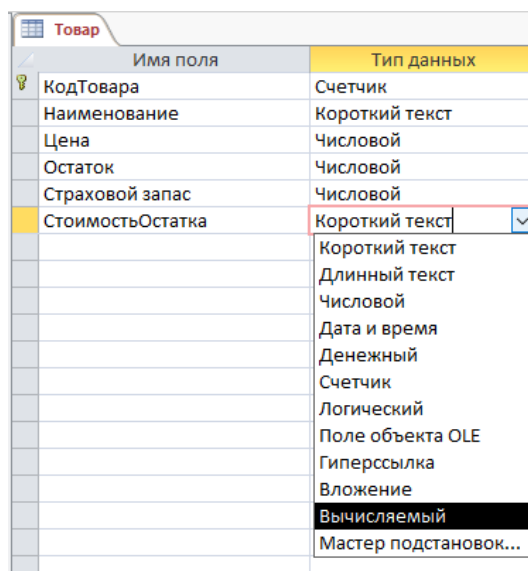


Рис.2.8.а. Создание вычисляемого поля

12. В появившемся окне *Построителя выражений* внесите формулу  $[Цена] * [Остаток]$ , выбрав соответствующие поля двойным щелчком ПКМ в поле *Категория выражений* (рис. 2.8.б). Нажмите ОК.

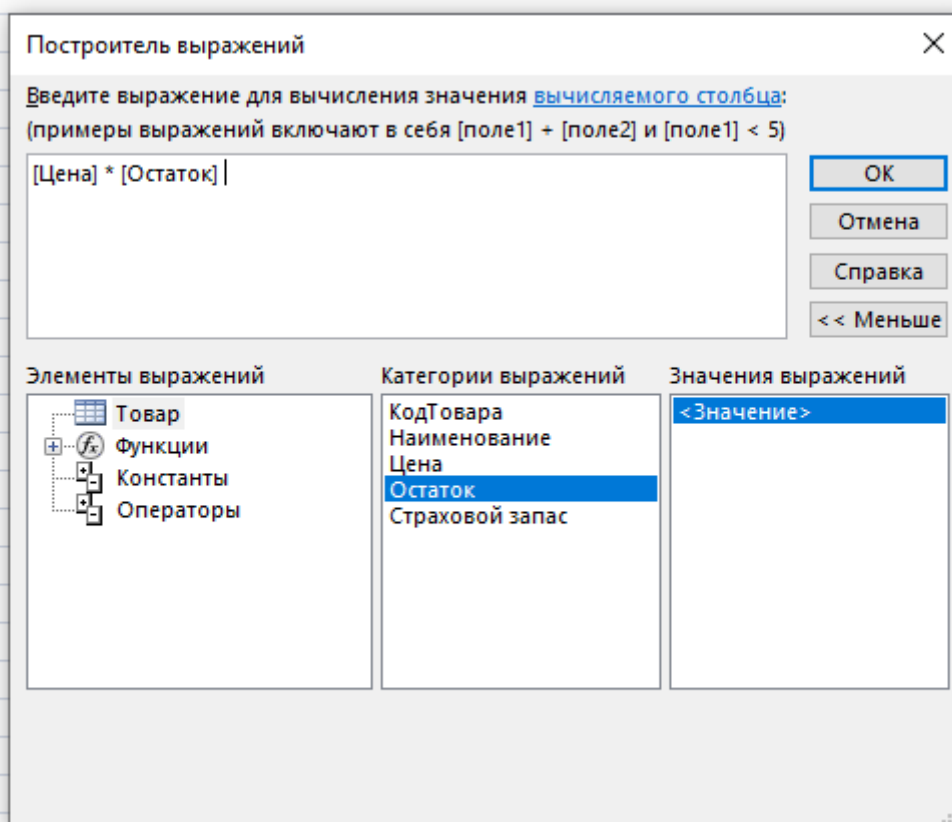
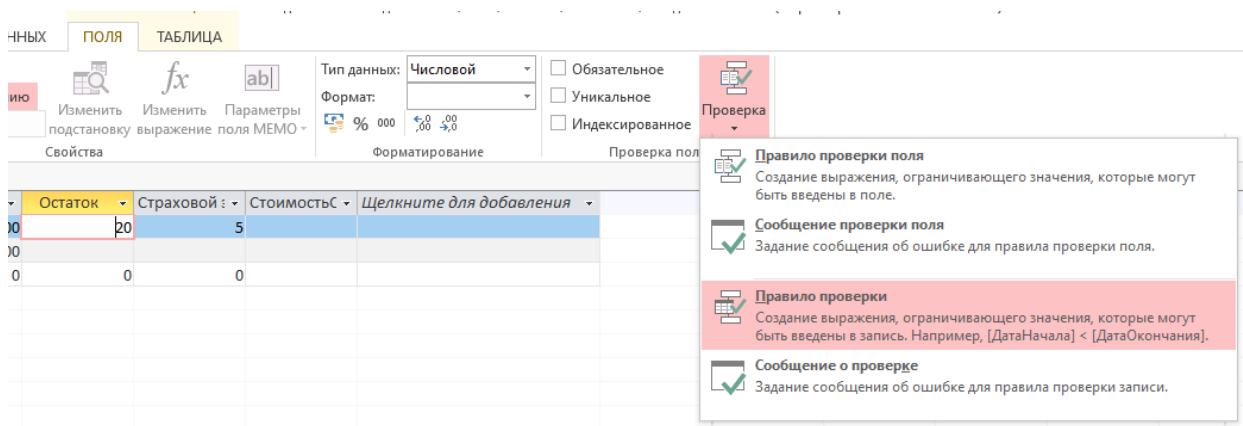


Рис.2.8.б. Создание вычисляемого поля

13. Так как остаток товара не может быть меньше страхового запаса, необходимо задать ограничение на значение этого поля. Для этого: выделите поле *Остаток*, выберите пункт *Проверка* вкладки *Поля*, выберите пункт *Правило проверки*



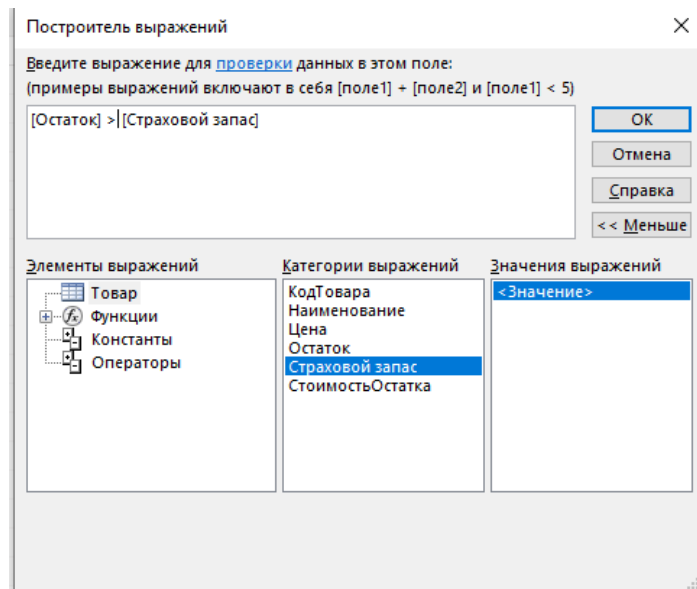
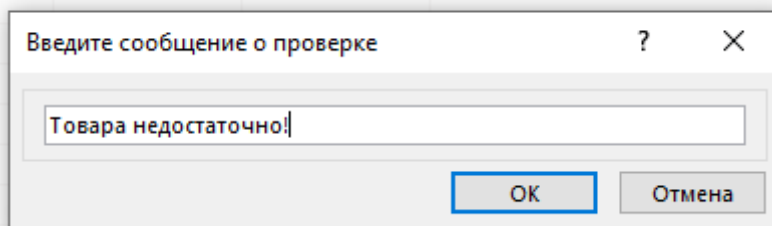


Рис.2.9.а. Задание ограничения на соотношение полей

С помощью *Построителя выражений* задайте требуемое ограничение на соотношение полей *Остаток* и *Страховой запас* (рис.2.9.а). Выберите пункт *Сообщение о проверке* и наберите текст сообщения об ошибке.



*Замечание.* Если требуется задать ограничение на значение одного поля, необходимо выбирать пункт *Правило проверки поля*.

**Самостоятельно!** Задать ограничение на значение поля *Цена* (Например, больше нуля).

14. Заполните созданные таблицы минимум 5-ю записями. Сохраните созданные таблицы.

15. Создайте таблицу *Продажи*.

В таблице *Продажи* значения полей *ИмяКлиента* и *Наименование* берется из таблиц *Клиенты* и *Товары* соответственно. Для этого необходимо использовать в качестве типа поля **Мастер Подстановок** (см.Приложение).

Рассмотрим создание поля этого типа на примере поля *ИмяКлиента*.

Выполните следующие действия:

- В режиме *Конструктора* выберите в списке полей то поле, для которого нужно создать список значений (*ИмяКлиента*). Щелкните кнопкой

мышью в ячейке с названием типа данных поля. Нажмите кнопку и в появившемся списке выберите значение *Мастер подстановок*. На экране появится окно *Создание подстановки* (рис.2.9).

- Установите переключатель в положение *Объект «столбец подстановки» будет использовать значения из таблицы или запроса* и нажмите кнопку *Далее*.

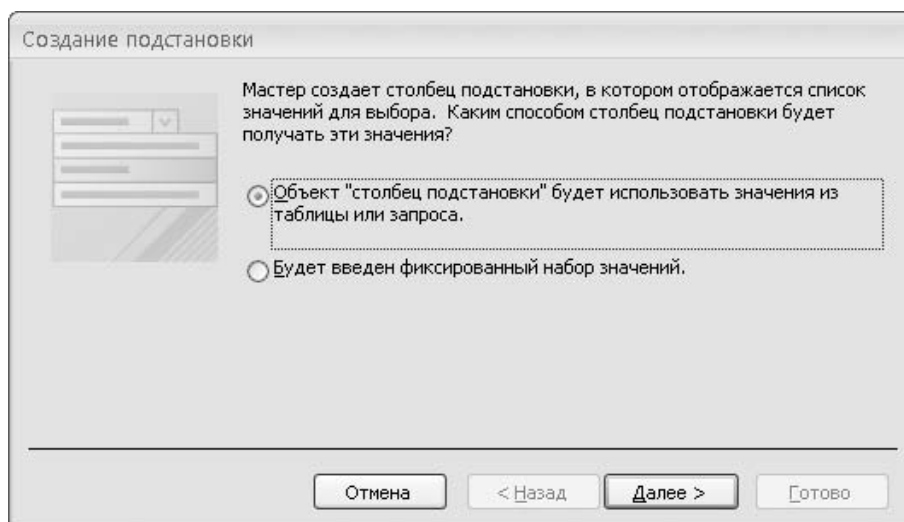




Рис.2.9. Использование Мастера Подстановок

- Установите переключатель *Показать* (рис.2.10) в положение *Таблицы* и щелкните кнопкой мыши на названии нужной таблицы (*Клиенты*). Нажмите кнопку *Далее*.
- Выберите те поля таблицы, значения которых необходимо отображать в подстановке (рис.2.11). В качестве такого поля выбирается поле, содержащее первичный ключ и, обычно, поле, поясняющее значение этого ключа (*КодКлиента* и *ИмяКлиента*). Чтобы добавить поле в подстановку, щелкните кнопкой мыши на названии поля в списке *Доступные поля* и нажмите кнопку.  Таким образом, добавятся все необходимые поля. Для удаления ошибочно добавленного поля щелкните кнопкой мыши на его названии в списке *Выбранные поля* и нажмите кнопку. 

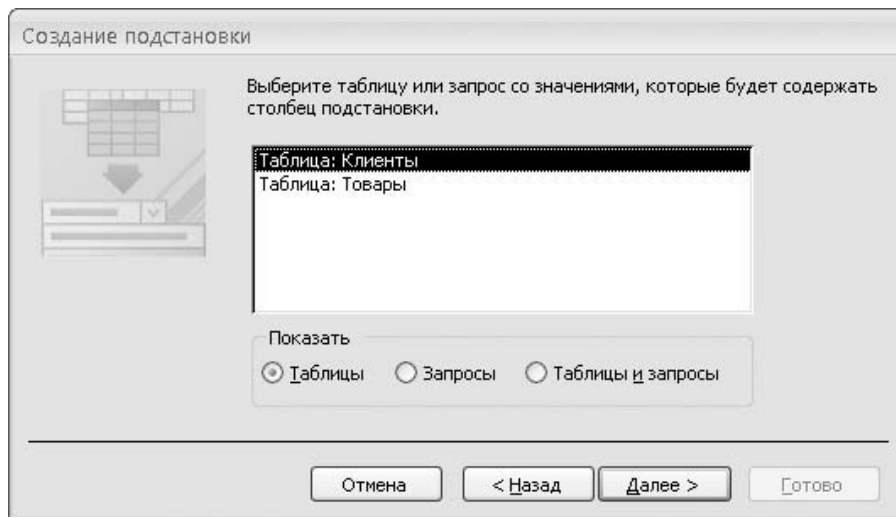


Рис.2.10. Выбор таблицы для подстановки

- После того как набор полей сформирован, нажмите кнопку *Далее*.
- Выберите режим упорядочения значений в подстановке (рис.2.12). По умолчанию записи будут отсортированы по возрастанию значений; если необходимо отсортировать их по убыванию, нажмите кнопку *по возрастанию* (после чего кнопка станет называться *по убыванию*). Нажмите кнопку *Далее*.
- Установите (если он снят) флажок *Скрыть ключевой столбец* (рис.2.13.). Затем подберите ширину столбцов в подстановке, так чтобы все значения отображались полностью. Нажмите кнопку *Далее*.
- В завершение создания подстановки (рис.2.14) при необходимости измените название поля, для которого создается подстановка.
- Отметьте галочкой пункт *Включить проверку целостности данных* и выберите пункт *Каскадное удаление*.

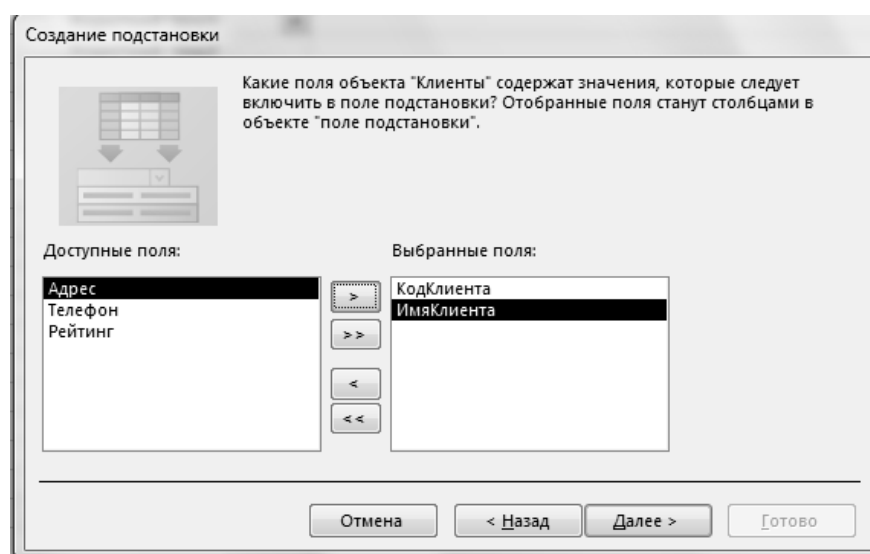


Рис.2.11.Выбор полей для подстановки

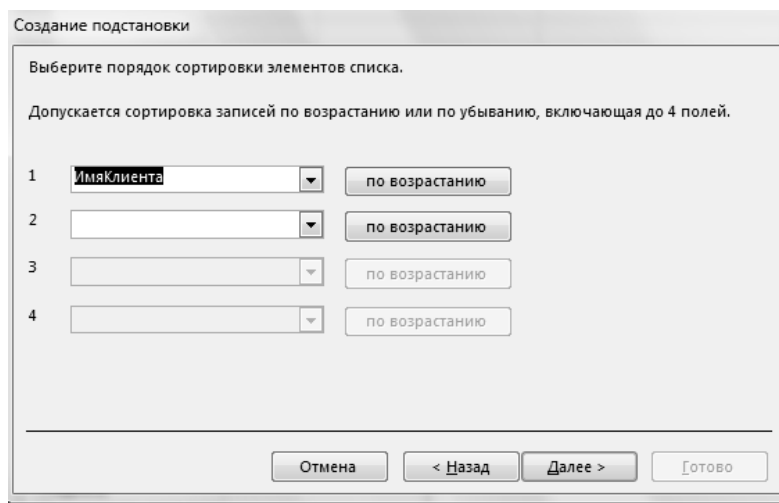


Рис.2.12. Задание режима упорядочения подставляемых записей

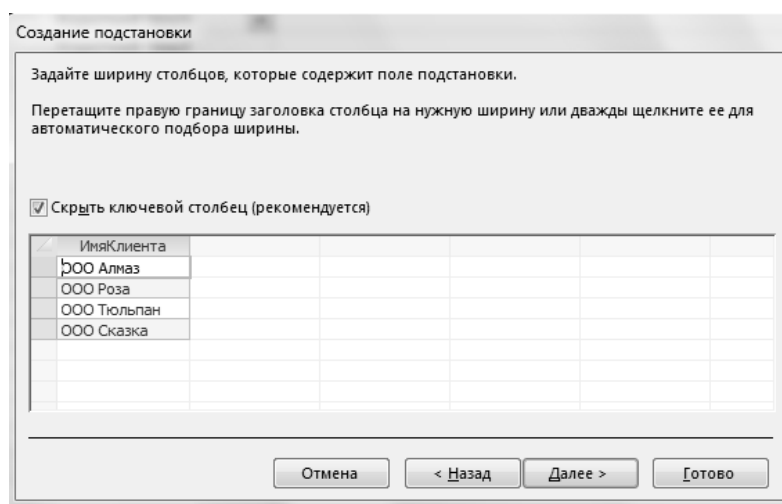


Рис.2.13. Оформление столбца подстановок

- Нажмите кнопку **Готово**.
- Перед созданием связи с таблицей — источником данных для подстановки — программа выведет панель с предложением сохранить текущую таблицу. Нажмите кнопку **Да** и введите название таблицы.

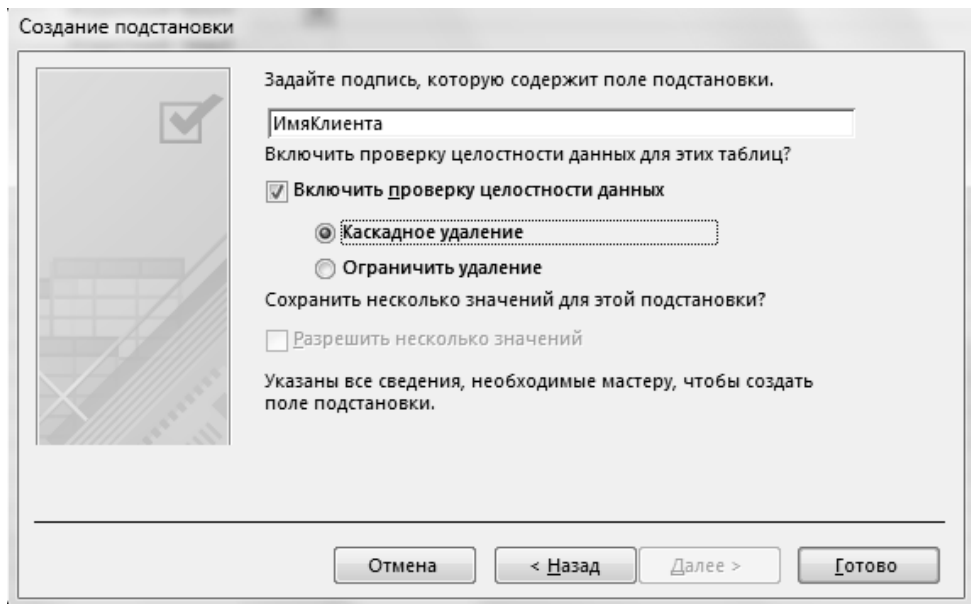


Рис.2.14. Задание подписи для поля подстановки

14. Сохраните изменения, выполненные в режиме конструктора, нажав сочетание клавиш **Ctrl+S**. Закройте таблицу.
15. Аналогично создайте подстановку для поля *Наименование*.
16. Выделите поле *Дата*. В разделе *Свойства поля* в строке *Значение по умолчанию* введите имя функции определения текущей даты *Дата()* (рис.2.14. а). (Имя функции может быть другим в конкретной реализации Access).

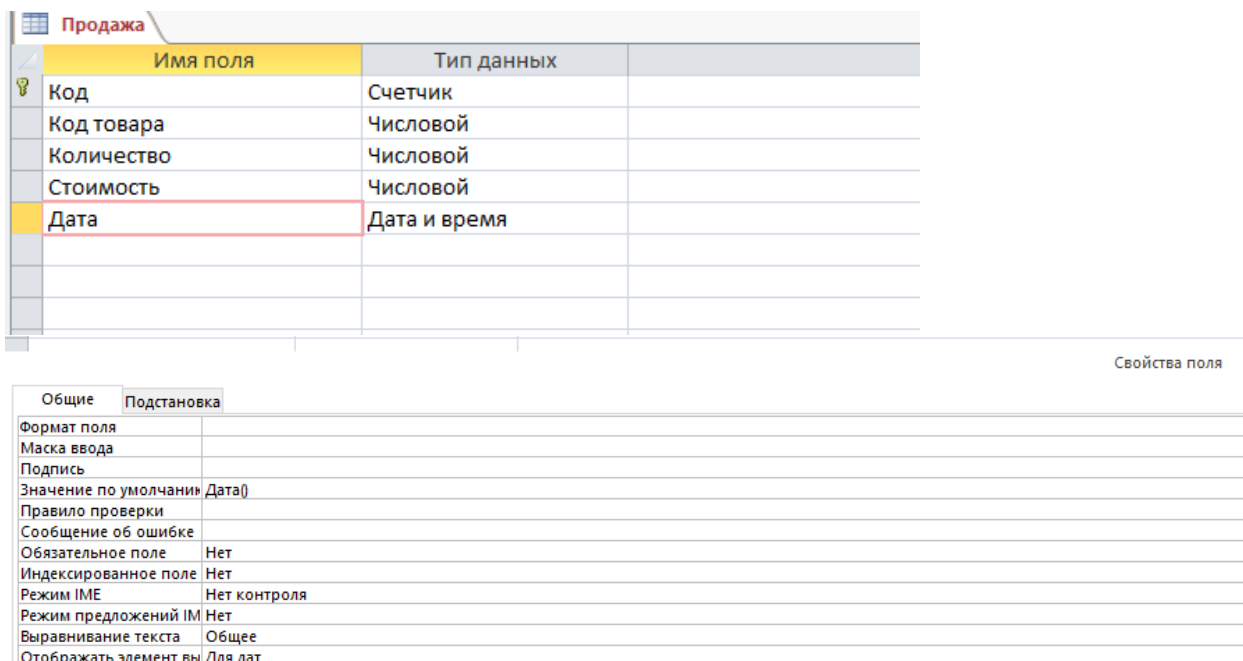


Рис.2.14.а). Задание значения по умолчанию для поля *Дата*



17. Аналогично создайте таблицу *Поставки*. Для полей *ИмяПоставщика* и *Наименование* используйте *Мастер подстановки* из соответствующих таблиц.

18. Заполните таблицы *Поставки* и *Продажи* 5-ю записями.

### **Настройка связей между таблицами**

Прежде чем приступить к настройке связей, следует отметить, что для установки связи между таблицами используются общие поля этих таблиц. Внешний ключ дочерней таблицы содержит значения первичного ключа родительской таблицы, и таким образом запись дочерней таблицы ссылается на одну из записей родительской таблицы.

Просмотр, создание, редактирование и удаление связей осуществляются на схеме данных. Чтобы открыть схему данных, нажмите кнопку *Схема данных* на вкладке *Работа с базами данных* (рис.2.15).

При первом вызове схемы данных, она может иметь другой вид и необходимо будет ее достроить вручную. Для этого выполните следующие действия.

- Перед созданием связи обе связываемые таблицы должны отображаться на схеме данных. Если таблица отсутствует на схеме данных, ее требуется вначале добавить. Для этого нажмите кнопку *Отобразить таблицу* на вкладке *Конструктор*. Затем в появившемся окне *Добавление таблицы* щелкните кнопкой мыши на названии таблицы и нажмите кнопку *Добавить*. После добавления всех нужных таблиц нажмите кнопку *Заккрыть*.

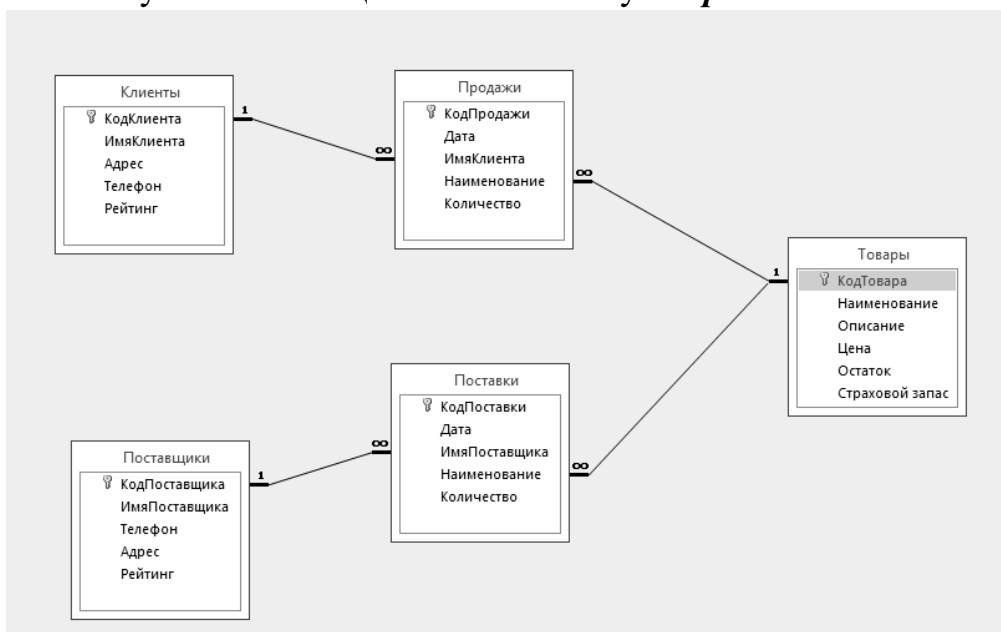


Рис.2.15. Схема данных

- Если связь между таблицами не отображается автоматически, щелкните кнопкой мыши на названии поля, которое является первичным ключом в

родительской таблице. Удерживая нажатой кнопку мыши, перетащите поле первичного ключа, наведите указатель на название того поля дочерней таблицы, которое должно стать внешним ключом (при этом в изображении указателя появится знак +), и отпустите кнопку мыши.

- Если связь между таблицами отображается, но нет обозначений «1-∞», необходимо подвести курсор к этой связи, правой клавишей мыши вызвать контекстное меню и выбрать пункт **Изменить связь**.
- На экране появится окно **Изменение связей** (рис.2.16). В этом окне указаны названия связываемых полей. Убедитесь, что поля для создания связи выбраны верно. В случае обнаружения ошибки выберите другие поля.
- Для обеспечения целостности данных необходимо установить флажки **Обеспечение целостности данных, каскадное удаление связанных записей и каскадное удаление связанных записей**.

Сохраните изменения схемы данных, нажав комбинацию клавиш **Ctrl+S**.

**Замечание.** Если после создания всех таблиц и связей вам потребуется изменить что-либо в вашей БД, то нужно иметь в виду, что сделать это можно будет только *после удаления всех связей*, в которых участвует редактируемая таблица.

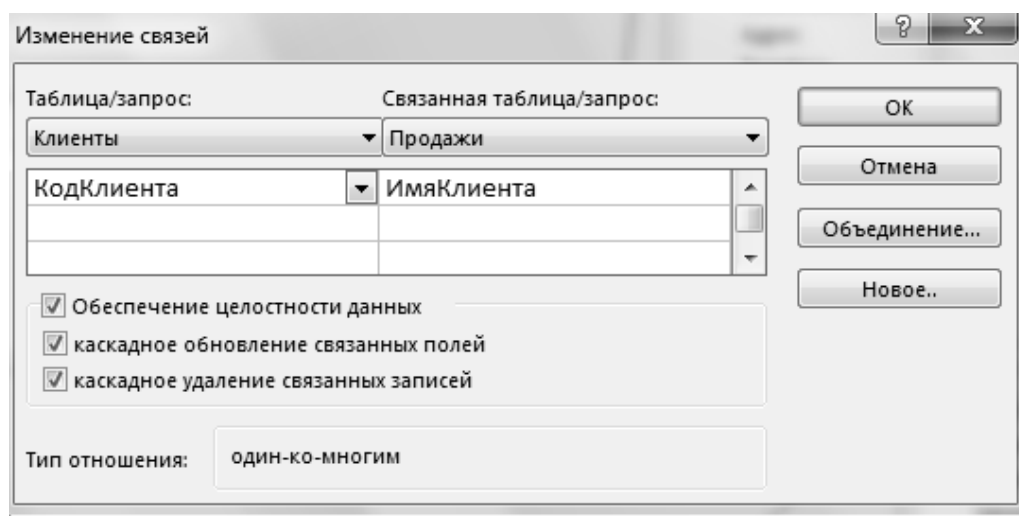


Рис.2.16. Задание режима обеспечения целостности БД

## 2.2. Лабораторная работа 2. Работа с запросами

### *Создание простого запроса (запрос на выборку)*

К данным таблиц можно обратиться, затем извлечь их, выполнить какие-либо вычисления — все это осуществляется с помощью запроса на выборку. Таким способом также можно получить любую информацию о данных, выполнить фильтрацию данных, внести данные в форму или отчет, осуществить автоматизацию управления данными.

Существует четыре пути создания запросов:

- с помощью мастера запросов;
- с использованием конструктора запросов;
- в режиме SQL-редактора;
- на основе существующего фильтра.

### *Создание запроса с помощью конструктора*

1. Откройте базу данных *Отдел\_Продаж*.
2. Перейдите на вкладку *Создание*.
3. Нажмите кнопку **Конструктор запросов** в группе *Другие*. Откроется окно **Добавление таблицы**. В появившемся окне есть три вкладки: **Таблицы**, **Запросы**, **Таблицы и запросы**. На вкладке **Таблицы** выберите таблицы *Товары*, *Клиенты*, *Продажи* с помощью кнопки **Добавить** (или двукратным нажатием правой клавишей мыши на нужной таблице). Затем нажмите кнопку **Закреть**, после чего на экране появится пустой бланк запроса на основании выбранных таблиц (рис.2.43).
4. Открывшееся окно конструктора запросов состоит из двух частей: верхняя содержит списки полей выбранных таблиц, а нижняя – бланк QBE для создания запроса. Каждый столбец бланка описывает одно поле, участвующее в запросе. Включение поля в запрос производится перетаскиванием его из списка полей таблиц при помощи мышки (или двукратным нажатием правой клавишей мыши) в нужный столбец бланка QBE.
5. Перетащите в бланк QBE поля из соответствующих таблиц, как показано на рис.2.44. Установка связей между таблицами производится автоматически, используя структуру связей, созданную при генерации проекта БД.
6. Условие для отбора нужных нам полей производится включением этого условия для заданного поля в строке QBE "*Условие отбора*". Несколько значений отбора вводятся в одну строку, разделенные логическими условиями AND или OR, либо вводятся в последующие ячейки строки "или". Установите для таблицы "*Клиенты*" условие отбора клиентов, например, *ООО Алмаз*, для чего в столбец описания поля внесите строку "*ООО Алмаз*" (рис.2.45).

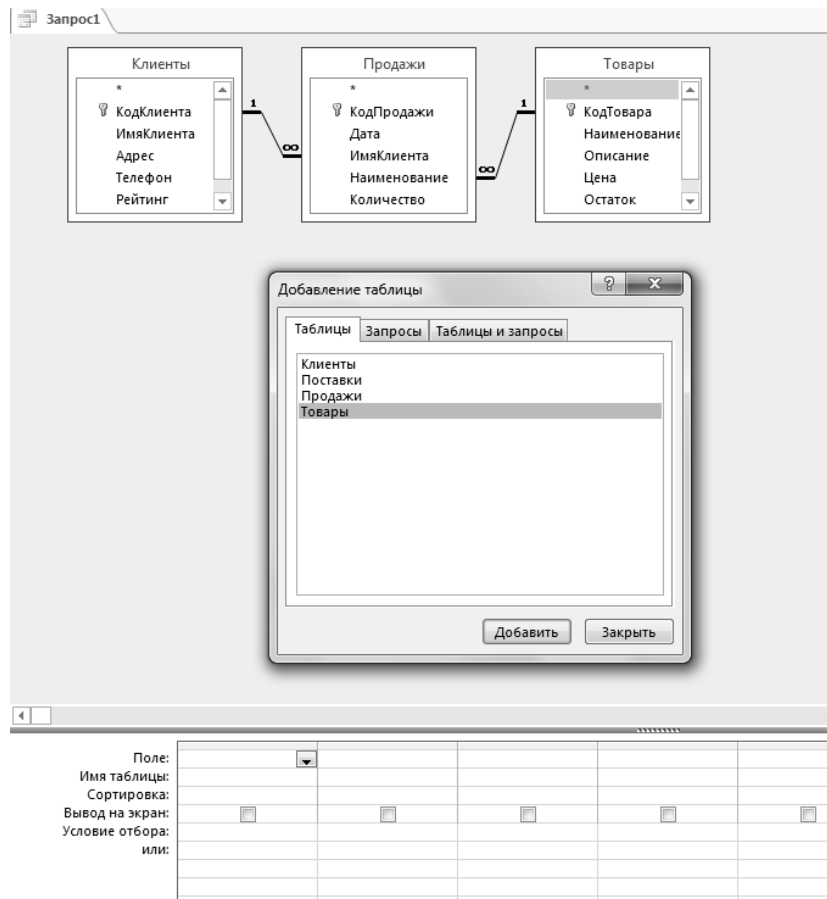


Рис.2.43. Добавление таблиц в запрос в режиме конструктора

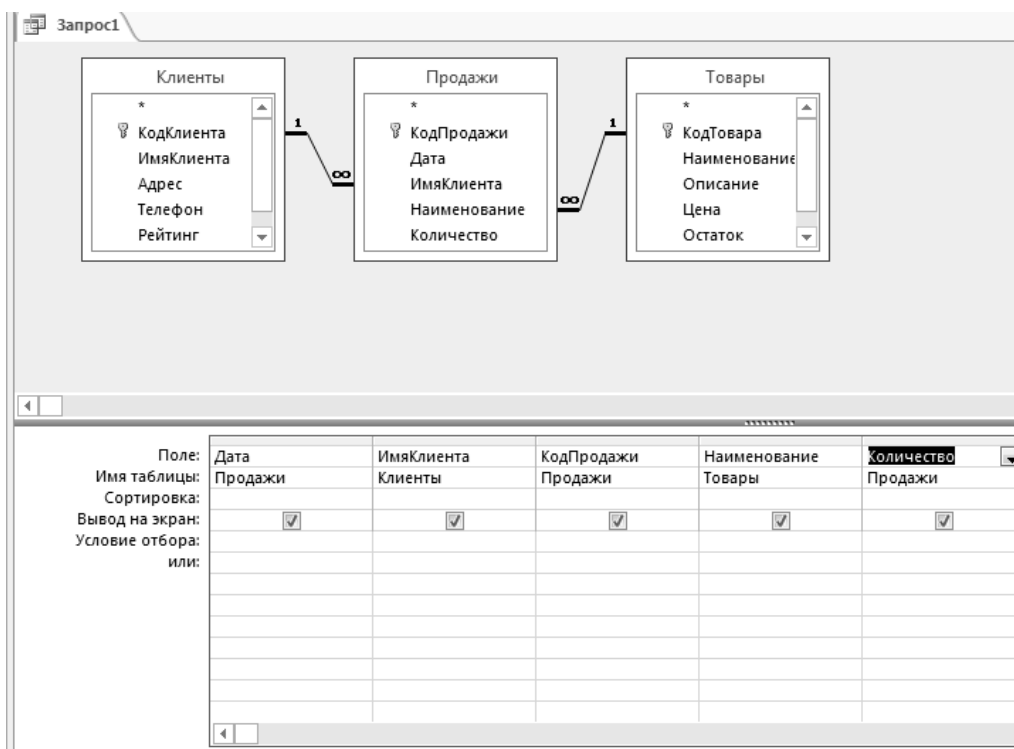


Рис. 2.44. Бланк запроса в конструкторе запросов

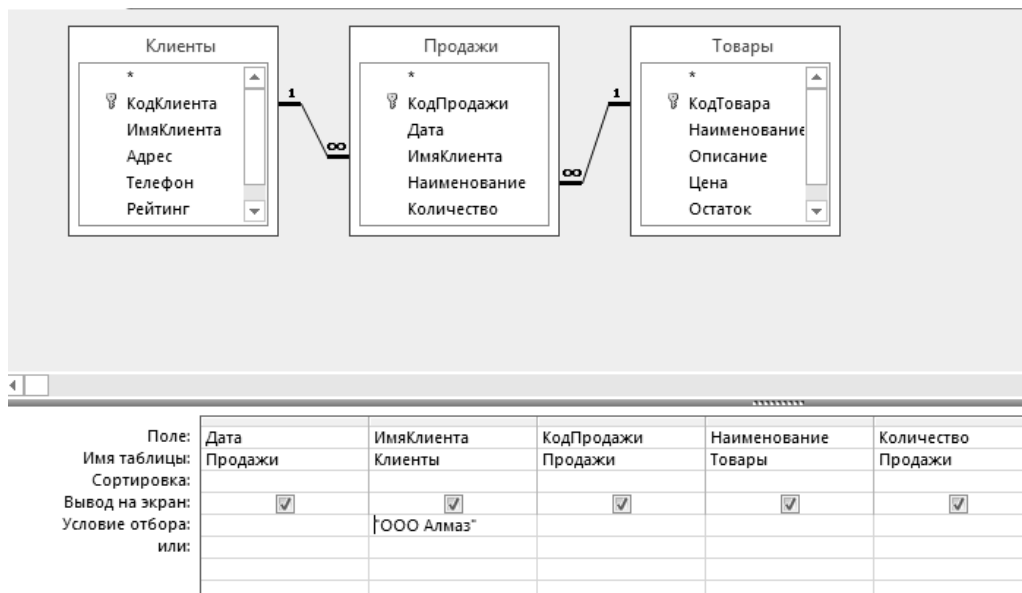


Рис.2.45. Запрос на вывод заказов клиента ООО Алмаз

Дата	ИмяКлиент	КодПродаж	Наименова	Количество
01.04.2017	ОО Алмаз	1	Морковь	10
06.04.2017	ООО Алмаз	4	Свекла	0
13.04.2017	ООО Алмаз	5	Морковь	0
*		(№)		

Рис.2.46. Результат работы простого запроса на выборку

7. Чтобы просмотреть результат запроса нажмите на панели кнопку

8. Результат запроса показан на рис.2.46.

9. Сформируйте запрос на выборку по условию «Выбрать клиентов ООО Алмаз или ООО Сказка, заказавших Морковь» (рис.2.47).

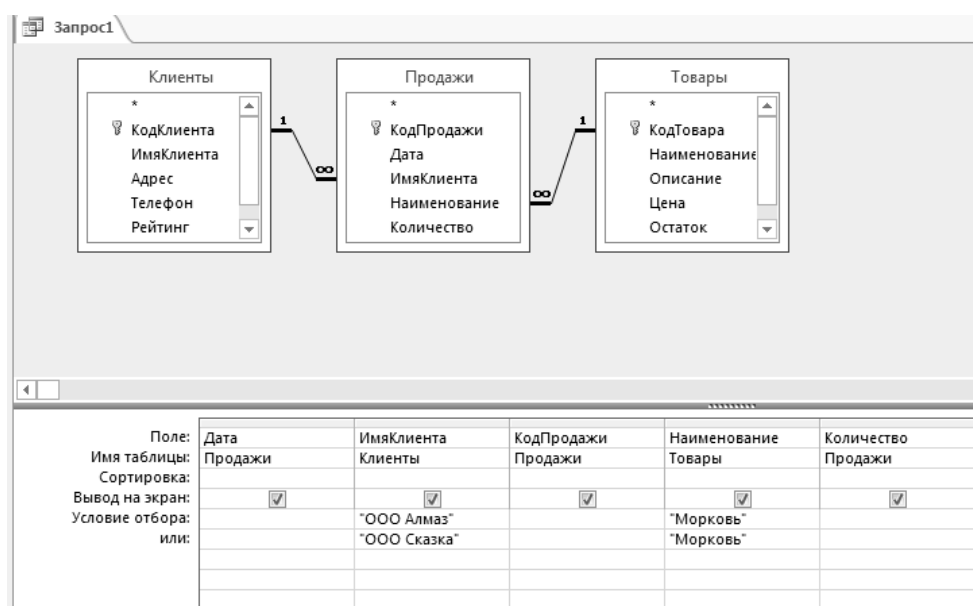


Рис.2.47. Запрос на выборку с использованием условий OR и AND

10. Результат запроса должен быть аналогичен показанному на рис.2.48

Дата	ИмяКлиент	КодПродаж	Наименова	Количество
01.04.2017	ОО Алмаз	1	Морковь	10
08.04.2017	ООО Сказка	3	Морковь	20
13.04.2017	ООО Алмаз	5	Морковь	0
*		(№)		

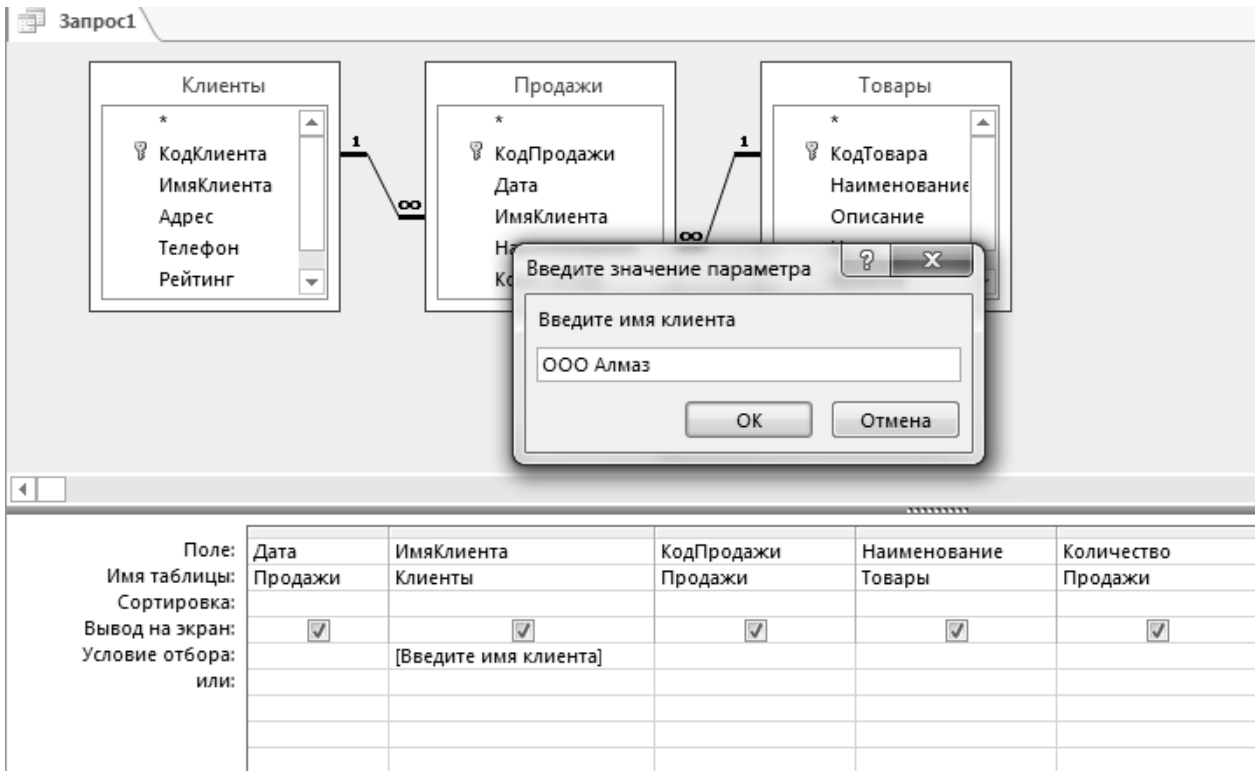
Рис.2.48. Результат запроса с использованием условий OR и AND

11. Сохраните последний запрос под именем *Выборка\_OR\_AND*

12. Если параметр запроса (допустим, фамилия клиента) заранее неизвестны, в Access предусмотрена возможность ввода параметров запросов непосредственно перед их исполнением. Чтобы установить ввод параметра вместо конкретного значения, в строке "Условие отбора" вводятся имя или фраза, заключенные в квадратные скобки. Все, введенное в квадратных скобках Access рассматривает как имя параметра, и выводится в диалоговом окне запроса как комментарий к вводимому параметру. В запросе можно использовать несколько параметров, но с уникальными именами.

13. Создадим запрос, выводящий список клиентов с соответствующими заказами. Для этого в запросе в строке "Условие отбора" поля *ИмяКлиента* введите выражение: `= [Введите имя клиента]` (рис.2.49).

14. Запустите запрос на выполнение с помощью кнопки 



Поле:	Дата	ИмяКлиента	КодПродажи	Наименование	Количество
Имя таблицы:	Продажи	Клиенты	Продажи	Товары	Продажи
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		[Введите имя клиента]			
или:					

Рис.2.49. Конструирование запроса с параметром

15. Результат запроса должен быть аналогичен представленному на рис.2.50.

Дата	ИмяКлиент	КодПродаж	Наименова	Количество
01.04.2017	ОО Алмаз		1 Морковь	10
06.04.2017	ООО Алмаз		4 Свекла	0
13.04.2017	ООО Алмаз		5 Морковь	0
*		(№)		

Рис.2.50. Результат работы запроса с параметром

Кроме стандартных операторов сравнения '=', '<', '>', '<=', '>=', Access поддерживает также BETWEEN, IN, LIKE (см. Приложение).

18. Создайте запрос, выдающий список клиентов с запросом фамилии и сделавших заказ, например, в промежутке времени с 01.04.2017 по 15.04.2017 (рис.2.51). Для этого в строке *Условие отбора* введите выражение **Between 01.04.2017 And 15.04.2017**

19. *Самостоятельно!* Создайте запрос, выдающий список клиентов, сделавших заказ с запросом фамилии и с запросом интервала времени. Сохраните этот запрос с именем *Выбор\_дата\_Клиент*.

Аналогично создайте запрос *Выбор\_дата\_Поставщик*.

Поле:	Дата	ИмяКлиента	КодПродажи	Наименование	Количество
Имя таблицы:	Продажи	Клиенты	Продажи	Товары	Продажи
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Between #01.04.2017# And #15.04.2017#	[Введите имя клиента]			
или:					

Рис.2.51. Запрос на выборку с параметром и заданным интервалом времени

## Запросы с вычисляемыми полями

В таблицах могут присутствовать вычисляемые поля, то есть поля, содержащие значения, полученные в результате операций над полями одной или нескольких таблиц. Созданию вычисляемого поля, значение которого рассчитывается исходя из данных одной таблицы, осуществляется достаточно просто. Для этого необходимо задать этому полю тип Вычисляемое (см. Приложение) и сконструировать необходимую формулу для вычисления его значений. Если же для вычисляемого поля необходимы данные из нескольких таблиц, приходится использовать специальные запросы.

1. Создайте запрос на базе таблиц *Продажи*, *Клиенты* и *Товары*, с помощью которого будет вычисляться дополнительное поле *Сумма*.
  2. Чтобы создать вычисляемое поле “Сумма”, содержащее информацию о стоимости заказа, в новом столбце (следующим за последним заполненным столбцом), в строке “Поле” введите выражение *Сумма:[Количество]\*[Цена]* (Рис.2.52).
  3. Запустите запрос на выполнение. Результат должен быть аналогичен представленному на рис.2.53
- Сохраните запрос с именем *Сумма\_Продажи*.

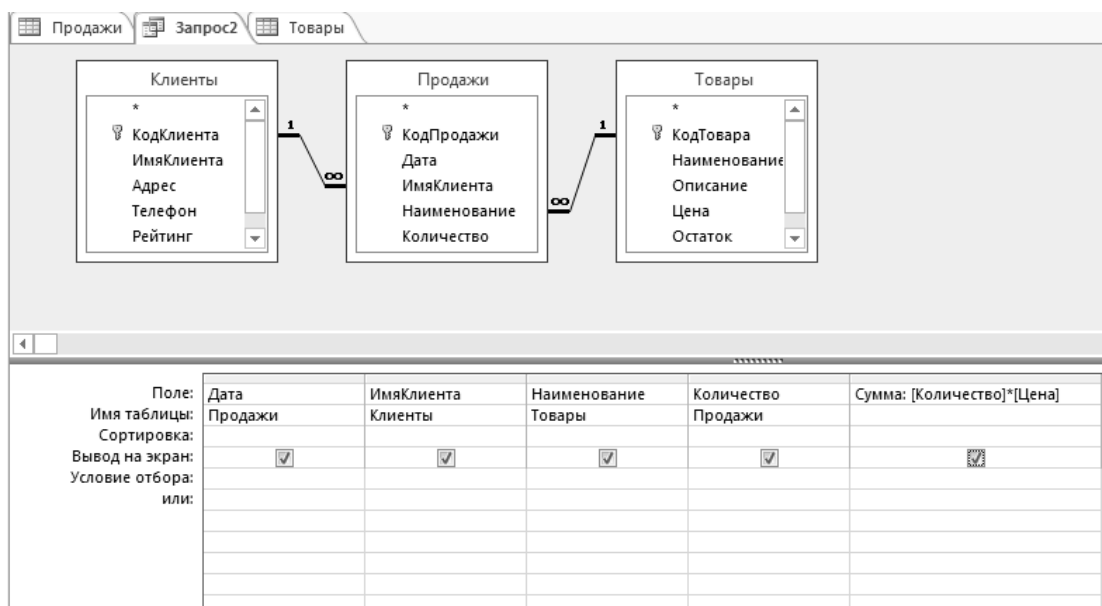


Рис.2.52. Запрос с вычисляемым полем


4. *Самостоятельно!* Измените запрос *Сумма\_Продажи* таким образом, чтобы получить список заказов на текущую дату. Сохраните запрос под именем *Сумма\_Продажи\_тек\_дата*.
5. *Самостоятельно!* Создайте запросы *Сумма\_Покупки* и *Сумма\_Покупки\_тек\_дата*.



Дата	ИмяКлиент	Наименова	Количество	Сумма
01.04.2017	ОО Алмаз	Морковь	10	500,00р.
06.04.2017	ООО Сказка	Свекла	10	700,00р.
08.04.2017	ООО Сказка	Морковь	20	1 000,00р.
06.04.2017	ООО Алмаз	Свекла	30	2 100,00р.
13.04.2017	ООО Алмаз	Морковь	20	1 000,00р.
06.04.2017	ООО Роза	Картофель	40	1 200,00р.

Рис.2.53. Результат выполнения запроса

### 2.3. Лабораторная работа 3. Создание сложных запросов

1. Откройте запрос *Сумма\_Продажи* в режиме **Конструктора**. Перенесите в бланк запроса поля *ИмяКлиента*, *Наименование*, *Количество*, *датаПродажи*. Переместите курсор в следующее поле и вызовите **Построитель выражений**, нажав кнопку  **Построитель** вкладки **Конструктор**.

В появившемся окне **Построителя** в поле *Элементы выражений* выберите *Встроенные функции*. В *Категории выражений* - *Дата и время*. В поле *Значения выражений* функцию *MonthName* (рис.2.53 а). Эта функция преобразует номер месяца в его название. Чтобы выделить из поля *Дата* номер месяца, необходимо использовать функцию *Month*.

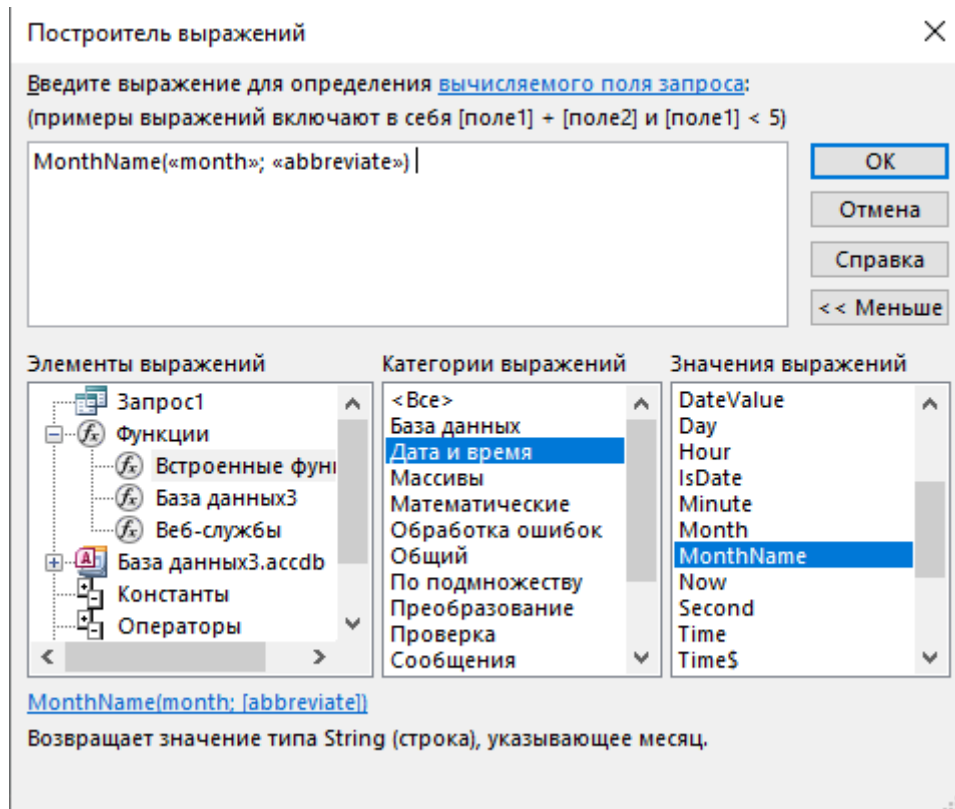


Рис.2.53.а. Создание вычисляемого поля.

Выделите первый аргумент функции *MonthName* и замените его функцией *Month*. Второй аргумент удалите (он определяет сокращение названия поля).

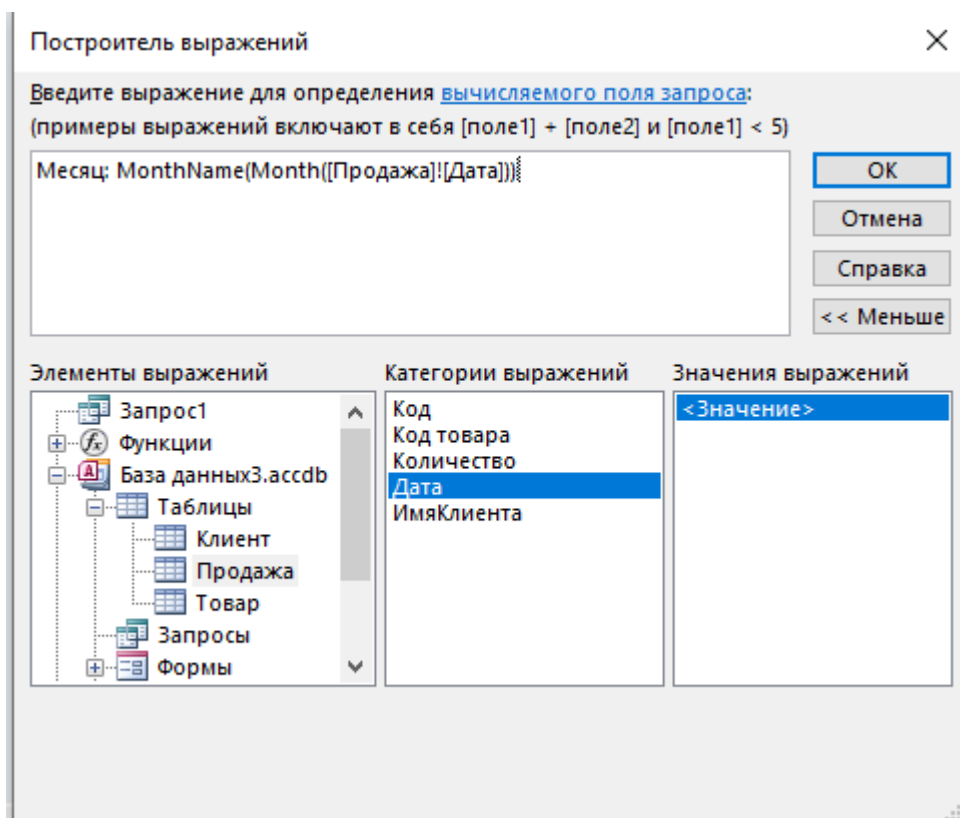


Рис.2.53.б. Создание вычисляемого поля.

Выделите аргумент функции *Month* «Дата». Замените его полем *Дата* из таблицы *Продажи*. Отредактируйте получившееся выражение так, как показано на рис. 2.53.б.

Нажмите ОК. Выполните запрос. Результат должен получиться аналогичным представленному на рис. 2.53.в.

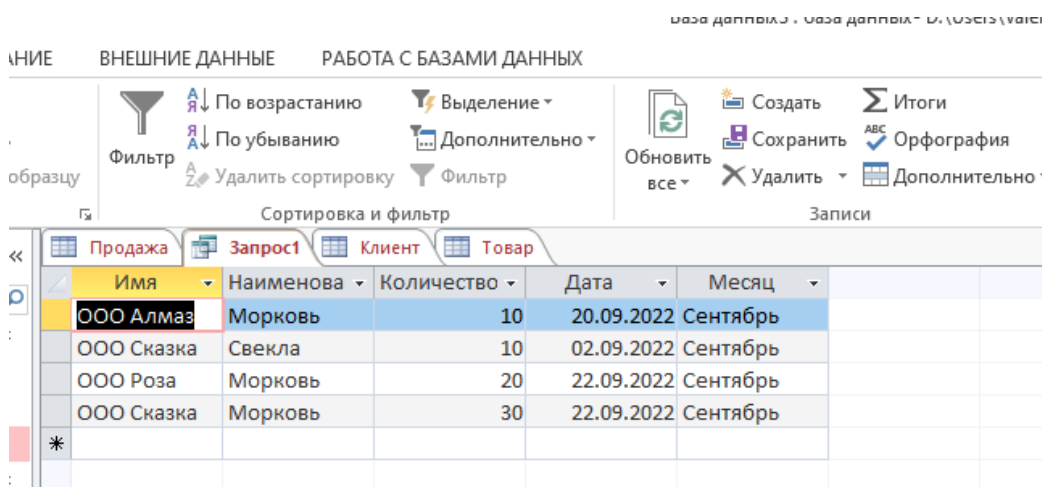


Рис.2.53.в. Результат запроса.

2. **Самостоятельно!** Создайте запрос на увеличение стоимости товара на процент, который пользователь назначает.

Используйте запрос *Сумма\_Продажи*.

Создайте вычисляемое поле *СуммаПродажи* =  $[Сумма] * (1 + (Процент) / 100)$ . Предусмотрите возможность ввода величины процента.

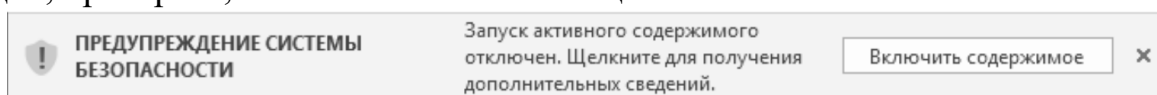
### Запросы на изменение БД

Существует четыре вида запросов на изменение:

- *запрос на создание таблицы* — создает новую таблицу на основе данных, которые уже размещены в одной или нескольких таблицах;
- *запрос на удаление* — удаляет записи из одной или нескольких таблиц на основании критериев, заданных в запросе;
- *запрос на добавление* — добавляет записи целиком или только указанные поля в одну или несколько таблиц;
- *запрос на обновление* — изменяет данные в существующих таблицах, обновляя записи с использованием критериев.

### **Создание таблиц с помощью запроса.**

Запросы на изменение данных в записях таблицы базы (обновление, добавление и удаление), также, как и запросы на создание таблицы из записей, сформированных в нем, могут выполняться некорректно, если включена блокировка (по умолчанию). Если при выполнении подобного запроса ничего не происходит, проверьте, не появляется ли сообщение



Если отображается это сообщение, нажмите на кнопку **Включить содержимое**. После этого выполнение запроса будет доступным.

В лабораторной работе 3 мы создали запросы с вычисляемым полем (*Сумма\_Продажи* и *Сумма\_Покупки*). Но результат этих запросов мы можем только посмотреть на экране или распечатать. Чтобы иметь возможность использовать его в дальнейшей работе, необходимо этот результат сохранить в специальной таблице. Сделаем это следующим образом.

1. Откройте базу данных *Отдел\_Продаж*
2. На вкладке **Создание** в группе *Другие* нажмите кнопку **Конструктор запросов**.
3. В окне **Добавление таблицы** дважды щелкните кнопкой мыши на названиях таблиц, из которых нужно получить данные: *Клиенты*, *Продажи* и *Товары*. Нажмите кнопку **Закрыть**.

4. В каждой таблице дважды щелкните кнопкой мыши на полях, которые необходимо добавить в выборку. Выберите поля *КодПродажи*, *Дата* и *Количество* из таблицы *Продажи*, поля *ИмяКлиента* из таблицы *Клиенты* и *Наименование* из таблицы *Товары*.

5. Создайте вычисляемое поле *Сумма:[Количество]\*[Цена]* (рис.2.54).

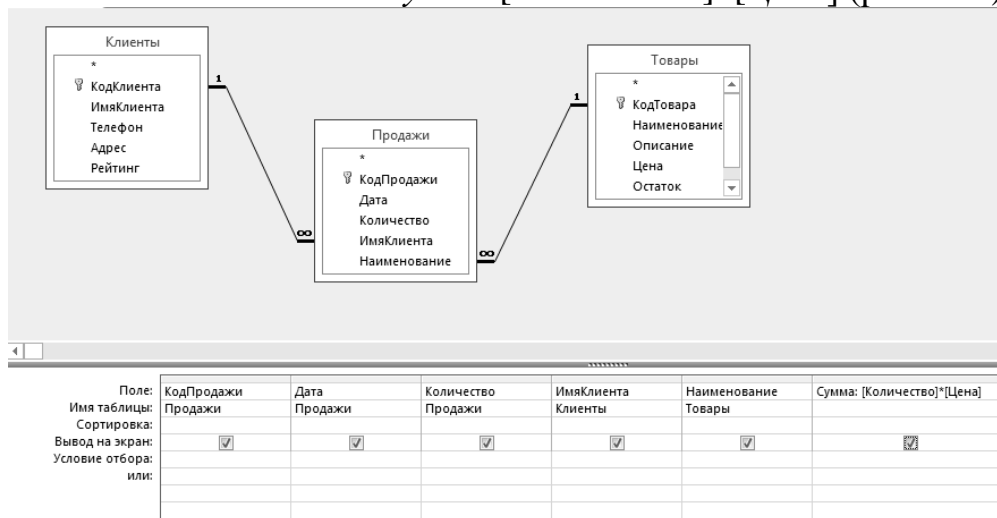



Рис.2.54. Создание запроса на добавление поля в таблицу

6. Выполните запрос, нажав на кнопку  .

7. Перейдите в режим *Конструктора*.

8. На вкладке *Конструктор* в группе *Тип запроса* выберите команду *Создание таблицы*. Откроется окно *Создание таблицы* (рис.2.55).

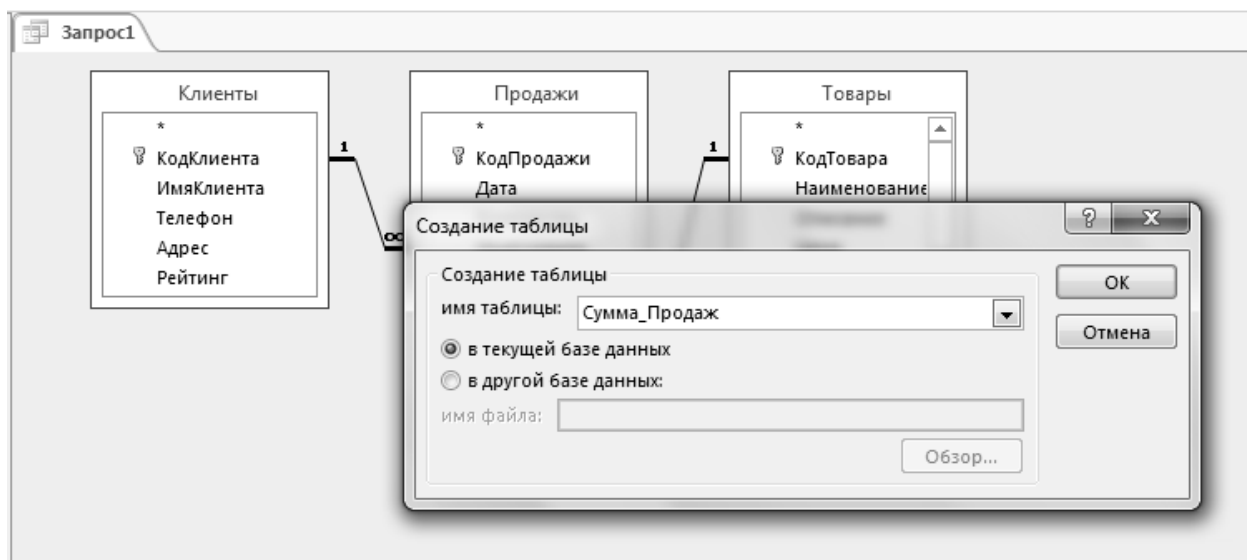


Рис.2.55. Задание имени создаваемой таблицы

9. Введите имя новой таблицы — *Сумма\_Продаж*. Установите переключатель в положение *в текущей базе данных*. Нажмите кнопку *ОК*.

10. Нажмите кнопку *Выполнить*  Откроется окно подтверждения операции. Нажмите кнопку *Да*.

11. В структуре базы данных появилась новая таблица *Сумма\_Продаж*, содержащая те же поля и данные в них, которые содержит таблица *Продажи*, плюс еще одно поле *Сумма*.

### **Создание запроса на обновление таблицы**

Пусть при заказе на сумму более 50000 руб. клиент получает скидку в 10%. Таким образом, стоит задача обновить значение поля *Сумма* таблицы *Сумма\_Продаж* с учетом скидки. Выполним следующие действия.

1. На вкладке **Создание** в группе **Другие** нажмите кнопку **Конструктор запросов**.

3. В окне **Добавление таблицы** дважды щелкните кнопкой мыши на названии таблицы *Сумма\_Продаж*. Нажмите кнопку **Заккрыть**.

4. Дважды щелкните кнопкой мыши на полях, которые необходимо добавить в выборку. Выберите поля *КодПродажи* и *Сумма*.

5. В бланке запроса в строке **Условие отбора** для поля *Сумма* введите условие выборки  $> 50000$ .

Результирующий бланк запроса на выборку показан на рис. 2.56.

6. Выполните запрос. Если результат выборки содержит только записи, где сумма заказа больше 50000, запрос составлен правильно.

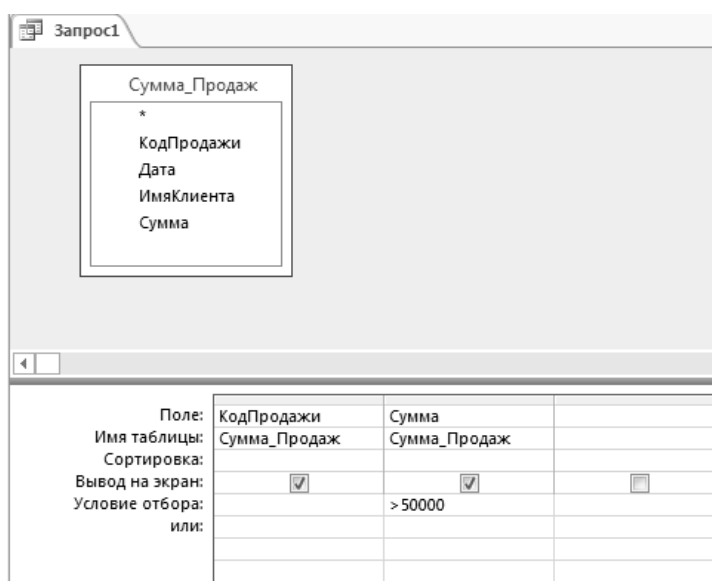


Рис.2.56 Запрос на выборку по условию

7. Перейдите в режим **Конструктора**. На вкладке **Конструктор** в группе **Тип запроса** выберите команду **Обновление**.

8. В бланке запроса в строке **Обновление** для поля *Сумма* введите формулу

$[Сумма\_Продаж]![Сумма]-[Сумма\_Продаж]![Сумма]*0,1$ . Теперь значение в поле *Сумма* для каждой записи, соответствующей условию выборки, будет уменьшено на 10 %.

9. Результирующий бланк запроса на обновление показан на рис.2.57.

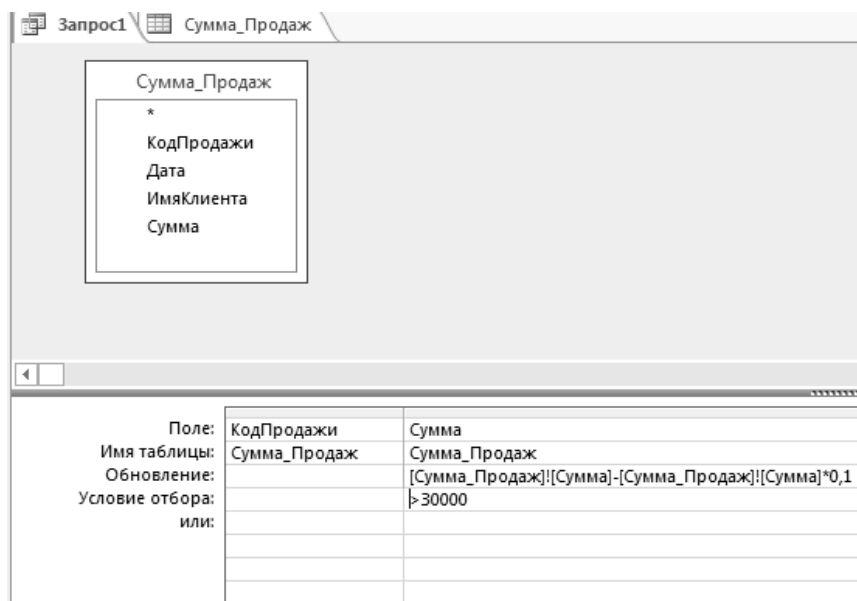


Рис.2.57 Запрос на обновление значения поля *Сумма*

10. Нажмите кнопку **Выполнить**. Откроется окно подтверждения операции обновления. Нажмите кнопку *Да*.
11. Сохраните запрос под именем *Скидки*.
12. Откройте таблицу *Сумма\_Продаж* и убедитесь, что данные в поле *Сумма* изменились.
13. *Самостоятельно!*

Так как количество товара на складе постоянно меняется, при продажах оно уменьшается, а при покупках увеличивается, необходимо предусмотреть возможность обновления поля *Остаток* в таблице *Товары*. В связи с этим, необходимо разработать два запроса на обновление *Обновление\_остатка\_минус* и *Обновление\_остатка\_плюс*.

Для создания этих запросов нужно использовать таблицы *Клиенты*, *Продажи*, *Товары* и *Поставщики*, *Поставки*, *Товары* соответственно.

### Создание запросов на удаление записей

Запрос на удаление записей используется, если требуется удалить сразу большое число записей. При этом, если удаляются записи в подчиненной таблице (таблицы *Продажи* и *Покупки*) удаление происходит без проблем. Если же удаление происходит в базовой таблице (*Товары*, *Клиенты* и *Поставщики*), то удалить записи можно только в том случае, когда на схеме данных в окне **Изменение связей** установлен флажок *Каскадное удаление связанных полей*.

Удалим в таблице *Продажи* записи, относящиеся к клиенту с именем «Козлов».

1. Создайте новый запрос в режиме **Конструктора**.
2. Добавьте таблицы *Продажи* и *Клиенты*.
3. В таблице *Продажи* дважды щелкните на поле «\*», в таблице *Клиенты* – на поле *ИмяКлиента*.
4. В бланке запроса в строке *Условие отбора* в поле *Имя* задайте “Козлов”
5. Запустите запрос на выполнение и убедитесь, что выбранные записи содержат только клиента с именем «Козлов».
6. Вернитесь в режим **Конструктора**.
7. На вкладке **Конструктор** в группе **Тип запроса** нажмите кнопку **Удаление**.

В бланке запроса в строке *Удаление* в столбце со значком \* должно быть установлено значение *Из*, а в столбце с условием — значение *Условие*. (рис.2.58)

8. Нажмите кнопку **Выполнить**. Откроется окно подтверждения операции. Нажмите кнопку *Да*.

9. Откройте таблицу *Продажи* и посмотрите на внесенные в данные изменения. (Теперь в таблице не осталось записей клиента с именем «Козлов»).

10. *Самостоятельно!* создайте запрос на удаление по запросу имени удаляемого клиента. Сохраните этот запрос под именем *Удаление\_продажи\_Клиент*.

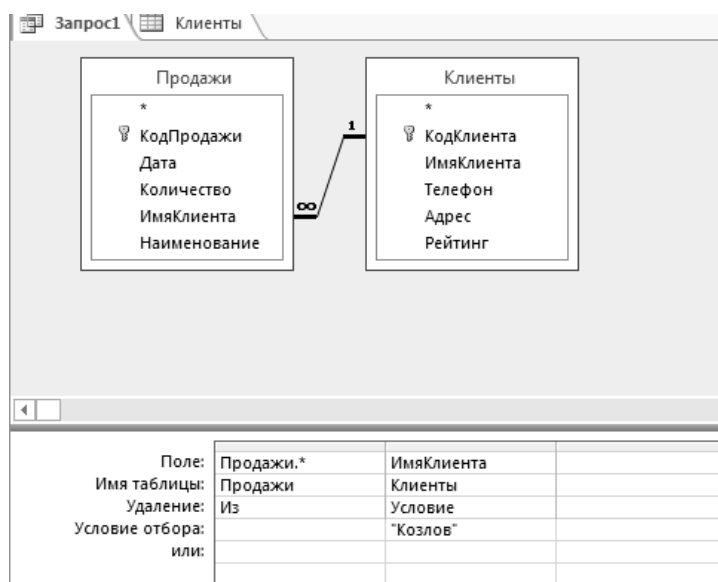


Рис.2.58. Запрос на удаление записей

11. Аналогично создайте запрос на удаление поставщика *Удаление\_поставки\_Поставщик*.

### **Использование групповых операций в запросах**

Групповые операции позволяют выделить группы записей с одинаковыми значениями в указанных полях и вычислить итоговые данные для каждой из групп по другим полям, используя одну из статистических функций. Статистические функции применимы, прежде всего, к полям с типом данных Числовой (Number), Денежный (Currency), Дата и время (Date/Time).

В Access предусматривается девять статистических функций:

- \_ Sum — сумма значений некоторого поля для группы;
- \_ Avg — среднее от всех значений поля в группе;
- \_ Max , Min — максимальное, минимальное значение поля в группе;
- \_ Count — число значений поля в группе без учета пустых значений;
- \_ StDev — среднеквадратичное отклонение от среднего значения поля в группе;
- \_ Var — дисперсия значений поля в группе;
- \_ First и Last — значение поля из первой или последней записи в группе.

Результат запроса с использованием групповых операций содержит по одной записи для каждой группы. В запрос, прежде всего, включаются поля, по которым производится группировка, и поля, для которых выполняются статистические функции. Кроме этих полей в запрос могут включаться поля, по которым задаются условия обора.

Рассмотрим конструирование однотабличного запроса с групповой операцией на примере таблицы *Продажи*.

### *Запрос с функцией Sum*

Определим, какое суммарное количество каждого из товаров продано.

1. Создайте в режиме конструктора запрос на выборку из таблицы *Продажи*.
  2. Перетащите в бланк запроса поле *Наименование*. По этому полю будет производиться группировка записей таблицы.
  3. Перетащите в бланк запроса поле *Количество*, по которому будет подсчитываться суммарное количество каждого из товаров.
  4. Выполните команду **Итоги** ( $\Sigma$ ) из группы **Показать или скрыть**. В бланке запроса появится новая строка *Групповая операция* со значением *Группировка* в обоих полях запроса.
  5. В столбце *Количество* замените слово **Группировка** на функцию *Sum*. Для этого вызовите список и выберите эту функцию. Бланк запроса примет вид, показанный на рис.2.59.
  6. Проверьте работу запроса, нажав кнопку **Выполнить**. Если все в порядке, перейдите в режим **Конструктора** и в окне *Свойств поля* на вкладке *Общие* в строке *Подпись* введите «*Всего*» (рис.2.60).
  7. Сохраните запрос под именем *Всего\_Продажи*.
  8. Откройте запрос *Всего\_Продажи* в режиме **Конструктора**.
- Самостоятельно!* Переделайте запрос таким образом, чтобы можно было получить общее количество каждого товара с запросом имени клиента. ( Следует иметь в виду, что в этом случае вводить нужно не *ИмяКлиента*, а *КодКлиента*). Сохраните запрос под именем *Всего\_Продажи\_Клиент*.



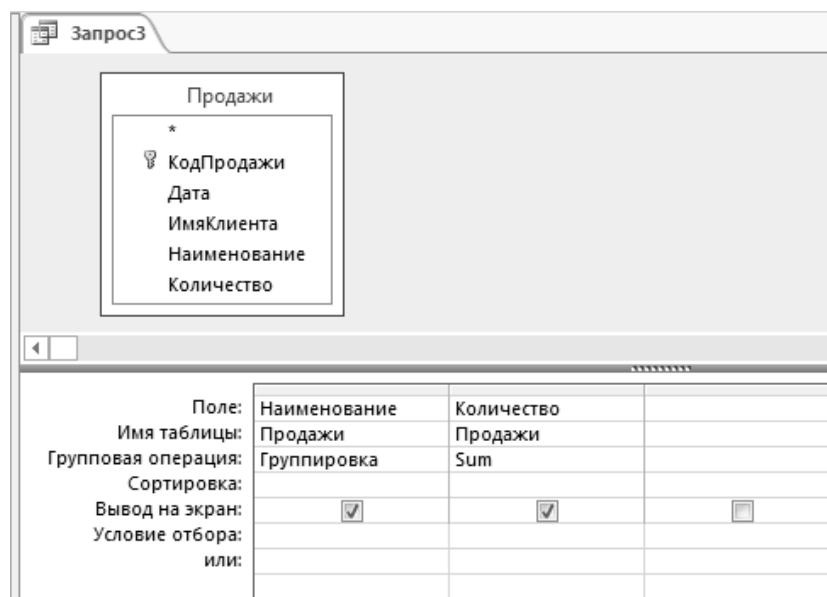


Рис.2.59. Запрос с группировкой по коду товара и суммированием количества в группе

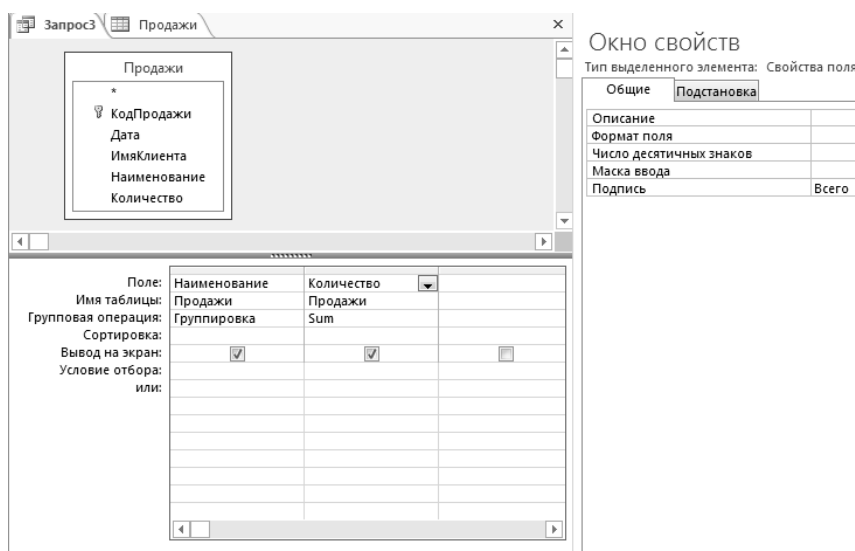


Рис.2.60. Изменение имени поля «Количество»

### Запрос с функцией Count

Определим, сколько раз поставлялся каждый товар.

1. Создайте запрос на выборку на основе таблицы *Поставки*.
2. Из списка полей перетащите в бланк запроса поле *ИмяПоставщика*. По сути, смысл задачи сводится к подсчету в таблице числа строк с одинаковыми именами поставщиков, поэтому неважно, по какому полю будет вычисляться функция *Count*. Перетащите в бланк запроса любое поле, например, опять *ИмяПоставщика*.
3. Выполните команду **Итоги** ( $\Sigma$ ) из группы **Показать или скрыть**. Замените слово **Группировка** в одном из столбцов с именем *ИмяПоставщика*.

на функцию *Count*. Бланк запроса примет вид, показанный на рис. 2.61.

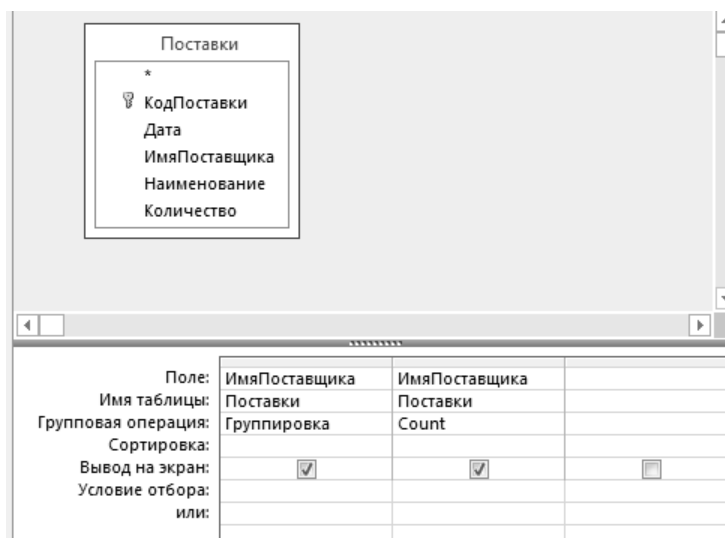


Рис. 2.61. Запрос для подсчета числа поставок

4. Сохраните запрос под именем *Число\_поставок*. Выполните запрос. Результат запроса должен быть аналогичен представленному на рис. 2.62.

ИмяПоставщ	Count-Имя
Иванов	3
Петров	3


Рис. 2.62. Результат подсчета числа поставок

**Самостоятельно!** Создайте два запроса *Сумма\_поставок\_баланс* и *Сумма\_продаж\_баланс*. Используйте запросы *Сумма\_Поставок* и *Сумма\_Продаж* соответственно. Группировку используйте по месяцам. Результатами запросов должны быть таблицы с двумя столбцами : *Месяц* и *Сумма*. Создайте запрос *Баланс\_месяц*. Этот запрос создается на основе двух предыдущих запросов. Для этого в бланк запроса нужно добавить не таблицу, а соответствующие запросы. В результате выполнения запроса должна появиться таблица с двумя столбцами: *Месяц*, *Баланс*.

## 2.4. Лабораторная работа 4. Работа с формами в Access

### Создание простой формы

#### Создание формы с помощью мастера форм

1. Откройте учебную базу данных *Отдел\_Продаж*.
2. На вкладке **Создание** выберите инструмент **Мастер форм**. Откроется окно **Создание форм** (рис.2.17).
3. В открывшемся слева списке выберите таблицу *Клиенты*
4. Переместите в список выбранных полей все поля, нажав клавишу  (рис.2.18).
5. Нажмите кнопку *Далее*. Откроется окно выбора внешнего вида формы (рис. 2.19).

Существуют четыре вида формы:

- в один столбец — все поля созданной формы будут расположены в один столбец;
- ленточный — поля формы будут расположены в одну или несколько строк;
- табличный — форма будет выглядеть как обычная таблица;
- выровненный — ленточная форма, поля которой выровнены по ширине

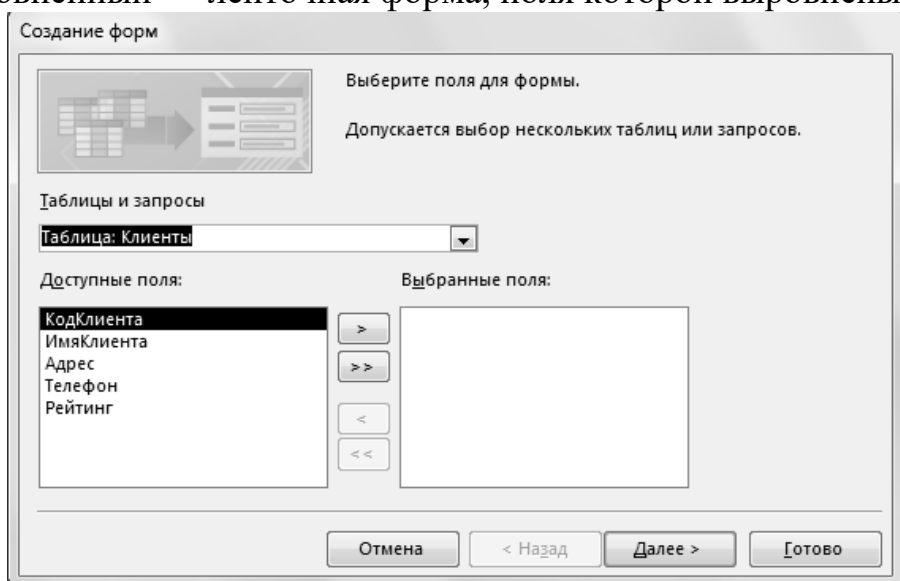


Рис.2 17. Создание формы в режиме **Мастера**

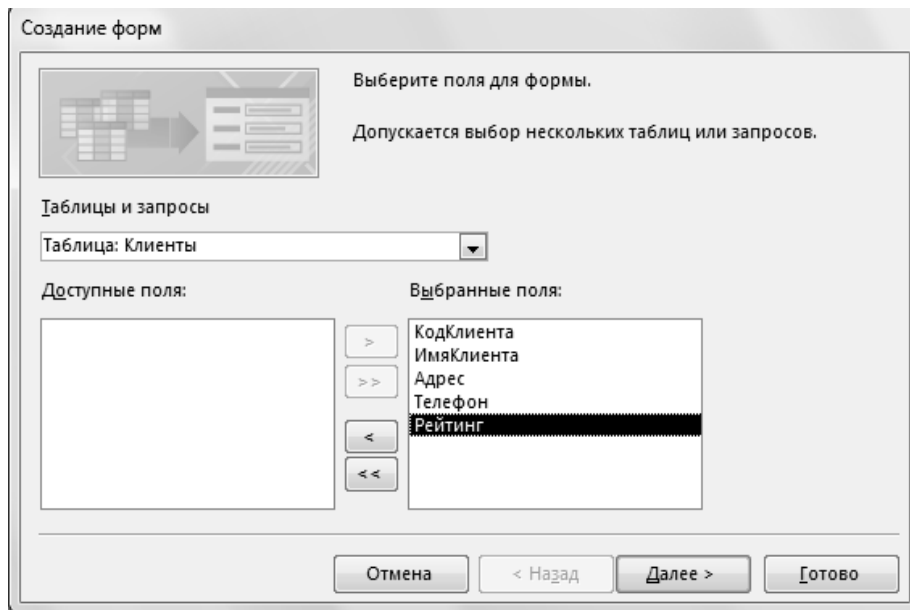


Рис.2 18. Выбор полей для формы

. 6. Установите переключатель в положение нужного вида формы (например, **в один столбец** ) и нажмите кнопку *Далее*.

➤ *Для Access 7.* Откроется окно выбора стиля формы (рис.2.20)

Чтобы посмотреть, как выглядит тот или иной стиль, щелкните на его названии кнопкой мыши. В левой части окна вы увидите, как изменится оформление формы. Выберите подходящую схему оформления (например, **Модульная**) и нажмите кнопку *Далее*.

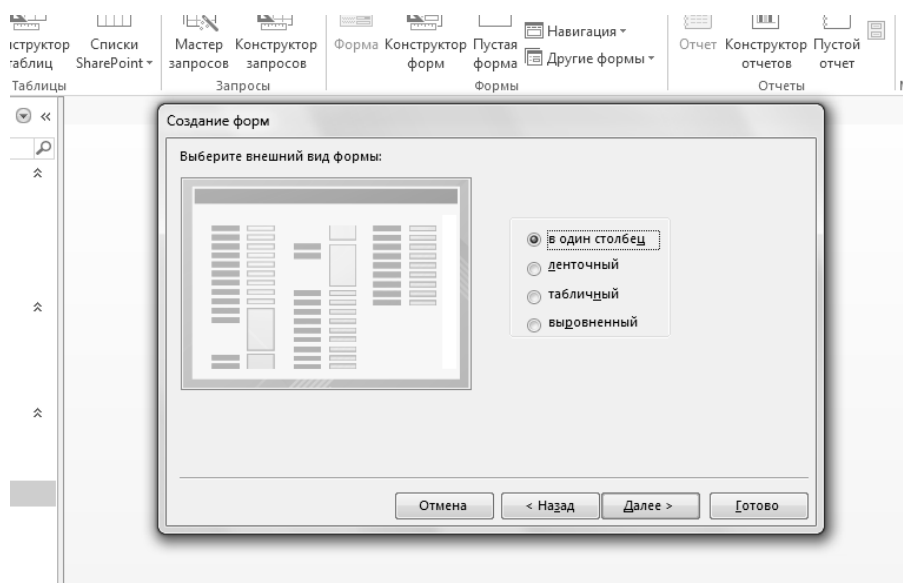


Рис.2 19. Окно выбора внешнего вида формы

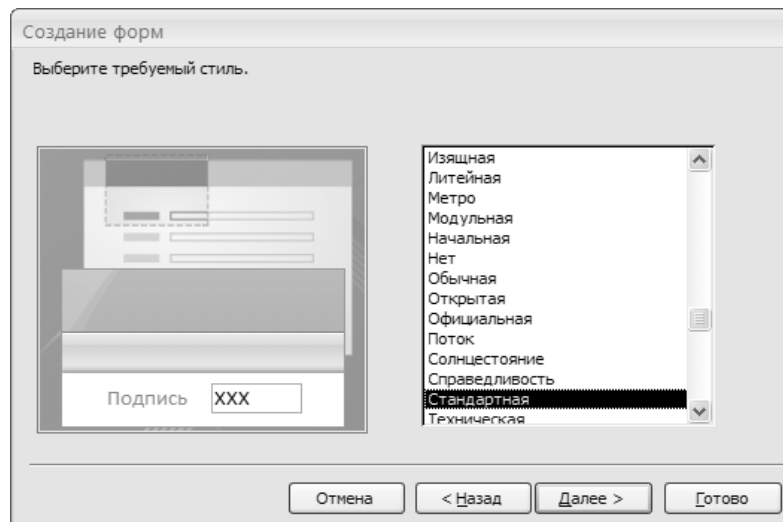


Рис. 2.20. Окно выбора стиля формы ( Access 7)

7. В открывшемся окне мастера форм (рис. 2.21) в текстовое поле введите имя новой формы и нажмите кнопку *Готово*.

Новая форма откроется в режиме просмотра (рис.2.22).

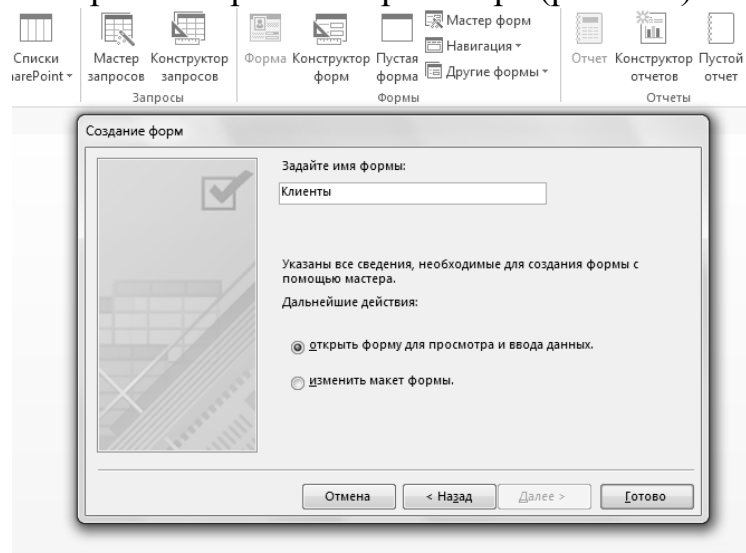


Рис. 2.21. Окно для задания имени формы

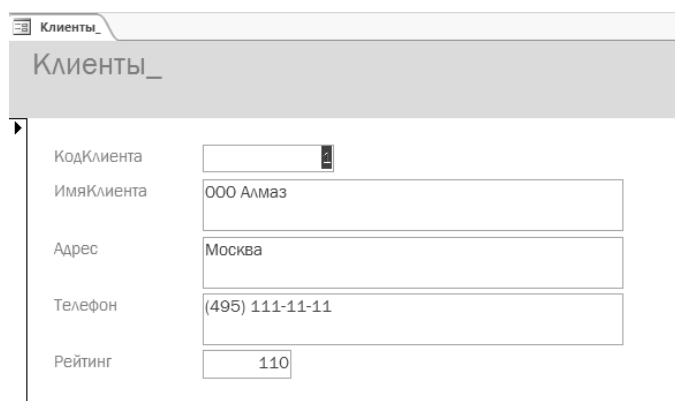


Рис. 2.22. Форма в режиме просмотра

8. Перейдите в режим **Конструктора**. Выделите поле *Рейтинг*.

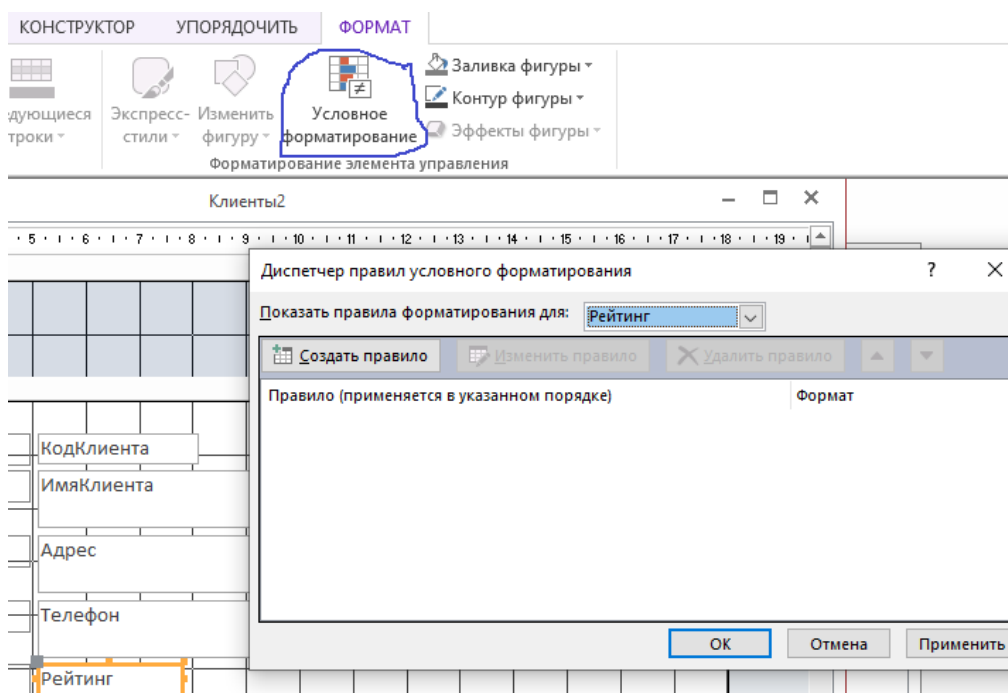


Рис.2.22.1. Форматирование поля *Рейтинг*

9. На вкладке *Формат* Выберите пункт *Условное форматирование* (рис.2.22.1)

10. В появившемся окне выберите *Создать правило*.

11. В новом окне сформируйте правило, согласно которому рейтинги меньше 70 (например) будут выделяться красным цветом (рис.2.22.2).

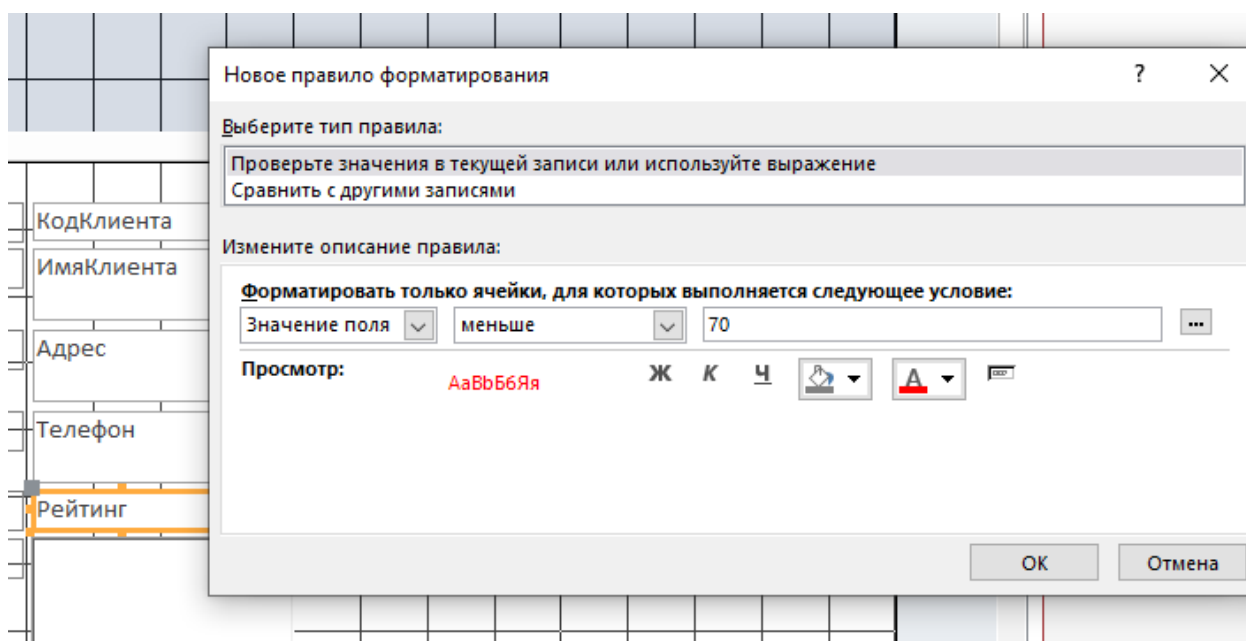


Рис.2.22.2. Задание правила форматирования для поля *Рейтинг*

12. В области заголовка формы правой клавишей мыши вызовите контекстное меню (рис.2.23) и оформите внешний вид формы.

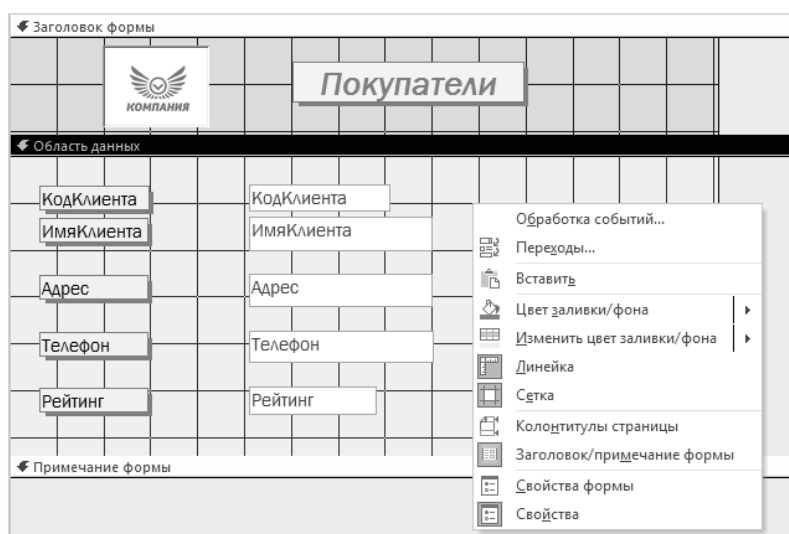


Рис.2.23. Вызов контекстного меню для оформления заголовка формы

13. Выделите по очереди все элементы формы, щелкнув на них мышкой, и вызывая правой клавишей контекстное меню, оформите внешний вид формы по своему желанию.

11. Перейдите в режим формы и оцените результат. Если он вас не удовлетворяет, повторите свои действия.

12. Сохраните форму.

13. Аналогичным образом создайте формы для таблиц *Поставщики* и *Товары*.

**Самостоятельно!** Для формы *Товары* сформируйте правило, согласно которому будет выделяться цветом остаток товара, близкий к страховому запасу.

### **Создание подчиненной формы**

1. Перейдите на вкладку **Создание** и в группе **Формы** выберите инструмент **Мастер форм**.

2. В диалоговом окне выберите таблицу *Клиенты* и перенесите в список **Выбранные поля** поле *ИмяКлиента*.

3. Затем выберите таблицу *Продажи* и перенесите в список все поля из этой таблицы, кроме поля *ИмяКлиента* (рис.2.24). Нажмите кнопку **Далее**.

4. В появившемся окне выберите **Подчиненная форма** (рис.2.25). Нажмите **Далее**.

5. В появившемся окне выберите вид подчиненной формы. Пусть это будет **Ленточный**. Нажмите **Далее**.

6. В появившемся окне задайте имя создаваемой формы. Пусть это будет *Клиенты\_Продажи*. Нажмите **Готово**.

7. Новая форма откроется в режиме просмотра.
8. Перейдите в режим **Конструктора** и создайте свой дизайн для формы *Клиенты\_Продажи*. Вставьте в заголовок рисунок, нажав кнопку **Эмблема**. С помощью кнопки **Дата и время** вставьте текущую дату. Выделите появившееся в форме поле с датой и перетяните его на верхний колонтитул (если оно расположено в другом месте формы)

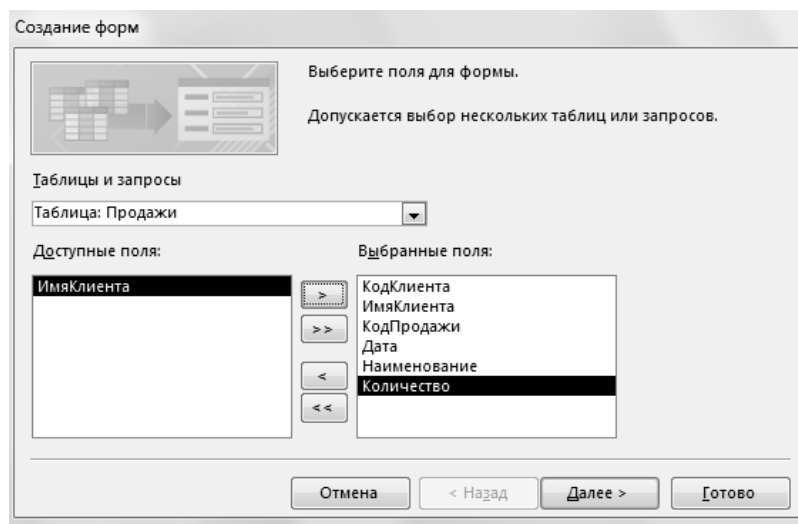


Рис.2.24. Создание подчиненной формы

9. Выделите поле *Дата* и в окне *Свойства* на вкладке *Данные* задайте *Значение по умолчанию- Дата()* (рис.2.26). Эта функция автоматически заполняет это поле значением текущей даты.

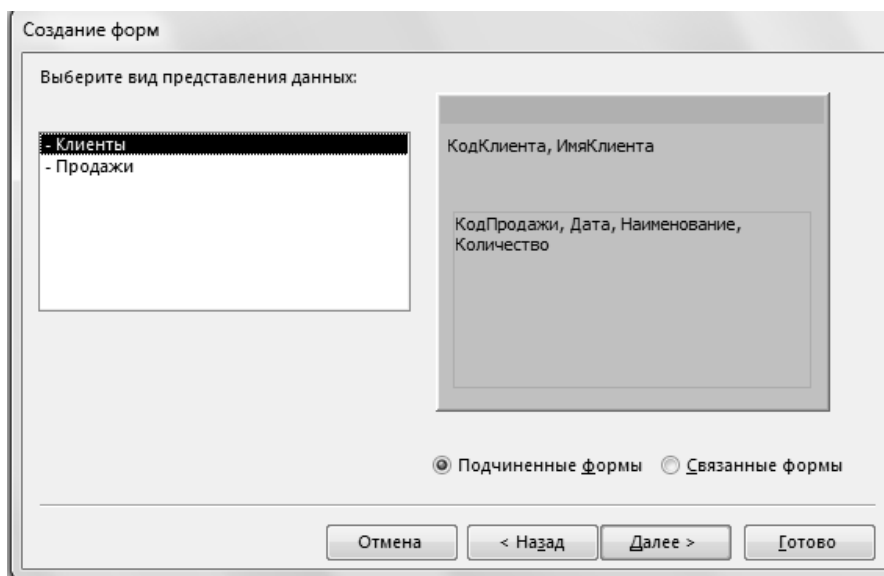


Рис.2.25. Выбор вида представления данных в форме





Рис.2.26. Задание свойств полю Дата

10. Сохраните изменения, сделанные для макета формы (рис.2.27).

11. Аналогично создайте форму *Поставщики\_Поставки*

### Создание элементов управления формой.

Для работы с полями форм и работы с самой формой удобно использовать **кнопки**. Для того, чтобы создать кнопку нужно выполнить следующие действия:

1. Откройте форму *Клиенты* в режиме конструктора.

На вкладке **Конструктор** из группы **Элементы управления** выберите инструмент **Кнопка** (рис.2.28).

2. Нарисуйте на форме в нужном месте кнопку требуемого размера. Появится окно выбора действий, для которых кнопка будет использоваться. Пусть это будет кнопка для добавления записи (то есть добавление нового клиента) (рис.2.29). В окне слева расположен образец рисунка на этой кнопке.

3. Нажмите *Далее*.

4. Появляется следующее окно (рис.2.30). Выберите, что вы хотите разместить на создаваемой кнопке рисунок или надпись. Пусть это будет надпись «*Добавить клиента*». Нажимаем *Далее*.

5. В появившемся окне задайте имя кнопки, под которым она будет использоваться при создании ссылок на нее. Пусть это имя будет «*Добавить*».

Нажмите *Готово*. Теперь можно создать дизайн для этой кнопки с помощью действий, описанных выше.

6. Создайте кнопку для закрытия формы. Сохраните изменения. Перейдите в режим формы (рис.2.31) и проверьте работоспособность разработанной формы.

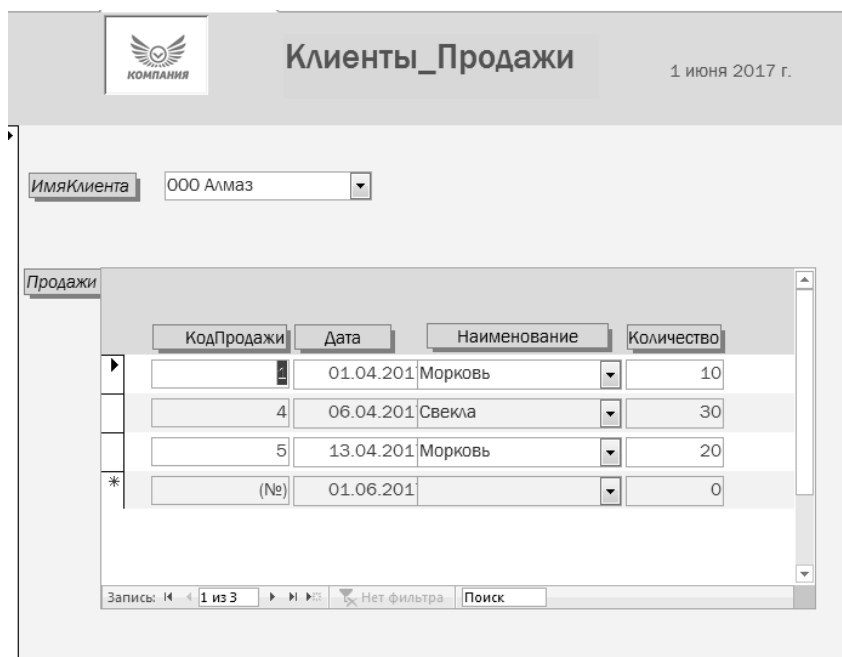


Рис.2.27. Форма Клиенты\_Продажи

7. Аналогичным образом добавьте такие же кнопки в формы *Товары* и *Поставщики*.
8. Добавьте несколько записей в таблицы *Клиенты*, *Товары* и *Поставщики* с использованием соответствующих форм.
9. Создайте кнопки для добавления записей и для выхода на формах *Клиенты\_Продажи* и *Поставщики\_Поставки*. При этом, кнопку «Добавить запись» в формах *Клиенты\_Продажи* и *Поставщики\_Поставки* необходимо разместить в поле «Примечание формы», а «Выход» в «Область данных»(рис.2.32).

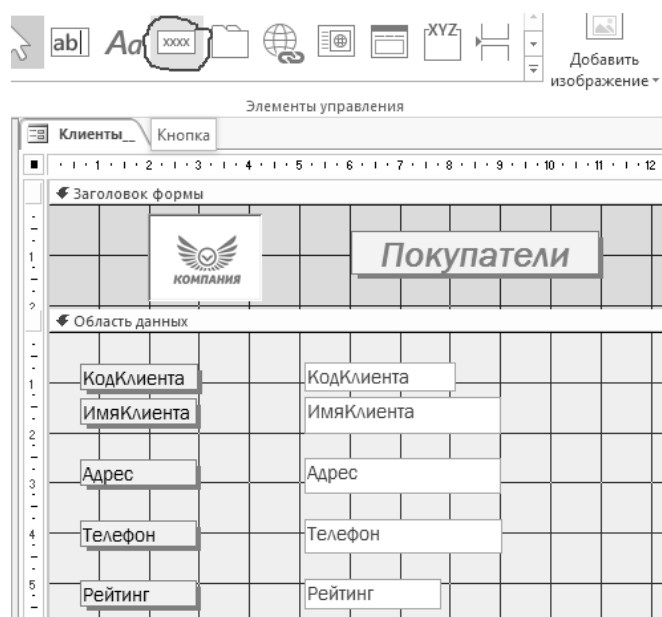


Рис.2.28. Выбор инструмента «Кнопка» для формы **Клиенты** в режиме конструктора

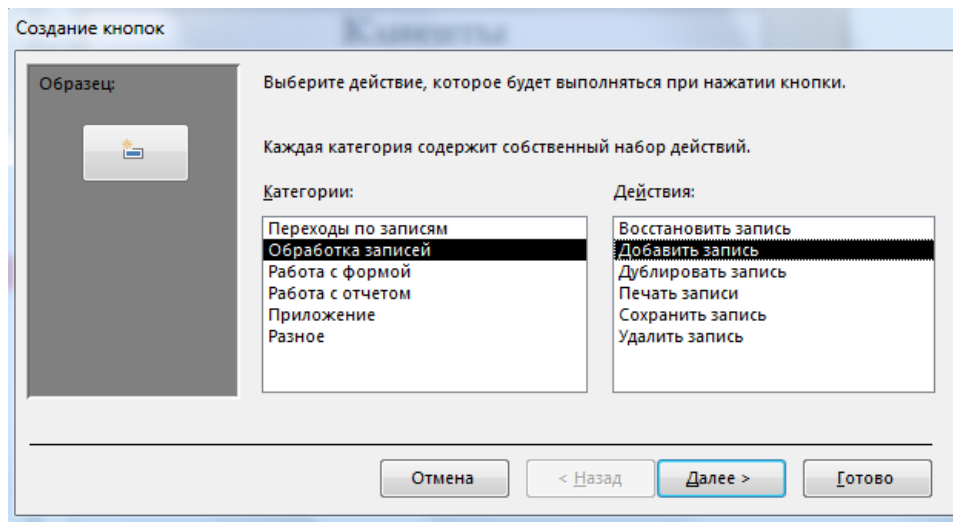


Рис.2.29. Окно выбора действий при нажатии кнопки

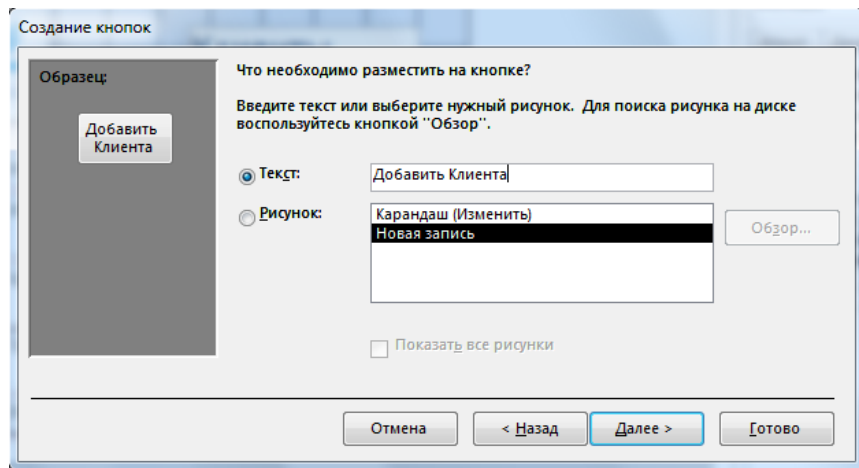


Рис. 2.30. Окно выбора надписи на кнопке

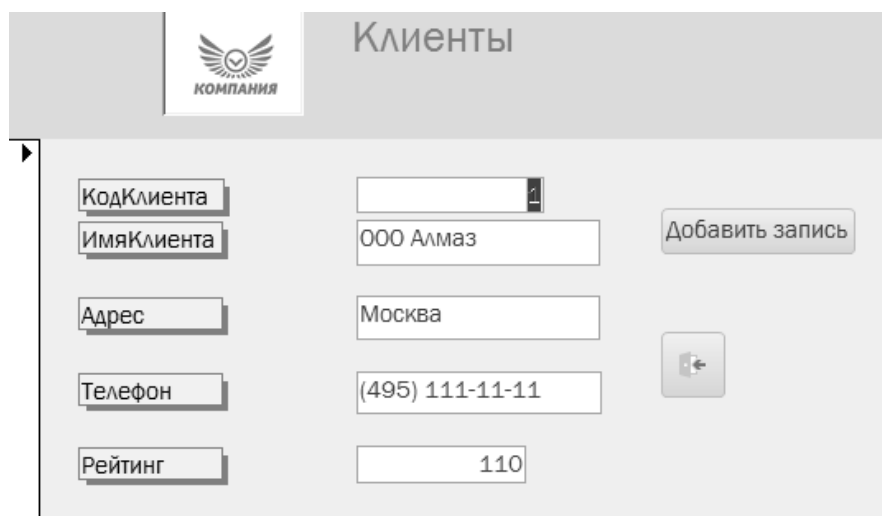


Рис. 2.31. Форма с кнопками Добавить запись и Выход

The screenshot shows a software interface for 'Клиенты\_Продажи'. At the top, there is a header with the company logo and the date '21 августа 2017 г.'. Below the header, there is a field for 'ИмяКлиента' with the value 'ООО Роза'. The main part of the form is a table titled 'Продажи' with the following columns: 'КодПродажи', 'Дата', 'Наименование', and 'Количество'. The table contains three rows of data:

КодПродажи	Дата	Наименование	Количество
6	06.04.201	Картофель	40
8	03.05.201	Свекла	50
(№)	21.08.201		0

Below the table, there is a button labeled 'Добавить запись'. At the bottom of the form, there is a status bar with the text 'Записи: 1 из 2' and 'Нет фильтра Поиск'.

Рис. 2.32. Подчиненная форма с кнопками *Добавить запись* и *Выход*

Формы «Клиенты\_Продажи» и «Поставщики\_Поставки» являются основными рабочими формами и служат для занесения записей о соответствующих сделках в отделе продаж.

В лабораторной работе 1 был использован пример проверки значения поля при вводе данных в таблицу. Если на основании этой таблицы построить форму и через нее осуществлять ввод данных, то заданная проверка будет также работать. Но для этого необходимо запускать запрос на обновление поля *Остаток (ЛР 3)*. В этом случае удобнее пользоваться ограничением не на поля таблицы, а на поля формы.

1. Откройте форму *Клиенты\_Продажи* в режиме **Конструктора**. В этой форме необходимо контролировать значения поля *Количество*, так как мы не можем продать товар, если его остаток на складе в результате этой продажи будет меньше страхового запаса.

1. Для того чтобы осуществить эту проверку нам потребуются дополнительные поля из таблицы *Товары*.
2. Выделите на подчиненной форме *Область данных* (рис.2.32.1). Выберите пункт *Добавить поля*. В появившемся *Списке полей* Выберите таблицу *Товары*.
3. Перетащите в подчиненную форму поля *Остаток*, *Страховой запас* и *Цена*.
4. Выделите выбранные поля и откройте *Окно Свойств*. На вкладке *Макет* в строке «Вывод на экран» задайте значение «Нет»(рис.2.32.2.)

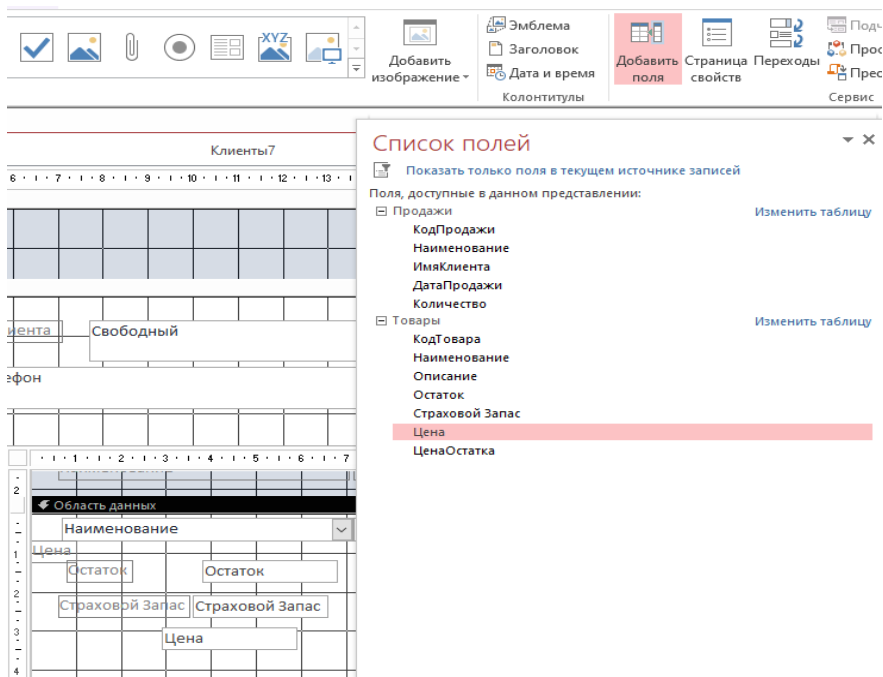


Рис. 2.32.1. Добавление полей из таблицы Товары

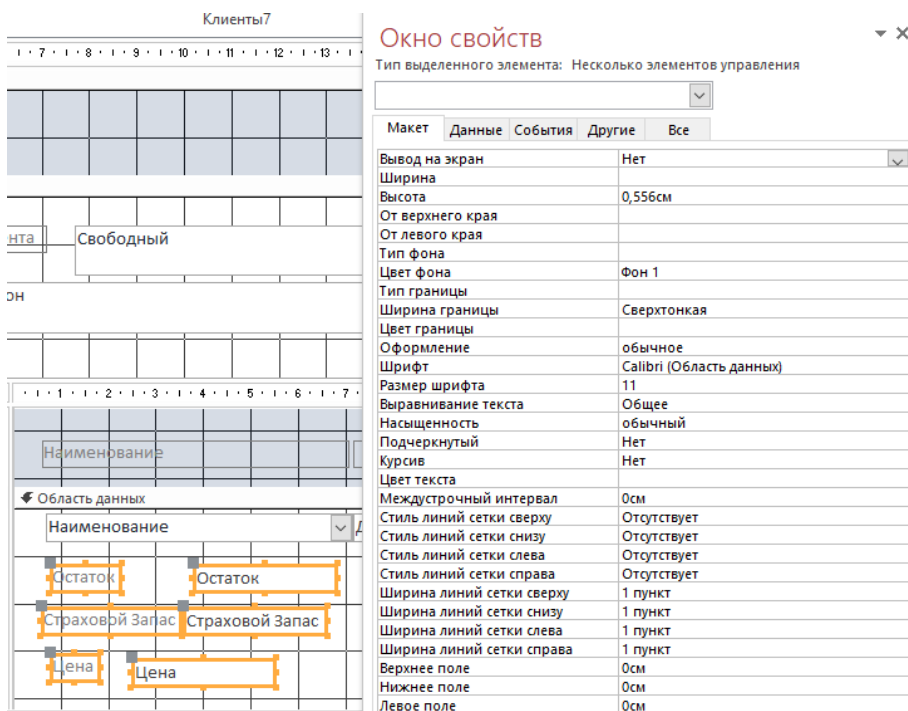


Рис. 2.32.2. Добавление полей из таблицы Товары

5. Выделите поле *Количество*. В свойствах поля на вкладке *Данные* в строке *Правило проверки* запишите  $\langle [Остаток] - [Страховой запас] \rangle$ , а в строке *Сообщение об ошибке* «Недостаточно товара на складе!». (рис.2.32.3)

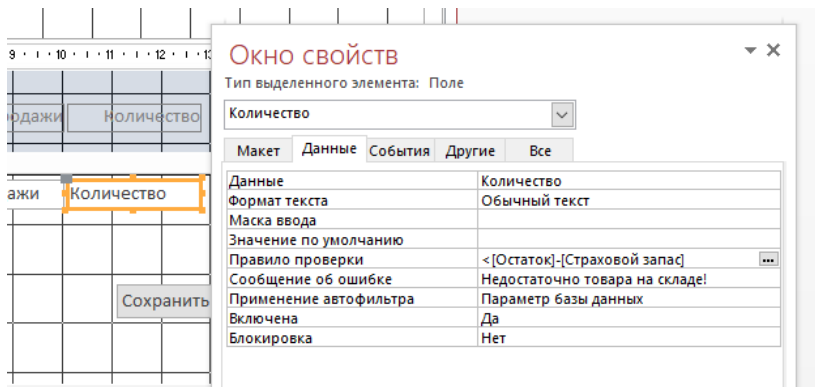


Рис. 2.32.3. Задание правила проверки поля *Количество*

Теперь, при попытке некорректного ввода количества продаваемого товара, появится сообщение и ввод нужно будет отменить (рис.2.32.4).

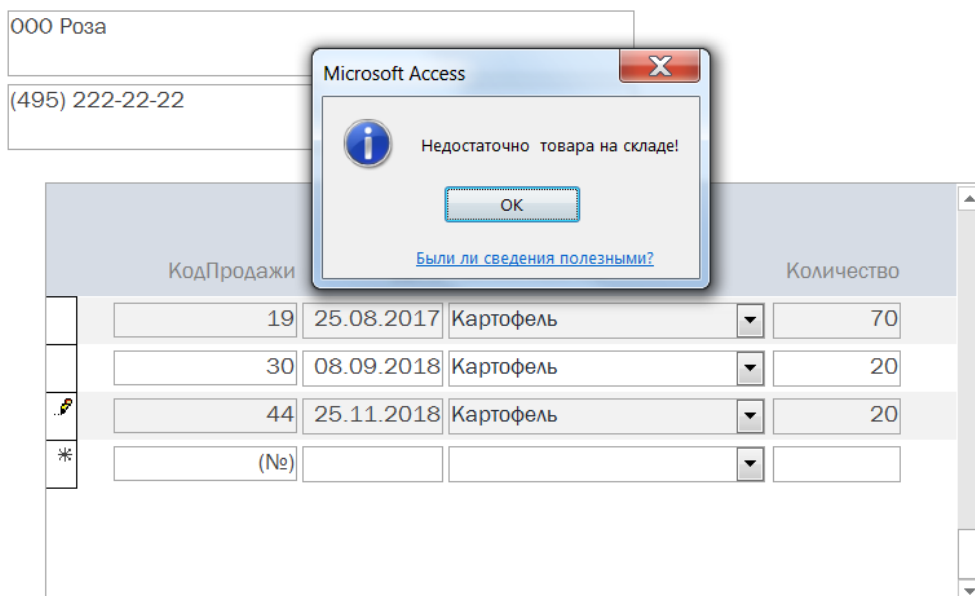


Рис.2.32.4. Результат действия ограничений на значение поля формы

6. Проверьте работоспособность формы.

**Самостоятельно !** Создайте кнопку с помощью которой будет вызываться запрос *Обновление\_остатка\_минус*. Задайте подпись к этой кнопке «*Обновить*». Кнопка создается в подчиненной форме. Аналогичную кнопку создайте в форме *Поставщики\_Поставки*.

### **Создание вычисляемых полей в форме**

1. Откройте форму *Клиенты\_Продажи* в режиме **Конструктора**.
2. Выделите подчиненную форму и увеличьте ее ширину так, чтобы поместилось еще одно поле.
3. Выбрав элемент управления *Поле* создайте дополнительное поле на подчиненной форме (рис. 2.32.5)

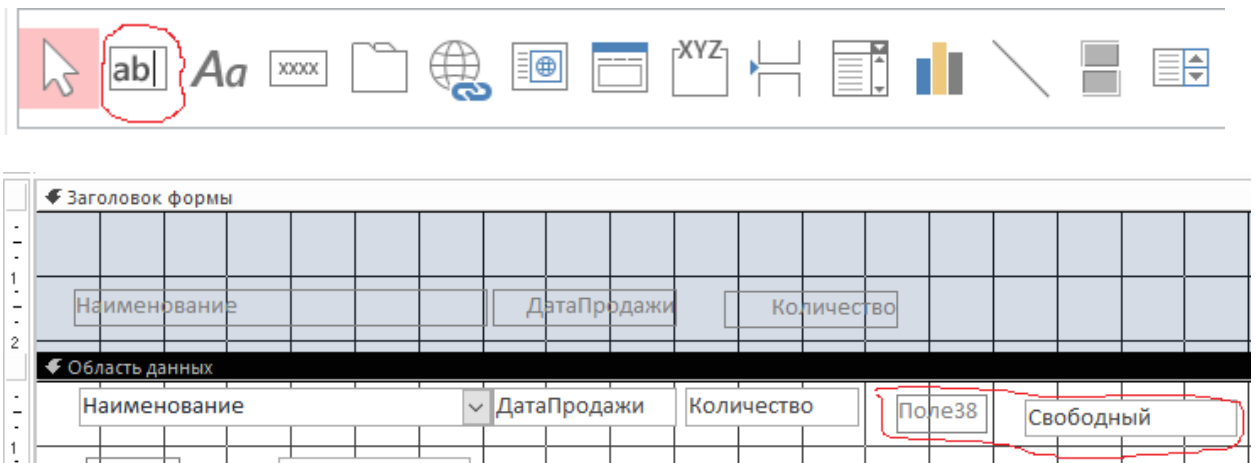


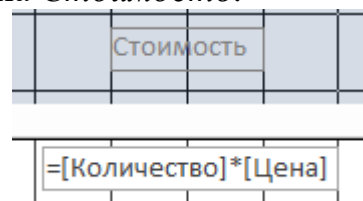
Рис.2.32.5. Создание вычисляемого поля

4. В поле *Свободный* занесите формулу  $=\text{Количество} * \text{Цена}$  (рис.2.32.6)



Рис.2.32.6. Создание вычисляемого поля

5. Выделите поле (Поле38), вырежьте его, перенесите в заголовок, вставьте и измените подпись на *Стоимость*.



6. Перейдите в режим формы и проверьте работоспособность. Теперь в форме против каждой записи о покупке должна высчитываться ее стоимость.
7. Вернитесь в режим **Конструктора**. Добавим в форму подсчет общей стоимости покупок по каждому клиенту.
8. Создайте еще одно поле аналогично предыдущему. В поле *Свободный* запишите формулу  $Sum = ([\text{Количество}] * [\text{Цена}])$  (рис.2.32.7.)
9. Выделите подчиненную форму и в *Окне свойств* посмотрите ее полное название (рис. 2.32.8). Запомните его.
10. Перейдите в *Примечание формы*. Создайте еще одно свободное поле.
11. Заполните его аналогично тому, как показано на рис.2.32.9).

12. Перейдите в режим формы. Должен получиться результат, аналогичный тому, что на рис. 2.32.10
13. **Самостоятельно!** Перейдите в режим *Конструктора* и сделайте поле с вычислением общей стоимости на **подчиненной форме** невидимыми.

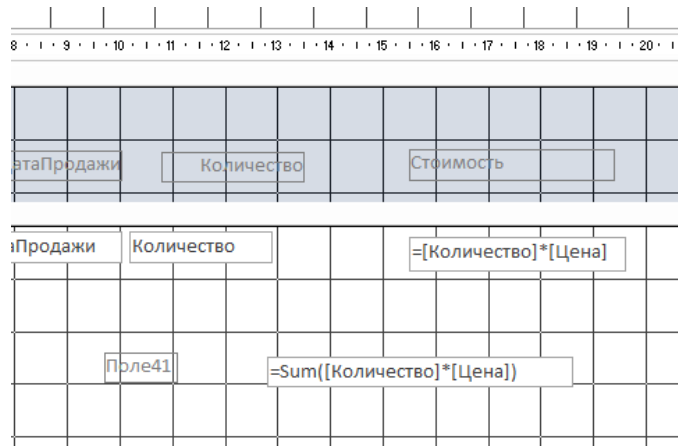


Рис.2.32.7. Создание вычисляемого поля для подсчета общей стоимости

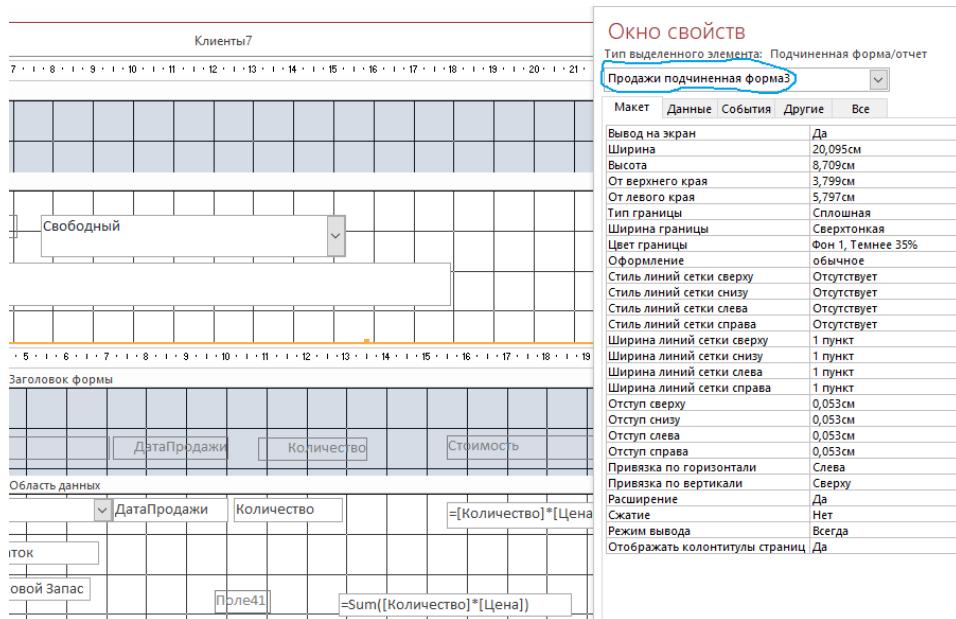


Рис.2.32.8. Полное название подчиненной формы



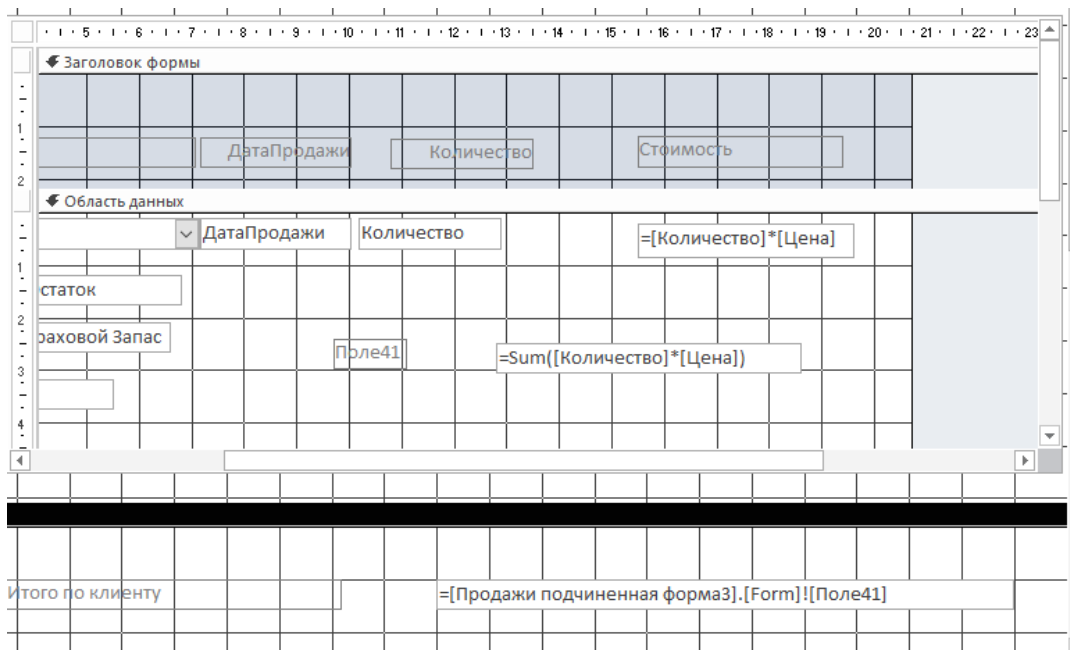
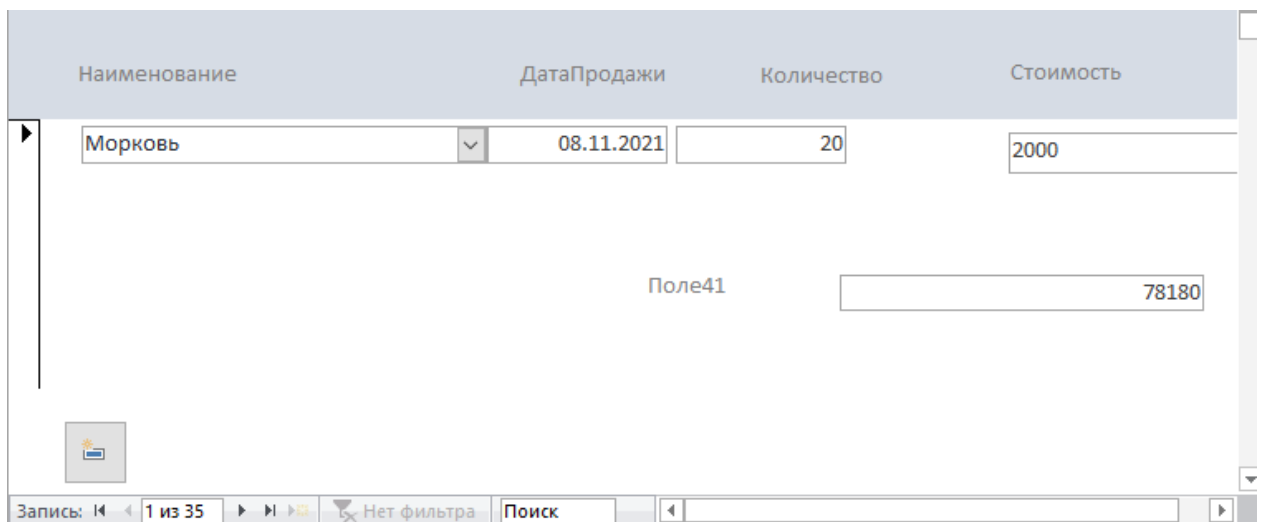


Рис.2.32.9. Перенос формулы в Примечание



Итого по клиенту 78180

Рис.2.32.10. Результат вычислений

### Создание всплывающих форм

Всплывающие формы бывают модальными и немодальными. Создадим немодальную форму.

1. Создайте форму *Рейтинг клиента*. Для этого выберите вкладку **Создание**. На этой вкладке в пункте **Другие формы** выберите *Модальное диалоговое окно*(рис.2.33).

2. Выберите пункт *Добавить поле* вкладки *Конструктор*.

3. Выберите поле *КодКлиента* таблицы *Клиенты*. Дважды щелкните на имени поля. Поле будет помещено на форму.

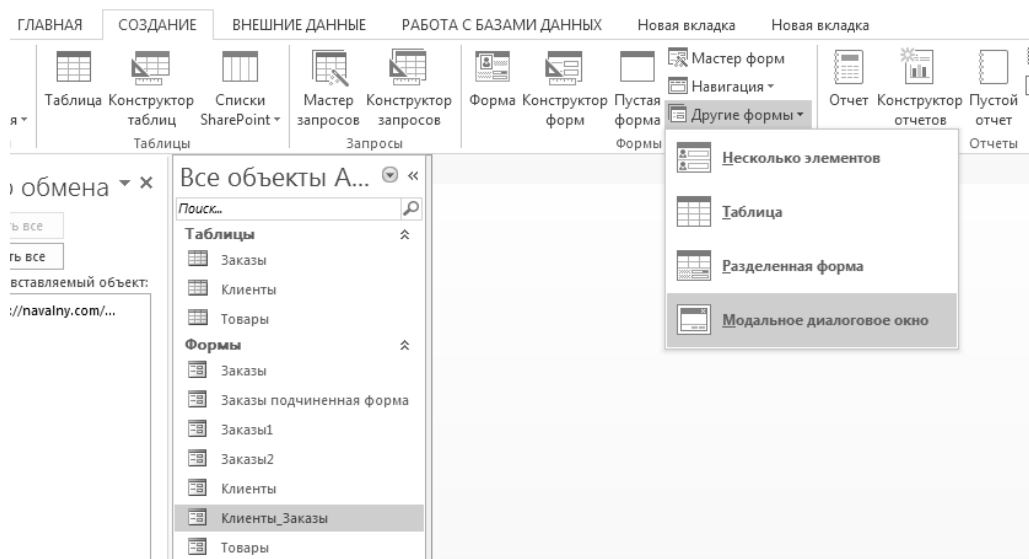


Рис.2.33. Создание модального диалогового окна

4. Аналогично перенесите поля *ИмяКлиента*, *Рейтинг* и *Адрес*. Создайте дизайн формы. Для удобства воспользуйтесь пунктом меню *Сетка* (рис.2.33)



Рис. 2.33. Разработка немодальной формы в режиме Конструктора

5. Сделайте поле *ИмяКлиента* невидимым. Для этого выделите это поле, вызовите контекстное меню и выберите пункт *Свойства*. На вкладке *Макет* измените свойство *Вывод на экран* на «НЕТ»(рис.2.34).

6. Вызовите контекстное меню, щелкнув правой клавишей мыши в области данных. Выберите пункт *Свойства формы*. На вкладке *Другие* установите свойства *Модальное окно*-«НЕТ», *Всплывающее окно* – «ДА».

7. Сохраните форму под названием *Рейтинг\_клиента*, закройте форму

8. Откройте форму *Клиенты\_Продажи* в режиме *Конструктора*.

9. Создайте в форме кнопку для открытия немодального всплывающего окна. Для этого перетащите в нее кнопку из раздела *Элементы управления* вкладки *Конструктор*. Задайте ей действие по открытию формы (см.выше) и в открывшемся окне выберите созданную ранее модальную форму *Рейтинг\_клиента* (рис.2.35). Нажмите *Далее*.

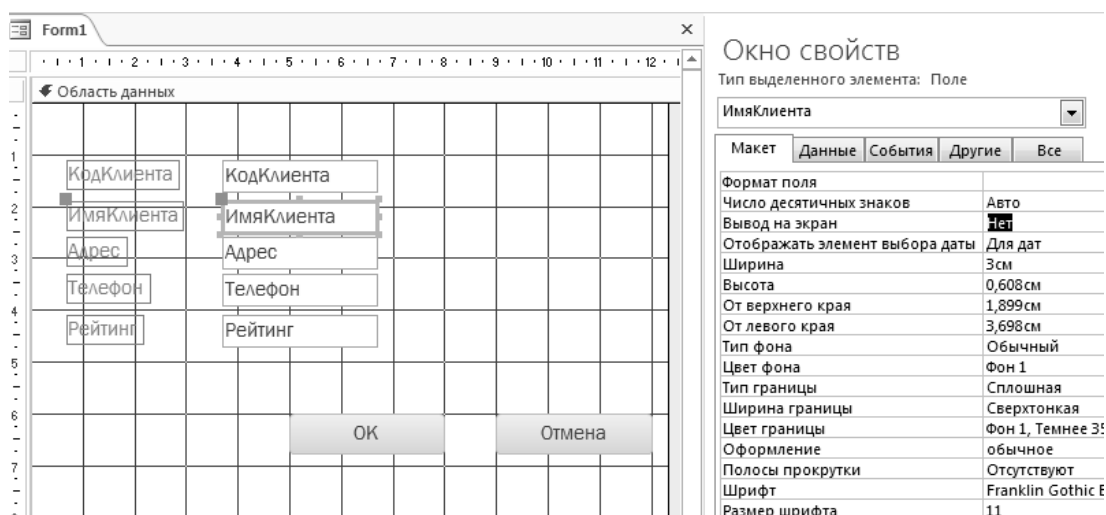


Рис. 2.34. Изменение свойств поля ИмяКлиента

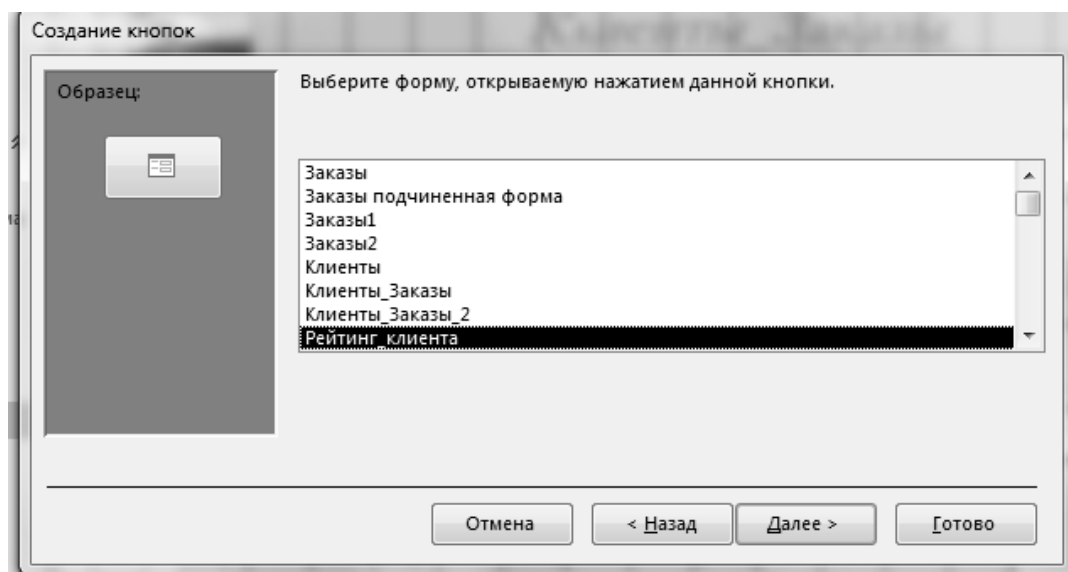



Рис.2.35. Выбор модальной формы

10. В открывшемся окне выберите пункт *Открыть форму для отобранных записей* (рис.2.36). Нажмите *Далее*.

11. В открывшемся окне выберите слева и справа поля *КодКлиента*. Нажмите клавишу . Нажмите *Далее*.

12. Вместо рисунка задайте надпись «*Сведения о клиенте*». Сохраните изменения и откройте форму в режиме просмотра.

13. Проверьте работоспособность формы. (рис.2.38)

14. Аналогично создайте кнопку «Сведения о Поставщике» в форме *Поставщики\_Поставки*.

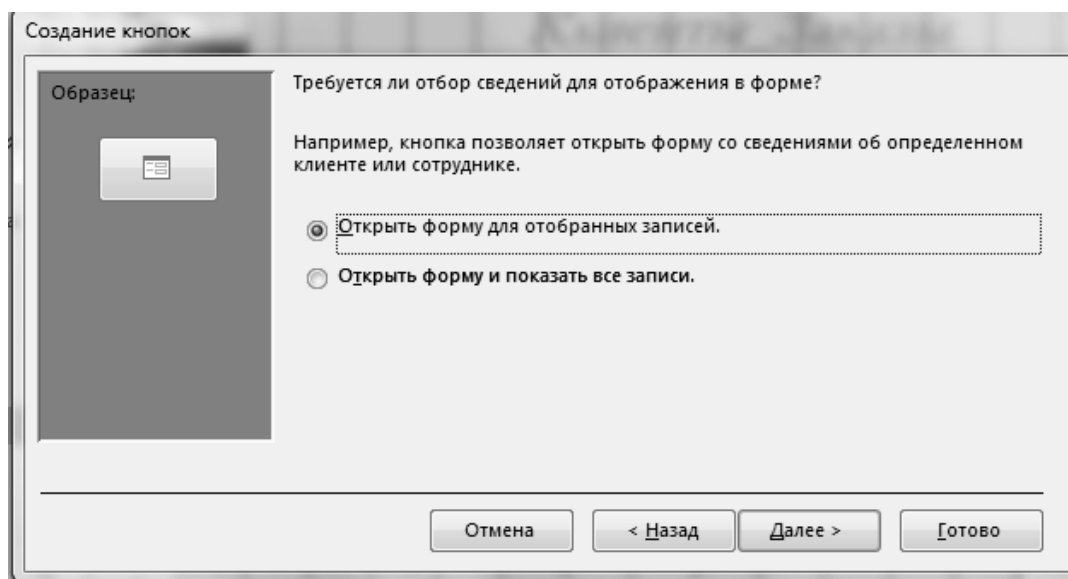


Рис.2.36. Выбор режима отображения записей

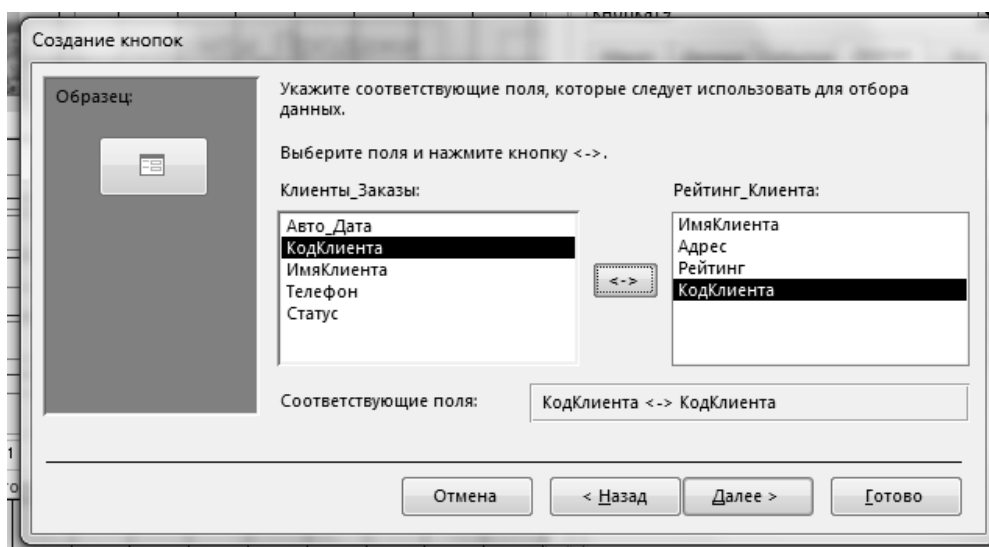


Рис.2.37. Установление соответствия полей

### **Задание списка клиентов**

Для корректной работы форм *Клиенты\_Продажи* и *Поставщики\_Поставки* необходимо предусмотреть возможность выбора соответствующего клиента или поставщика из списка. Это позволит избежать появления некорректных записей в базе. Для этого необходимо изменить разработанные формы следующим образом:

- Откройте форму *Клиенты\_Продажи* в режиме *Конструктора*. Выделите поле *ИмяКлиента* и удалите его.

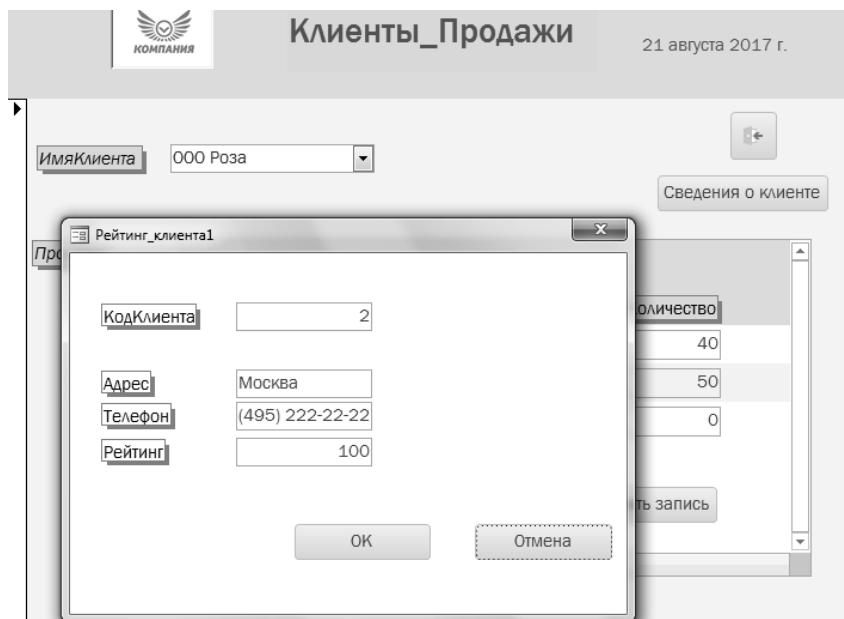


Рис.2.38. Форма в режиме просмотра.

- На вкладке *Конструктор* выберите пункт *Поле со списком*

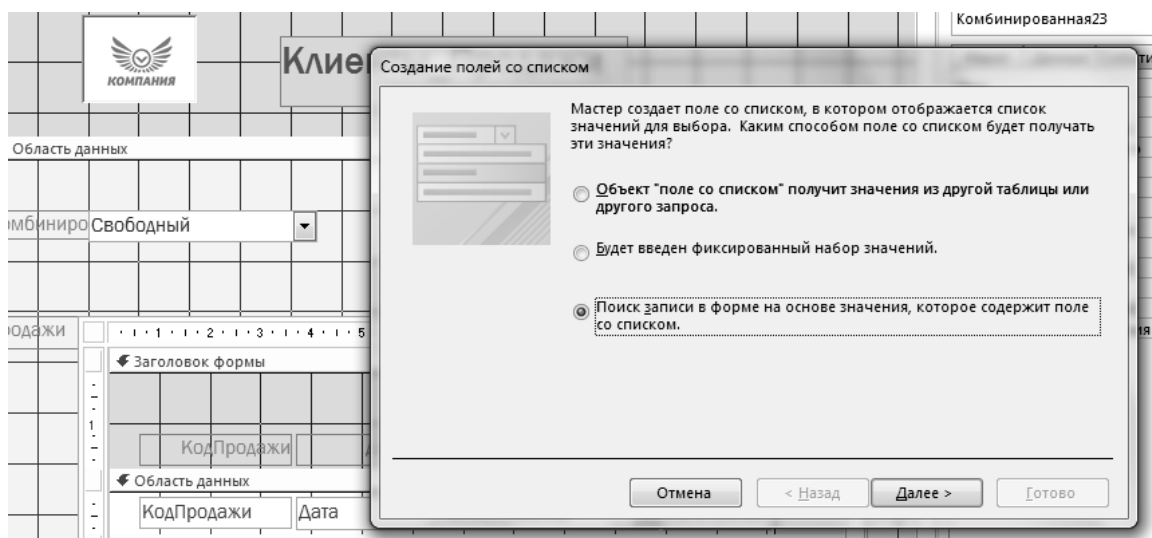


Рис.2.40 Выбор способа получения значений для поля со списком

- В появившемся окне выберите третий способ получения значений для поля со списком (рис. 2.40). Нажмите *Далее*.
- Выберите два поля, как показано на рис.2.41. Нажмите *Далее*.
- Поставьте галочку «Скрыть ключевой столбец».
- Закончите оформление формы и перейдите в Режим просмотра.
- Проверьте работоспособность формы, занеся в нее несколько записей.

**Самостоятельно!** Произведите аналогичные действия с формой *Поставщики\_Поставки*.

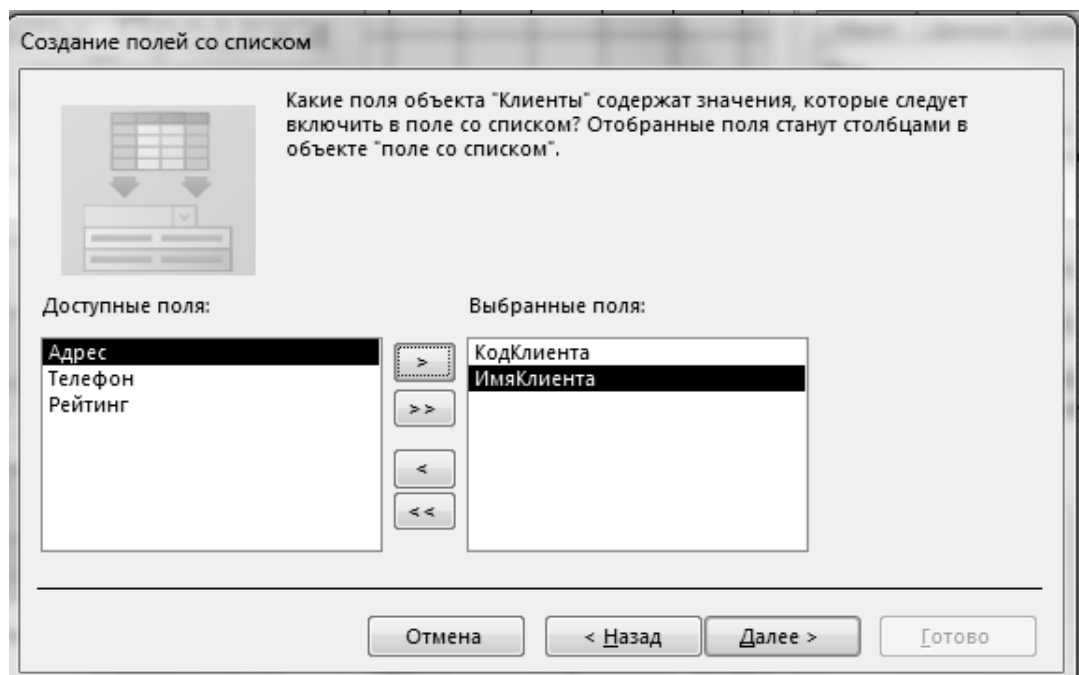


Рис.2.41 Выбор полей для получения знаний для поля со списком

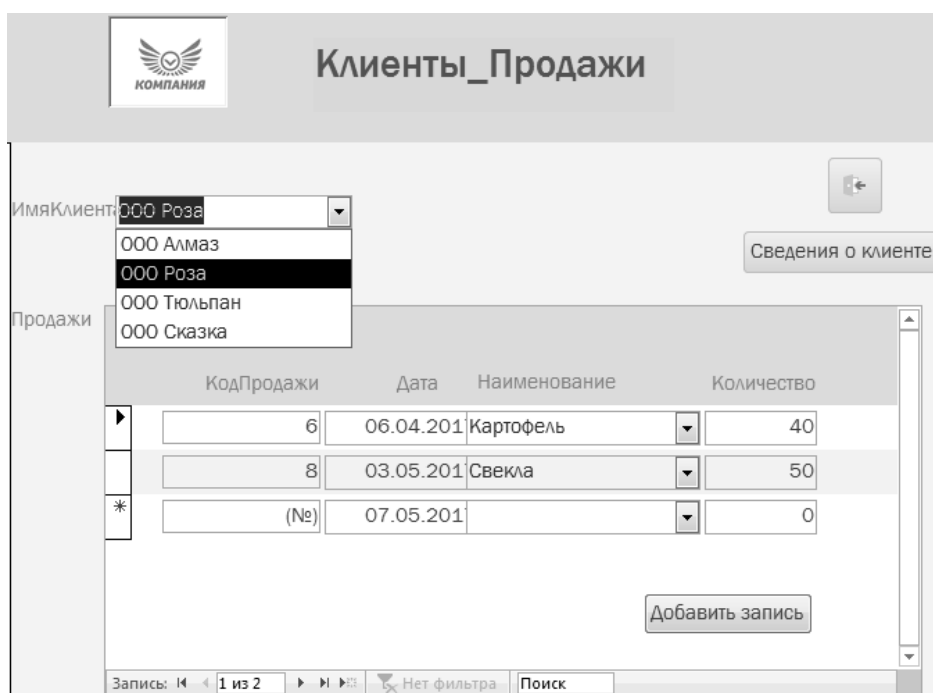


Рис.2.42 Форма с «Поле со списком»

## 2.5. Лабораторная работа 5. Создание отчетов.

Процесс создания отчета необходимо начинать с выбора полей, которые должны входить в отчет, а также определения таблиц или запросов, в которых эти поля находятся. Возможно, что при создании отчета вам могут потребоваться данные какой-нибудь одной таблицы, но чаще всего в нем должна содержаться информация из нескольких таблиц.

Наиболее простым способом, позволяющим быстро решить эту задачу, является использование инструмента автоматического генерирования отчетов. Данное средство позволяет сформировать отчет, включающий в себя все необходимые записи используемой таблицы или запроса, что, в свою очередь, дает возможность быстро просматривать необходимые базовые данные. После этого отчет может быть изменен и сохранен в нужном виде.

Автоматически создать отчет можно следующим образом.

1. Выберите источник записей, на основе которых требуется создать отчет. Для этого в области переходов выделите нужную таблицу или запрос. Пусть это будет таблица *Клиенты*.
2. Нажмите кнопку **Отчет**, расположенную на вкладке **Создание** в группе **Отчеты**.
3. Программа Access автоматически создаст отчет на основе выбранных вами данных и отобразит его в режиме макета (рис.2.63).

КодКлиента	ИмяКлиента	Телефон	Адрес	Рейтинг
1	Иванов	(921)356-45-36	Москва	100
2	Петров	(921)456-33-21	Киев	200
3	Сидоров	(921)567-43-21	Петербург	300
4	Воронин	(921)345-67-89	Кириши	250
5	Козлов	(921)333-77-99	Москва	200
6	Мичурин	(921)345-67-89	Киров	127
7	Соловьев	(921)456-32-17	Киров	500

Рис.2.63. Отчет в режиме макета

4. **Самостоятельно!** Выделите поле *Рейтинг* и задайте для него правило выделения фоном клиентов с рейтингом, меньше заданного, аналогично тому, как вы делали в форме *Клиенты*. Сохраните отчет под именем *Клиенты\_отчет*.

Аналогично создайте отчеты *Товары\_отчет* и *Поставщики\_отчет* по таблицам *Товары* и *Поставщики* соответственно.

5. Создайте отчет на основе запроса *Выбор\_дата\_клиент* (см.лабораторная работа 3, п.19).

## Создание отчета с использованием мастера

1. На вкладке **Создание** в группе **Отчеты** нажмите кнопку **Мастер отчетов**.
2. В появившемся окне **Создание отчетов** выберите запрос **Сумма\_Продаж** а в нем поля **ИмяКлиента**, **ДатаПродажи**, **Наименование**, **Сумма**, **Количество**, **КодПродажи**( именно в этом порядке!), (рис.2.64). Нажмите кнопку **Далее**.

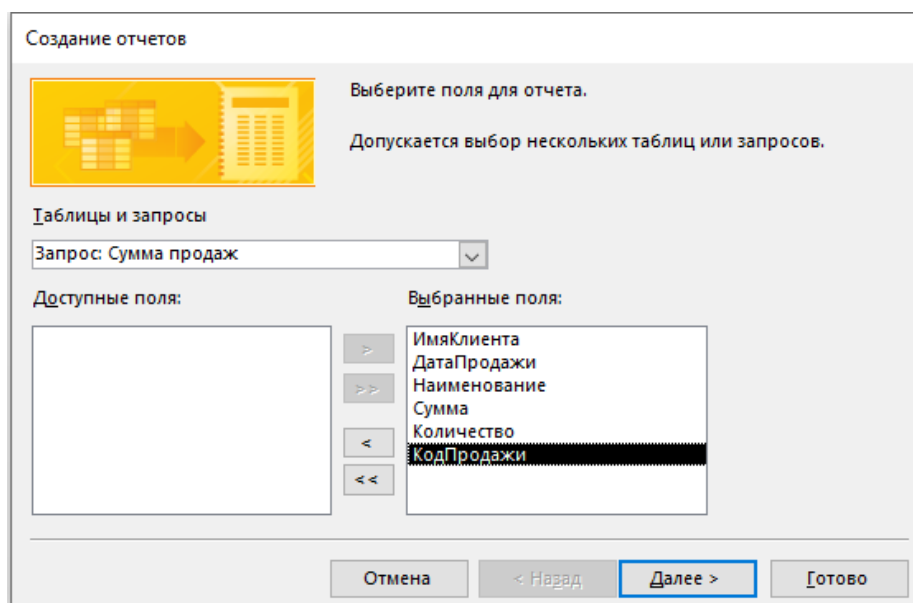


Рис.2.64. Выбор источников для создания отчета

3. Появившееся окно должно иметь вид, представленный на рис.2.65.

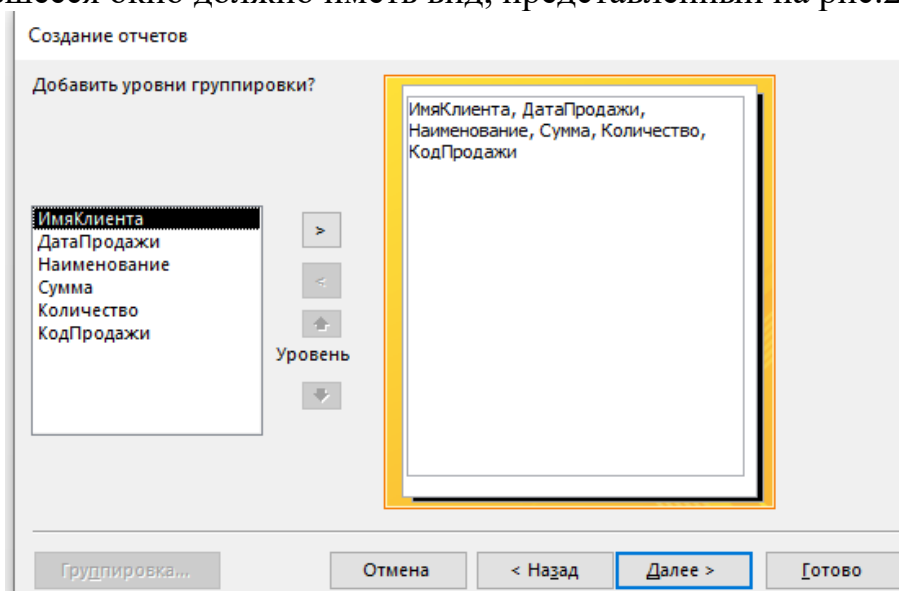


Рис.2.65. Выбор полей для создания отчета

5. В появившемся окне в левой его половине, последовательно дважды щелкните мышкой на полях **ИмяКлиента**, затем **ДатаПродажи**.



6. Результат должен быть аналогичен представленному на рис.2.66.

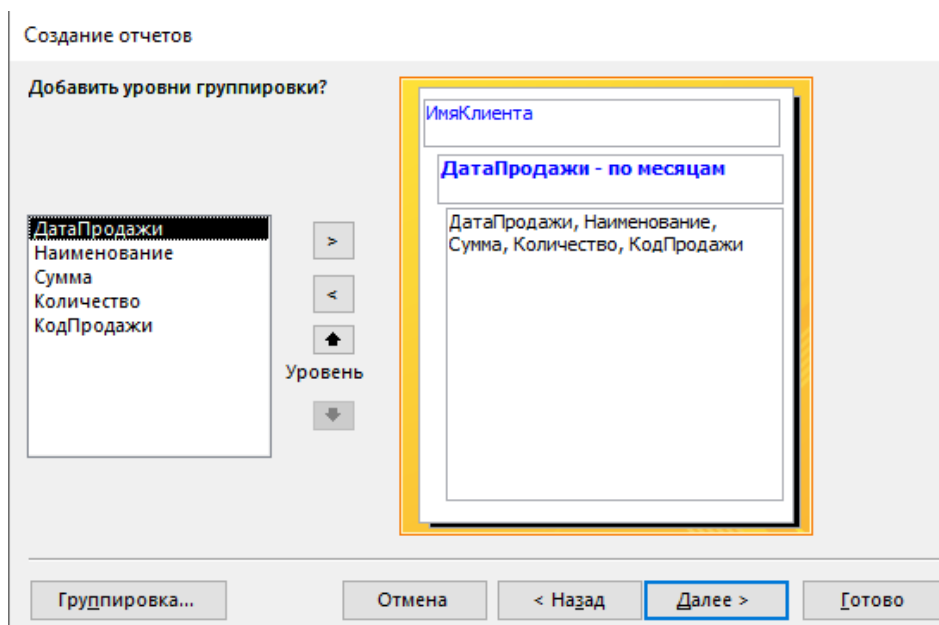


Рис.2.66.

7. Нажмите кнопку *Группировка*.

В открывшемся окне ничего не меняйте. Нажмите ОК (рис.2.67), затем *Далее*.

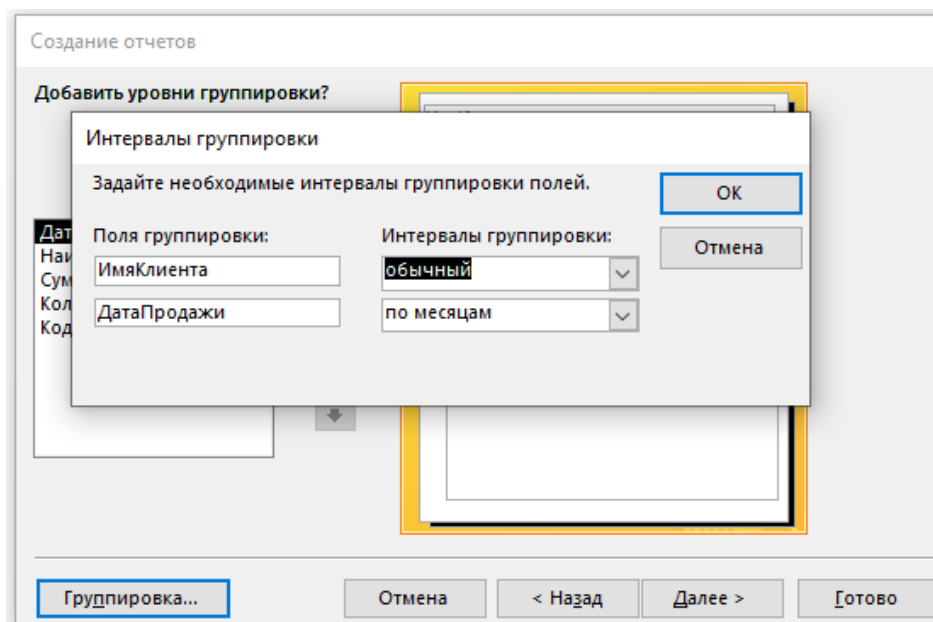


Рис.2.67.

8. Выберите поля *Дата* и *Сумма* для сортировки по возрастанию (рис.2.68).

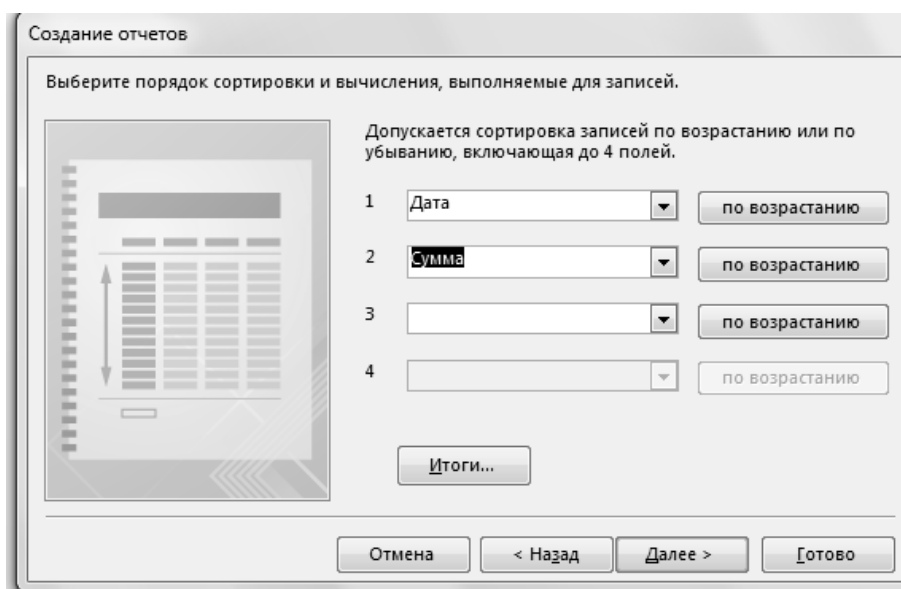


Рис.2.68

9. Нажмите кнопку *Итоги*. В открывшемся окне поставьте галочку в поле *Сумма* для вычисления общей суммы (*Sum*) заказа по месяцам по каждому клиенту (рис.2.69)

10. Нажмите *ОК*, затем *Далее*.

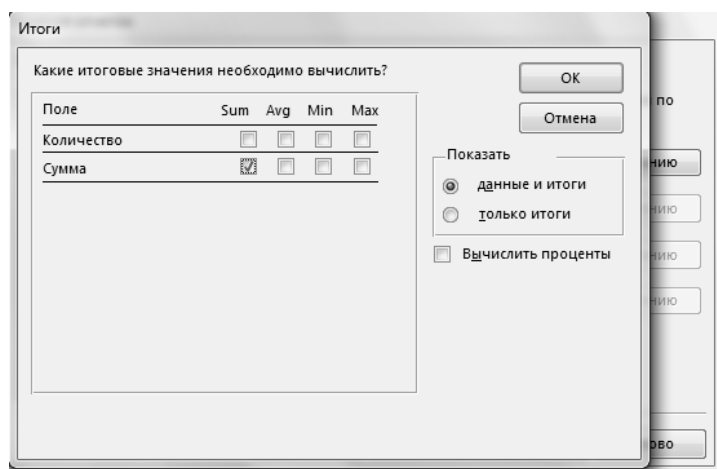


Рис.2.69

11. В открывшемся окне выберите вид макета «ступенчатый» и ориентацию «альбомная». Нажмите *Далее*.

12. В открывшемся окне задайте имя отчету *Продажи отчет*. Нажмите *Готово*.

13. Отчет должен быть аналогичен, представленному на рис.2.70.

Если в отчете значения некоторых полей не помещаются (#####), перейдите в режим **Конструктора** и подкорректируйте размеры поля.

14. Разработайте дизайн отчета.

Сумма продаж7						
ИмяКлиента	ДатаПродажи - по месяцам	ДатаПродажи	Сумма	Наименование	Количество	дПродажи
Анна	Декабрь 2021					
		05.12.2021	800,00 Р	Картофель	10	84
		16.12.2021	20 000,00 Р	Свекла	100	88
		16.12.2021	20 000,00 Р	Свекла	100	86
		16.12.2021	20 000,00 Р	Свекла	100	89
Итоги для 'ДатаПродажи' = 16.12.2021 (4 записей)						
Sum					60800	
Итоги для 'ИмяКлиента' = Анна (4 записей)						
Sum					60800	
Пингвин	Октябрь 2021					
		24.10.2021	800,00 Р	Картофель	10	8
		24.10.2021	2 400,00 Р	Картофель	30	9
		31.10.2021	800,00 Р	Картофель	10	32
		31.10.2021	800,00 Р	Картофель	10	30
Итоги для 'ДатаПродажи' = 31.10.2021 (4 записей)						
Sum					4800	
	Ноябрь 2021					
		08.11.2021	800,00 Р	Картофель	10	35
		08.11.2021	1 000,00 Р	Морковь	10	37
		20.11.2021	800,00 Р	Картофель	10	46
		20.11.2021	1 000,00 Р	Морковь	10	40

Рис.2.70.

15. **Самостоятельно!** Создайте отчеты, в которых содержится следующая информация:

- по каждому клиенту – общий объем продаж ( колич.)(рис. 2.70.а)
- по каждому клиенту – максимальная сумма продажи ( руб) (рис. 2.70.б)
- по каждому товару – общая стоимость продаж ( руб.)(рис. 2.70.в)
- по каждому товару – общий объем продаж (( колич.) (рис. 2.70.г)
- по каждому товару – максимальный объем продажи (колич.) (рис. 2.70.д)

*Аналогичные отчеты необходимо создать по поставщикам.*

- по каждому поставщику- общая стоимость поставок(руб.)
- по каждому поставщику – общий объем поставки( колич.)
- по каждому поставщику – максимальная сумма поставки ( руб)
- по каждому товару – общая стоимость поставок ( руб.)
- по каждому товару – общий объем поставок ( колич.)
- по каждому товару – максимальный объем поставки (колич.)

ИмяКлиента	ДатаПродажи - по месяцам	ДатаПродажи	Наименование	Количество	Сумма	КодПродажи
Анна	Декабрь 2021					
		05.12.2021	Картофель	10	800,00 Р	84
		16.12.2021	Свекла	100	20 000,00 Р	88
		16.12.2021	Свекла	100	20 000,00 Р	86
		16.12.2021	Свекла	100	20 000,00 Р	89
Итоги для 'ДатаПродажи' = 16.12.2021 (4 записей)						
Sum				310		
Итоги для 'ИмяКлиента' = Анна (4 записей)						
Sum				310		

Рис. 2.70.а.

ИмяКлиента	ДатаПродажи - по месяцам	ДатаПродажи	Наименование	Сумма	Количество	КодПродажи
Анна	Декабрь 2021					
		05.12.2021	Картофель	800,00 Р	10	84
		16.12.2021	Свекла	20 000,00 Р	100	88
		16.12.2021	Свекла	20 000,00 Р	100	86
		16.12.2021	Свекла	20 000,00 Р	100	89
Итоги для 'ДатаПродажи' = 16.12.2021 (4 записей)						
Sum					20000	
Итоги для 'ИмяКлиента' = Анна (4 записей)						
Max					20000	

Рис. 2.70.б.

Наименование	ДатаПродажи - по месяцам	ДатаПродажи	ИмяКлиента	Сумма	Количество	КодПродажи
Картофель	Сентябрь 2021					
		26.09.2021	Чайка	800,00 Р	10	5
		26.09.2021	Чайка	800,00 Р	10	3
Итоги для 'ДатаПродажи' = 26.09.2021 (2 записей)						
Sum					1600	
	Октябрь 2021					
		24.10.2021	Чайка	4 000,00 Р	50	10
		24.10.2021	Пингвин	800,00 Р	10	8
		24.10.2021	Пингвин	2 400,00 Р	30	9
		31.10.2021	Чайка	800,00 Р	10	22
		31.10.2021	Чайка	800,00 Р	10	33
		31.10.2021	Чайка	800,00 Р	10	29
		31.10.2021	Чайка	800,00 Р	10	28
		31.10.2021	Чайка	880,00 Р	11	25

Рис. 2.70.в

Наименование	ДатаПродажи - по месяцам	ДатаПродажи	ИмяКлиента	Сумма	Количество	КодПродажи
Картофель	Сентябрь 2021					
		26.09.2021	Чайка	800,00 Р	10	5
		26.09.2021	Чайка	800,00 Р	10	3
Итоги для 'ДатаПродажи' = 26.09.2021 (2 записей)						
Sum					20	
	Октябрь 2021					
		24.10.2021	Чайка	4 000,00 Р	50	10

Рис. 2.70.г.

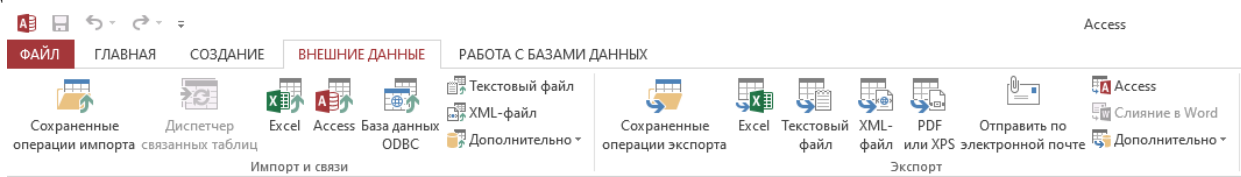
Наименование	ДатаПродажи - по месяцу	ДатаПродажи	ИмяКлиента	Количество	Сумма	КодПродажи
Картофель	Сентябрь 2021					
		26.09.2021	Чайка	25	2 000,00 Р	5
		26.09.2021	Чайка	10	800,00 Р	3
Итоги для 'ДатаПродажи' = 26.09.2021 (2 записей)						
Max				25		

Рис. 2.70.д.

## Экспорт отчета в Word или Excel

При необходимости отчет можно экспортировать в Word или Excel. Это позволит пользователям, не имеющим установленной СУБД Access на своем компьютере увидеть отчеты. Следует отметить, что при экспорте в Word сохраняется оформление страниц отчета (форматирование текста, колонтитулы), а при экспорте в Excel – форматирование текста и уровни группировки данных. Графические объекты (линии, рисунки и т.д.) при экспорте отчетов не сохраняются.

Для выполнения операции экспорта предусмотрена одноименная панель вкладки **Внешние данные**.



1. Выделите на панели объектов Access любой отчет, например *Сумма Продаж*. На вкладке **Внешние данные** в разделе Экспорт выберите пункт Excel.
2. Выберите место назначения файла, отметьте галочкой пункт, как показано на рис. 2.70.ж.

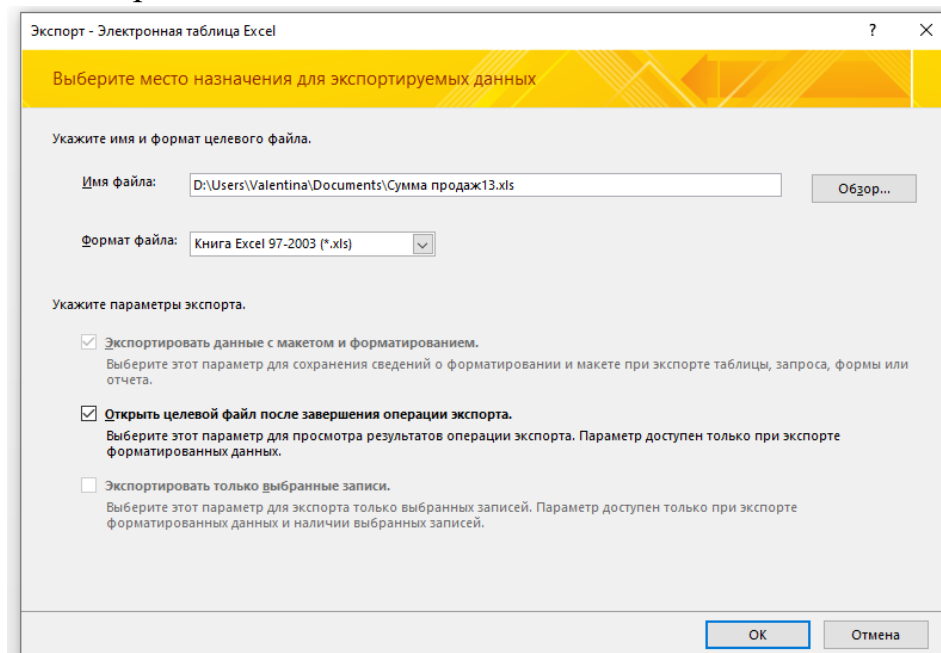


Рис. 2.70.ж.

3. При необходимости откорректируйте внешний вид отчета: поменяйте формат ячеек Сумма на денежный, измените цвет шрифта и т.п. (рис.2.70.з)

1	Наименование	ДатаПродажи - по месяца	ДатаПродажи	ИмяКлиента	Количество	Сумма	КодПродажи
2	Картофель	Сентябрь 2021					
4			26.09.2021 0:00	Чайка	25	2 000,00	5
5			26.09.2021 0:00	Чайка	10	800,00	3
6					Max - Количество: 25		
7		Октябрь 2021					
8			24.10.2021 0:00	Чайка	50	4 000,00	10
9			24.10.2021 0:00	Пингвин	10	800,00	8
10			24.10.2021 0:00	Пингвин	30	2 400,00	9
11			31.10.2021 0:00	Чайка	10	800,00	22
12			31.10.2021 0:00	Чайка	10	800,00	33
13			31.10.2021 0:00	Чайка	10	800,00	29
14			31.10.2021 0:00	Чайка	10	800,00	28
15			31.10.2021 0:00	Чайка	11	880,00	25
16			31.10.2021 0:00	Пингвин	10	800,00	32
17			31.10.2021 0:00	Чайка	10	800,00	23
18			31.10.2021 0:00	Чайка	100	8 000,00	19
19			31.10.2021 0:00	Чайка	10	800,00	18
20			31.10.2021 0:00	Чайка	10	800,00	17
21			31.10.2021 0:00	Чайка	10	800,00	12
22			31.10.2021 0:00	Чайка	20	1 600,00	11
23			31.10.2021 0:00	Чайка	10	800,00	24
24			31.10.2021 0:00	Пингвин	10	800,00	30
25					Max - Количество: 100		
26		Ноябрь 2021					

Рис. 2.70.ж.

## 2.6. Лабораторная работа 6. Создание макросов

Для успешного выполнения следующей лабораторной работы, которая будет заключительной, необходимо рассмотреть еще один вопрос, ранее не рассмотренный, а именно – использование *Макросов*.

Макрос (от слова "макрокоманда") — программа, состоящая из последовательности макрокоманд.

Макрокоманда — это инструкция, ориентированная на выполнение определенного действия над объектами Access и их элементами.

Например, макрокомандой можно открыть форму, отчет, напечатать отчет, запустить на выполнение запрос, применить фильтр, присвоить значение, создать свое меню, организовать выполнение различных ветвей алгоритма в зависимости от условий.

Имеющийся в Access набор макрокоманд (более 50) реализует практически любые действия, которые необходимы для решения задачи.

Макросы бывают *изолированные* и *встроенные (внедренные)*.

*Изолированный макрос* представлен как отдельный объект наряду с другими объектами и он отображается в области навигации в группе *Макросы* (Macros).

*Встроенный макрос* не отображается как объект в группе *Макросы*. Этот макрос *всегда связывается с событием* и сам становится компонентом формы или отчета.

## Изолированные макросы

Изолированные макросы могут запускаться на выполнение прямо из области навигации. Возможно решение задач с помощью ряда взаимосвязанных макросов, первый из которых будет запускаться на выполнение из области навигации. Пользователь запускает главный макрос на выполнение и далее все управление выполнением задачи осуществляется изнутри макроса. Макрос сам открывает нужные объекты, выбирает и обрабатывает данные, вызывает другие макросы, следуя алгоритму, приводящему к решению задачи (вложенные макросы). При необходимости из макроса может быть инициирован диалог с пользователем. Для перехода по различным ветвям макроса используется блок управления Если (If).

Создание *изолированного* макроса.

1. На вкладке *Создание* выберите пункт *Макрос*.

В появившемся окне (рис.2.71.1) обратите внимание на кнопку *Каталог макрокоманд* вкладки *Конструктор* (рис 2.71.2). Для постоянного отображения в окне конструктора макросов списка всех доступных макрокоманд эта кнопка должна быть нажата (подсвечиваться). Кроме того, в каталоге имеется раздел *В этой базе данных*, где представлены не только все макросы из области навигации, но и макросы, внедренные в формы и отчеты. Содержимое этих макросов также может быть скопировано в другие конструируемые макросы.

2. В окне макроса в поле *Добавить новую макрокоманду* в раскрывающемся списке выберите макрокоманду *Открыть отчет*.

3. В открывшемся окне заполните форму (рис.2.71.3)

4. Проверьте работоспособность макроса. Если результат вас устроил, сохраните макрос. Этот макрос должен появиться в области навигации.

5. Запустите макрос из области навигации. Результат должен быть аналогичен п.4.

6. Откройте форму *КлиентыПродажи* в режиме *Конструктора*.

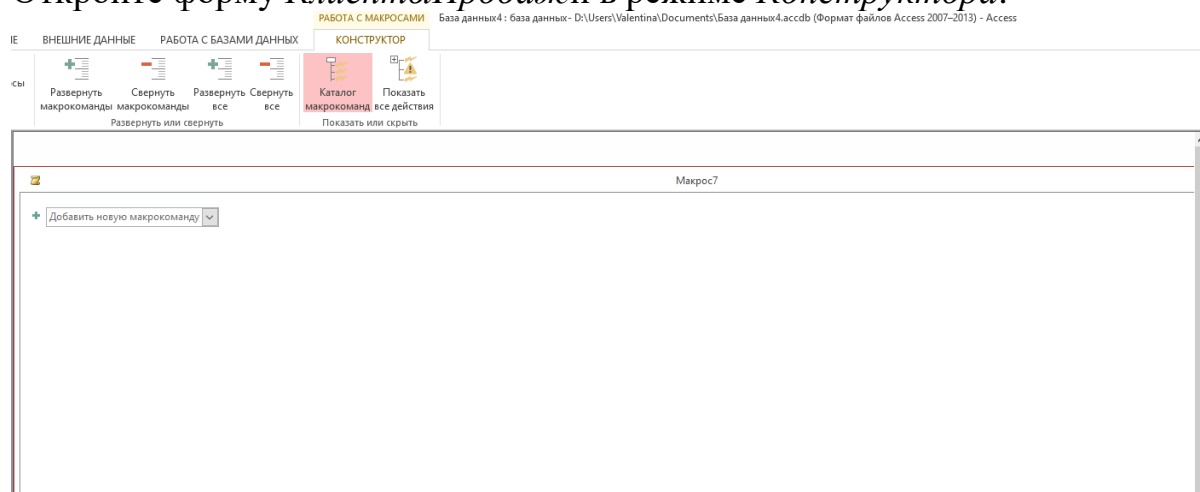


Рис. 2.71.1. Окно создания изолированного макроса

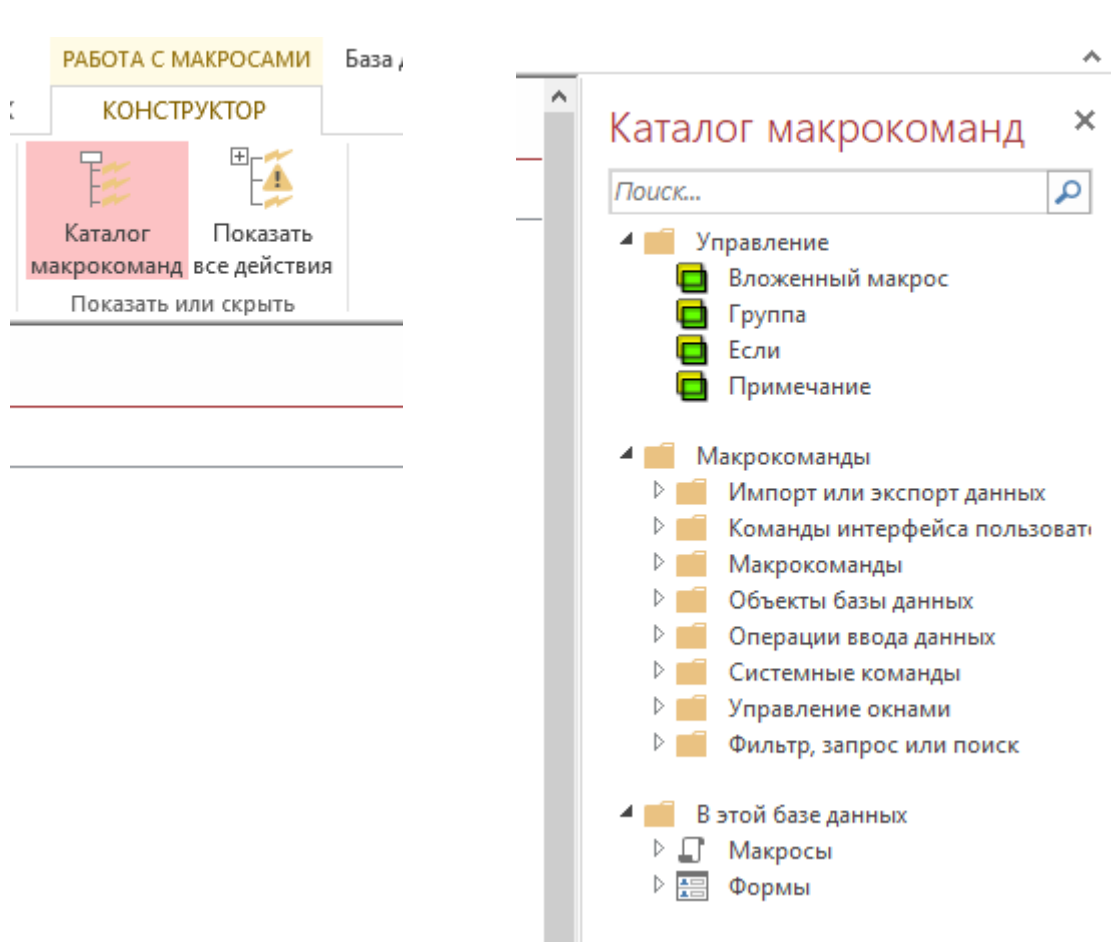


Рис. 2.71.2

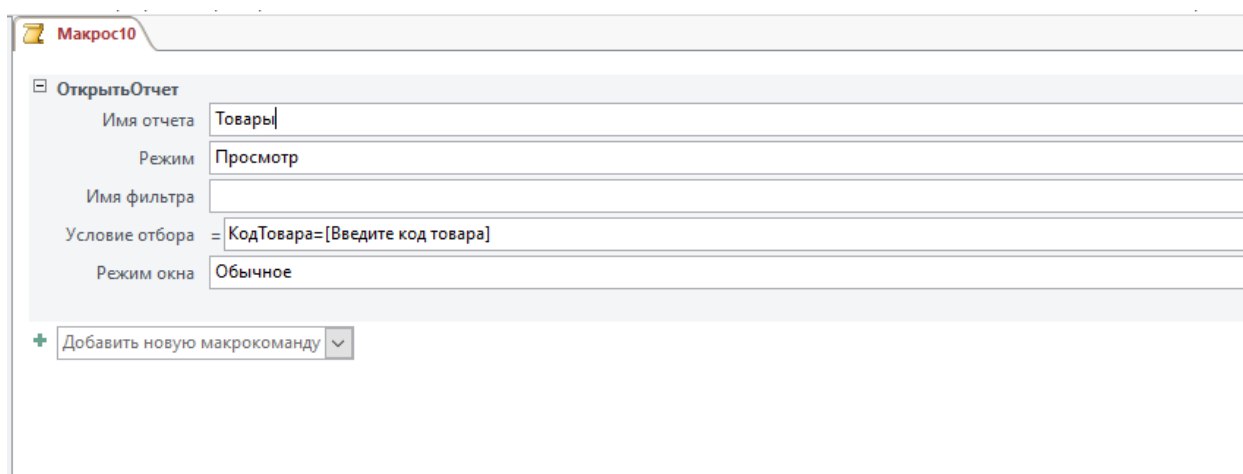


Рис. 2.71.3.

7. Создайте в главной форме кнопку с надписью «Посмотреть товар», связав ее с созданным макросом (рис.2.71.4).
8. Проверьте работоспособность кнопки.



ИмяКлиента

Телефон

Продажи

КодПродажи	Наименование	ДатаПродажи	Количество
37	Морковь	08.11.2021	10

Посмотреть товар

Рис. 2.71.4.

Когда пользователь работает в ленточной форме со списком (товаров, сотрудников, заказчиков и т.п.), то удобно иметь в каждой строке формы кнопку «*Подробнее*», при нажатии на которую будет открываться форма с детальной информацией по выбранному товару (сотруднику, заказчику и т.п.).

1. Создайте простую ленточную форму *ТоварыСправочник* (рис.2.71.5.)

Товары2

### ТоварыСправочник

КодТовара	Наименование
1	Морковь
2	Картофель
3	Свекла
*	(№)

Рис. 2.71.5.

2. Перейдите в режим *Конструктора* и сдвиньте шапку и поля таблицы вправо, чтобы поместилась кнопка «*Подробнее*» (рис.2.71.6).

- Затем на панели *Элементы управления* вкладки *Конструктор* нажмите кнопку *Использовать мастера* и вставьте слева от поля *КодТовара* управляющую кнопку (в область данных, а не в область заголовка!), как показано на рис. 2.71.7.

*Примечание.* Кнопку *Использовать мастера* можно найти, раскрыв список *Дополнительные параметры*:

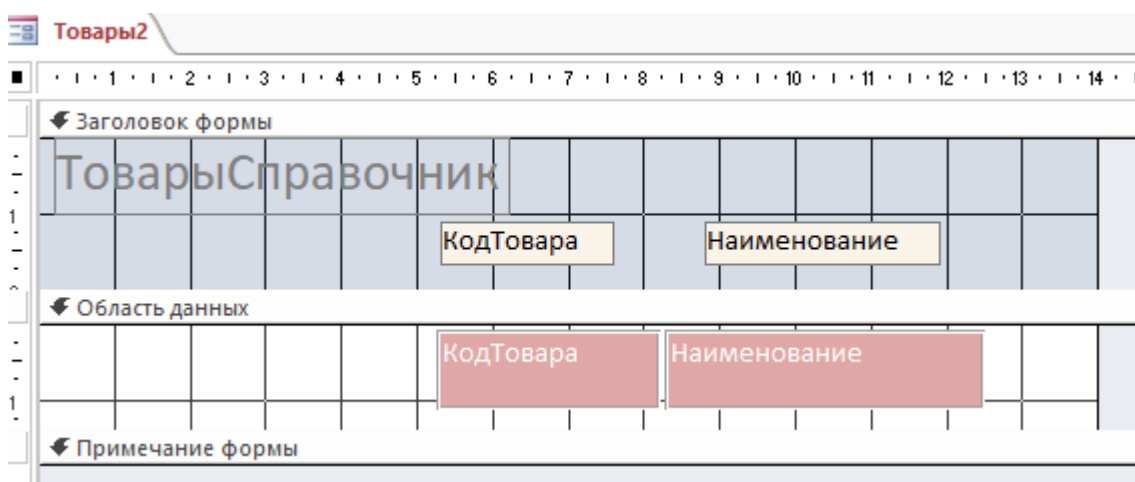
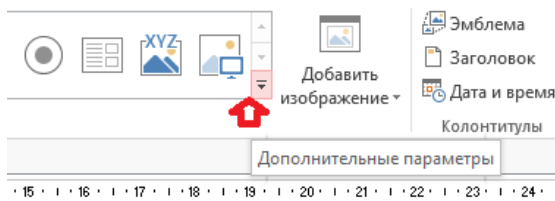


Рис. 2.71.6.

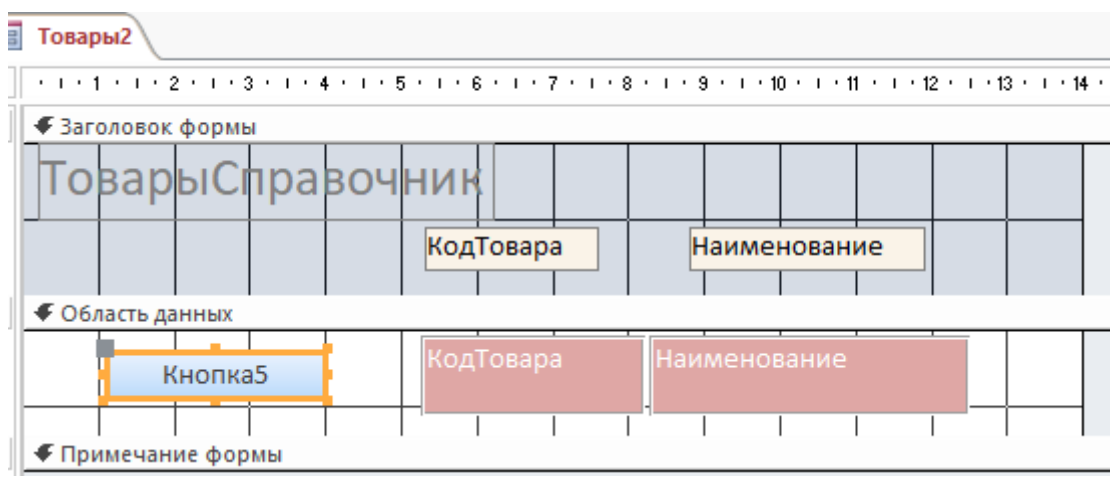


Рис. 2.71.7.

- Переименуйте заголовок кнопки на *«Подробнее»*.
- Щелкните правой кнопкой мыши на кнопке *«Подробнее»* и выберите в контекстном меню команду *Обработка событий* (рис. 2.71.8).

6. В появившемся окне выберите *Макросы* и нажмите ОК. Будет создан пустой макрос.

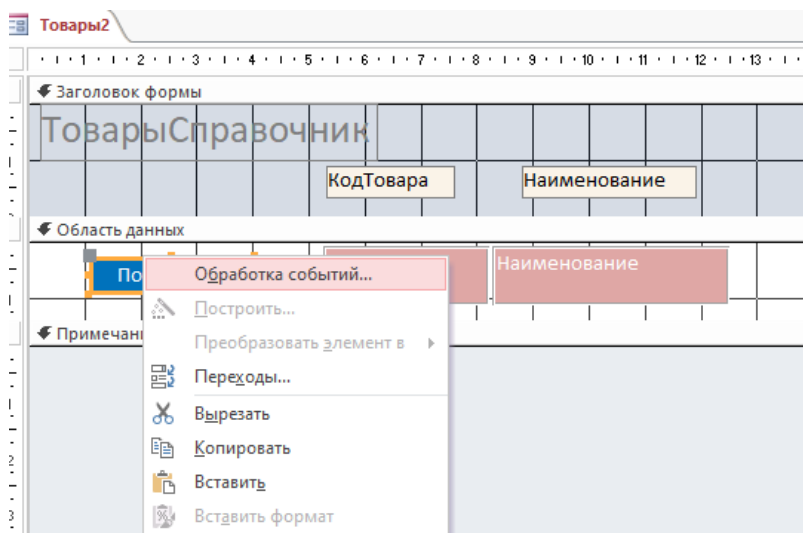


Рис. 2.71.8.

**Примечание.** Этот макрос будет храниться в кнопке *Подробнее*, не отображаясь в списке объектов БД. Его можно открыть для редактирования только из окна свойств кнопки *Подробнее*, нажав кнопку **...** или командой *Обработка событий* (рис. 2.71.8).

7. Так как нам необходимо получить сведения о товаре, для которого нажата кнопка *Подробнее*, необходимо передавать в результирующую форму код этого товара. Чтобы это реализовать создадим временную переменную, например, *CurrentID* и присвоим ей значение *КодТовара*. Для этой цели используйте макрокоманду *ЗадатьВремПеременную*, введите соответствующие значения полей, как показано на рис. 2.71.9

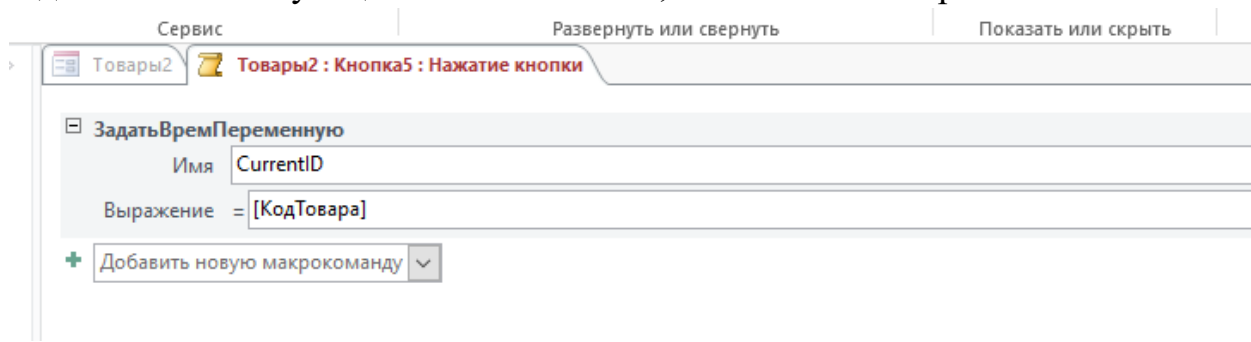


Рис. 2.71.9.

8. Выберите следующую макрокоманду *Открыть форму* (рис.2.71.10). В открывшемся окне выберите форму *Товары*. Параметру *Режим окна* выберите значение *Окно диалога*(чтобы пользователь не мог вернуться в основную форму, не закрыв это окно)

9. В поле *Условие отбора* введите строку `[КодТовара] = [TempVars]!CurrentID`. Это означает, что выбранная форма будет отображать только товары с кодом, равным значению переменной `CurrentID`. При этом, чтобы сослаться на нее, нужно указать название коллекции временных переменных `[TempVars]`.

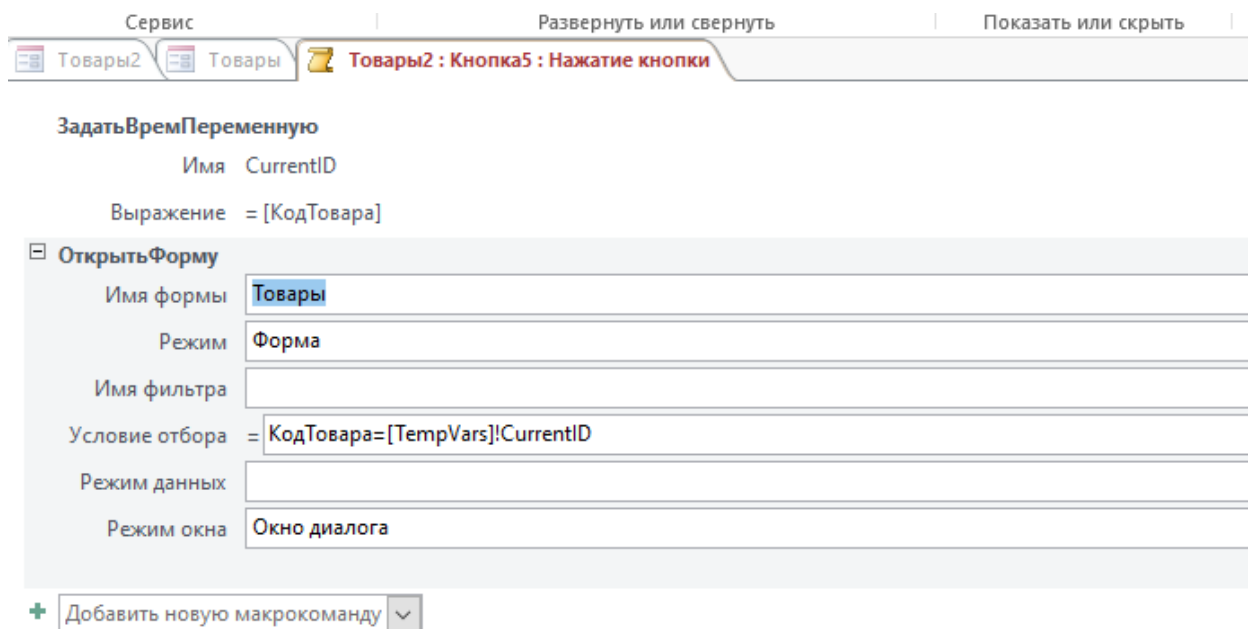


Рис. 2.71.10.

10. Выберите следующую макрокоманду *ОбновитьЗапись* (рис.2.71.11)

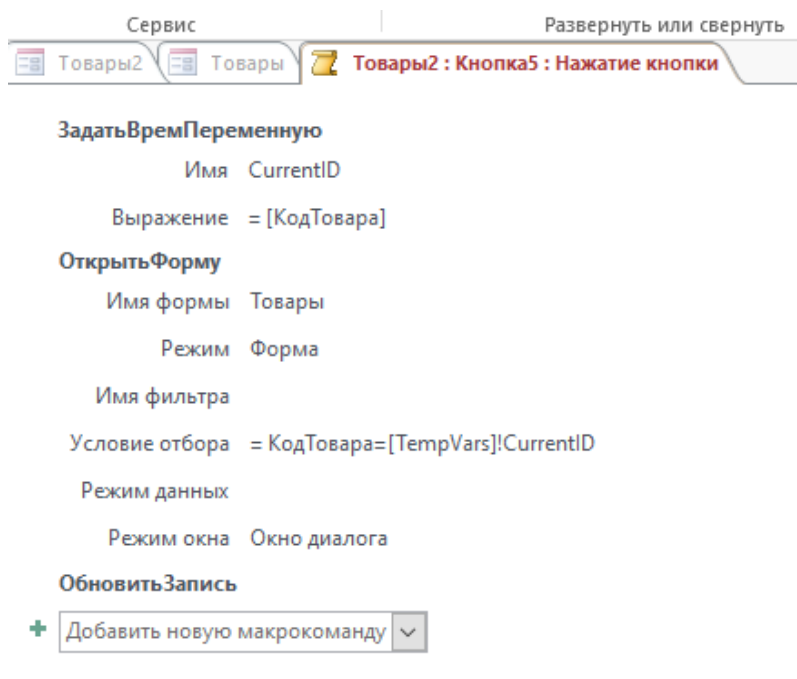


Рис. 2.71.11.

11. Сохраните макрос, перейдите в режим формы и проверьте ее работоспособность. Должно получиться примерно так, как представлено на рис. 2.71.12.

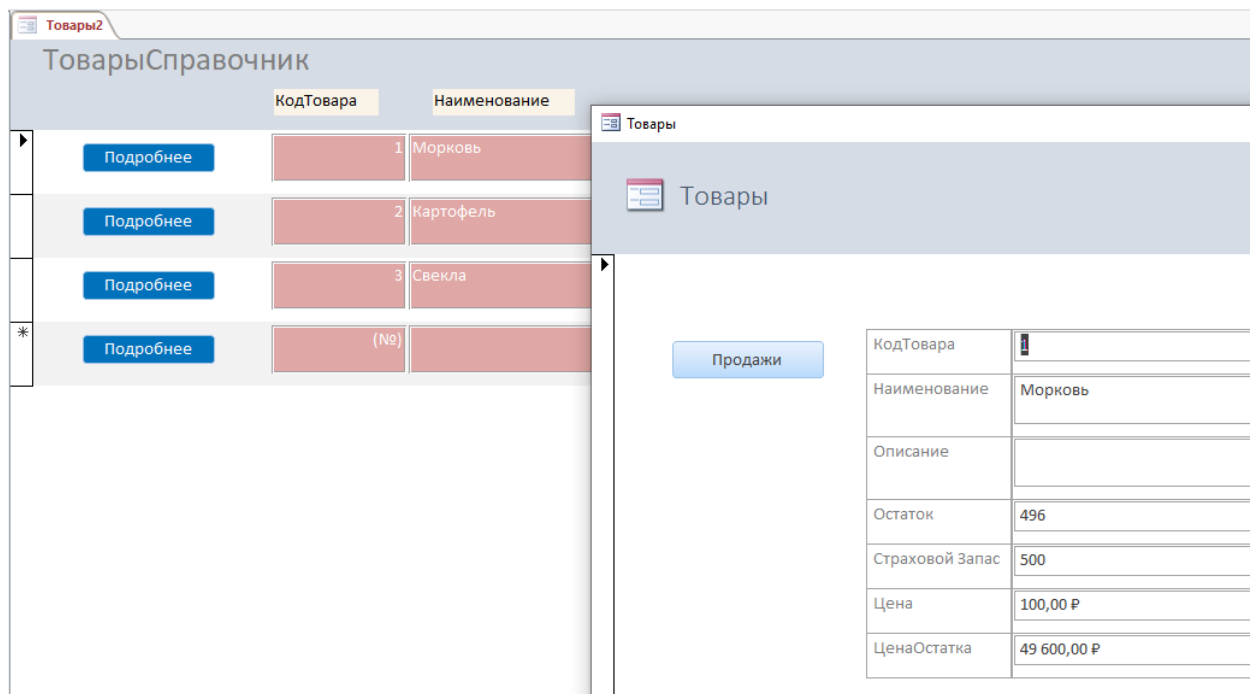


Рис. 2.71.12.

### Создание встроенного макроса

Создание встроенного макроса рассмотрим на примере макроса данных. С помощью этих макросов можно задавать логику БД непосредственно в таблице. Макросы данных связываются с событиями в базовой таблице и выполняются только при изменении, вставке или удалении записи.

Такие макросы позволяют проверять данные или выполнять вычисления. Привязывая логику к данным и сосредотачивая ее в исходных таблицах, макросы данных обеспечивают доступ к ней из объектов приложения.

Так, выполнение операций с записями в формах, построенных на базовых таблицах, инициирует выполнение макросов данных.

Макросы данных, по сути, позволяют реализовать в базе данных функции, которые в серверных базах данных реализуются с помощью особого вида хранимых процедур — триггеров. Триггер также связывается с определенной операцией в базовой таблице и каждый раз при выполнении этой операции вызывается.

Создадим макрос для изменения поля *Остаток* таблицы *Товары*. Мы уже познакомились с несколькими способами, как это можно сделать. Но нам не удалось окончательно избавиться от проблемы многократного повторения

уменьшения/увеличения остатка товаров, так как участвовали в этой процедуре все записи ( в лучшем случае - только на текущую дату). Использование макроса данных позволяет решить эту проблему окончательно.

1. Откройте таблицу *Продажи*
2. На вкладке *Таблица* выберите пункт *После вставки* (рис.2.71.13). Открывается окно макроса.
3. Выберите макрокоманду *Найти запись* (рис.2.71.14).
4. Выберите таблицу *Товары*
5. Заполните поле *Условие отбора*, как показано на рис. рис.2.71.14
6. Щелкните по подписи *Добавить псевдоним*. В поле *Псевдоним* внесите псевдоним изменяемой записи, например «*Изменение*».
7. Выберите макрокоманду *Изменить Запись*. Щелкните по подписи *Добавить псевдоним* и внесите тот же псевдоним.
8. Выберите макрокоманду *ЗадатьПоле*.
9. Заполните поля макрокоманды так, как показано на рис. 2.71.14.

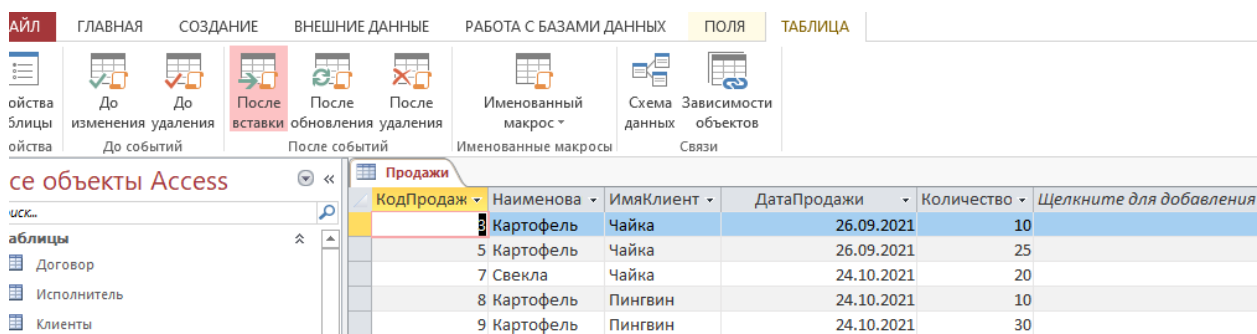


Рис. 2.71.13.

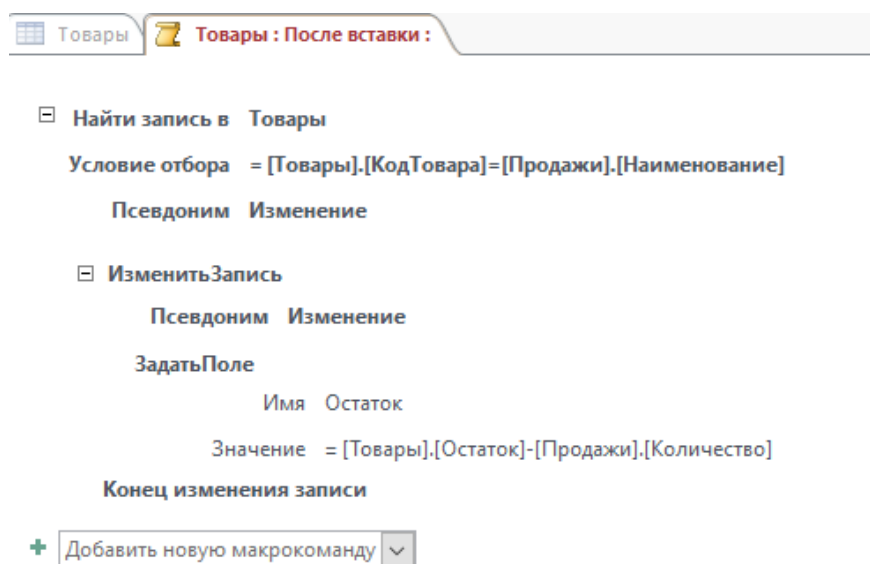


Рис. 2.71.14.

9. Проверьте работоспособность макроса. Для этого занесите запись в таблицу *Продажи* через форму *КлиентыПродажи*. Посмотрите, изменилось ли значение поля *Остаток* в таблице *Товары*.
10. Если все работает правильно, идем дальше.  
**Самостоятельно!** Создайте встроенный макрос на изменение поля *Остаток* в таблице *Товары* при покупке товара. Используйте таблицу *Поставки*

### Создание вложенного макроса

Access предоставляет возможность создавать в макросе вложенные макросы. Вложенный макрос имеет имя и может содержать любые макрокоманды.

Работать с объединенными макросами часто оказывается удобнее, чем с несколькими отдельными макросами. В области навигации макрос с вложенными макросами отображается как один объект. Целесообразно объединять несколько макросов в одном, если они связаны с решением одной задачи или используются при работе с одним объектом

Рассмотрим, как с помощью вложенных макросов, которые запускаются при инициировании пользователем событий в одной форме, можно отфильтровать записи в другой форме.

Пусть необходимо создать макрос, позволяющий при открытии формы *КлиентыПродажи*, при двойном щелчке на поле *Наименование*, выдавать информацию о соответствующем товаре.

1. Откройте форму *КлиентыПродажи* в режиме *Конструктора*
2. Задайте имя изолированному макросу, например, *Сведения*.
3. Выберите макрокоманду *Вложенный макрос*, затем макрокоманду *ОткрытьТаблицу*
4. Выберите имя таблицы – *Товары* (рис.2.71.15)

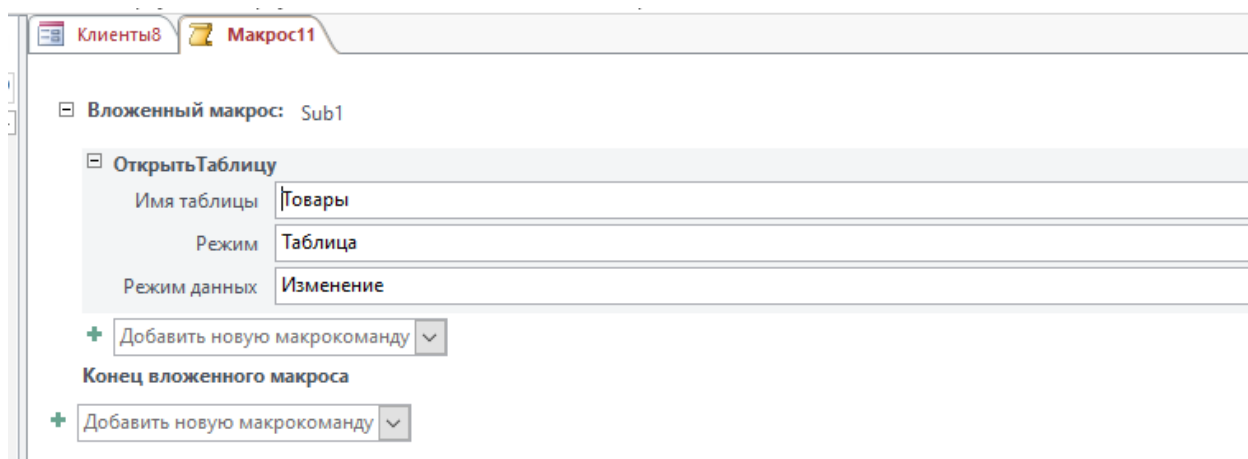


Рис. 2.71.15.

5. Выберите следующую макрокоманду *Вложенный макрос*.

Макрос Sub2 должен выполняться, когда пользователь дважды щелкает на поле *Наименование*.

Для этого (рис.2.71.16):

6. Выберите макрокоманду *ВыделитьОбъект*, которая сделает объект текущим и позволит выполнять для него макрокоманды.
7. В поле *Имя объекта* выберите таблицу *Товары*.
8. Выберите макрокоманду *Показать все записи*.
9. Выберите макрокоманду *Применить фильтр*, которая отфильтрует записи в таблице *Товары*
10. В поле *Условие отбора* с помощью *Построителя*, последовательно выберите из таблицы *Товары* поле *КодТовара*, затем из загруженных форм – **подчиненную форму** и в ней поле *Наименование*:

*Товары*![КодТовара] = *Формы*![Клиенты8]![Продажи подчиненная форма7].Form![Наименование]

11. Сохраните макрос

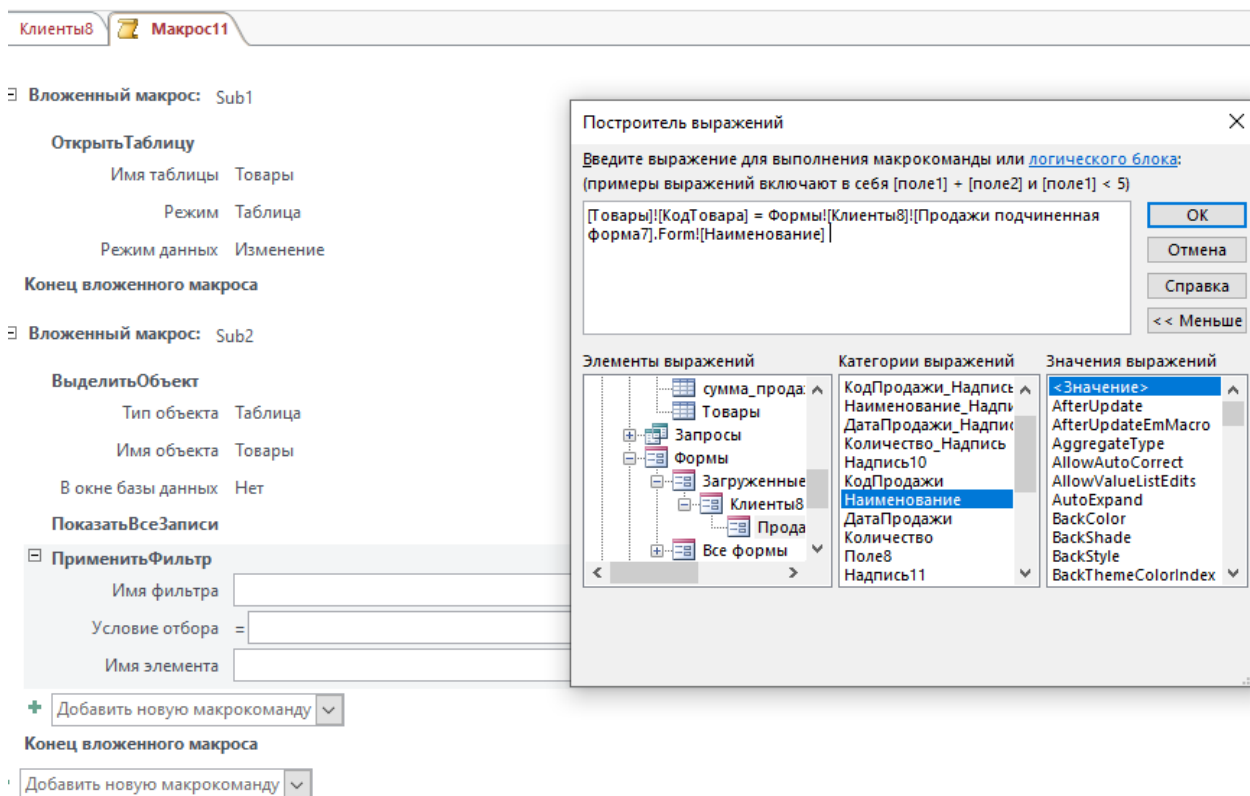


Рис. 2.71.16.

12. Закройте макрос.



13. У вас должна остаться форма *КлиентыПродажи*, открытая в режиме *Конструктора*.

14. Откройте *Свойства формы*. На вкладке *События* в строке *Открытие* выберите макрос *Сведения.Sub1* (рис.2.71.17)

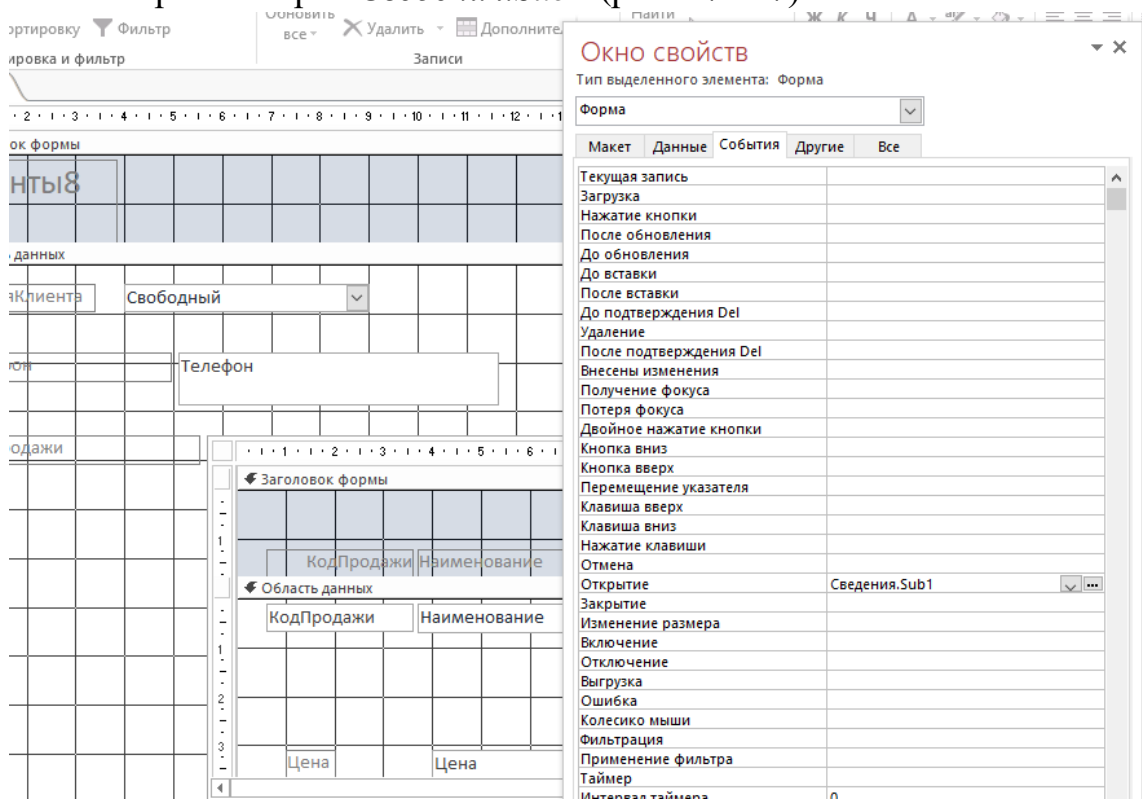


Рис. 2.71.17.

15. Выделите поле *Наименование*. На вкладке *События* в строке *Двойное нажатие кнопки* выберите макрос *Сведения.Sub2* (рис.2.71.18)

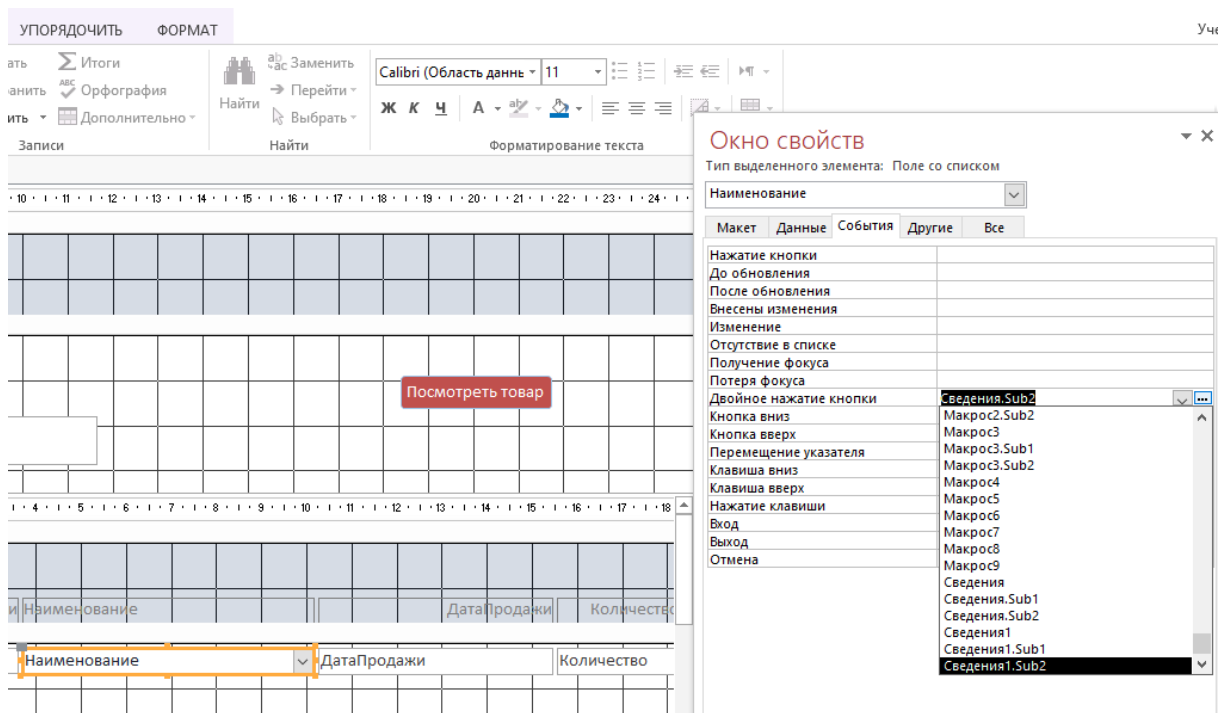


Рис.2.71.18

16. Сохраните макрос и все закройте.

17. Проверьте работоспособность формы *КлиентыПродажи* с макросом.

При двойном щелчке мыши на поле *Наименование* должна появиться строка из таблицы *Товары* для выбранного товара.

18. Если все работает правильно, идем дальше.

Создадим форму для отдела «Склад». Назначение формы – отслеживать наличие товара на складе, и фиксировать, какой товар находится в дефиците. Окончательно форма должна иметь вид, аналогичный представленному на рис.2.91.

1. Откройте форму *Товары* в режиме **Конструктора** и несколько измените ее. Сгруппируйте все поля и передвиньте их к правому краю формы.

2. На вкладке *Макет* свойство *Разрешить режим макета* установите «Нет». (Рис.2.71)

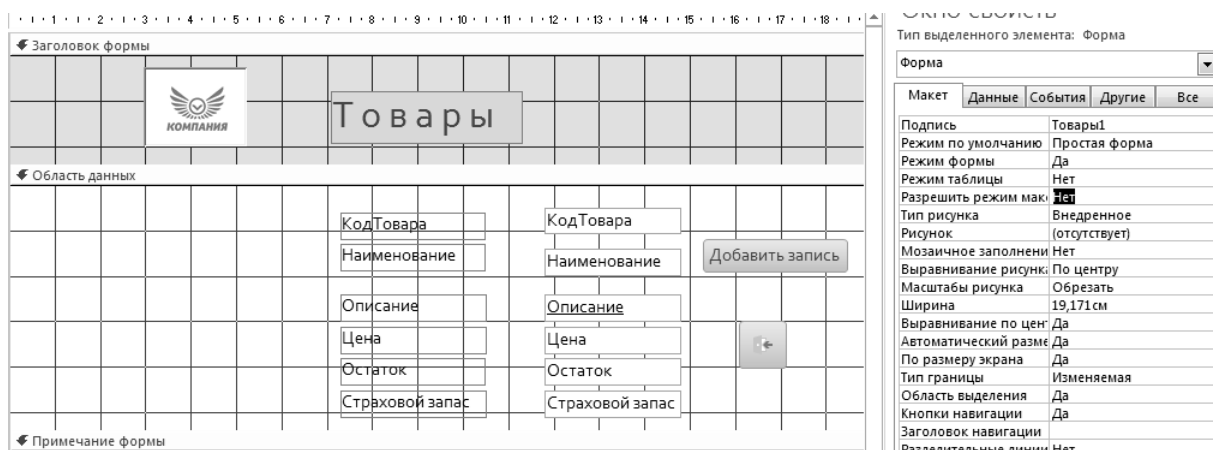


Рис.2.71. Форма «Товары» в режиме Конструктора

3. На вкладке *Конструктор* с помощью пункта *Группа переключателей* обозначьте прямоугольную область в левом верхнем углу формы (рис.2.72).

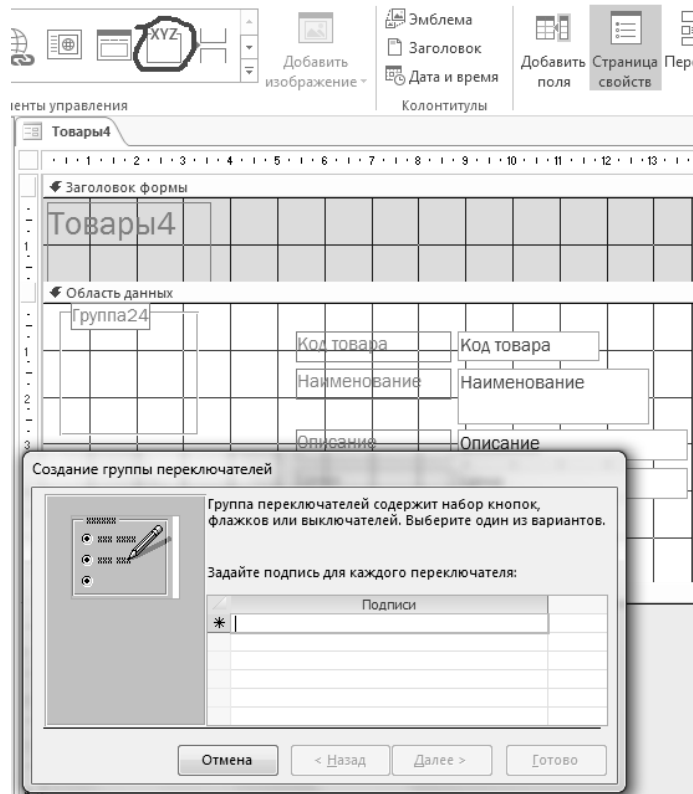


Рис.2.72. Задание группы переключателей

4. В появившемся окне задайте две подписи для двух переключателей (рис.2.73). Нажмите *Далее*.

5. В появившемся окне выберите переключатель «Все товары» по умолчанию. Нажмите *Далее*.

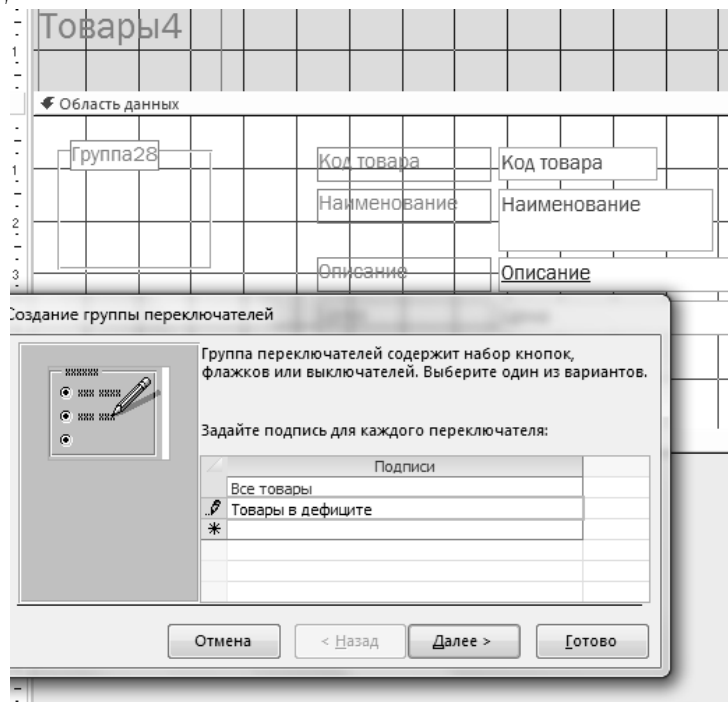


Рис.2.73. Задание подписей для переключателей

6. В появившемся окне ничего не изменяйте (рис.2.74). Нажмите *Далее*.
7. В следующем окне выберите пункт *Сохранить значения для дальнейшего применения*. Нажмите *Далее*.
8. В следующем окне выберите тип элемента управления. Пусть это будет *Переключатели*. Нажмите *Далее*, затем *Готово*.
9. Измените надпись над группой переключателей на *Показать* (Ри.2.75).
10. Выделите группу переключателей. В окне свойств на вкладке *Другие* в строке *Имя* напишите «*Выбор*». (Это имя нужно запомнить!).

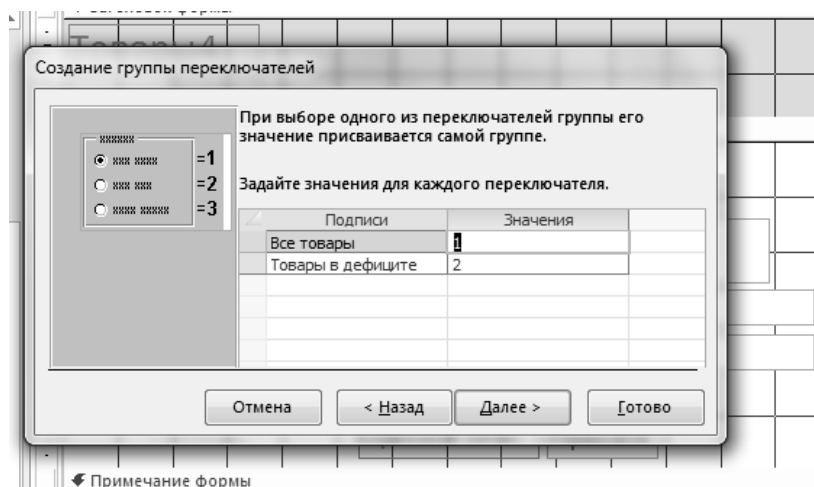


Рис.2.74. Задание значения для переключателей



Рис.2.75. Изменение подписи для группы переключателей

11. Создайте поле со списком. Для этого щелкните на пункте **Список** ( Не **Поле со списком!**) вкладки **Конструктор** и нарисуйте прямоугольник в левом нижнем углу формы (рис.2.76). В появившемся окне нажмите клавишу *Отмена*.

12. Выделите надпись над списком и замените в ней текст на «*Все товары*». Выделите эту надпись шрифтом(рис.2.77).

13. Выделите поле *Список*. В *Окне свойств* (клавиша F4) на вкладке *Другие* задайте имя *Список*.

14. Выделите поле «*Все товары*» и задайте ему в *Окне свойств* имя *Заголовок*. (Запомните, важно!).

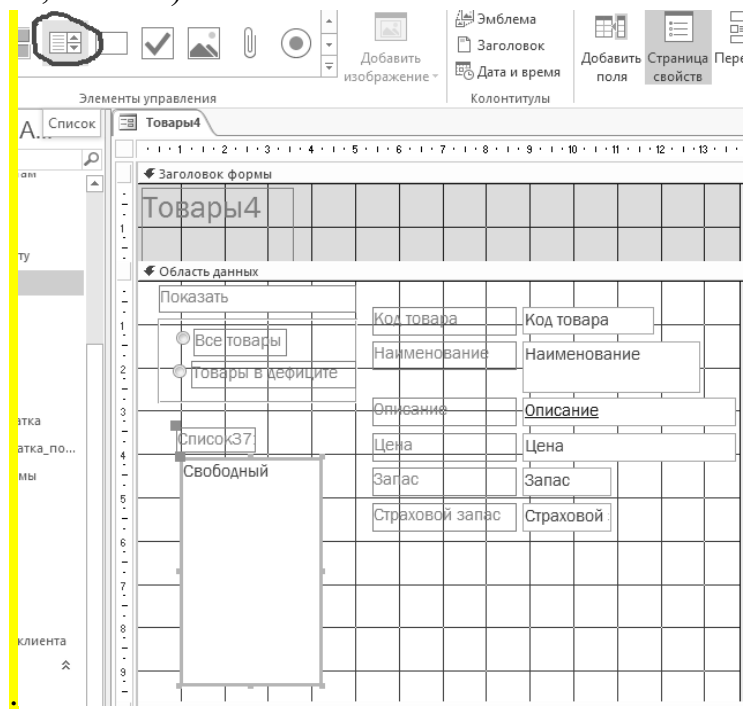


Рис.2.76.Создание поля со списком



Рис.2.77. Изменение надписи над списком

15. Выделите первый переключатель. На вкладке *Данные* проверьте значение параметра. Оно должно быть равно 1. У второго переключателя - равно 2. Если все верно, идем дальше.

16. Програмуем наш переключатель. При значении переключателя равным 1, должен выводиться список всех товаров на складе, при значении 2 – список тех товаров, запас которых меньше страхового запаса.

17. Выделите группу переключателей. В окне свойств на вкладке *События* выберите пункт *После обновления* и нажмите на ... (рис.2.78).

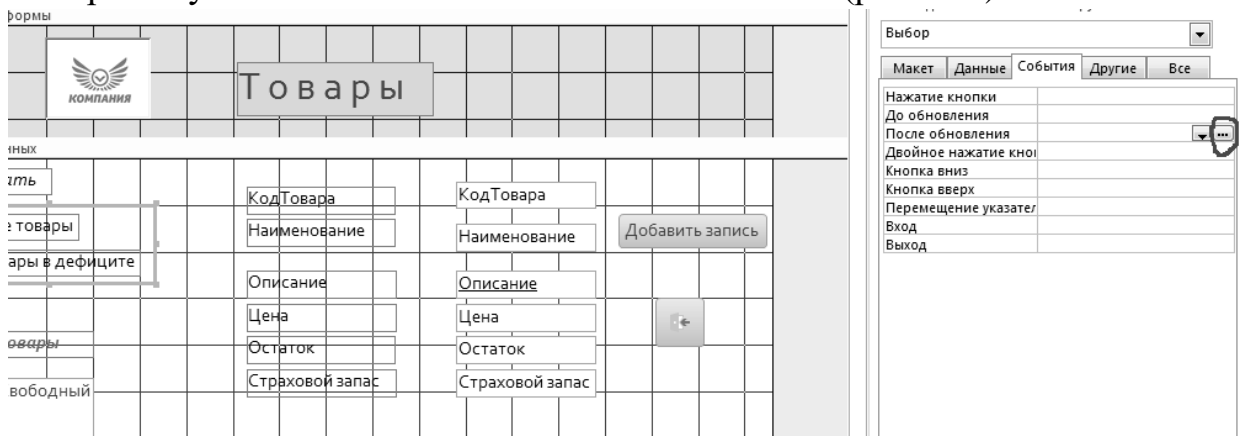


Рис.2.78. Программирование переключателей

18. В появившемся окне выберите *Макросы*. Нажмите *ОК*.

19. В появившемся окне *Макроса* в выпадающем меню выберите макрокоманду *Если* (рис.2.79).

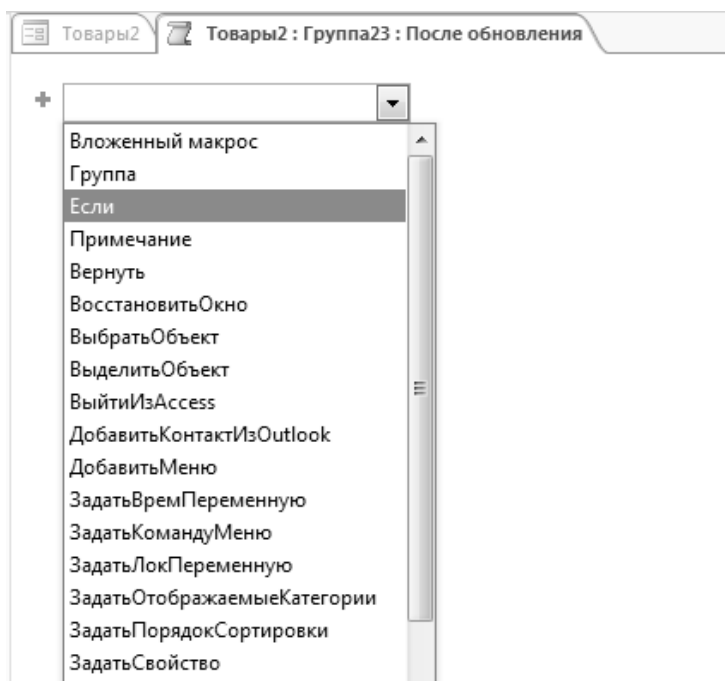


Рис.2.79. Создание макроса

В появившемся поле «*Если*» напишите  $[Выбор] = 1$  (рис.2.80).

20. Нажмите *Tab*. Курсор переходит на *Построитель выражений*. Он нам не нужен. Нажимаем еще раз *Tab* переходим по «**то**»

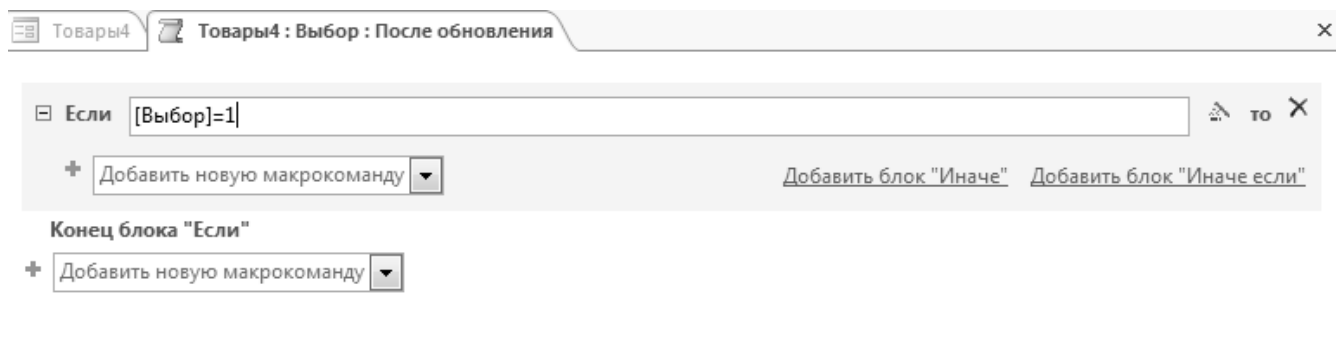


Рис.2.80. Создание макроса (продолжение)

21. В выпадающем меню выбираем макрокоманду *ЗадатьСвойство* (рис.2.81).

22. Заполняем появившуюся форму, как показано на рис.2.81.

23. В следующем окне *Добавить новую макрокоманду* выберите *ПоказатьВсеЗаписи*.

24. Далее добавьте блок «Иначе если»

25. Выделите блок *ЗадатьСвойство* и скопируйте его (рис.2.83).

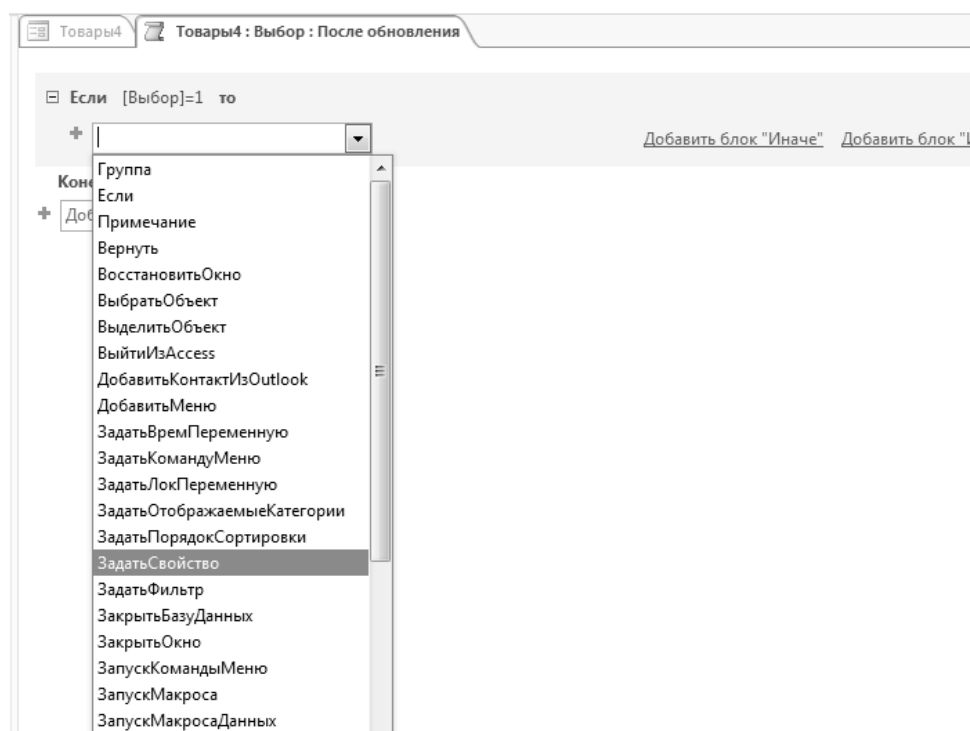


Рис.2.81. Создание макроса (продолжение)

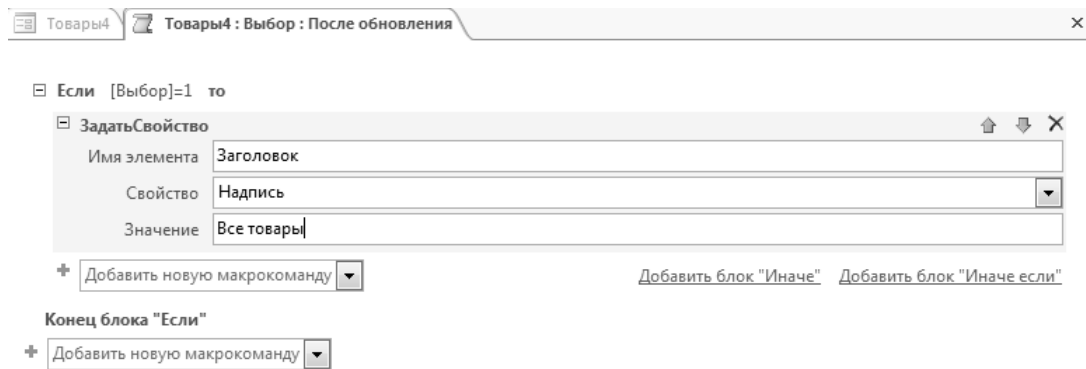


Рис.2.82. Создание макроса (продолжение)

26. Измените поле *Значение* на «Товары в дефиците»
27. Выберите макрокоманду *ЗадатьФильтр*.
28. Заполните поле *Условие отбора* условием  $[Остаток] < [Страховой запас]$  (рис.2.84).
29. За концом блока «Если» добавьте макрокоманду *Обновление* (рис.2.85). Имя элемента *Список*. Закройте окно Макроса.
30. Выделите поле со списком. В окне свойств на вкладке *Данные* в свойстве *Источник строк* выберите .... В появившемся окне добавьте таблицу *Товары*(рис.2.86)

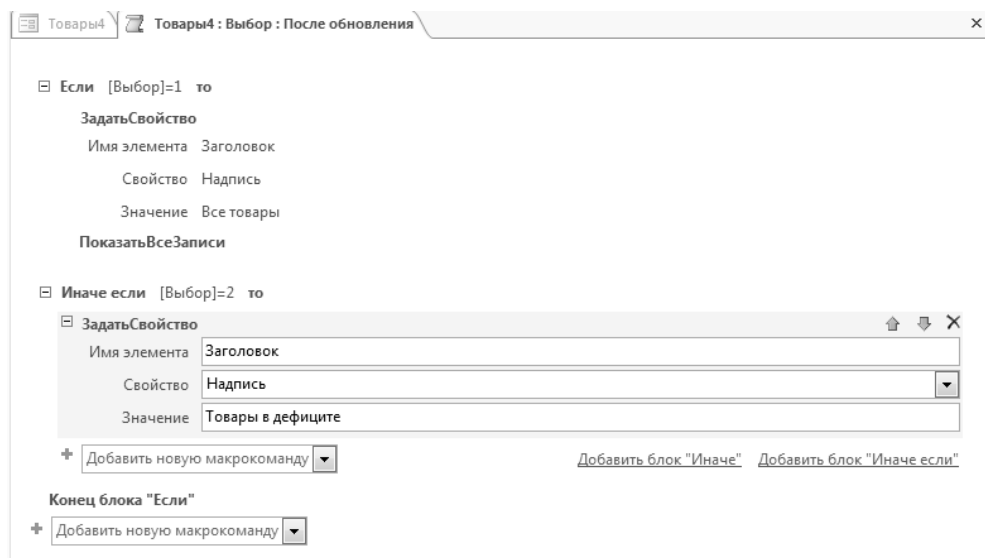


Рис.2.83. Создание макроса (продолжение)

31. Выбираем поля *Код товара*, *Наименование*. Проверяем работоспособность, нажав на ! Если получили список товаров, то все работает нормально. Возвращаемся в *Конструктор*.
32. В следующем поле правой клавишей мыши вызовите построитель выражений. В открывшемся окне выберите *Формы* → *Загруженные формы* → *Товары*.



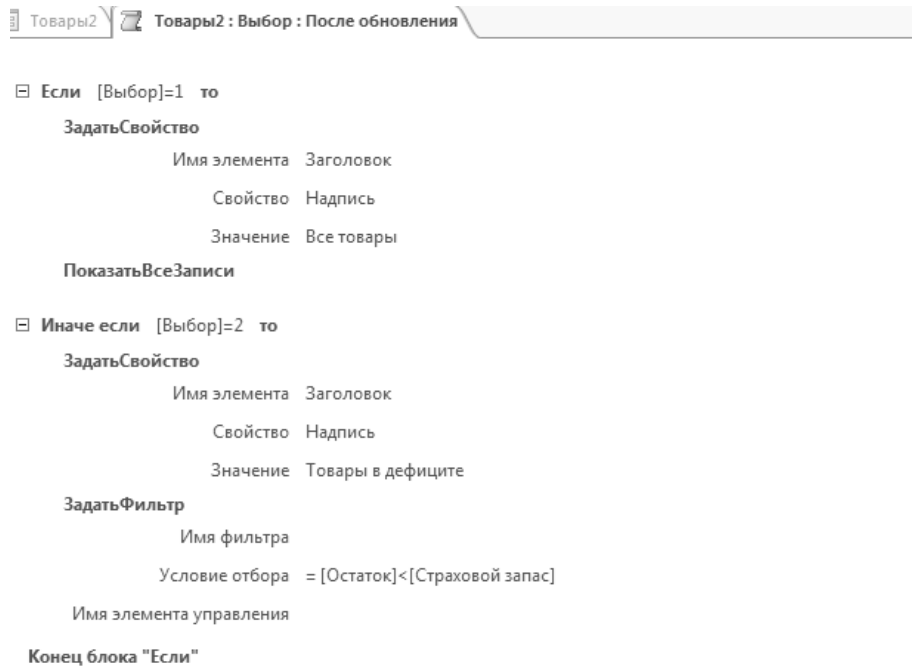


Рис.2.84. Создание макроса (продолжение)

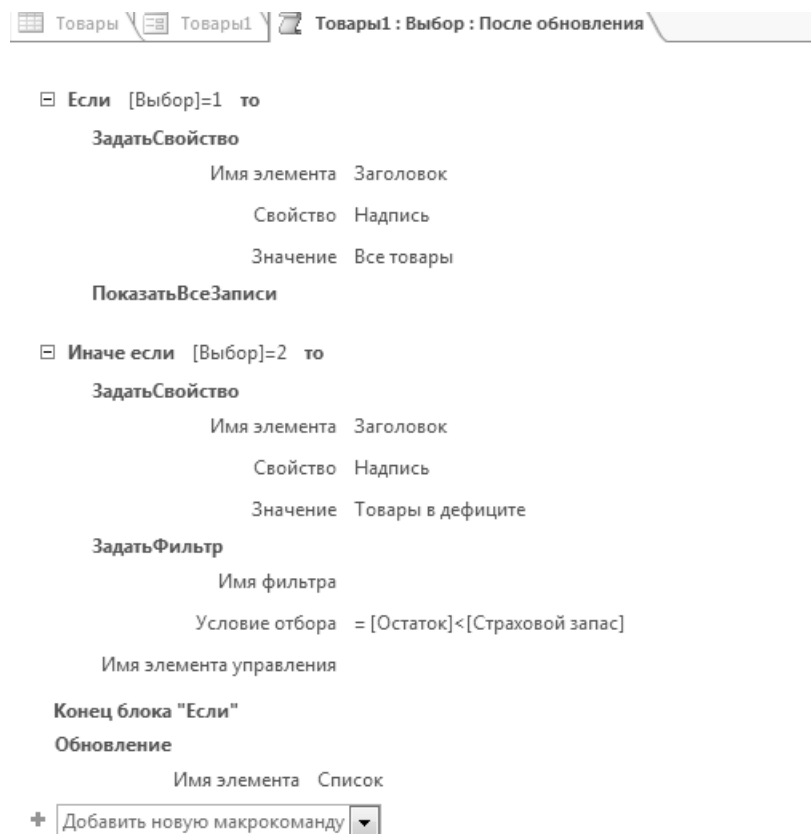


Рис.2.85. Создание макроса (окончание)

33. В поле *Категория выражений* дважды щелкните на поле *Выбор* (рис.2.87). Нажмите ОК.

34. В условиях отбора созданного поля поставьте цифры 1 или 2 (рис.2.88)

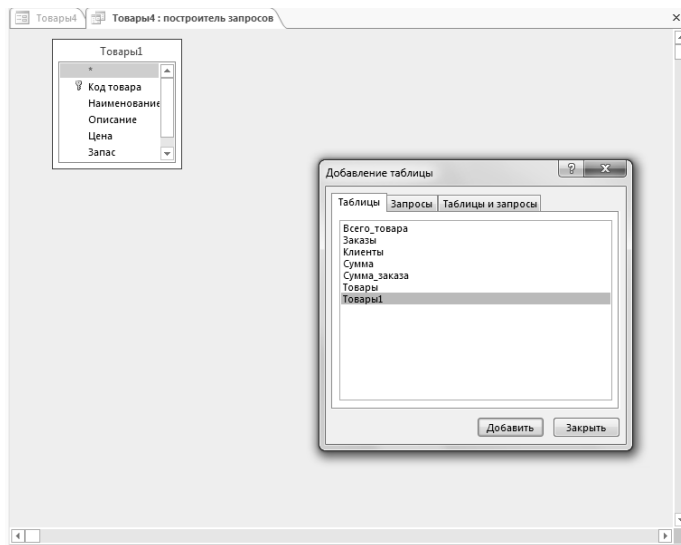


Рис.2.86. Выбор источника строк для поля со списком

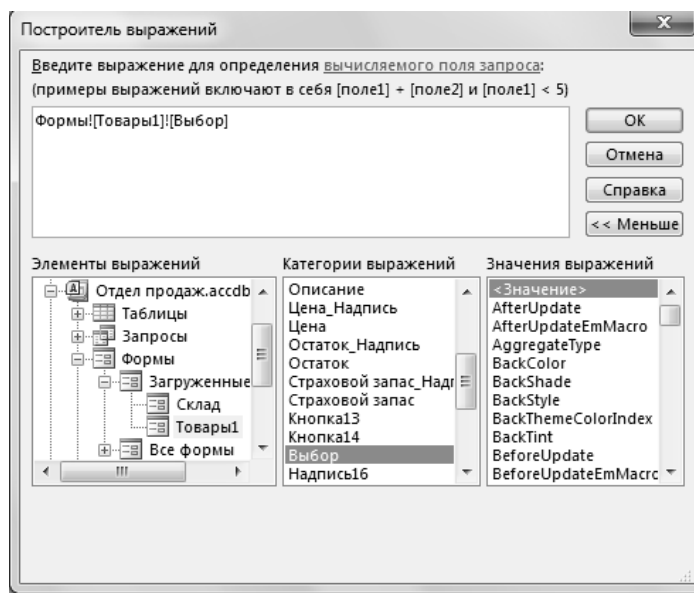


Рис.2.87. Использование «Построителя выражений»

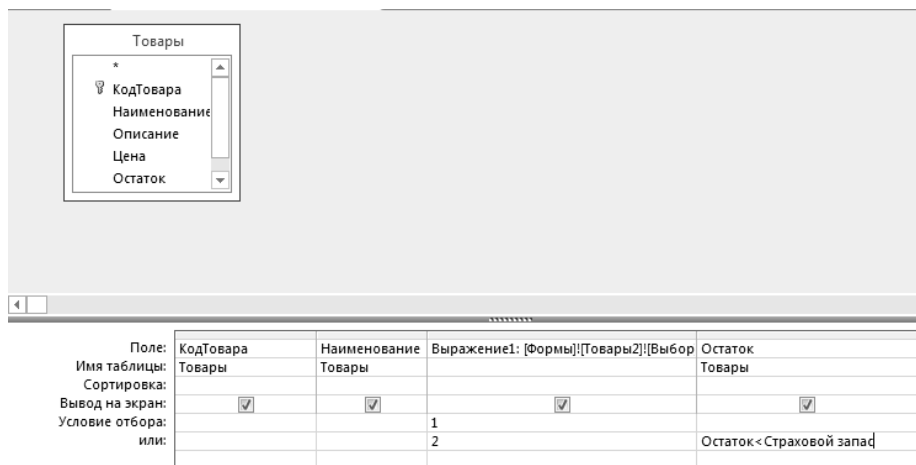


Рис.2.88. Запрос на выборку по условию

35. В следующую графу перенесите поле *Остаток* и введите условие отбора  $[Остаток] < [Страховой\ запас]$  для строки с условием «2» для *Выражения 1*. (рис.2.88). Закройте построитель.

36. Находимся в форме в режиме *Конструктора*. Выделяем поле со списком. В *Окне свойств* на вкладке *Макет* значение свойства *Число столбцов* выбираем равное 2, а *Ширину столбца*, равную 0.

37. Переходим в режим формы. Должно получиться так, как показано на рис 2.89.

38. Если все так, идем дальше.

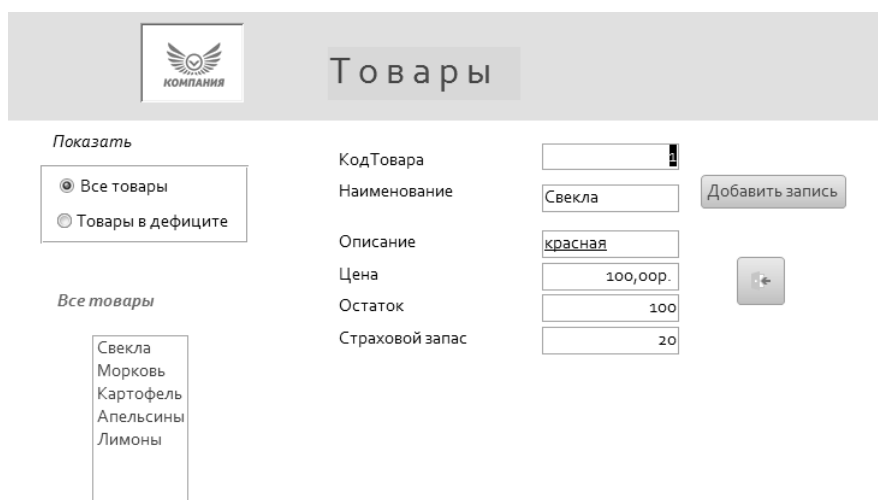


Рис.2.89. Форма с макросом для переключателей

39. Снова переходим в режим *Конструктора*. Выделяем поле со списком и в *Окне свойств* на вкладке *События* выбираем свойство *После обновления*. Щелкаем мышкой на ....Открываем еще один *Макрос*.

40.Выбираем макрокоманду *КЭлементуУправления* и имя элемента выбираем *Код товара*.

41.Следующая макрокоманда *НайтиЗапись*. В строке *Образец поиска* пишем  $=[Список].column(0)$  (рис. 2.90)

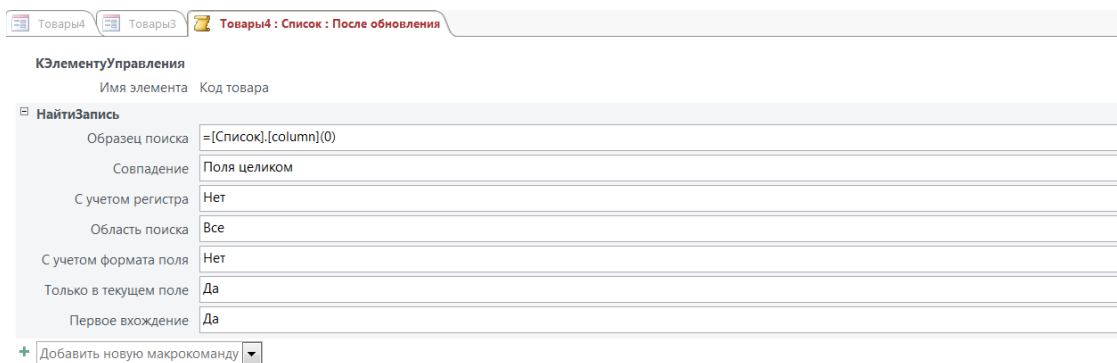


Рис.2.90. Макрос для поля со списком

42. Закройте построитель выражений. Перейдите в режим формы и проверьте ее работоспособность. Если все работает верно, создайте дизайн формы и сохраните ее (рис.2.91).

Рис.2.91. Готовая форма для отдела «Склад»

## 2.7. Лабораторная работа 7. Разработка интерфейса базы данных

Данная лабораторная работа является заключительной в серии лабораторных работ и обобщает навыки, полученные в предыдущих работах. При выполнении лабораторной работы предполагается большая доля самостоятельности, и ее можно считать в некотором смысле «зачетной» работой.

### Создание окна регистрации

Для начала необходимо создать пользовательское диалоговое окно, которое будет отображаться при открытии приложения. В окне необходимо обеспечить ввод имени исполнителя, приступающего к работе, вход в основную форму управления приложением и сохранение кода исполнителя с помощью встроеного макроса. Сохраненный код позволит при открытии других объектов выбирать из них сведения, касающиеся только этого исполнителя.

1. Сначала необходимо дополнительно создать таблицу *Исполнитель*, в которую заносятся сотрудники, работающие с Приложением. Примерный вид таблицы представлен на рисунке 2.92.1.

Код	Фамилия	Имя	Отчество	Должность	Адрес эл_п	Телефон	Ще
	Иванов	Иван	Иванович	ст.инженер	ivanov@mail.r	7(921)555-55-5	
2	Петров	Петр	Петрович	инженер	petrov@mail.r	7(921)666-77-7	
(№)							

Рис.2.92.1 - Таблица "Исполнитель"

2. Затем создадим в *Конструкторе* пустую форму. В пустой форме добавляем поле со списком и в качестве источника строк выбираем таблицу *Исполнитель*. Выбираем поля: фамилия, имя, отчество, должность, код

3. Поскольку в поле со списком желательно отображать фамилию, имя и отчество исполнителя в едином поле, объединяем эти поля в вычисляемом поле. Для этого в режиме конструктора запросов создаем вычисляемое поле, записав выражение `[Фамилия] & " " & [Имя] & " " & [Отчество]`. (рис.2.92.2). Поля с фамилией, именем и отчеством удаляем. Закрываем конструктор запросов, сохранив сделанные в нем изменения.

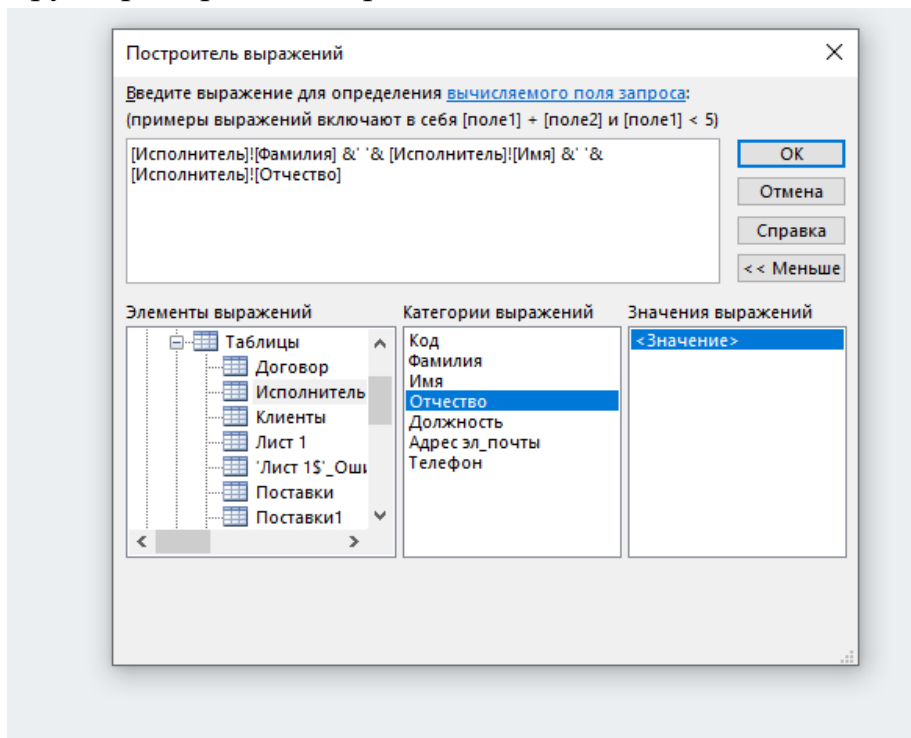


Рис.2.92.2.

4. В окне свойств поля со списком на вкладке *Макет* измените *Число столбцов* на 3 (*ФИО, Должность, Код*).

5. В строке *Ширина столбцов* задайте ширину столбцов (в см.) Настройте ширину самого поля со списком. В строке *Заголовки столбцов* выберите ДА. Измените надпись поля со списком на «*Выберите исполнителя*». (рис.2.92.3)

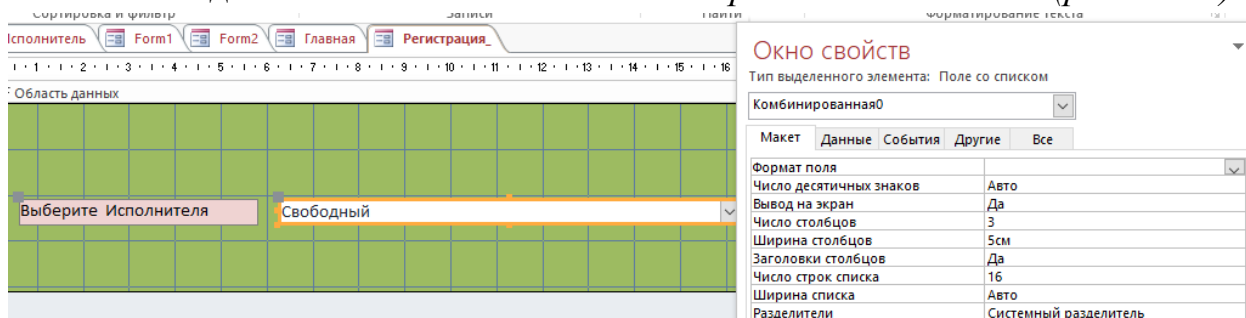


Рис.2.92.3.

6. Во вкладке *Другие* посмотрите и запомните имя поля *Свободный*(рис.2.92.4)

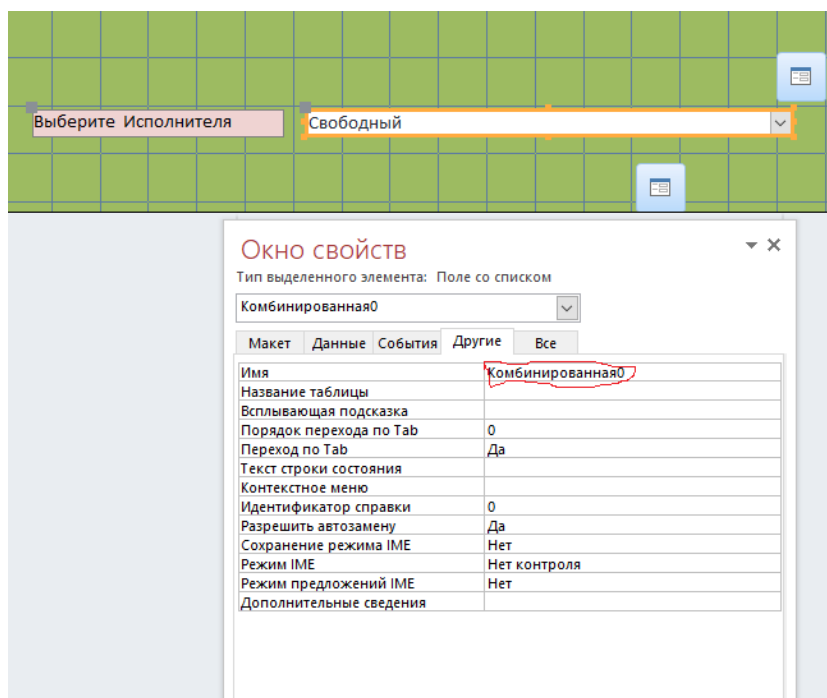


Рис.2.92.4.

7. Сохраните изменения и перейдите в режим формы. Должно получиться так, как показано на рис.2.92.5

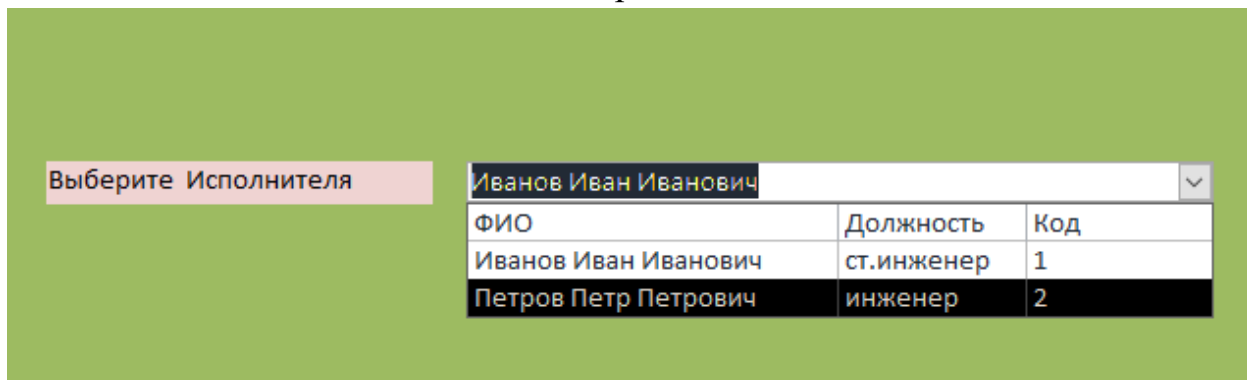


Рис.2.92.5.

8. Сохраните форму под именем *Регистрация*. С помощью этой формы пользователь получает доступ к вашему *Приложению*.
9. Интерфейс БД реализуется в виде иерархического меню. На верхнем (1-ом) уровне находится специальная форма, называемая *Главной кнопочной формой*.
10. Форма *Регистрация* должна давать возможность открывать *Главную кнопочную форму*.
11. Создайте в Конструкторе пустую форму. В окне *Свойств формы* на вкладке *Другие* задайте свойство *Всплывающая форма* – ДА, на

- вкладке *Макет* задайте *Тип границы* – «Окна диалога». Сохраните форму под именем *Главная*.
12. Откройте форму *Регистрация* в режиме *Конструктора*.
  13. Создайте на этой форме кнопку, выбрав категорию *Работа с формой* и действие *Открытие формы*. Выберите форму *Главная*.
  14. В окне *Свойств* на вкладке *События* в строке *Нажатие кнопки* откройте *Внедренный макрос*
  15. Измените макрос так, как показано на рис. 2.92.6

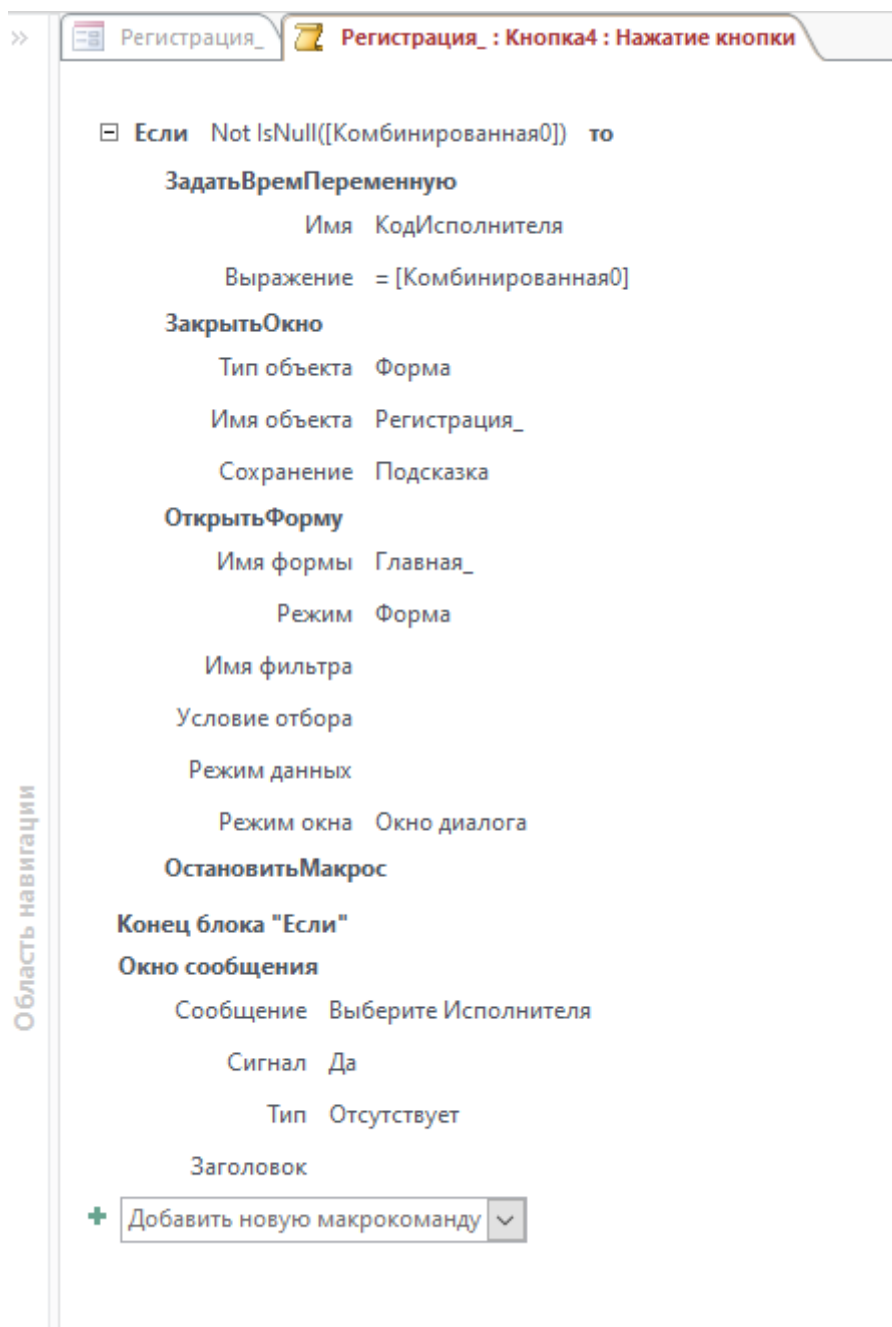


Рис.2.92.6.

16. Сохраните изменения. Проверьте работоспособность формы. Если поле Исполнитель будет пустым, должно появиться сообщение (рис. 2.92.7), если исполнитель выбран, должна открыться главная форма(рис.2.92.8)

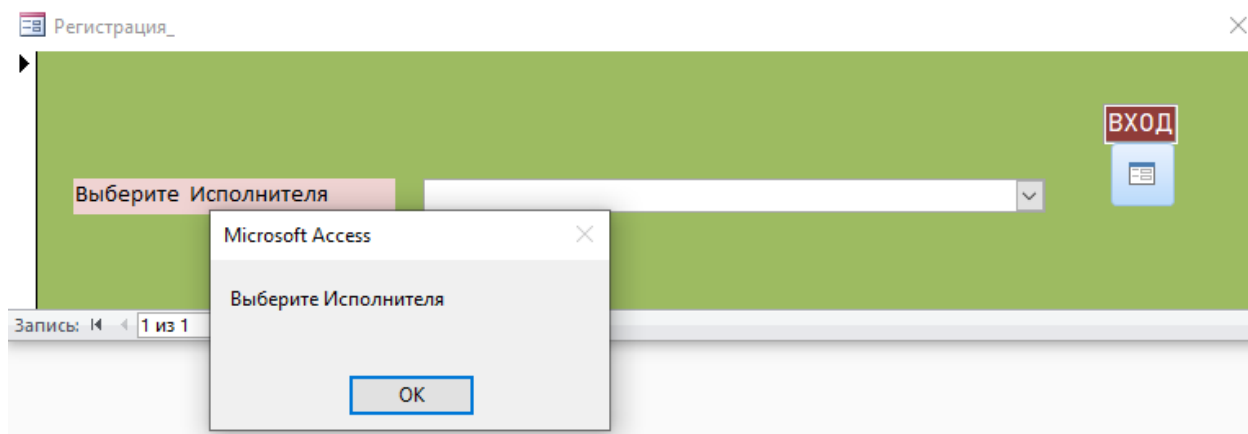


Рис.2.92.7.

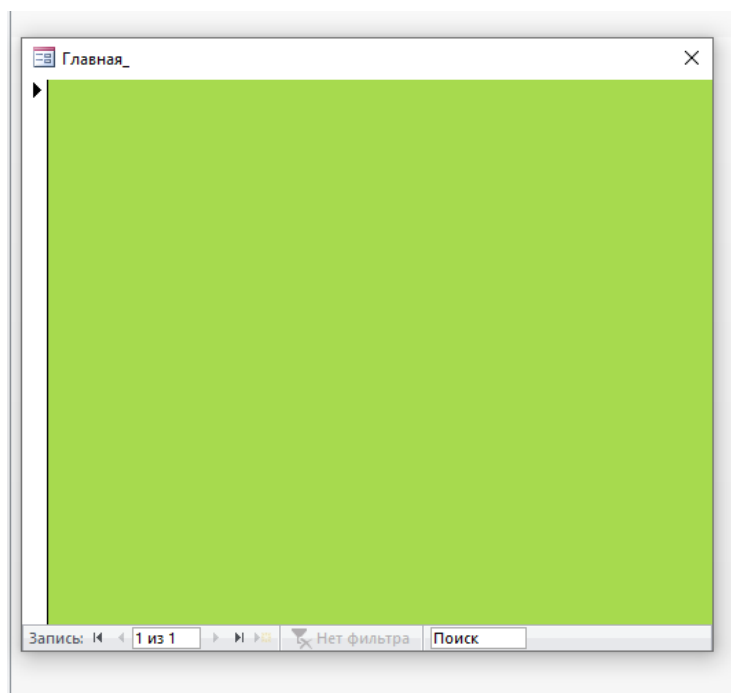


Рис.2.92.8.

17. Для того чтобы форма отображалась сразу после открытия базы данных, создайте автоматически запускающийся макрос с именем Autoexec и включите в него макрокоманду *ОткрытьФорму*— «Регистрация».

18. Закройте вашу БД. Затем откройте ее снова. Должно появиться окно регистрации.



Теперь вернемся к форме *Главная* и построим ее.

1. Откройте формы *Главная* и *Регистрация* в режиме *Конструктора*.
2. Скопируйте поле со списком из формы *Регистрация*.
3. В форме *Главная* создайте *Заголовок* и вставьте скопированное поле со списком в заголовок. Чтобы в поле со списком отображался исполнитель, выбранный в окне регистрации, в свойствах поля на вкладке *Данные* измените *Значение по умолчанию* на: `= [TempVars] ! [КодИсполнителя]` (рис.2.92.9). Используемая здесь временная переменная была определена в макросе кнопки *Вход* из формы *Регистрация*

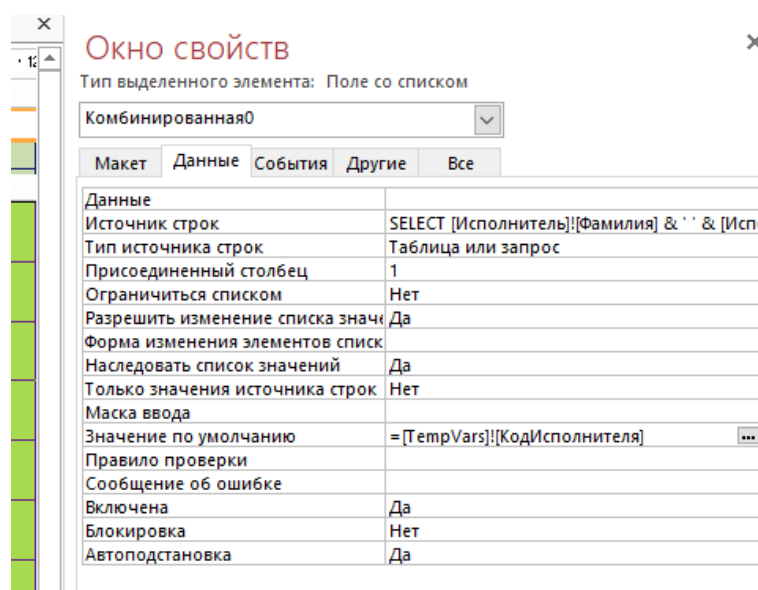


Рис.2.92.9.

4. Перейдите в режим формы. Сохраните все изменения и закройте обе формы.
5. Теперь, при открытии формы *Главная* в ее заголовке должно отображаться имя исполнителя, внесенное при регистрации (рис.2.92.10)

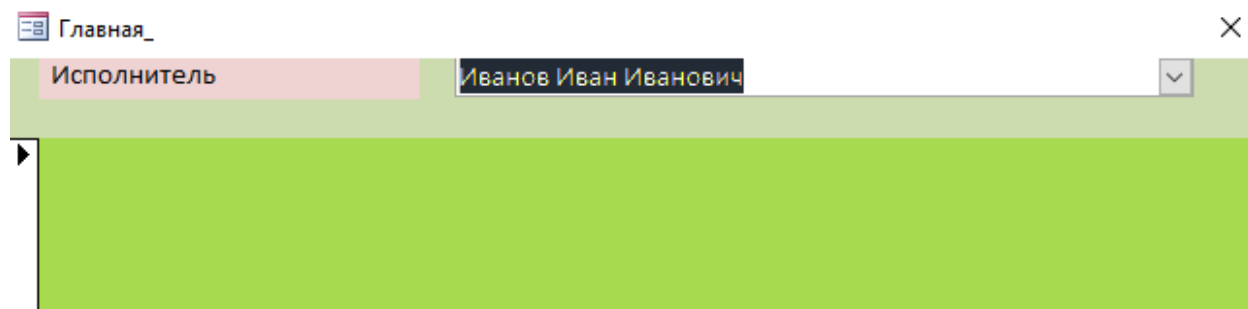


Рис.2.92.10.

Следующим шагом будет окончательное оформление главной формы. Интерфейс разрабатывается студентом самостоятельно. Необходимо только, чтобы присутствовали все пункты меню вызова необходимой информации. Пример оформления представлен ниже.

На главной форме в виде специальных кнопок оформляются пункты меню, предназначенные для выбора той информации, которая необходима для работы с базой.

Главная форма выглядит примерно так, как показано на рис.2.92. Эта форма создается в режиме **Конструктора**. В ней присутствует четыре кнопки. Кнопки *Клиенты*, *Продажи*, *Покупки* и *Склад* служат для вызова кнопочных форм следующего (нижнего) уровня.

При нажатии кнопки *Клиенты* появляется форма нижнего (2-ого) уровня, показанная на рис.2.93. Форма содержит две группы переключателей. Первая группа предназначена для работы с информацией по клиентам, являющимися *Покупателями*, вторая – для клиентов – *Поставщиков*. Переключатели имеют три значения:

1- *Добавить*. При выборе этого переключателя должна открываться форма, аналогичная представленной на рис.2.94. Для этого можно использовать форму *Клиенты*, созданную ранее, доработав ее дизайн и добавив необходимые кнопки.



Рис.2.92. Главная кнопочная форма

ООО ЯР Строй

КЛИЕНТЫ

Покупатели      Поставщики

Выбрать действие

Добавить       Добавить

Удалить       Удалить

Посмотреть всех       Посмотреть всех

←

Рис.2.93. Форма второго уровня «Клиенты»

КОМПАНИЯ

Покупатели

КодКлиента (№)

ИмяКлиента

Адрес

Телефон

Рейтинг 0

Добавить запись

←

Рис.2.94. Форма, вызываемая при значении переключателя «1».

The image shows a web interface for managing buyers. At the top left is a logo with a bird and the word 'КОМПАНИЯ'. To the right is a title box 'Покупатели'. Below are five rows of data entry fields, each with a label and a value: 'ИмяКлиента' (Client Name) is a dropdown menu with 'ООО Роза'; 'КодКлиента' (Client Code) is a text box with '2'; 'Адрес' (Address) is a text box with 'Москва'; 'Телефон' (Phone) is a text box with '(495) 222-22-22'; 'Рейтинг' (Rating) is a text box with '100'. To the right of the 'КодКлиента' field is a button 'Удалить запись' (Delete record). To the right of the 'Телефон' field is a button with a left-pointing arrow.

Рис.2.95. Форма, вызываемая при значении переключателя «2».

2- *Удалить*. При выборе этого переключателя открывается форма, представленная на рис.2.95. Ее можно получить, скопировав форму *Клиенты*, добавив в нее кнопку *Удалить запись* и изменив поле *ИмяКлиента* на «*Поле со списком*».

3-*Посмотреть всех*. При выборе этого переключателя создается отчет, в котором содержатся сведения о всех клиентах-покупателях.

Вторая группа переключателей проектируется аналогично, только она работает с формой *Поставщики*.

При программировании переключателей следует обратить внимание на то, что у каждой группы переключателей должно быть уникальное имя.

Возвращаемся к *Главной кнопочной форме* (рис.2.92). При выборе пункта *Продажа* происходит переход к форме нижнего уровня *Продажи* (рис.2.96).

На ней также имеется группа переключателей, с помощью которых вызываются соответствующие формы следующего уровня.

При выборе переключателя *Новая продажа* открывается форма *Клиенты\_Продажи*, разработанная ранее.



Рис.2.96. Форма второго уровня Продажи\_2.

При выборе переключателя *Удалить запись* происходит вызов формы *Клиенты\_Продажи*, в которой кнопка *Добавить запись* заменяется кнопкой *Удалить запись* (рис.2.98).

При выборе пунктов меню *Отчеты* удобно использовать специальный вид форм – формы навигации.

Форма навигации предназначена для создания на ней горизонтальных, вертикальных или и тех, и других вкладок одновременно. На этих вкладках могут размещаться отчеты.

Создаются формы навигации командами из списка *Навигация*, расположенного на ленте *Создание* в группе *Формы*.

Создайте форму навигации с горизонтальным расположением вкладок в *два уровня*.

Вы получите заготовку для формы, представленную на рис.2. 96.1.

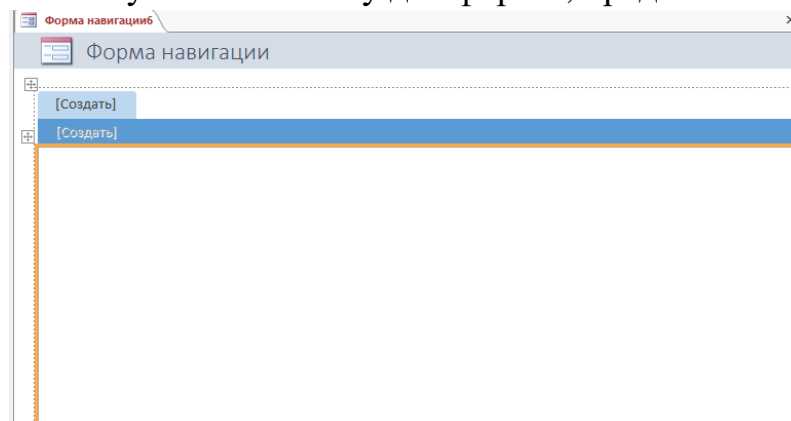


Рис.2.96.1

Вкладки навигации первого уровня используйте для открытия группы вкладок второго уровня. Дайте этим вкладкам имена, которые будут определять суть содержимого группы. К продажам относятся две группы отчетов : по клиентам и по товарам. Щелкните по вкладке *Создать* и задайте имя первой группе, например *Сведения по Клиентам*. Затем щелкните справа от этой вкладки . Появится еще одна вкладка *Создать*. Задайте ей имя *Сведения по товарам* (рис.2.92.2)

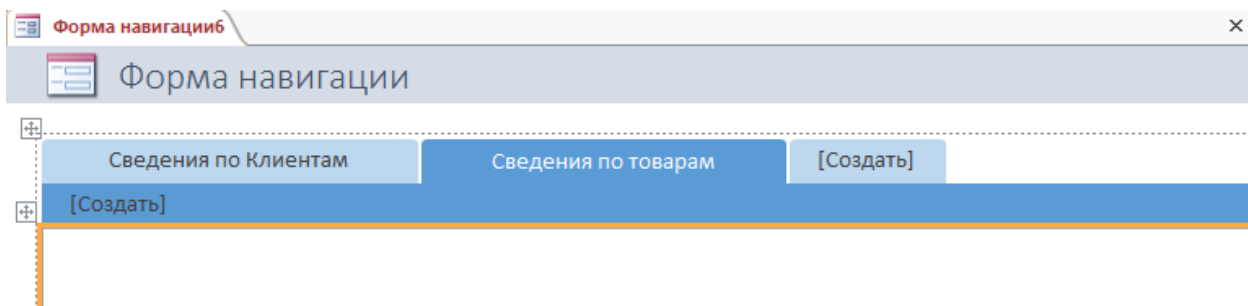


Рис.2.96.2

*Замечание!* Имена не должны совпадать с именами имеющихся в базе форм и отчетов.

Разместите на страницах вкладок второго уровня соответствующие отчеты. Для размещения нужных отчетов на страницах вкладок достаточно перетаскивать их из списка объектов в нужные места в ряду кнопок навигации или вводить их имена на кнопках навигации. (рис.2.96.3)

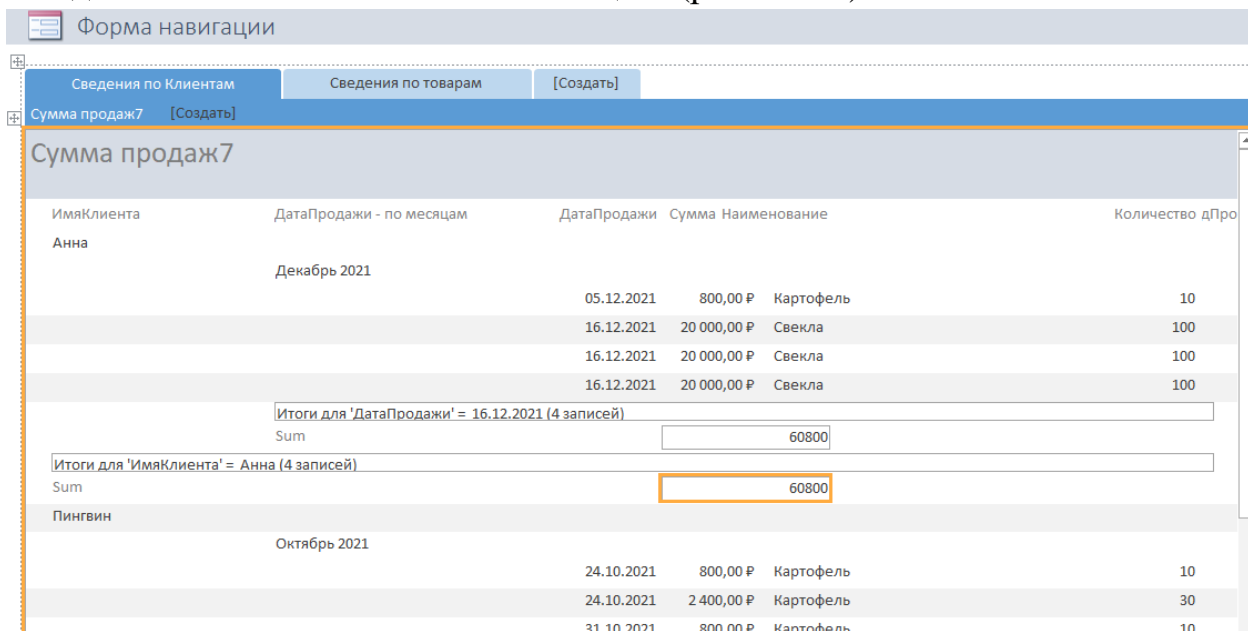


Рис.2.96.3

В каждой группе вкладок должно быть по три отчета: рис. 2.70 -2.70.б и рис.2.70.в-2.70.д соответственно.

Аналогично проектируются формы второго уровня при выборе на главной кнопочной форме пункта *Покупки*. При этом на форме *Поставщики\_покупки* необходимо добавить вычисляемые поля (кто этого не сделал ранее) аналогично форме *Клиенты\_Продажи*.

Клиенты\_Продажи 25 августа 2017 г.

ИмяКлиента: ООО Роза

Сведения о клиенте

Продажи

КодПродажи	Дата	Наименование	Количество
6	06.04.201	Картофель	40
10	25.08.201	Морковь	10
13	25.08.201	Свекла	10
*(№)	25.08.201		0

Сохранить Добавить запись

Рис.2.97. Форма второго уровня Клиенты\_Продажи\_2

Клиенты\_Продажи 27 августа 2017 г.

ИмяКлиента: ООО Тюльпан

Сведения о клиенте

Продажи

КодПродажи	Дата	Наименование	Количество
9	25.08.201	Свекла	30
11	25.08.201	Картофель	10
16	25.08.201	Свекла	20
*(№)	27.08.201		0

Удалить запись

Записи: 4 из 3 Нет фильтра Поиск

Рис.2.98. Форма второго уровня «Удалить запись»

И, наконец, при выборе пункта *Склад* на главной кнопочной форме открывается форма *Склад* (рис.2.91).

На этом выполнение лабораторной работы 7 заканчивается.

### 3. Общие рекомендации к выполнению цикла лабораторных работ

При выполнении лабораторных работ обязательным является наличие того или иного объекта базы данных (формы, отчета, запроса). Заполнение базы данных информацией и дизайн форм осуществляется студентом самостоятельно и при оценке работы учитывается творческий подход к проектированию объектов БД.

#### *Содержание отчета по лабораторным работам*

Отчет по лабораторной работе должен содержать:

- описание объекта базы данных, изучению которого посвящена данная лабораторная работа: назначение, способы создания, режимы работы;
- схему данных с объяснениями типов связей и обеспечением целостности данных
- скриншоты результатов создания объектов базы данных с объяснениями его работы.

### 4. Приложение

#### **Типы данных в Access**

Рассмотрим типы данных, используемые в Access, и особенности работы с ними, в том числе свойства, специфичные для полей с таким типом данных. Значения свойств вы можете просмотреть и изменить в нижней части конструктора таблицы, на вкладке **Общие**. Итак, поле таблицы может иметь один из следующих типов данных.

- **Счетчик**. Поле заполняется автоматически уникальными значениями —последовательными или случайными, в зависимости от значения свойства **Новые значения**. Свойство **Размер поля** для поля с типом **Счетчик** принимает значения **Длинное целое** или **Код репликации** (Код репликации Глобальный уникальный идентификатор (GUID)). Пример GUID: {4fbc3fb2-90f5-4067-af37-d7ee6eb9efb6}. В таблице Access может быть только одно поле с типом данных **Счетчик**, и оно обычно служит первичным ключом.
- **Текстовый**. В текстовом поле хранится символьная строка. Свойство **Размер поля** определяет максимальную длину строки и принимает значение от 0 до 255.



- **Поле МЕМО.** Поле предназначено для хранения большого объема текста — до 65 535 символов при вводе вручную и до 2 Гбайт (максимальный размер базы данных Access) при программном заполнении полей. Для типа данных **Поле МЕМО** вы можете выбрать значение свойства **Формат текста: Обычный текст** или **Формат RTF**.
- **Числовой.** Числовые поля содержат значения, с которыми можно выполнять математические операции: сложение, умножение, вычисление максимального и т. д. Свойство **Размер поля** определяет, является число целым или десятичным (имеющим дробную часть), а также максимальное и минимальное допустимое значение. Для поля, имеющего размер **Действительное**, свойство **Точность** определяет максимальное хранимое в базе суммарное количество знаков в целой и дробной части, свойство **Шкала** — максимальное количество знаков в дробной части.
- **Дата/время.** Поле предназначено для хранения даты и/или времени суток (в зависимости от формата).
- **Денежный.** Поле предназначено для хранения денежных сумм с точностью до четырех знаков после десятичного разделителя.
- **Логический.** Поле может принимать одно из двух значений, например: Да или Нет, Истина или Ложь.
- **Поле объекта OLE.** Поле предназначено для хранения присоединенных (вложенных) объектов, таких как документы Microsoft Office, изображения, аудио/видео и др. Для работы с такими полями на вашем компьютере должен быть установлен OLE-сервер. В базе данных формата Access 2007 (ACCBD) вместо типа данных **Поле объекта OLE** лучше использовать тип данных **Вложение**.
- **Гиперссылка.** В поле может храниться ссылка: на веб-страницу (пример: <http://www.microsoft.com>); на файл на FTP-сервере, в локальной сети или на компьютере (пример: C:\Docs\Sample.doc); на адрес электронной почты (пример: [mymail@list.ru](mailto:mymail@list.ru)) и т. п.

Значение поля имеет следующую структуру: *Отображаемый текст#Адрес#Метка или закладка#Всплывающая подсказка*. Например, значение *Мой документ#C:\Docs\Sample.doc#MyLabel#Переход на мою метку* означает, что при просмотре таблицы в поле отображается значение *Мой документ*, при подведении указателя мыши к значению поля появляется подсказка *Переход на мою метку* (рис. П.1), а при щелчке кнопкой мыши на ссылке открывается документ Sample.doc на той странице, где вставлена закладка с именем *MyLabel*.

Необязательно вводить ссылку вручную. Щелкните в поле правой кнопкой мыши, в появившемся меню выберите команду **Гиперссылка** ▶ **Изменить гиперссылку**. Откроется окно **Вставка гиперссылки**, в котором

можно выбрать адрес гиперссылки — файл или веб-страницу, а также ввести отображаемый в поле текст и текст всплывающей подсказки.

Редактирование гиперссылок имеет свои тонкости, поскольку для изменения значения нельзя просто щелкнуть кнопкой мыши на ячейке таблицы (при этом вы сразу перейдете по указанному адресу). Поэтому, чтобы внести исправления, щелкните кнопкой мыши на *предыдущую* ячейку, нажмите клавишу **Tab** для установки курсора в ячейку с гиперссылкой, а затем клавишу **F2**. После этого можно корректировать значение поля.

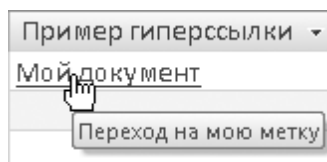


Рис.П.1.. Пример гиперссылки

- **Вложение**. Поле с типом данных **Вложение** (как и поле с типом данных **Поле объекта OLE**) предназначено для хранения любых присоединенных (вложенных) объектов, таких как документы Microsoft Office, изображения, аудио, видео и др. В поле с таким типом данных отображается количество присоединенных объектов и значок скрепки, при нажатии на котором появляется окно **Вложения** (рис.П.2.). В окне **Вложения** можно добавить и удалить вложенный объект, открыть его для редактирования в соответствующей программе (например, для редактирования документа в формате DOC запускается Microsoft Word), а также сохранить объект в файловой системе. Важное отличие вложения от гиперссылки состоит в том, что при добавлении файла как вложения в базе данных Access создается копия этого файла. При этом изменения, внесенные при редактировании вложенного объекта, никак не отражаются на содержимом исходного файла в файловой системе, а изменения в файле не переносятся автоматически на объект, хранимый в базе данных. Напротив, при создании гиперссылки на файл копия не создается, при щелчке кнопкой мыши на гиперссылке открывается сам файл («оригинал»).

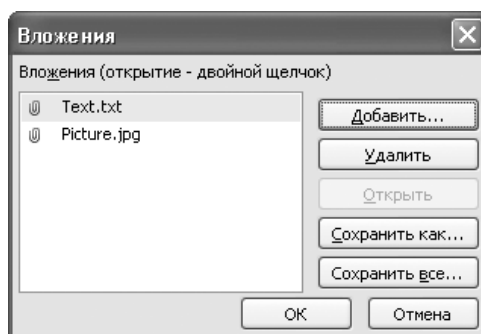


Рис.П.2. Окно Вложения

### **Форматы отображения данных**

Одно и то же значение поля, хранимое в базе данных, можно по-разному отобразить на экране (в таблице, отчете и др.). Если вам не подходит представление данных, установленное по умолчанию, вы можете изменить формат отображения. Для этого используется свойство **Формат поля** на вкладке **Общие** в нижней части конструктора таблицы.

#### **Формат текстового поля и поля МЕМО**

Для полей, содержащих текст, значением свойства **Формат поля** является строка, в которую вы можете включить специальные символы:

> — содержимое поля отображается прописными буквами (верхний регистр);

< — содержимое поля отображается строчными буквами (нижний регистр);

@ — отображается очередной символ из значения поля; если все символы уже выведены, то отображается пробел;

& — отображается очередной символ из значения поля; если все символы уже выведены, то ничего не отображается. Кроме того, после формата отображения текста через точку с запятой можно указать формат для пустого поля.

Например, вы зададите для поля *Телефон* формат (&&&) &&&-&&-&&;”Нет телефона”. При этом введенное значение 4951221212 отображается как (495) 122-12-12 (но в базе данных все равно хранится 4951221212!), а в поле, в которое не введено значение, показан текст *Нет телефона*. Если для поля *Статус* задан формат >&;”не определен”, то слово «срочный» выглядит как **СРОЧНЫЙ**, а в незаполненном поле выводится строка *не определен*.

#### **Формат числового и денежного поля**

Для числовых и денежных полей вы можете выбрать из списка одно из следующих значений свойства **Формат поля**:

- **Основной** — значение отображается в том же виде, в каком хранится в базе данных, например: 12,3456.
- **Денежный** — вместе с числом в поле выводится обозначение денежной единицы, например, 12,34р. (в базе данных хранится только число). Обозначение денежной единицы определяется региональными настройками Windows (Пуск □ ▶ Панель управления □ ▶ Язык и региональные стандарты). Формат **Денежный** и тип данных **Денежный** — это не одно и то же. Тип данных **Денежный** обеспечивает хранение числа с точностью до четырех знаков после разделителя, а формат **Денежный** — отображение денежной единицы. Поэтому поле с типом данных **Числовой** может иметь денежный формат, и наоборот, поле с типом **Денежный** — формат **Основной** или другой числовой формат.
- **Евро** — вместе с числом в поле выводится обозначение евро, например: 12,34€.

- **Фиксированный** — в поле отображается число без разделителей тысяч с фиксированным количеством знаков после разделителя, указанном в свойстве **Число десятичных знаков**. Например: 12345,67.
- **С разделителями разрядов** — формат аналогичен формату **Фиксированный**, только число отображается с разделителями тысяч, например: 12 345,67.

Разделитель тысяч (обычно пробел или запятая) определяется региональными настройками.

- **Процентный** — в поле выводится значение, умноженное на 100 и дополненное символом %.
- **Экспоненциальный** — число представлено в виде произведения мантиссы ( числа с единственным знаком в целой части) и степени числа десять. Формат удобен для отображения очень маленьких и очень больших по абсолютной величине чисел. Например, число  $0,0000001234 = 1,234 \times 10^{-7}$  отображается как **1,234E-7**, число  $12\,340\,000 = 1,234 \times 10^7$  — как **1,234E+7** (справа от буквы E отображается мантисса, слева — показатель степени).

Помимо свойства **Формат поля**, на отображение числовых и денежных значений влияет свойство **Число десятичных знаков**. Это свойство задает количество знаков, отображаемых после разделителя для всех форматов, кроме формата **Основной**. При этом «лишние» знаки не выводятся (в поле отображается округленное значение, и только при установке курсора в поле — полное значение), а числа с меньшим количеством знаков дополняются справа нулями. В формате **Основной** отображаются все хранимые знаки, независимо от значения свойства **Число десятичных знаков**.

#### Формат даты и времени

Для полей с типом данных *Дата/время* вы можете выбрать один из следующих встроенных форматов (значений свойства **Формат поля**).

- **Полный формат даты** — значение поля отображается полностью, например: **04.02.2008 14:55:00**. Если в поле хранится только дата, то и отображается только дата, время не выводится (пример: **32.03.2008**). Если в поле хранится только время, то не отображается дата (пример: **15:05:05**).
- **Длинный формат даты** — например, **4 января 2008 г.** Если в поле хранится только время, то отображается дата по умолчанию — **30 декабря 1899 г.**
- **Средний формат даты** — например, **04-январь-2008**. Если в поле хранится только время, то отображается дата по умолчанию — **30-дек-1899**.
- **Краткий формат даты** — например, **04.02.2008**. Если в поле хранится только время, то отображается дата по умолчанию — **30.12.1899**.

- **Длинный формат времени** — например, **23:47:12**. Если в поле хранится только дата, то отображается время по умолчанию — **00:00:00**.
- **Средний формат времени** — например, **11:47**. Время отображается в 12 часовом формате. Например, вместо **21:00** выводится **9:00** (девять вечера). Если в поле хранится только дата, то отображается время по умолчанию — **12:00**.
- **Краткий формат времени** — например, **23:12** (в 24-часовом формате). Если в поле хранится только дата, то отображается время по умолчанию — **0:00**.

### *Формат логического поля*

По умолчанию значение логического поля отображается в виде установленного (истинное значение) или снятого (ложное значение) флажка. Если вы хотите, чтобы значение отображалось в текстовом виде, выполните следующие действия. В нижней части конструктора перейдите на вкладку **Подстановка** и для свойства **Тип элемента управления** выберите из списка значение **Поле**. Вернитесь на вкладку **Общие** и для свойства **Формат поля** выберите из списка текстовые значения, которые будут отображаться в поле: *Истина/Ложь*, *Да/Нет* или *Вкл./Выкл*.

Если для логического поля выбран текстовый формат, то значения вводятся вручную. Для ввода истинного значения наберите -1, Истина, Да или Вкл., для ввода ложного значения — 0, Ложь, Нет или Выкл. Вы также можете настроить пользовательский формат логического поля и выбрать для вывода в поле любые фразы. Информация о создании пользовательских форматов для логического поля приведена в справочной системе Access в разделе **Программирование объектов** ▢ **Свойства** ▢ **Свойство "Формат (Format)" для логического типа данных**.

### *Подстановка*

Подстановкой называется список возможных значений поля. Если для поля создана подстановка, то значение поля можно выбрать из этого списка, вместо того чтобы вводить вручную. Источником значений для подстановки может быть другая таблица, запрос или созданный вами фиксированный (статический) список. Для создания подстановки удобно использовать мастер подстановок. Процедура создания подстановки зависит от типа источника данных.

*Создание подстановки на основе таблицы было рассмотрено выше. Аналогичным образом осуществляется подстановка на основе запросов.*

*Создание подстановки на основе фиксированного списка значений*

Если вам нужно создать подстановку, источником данных для которой служит список введенных значений, выполните следующие действия.

1. В режиме конструктора выберите в списке полей то поле, для которого нужно создать список значений. Щелкните кнопкой мыши в ячейке с названием типа данных поля.
2. Нажмите кнопку и в появившемся списке выберите значение **Мастер подстановок**. На экране появится окно **Создание подстановки** (см. рис.П.3).
3. Установите переключатель в положение **Будет введен фиксированный набор значений** и нажмите кнопку **Далее**.
4. Введите количество столбцов (рис.П.3). Затем щелкните кнопкой мыши в первой ячейке первого столбца и введите необходимое значение. Нажмите клавишу **Tab** для перехода к следующей ячейке. Введите таким образом весь набор значений для подстановки. Подберите ширину столбцов в подстановке так, чтобы все значения отображались полностью. Нажмите кнопку **Далее**.

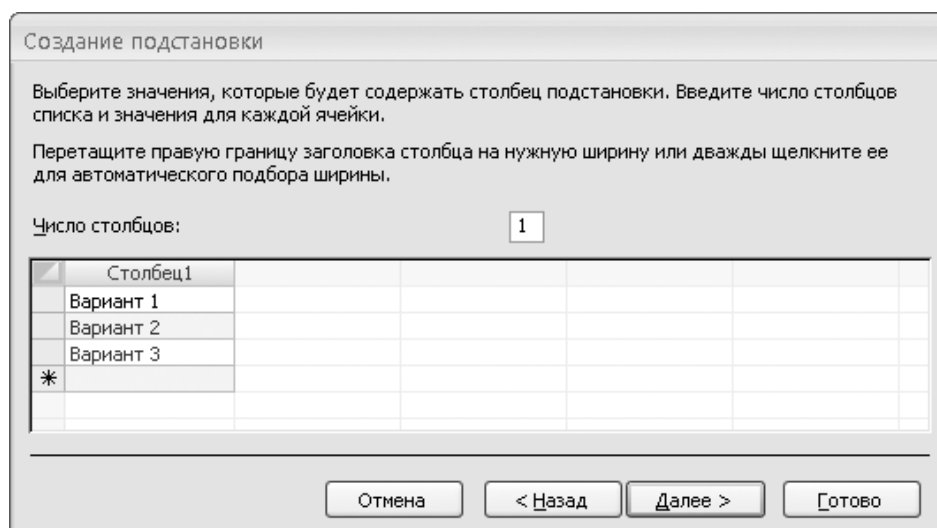


Рис.П.3. Ввод списка значений

5. В завершение создания подстановки при необходимости измените название поля, для которого создается подстановка.
6. Нажмите кнопку **Готово**.

### Использование мастера масок

Чтобы создать маску ввода с помощью мастера масок, выполните следующие действия.

2. Щелкните кнопкой мыши на названии свойства **Маска ввода**. Справа от значения свойства появится кнопка  .
2. Нажмите кнопку  . На экране вы увидите окно **Создание масок ввода** (рис. П.4).
3. В окне **Создание масок ввода** выберите нужную маску: одну из масок даты или времени либо маску пароля. Маска пароля позволяет создать поле для

ввода пароля: данные, введенные в такое поле, будут скрыты, вместо реальных символов отобразятся звездочки.

4. Если требуется протестировать ввод значения с помощью выбранной маски, введите в поле **Проба** пример значения поля.

5. Если выбрана маска даты/времени, которую не требуется редактировать, или маска пароля, нажмите кнопку **Готово**. Процесс создания маски на этом окончен. Если нужно изменить выбранную маску даты или времени, нажмите кнопку **Далее**.

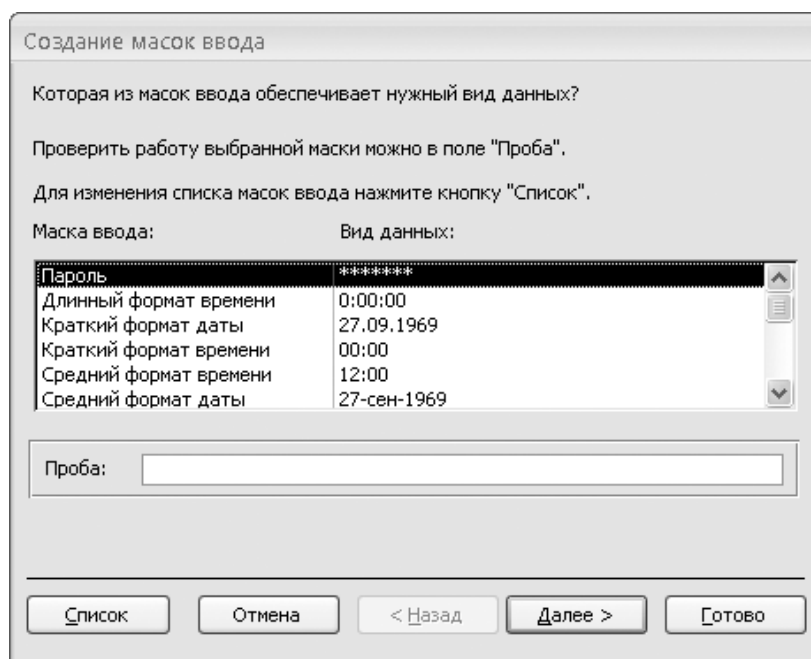


Рис.П.4. Мастер масок

При ручном вводе в значение свойства **Маска ввода** вы можете включить специальные символы и текстовые константы. Специальный символ определяет, какие символы разрешено или необходимо ввести в значении поля на той позиции, на которой в маске находится спецсимвол. Текстовые константы включаются в значение поля, их нельзя изменить или удалить.

Например, для поля *Телефон* можно использовать маску (999) 000-00-00. Эта маска означает, что в поле необходимо ввести семь цифр номера и до трех цифр кода. При сохранении программа Access автоматически исправит маску, она будет выглядеть так: \ (999) "000\ -00\ -00.

В маске могут быть использованы только специальные символы. Например, 0 — обязательный ввод цифры, & — обязательный ввод произвольного символа или пробела. Информацию об этих символах вы можете получить в справочной системе Access в разделе **Структура базы данных ▶ Создание маски ввода для ввода значений в поле или элемент управления в определенном формате**.

*Условие на значение*

Задание условия на значение поля — еще один способ автоматической проверки корректности вводимых данных, который вы можете использовать наряду с контролем соответствия типов данных и масками ввода.

Условие на значение вы можете создать вручную, введя значение свойства **Условие на значение**, или с помощью построителя выражений (рис.П.5), вызываемого нажатием кнопки справа от значения свойства. Помимо этого, вы можете задать значение свойства **Сообщение об ошибке** — текст сообщения, которое будет появляться в случае, если введенное в поле значение не соответствует условию.

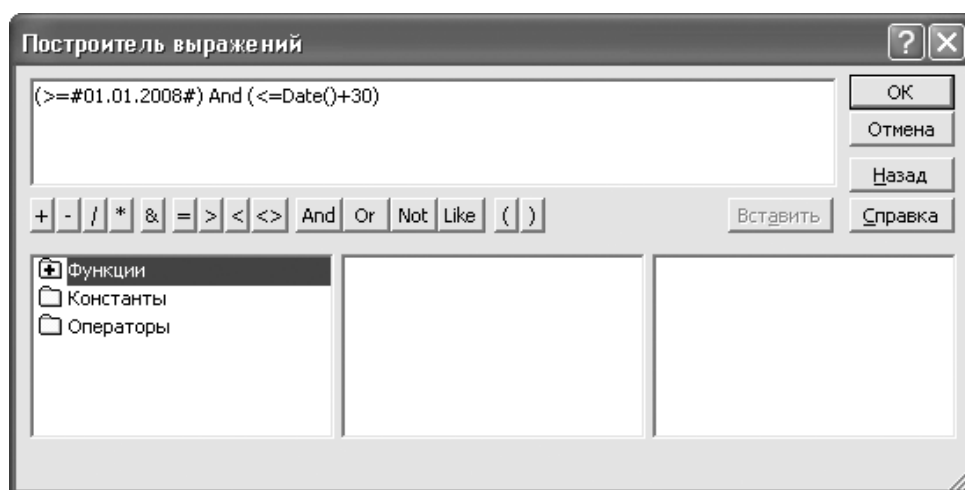


Рис.П.5. Построитель выражений

Перечислим операторы, наиболее часто используемые в условиях на значение.

< (>) — условие выполняется, если значение слева от знака < (>) меньше (больше), чем значение справа от знака.

<= (>=) — условие выполняется, если значение слева от знака <= (>=) меньше или равно (больше или равно), чем значение справа от знака

= — условие выполняется, если сравниваемые значения равны.

<> — условие выполняется, если сравниваемые значения не равны.

**BETWEEN** — условие выполняется, если значение слева от оператора находится в диапазоне, указанном справа от оператора. Например, условию **[Имя] BETWEEN "a" AND "к"** удовлетворяет значение *Иванов*, но не удовлетворяет значение *Петров*. Условие **BETWEEN 0 AND 1** равносильно условию **>=0 AND <=1**.

**IN** — условие выполняется, если значение слева от оператора совпадает с одним из элементов списка справа от оператора. Список заключается в круглые скобки, элементы списка указываются в кавычках через точку с запятой. Например: **[Рейтинг] IN ("500";"1000";"1500")**.

**LIKE** — условие выполняется, если значение слева удовлетворяет шаблону справа. Шаблон может содержать подстановочные знаки ? (на месте этого знака может быть произвольный символ) и \* (на месте этого знака может быть любое количество произвольных символов). Например, условие **Like**



“?.12.\*” означает, что вводимая дата должна относиться к декабрю. Такое условие аналогично заданию маски ввода (см. подразд. «Маска ввода»).

**IS NOT NULL** — условие выполняется, если в поле введено любое значение (то есть поле не пустое). Такое условие аналогично заданию обязательности поля (см. подразд. «Обязательное поле»).

**NOT** — условие выполняется в том случае, если *не* выполняется условие, находящееся справа от оператора **NOT**. Например, условие **[Рейтинг] NOT =0** равносильно условию **[Рейтинг]<>0**.

**AND** — условие выполняется, если выполнены оба условия справа и слева от оператора. Например, если задано условие **>= #02.02.2008# AND <= Date()+30**, то вводимая дата должна быть не ранее начала 2008 года, но не позднее чем через 30 дней после текущей даты.

**OR** — условие выполняется, если выполнено хотя бы одно из условий справа и слева от оператора. Например, условие **0 OR >=500** выполняется, если вводимое число равно нулю или не менее 500.

### *Значение по умолчанию*

Значение по умолчанию автоматически устанавливается в поле при создании новой записи. Если вы зададите значение по умолчанию, это позволит вам ускорить ввод данных в том случае, когда во многих записях не потребуется менять значение по умолчанию на какое-либо другое. Чтобы создать значение по умолчанию, введите значение свойства **Значение по умолчанию**. Например, чтобы в поле **Дата** по умолчанию устанавливалась текущая дата, введите значение **Date()**.

### *Обязательное поле*

Если поле обязательно должно быть заполнено, то в свойстве **Обязательное поле** выберите из списка значение *Да*. Программа Access не позволит сохранить запись, если не введено значение обязательного поля. Завершим обзор свойств полей таблицы рассмотрением свойства, позволяющего создать индекс для поля.

### Библиографический список

1. Ю. Бекаревич, Н. Пушкина. Microsoft Access 2013. БХВ-Петербург, 2014, 458с.
2. Дейт К. Дж. Введение в системы баз данных - 8-е изд. М.: «Вильямс», 2006. — 1328 с.
3. В.А.Галанина. Базы данных. Введение в теорию реляционных баз данных. СПб: ГУАП, 2008, 107.
4. Коннолли Т., Бегг К. Базы данных: проектирование, реализация, сопровождение. Теория и практика.—3-е изд.: Пер. с англ.: Уч. пос. М.: Изд. дом "Вильямс", 2003, 1440 с.
5. Грабер М. Введение в SQL.. М.:Лори, 2008, 378 с

6. Лори Ульрих Фуллер, Кен Кук. Access 2010 для чайников — М.: Диалектика», 2010. — С. 384.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. Схема данных в СУБД Access.....	3
2. Цикл лабораторных работ с использованием СУБД Access.....	5
2.1. Лабораторная работа 1. Работа с таблицами.....	5
2.2. Лабораторная работа 2. Работа с формами .....	15
2.3. Лабораторная работа 3. Работа с запросами.....	29
2.4. Лабораторная работа 4. Создание сложных запросов .....	35
2.5. Лабораторная работа 5. Создание отчетов.....	43
2.6. Лабораторная работа 6. Создание макросов .....	48
2.7. Лабораторная работа 7. Разработка интерфейса базы данных... .	58
3. Общие рекомендации к выполнению цикла лабораторных работ..	63
4. Приложение.....	64
Библиографический список.....	73