

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Емельянов Сергей Геннадьевич  
Должность: ректор  
Дата подписания: 04.02.2021 19:02:25  
Уникальный программный ключ:  
9ba7d3e34c012eba476ffd2d064cf2781953be730df2374d16f3c0ce536f0fc6

1

## МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра информатики и прикладной математики

УТВЕРЖДАЮ

Первый проректор –

проректор по учебной  
работе



Е.А. Кудряшов

21 октября 2012 г.

### РАБОТА С БАЗАМИ ДАННЫХ В MICROSOFT ACCESS 2007

Методические указания к лабораторным работам по курсу  
«Информатика»

КУРСК 2012

УДК 004.9

Составитель Е.И.Аникина

Рецензент

Кандидат технических наук, доцент кафедры информатики  
и прикладной математики *Н.Н. Бочанова*

**Работа с базами данных в Microsoft ACCESS 2007:**  
методические указания к лабораторным работам по курсу  
«Информатика»/Юго-Зап. гос. ун-т; сост. Е.И.Аникина. Курск,  
2012. 46 с.: Ил.47.

Содержат краткое введение в теорию баз данных,  
описание основ технологии создания базы данных и ее  
поддержки в актуальном состоянии средствами пакета Microsoft  
ACCESS 2007, примеры создания таблиц, форм и поисковых  
запросов, а также задания и рекомендации для выполнения  
лабораторных работ по курсу «Информатика» для студентов  
технологических специальностей.

Текст печатается в авторской редакции

Подписано в печать . Формат 60x84 1/16. Печать  
офсетная

Усл. печ. л. 2,7 . Уч.-изд. л. 2,4 . Тираж 100 экз. Заказ .  
Бесплатно

Юго-Западный государственный университет  
305040, Курск, ул.50 лет Октября, 94.

## ВВЕДЕНИЕ

### Общие сведения о базах данных

**База данных** — это средство сбора и организации информации. В базах данных могут содержаться сведения о людях, продуктах, заказах и т. д. Многие базы данных изначально представляют собой список в текстовом процессоре или электронной таблице. По мере того как список разрастается, в нем накапливаются излишние и противоречивые данные. В форме списка эти данные становится все труднее понять, а возможности поиска или извлечения подмножеств данных для просмотра весьма ограничены. Когда возникают подобные проблемы, полезно перенести информацию в базу данных, созданную с помощью системы управления базами данных (СУБД), например Office Access 2007.

Компьютерная база данных представляет собой хранилище объектов, в состав которых входят таблицы, формы, отчеты, запросы, макросы и модули. В одной базе данных может храниться несколько таблиц. Например, система заказов товаров, в которой используются три таблицы (КЛИЕНТЫ, ТОВАРЫ и ЗАКАЗЫ), — это не три базы данных, а одна, содержащая три таблицы. В базе данных Access таблицы сохраняются в одном файле вместе с другими объектами, такими как формы, отчеты, макросы и модули. Приложение Access 2007 можно использовать для создания файлов в более ранних форматах файлов (например, Access 2000 и Access 2002-2003). Базы данных, созданные в формате Access 2007, имеют расширение имени файла ACCDB, а базы данных, созданные в более ранних форматах Access, — расширение MDB.

## Компоненты базы данных Access

### Таблицы

По внешнему виду таблица базы данных похожа на электронную таблицу, в которой данные располагаются в строках и столбцах. Основное различие между хранением данных в электронной таблице и в базе данных — способ организации данных.

Таблица содержит данные по определенной теме, например, сведения о сотрудниках или товарах. Каждая *запись* в таблице включает данные об одном элементе, например о конкретном сотруднике. Запись состоит из *полей* и включает такие сведения, как имя, адрес и телефонный номер.

Код	Организация	Имя	Фамилия
1	Организация А	Ольга	Костерина
2	Организация В	Григорий	Верный
3	Организация С	Владимир	Егоров

1 –Запись, 2 - Поле

Рис.1. Структура таблицы ACCESS

База данных может включать множество таблиц, в которых хранятся данные по различным темам. Каждая таблица может состоять из множества полей различного типа, включая текст, числа, даты и рисунки.

Определение структуры базы данных необходимо всегда начинать с создания ее таблиц. Таблицы создаются раньше любых других объектов базы данных.

## **Формы**

Формы иногда называются окнами ввода данных. Это интерфейсы, которые используются для работы с данными и часто содержат кнопки для выполнения различных команд. Базу данных можно создать без помощи форм, просто вводя в таблицу данные в режиме таблицы. Однако большинство пользователей баз данных предпочитают просматривать, вводить и редактировать данные таблиц при помощи форм.

Формы позволяют работать с данными в удобном формате; кроме того, в них можно добавлять функциональные элементы, например кнопки команд. Программным путем этим кнопкам можно назначить выполнение разнообразных задач, таких как определение данных, отображаемых в форме, или открытие других форм или отчетов. Например, можно создать форму с именем «Форма клиента» для работы с данными клиента. В форме клиента может присутствовать кнопка, открывающая форму заказа, в которой создается новый заказ для данного клиента.

Формы также позволяют задавать условия работы других пользователей с информацией, содержащейся в базе данных. Например, можно создать форму со строго ограниченным набором отображаемых полей и разрешенных операций. Это помогает защитить данные и гарантировать правильность их ввода.

## **Отчеты**

Отчеты служат для сбора и представления данных, содержащихся в таблицах. Обычно отчет позволяет ответить на определенный вопрос, например: «Сколько денег было получено от каждого клиента в этом году?», «В каких городах есть клиенты нашей компании?» Каждый отчет можно отформатировать так, чтобы представить сведения в наиболее удобном виде.

Отчет можно запустить в любое время, и он всегда будет отражать текущие сведения в базе данных. Обычно отчеты

форматируют для печати, но их можно также просматривать на экране, экспортировать в другую программу или отправлять в виде сообщений электронной почты.

## **Запросы**

Запросы являются основным рабочим инструментом базы данных и могут выполнять множество различных функций. Самая распространенная функция запросов — извлечение определенных данных из таблиц. Данные, которые необходимо просмотреть, как правило, находятся в нескольких таблицах; запросы позволяют представить их в одной таблице. Кроме того, поскольку обычно не требуется просматривать все записи сразу, с помощью запросов можно, задав ряд условий, «отфильтровать» только нужные записи. Часто запросы служат источником записей для форм и отчетов.

Некоторые запросы предусматривают возможность обновления: это означает, что данные в основных таблицах можно изменять через таблицу запроса. Работая с запросом с возможностью обновления, следует помнить, что изменения фактически вносятся не только в таблицу запросов, но и в соответствующие таблицы базы данных.

Существует два основных вида запросов: запросы на выборку и запросы на изменение. Запрос на выборку просто извлекает данные и дает возможность пользоваться ими. Результаты такого запроса можно просмотреть на экране, распечатать или скопировать в буфер обмена. Кроме того, их можно использовать в качестве источника записей для формы или отчета.

Запрос на изменение, как следует из его названия, выполняет действия с данными. Запросы на изменение можно использовать для создания новых таблиц, добавления данных в существующие таблицы, обновления или удаления данных.

## **Макросы**

Макросы в приложении Access можно рассматривать как упрощенный язык программирования, который позволяет

добавлять функциональные возможности в базу данных. Например, кнопке команды в форме можно назначить макрос, который будет запускаться при нажатии этой кнопки. Макрос содержит последовательность действий для выполнения определенной задачи, например для открытия отчета, выполнения запроса или закрытия базы данных. Большинство операций с базой данных, выполняемых вручную, можно автоматизировать с помощью макросов, которые позволяют существенно экономить время.

## **Модули**

Модули, как и макросы, являются объектами, которые можно использовать для добавления функциональных возможностей в базу данных. Макросы создаются в приложении Access путем выбора макрокоманд из списка, а модули пишутся на языке программирования. Модуль представляет собой набор описаний, операторов и процедур, которые хранятся в одном программном блоке.

## Лабораторная работа №1

### **СОЗДАНИЕ БАЗЫ ДАННЫХ СРЕДСТВАМИ СУБД ACCESS 2007**

#### **ЗАДАНИЕ**

##### *Задание 1*

Получите у преподавателя вариант задания. Создайте средствами СУБД ACCESS базу данных, соответствующую вашему варианту задания.

1.1 Создайте структуры трех таблиц из вашего варианта задания. *Данные в таблицы пока не вводите!*

1.2. Создайте схему данных.

##### *Задание 2*

Введите в каждую из таблиц вашей базы данных по 5 записей (используйте режим таблицы).

##### *Задание 3*

3.1. Проверьте обеспечение сущностной целостности данных. Попробуйте ввести повторяющиеся данные в ключевое поле любой из ваших таблиц. Как на это реагирует система?

3.2. Попробуйте ввести текстовые данные в числовое поле. Как на это реагирует система?

3.3. Проверьте обеспечение ссылочной целостности данных. Попробуйте удалить запись из таблицы связи. Как на это реагирует система?

#### **ТЕХНОЛОГИЯ РАБОТЫ**

Технологию создания базы данных продемонстрируем на примере создания базы данных **Заказ товаров по каталогу**, данные в которой будут храниться в трех таблицах : **КЛИЕНТЫ**, **ТОВАРЫ** и **ЗАКАЗЫ**.

Таблица Клиенты

Код клиента	Фамилия Имя Отчество	Адрес	Телефон



Таблица Товары

Код товара	Наименование	Размер	Цвет	Цена

Таблица Заказы

Код клиента	Код товара	Количество единиц товара	Дата приема заказа	Дата выполнения заказа

## Создание структуры базы данных

Работа с новой базой данных начинается с создания структуры базы данных. Для каждой таблицы задается **структура записи** (имена полей и типы данных в каждом поле). Структура записи – это «заготовка» будущей таблицы, её каркас. Для обеспечения целостности данных между таблицами задаются связи путем построения **Схемы данных**. После этого в таблицы вводятся данные.

Запускаем на выполнение ACCESS 2007 и видим на экране страницу **Приступая к работе с Microsoft Office Access**.

В разделе Новая пустая база данных выбираем команду Новая база данных.

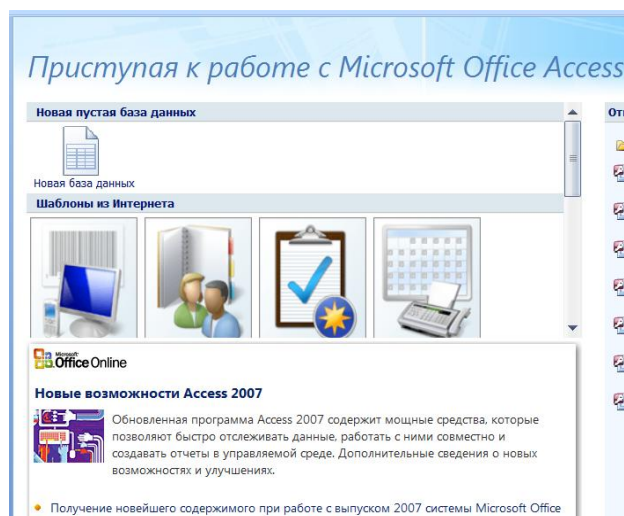


Рис.2. Страница Приступая к работе с Microsoft Office Access

В области Новая база данных справа от страницы Приступая к работе с Microsoft Office Access в поле Имя файла вводим имя файла.

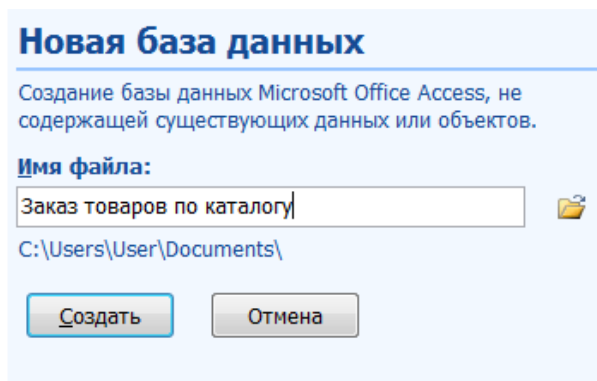



Рис.3. Ввод имени файла базы данных

Чтобы сохранить файл в своей папке, отличной от используемой по умолчанию, нажимаем кнопку Открыть  (рядом с полем Имя файла), переходим к нужной папке и нажимаем кнопку ОК, а затем - кнопку Создать.

Приложение Access создаст базу данных с пустой таблицей с именем «Таблица1» и открывает эту таблицу в режиме таблицы.

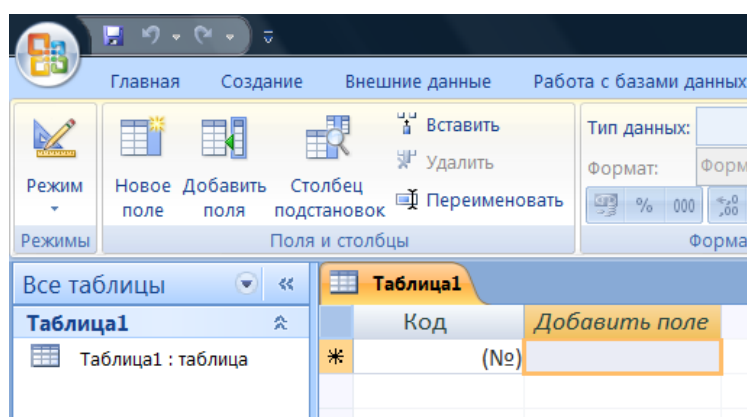
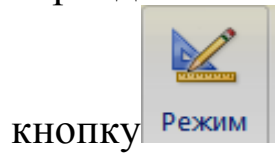


Рис.4. Таблица1 в режиме таблицы

Таблицы являются основой базы данных. В них хранятся данные. Начнем с создания таблицы КЛИЕНТЫ. Для этого

перейдем в режим Конструктора таблиц, нажав на



кнопку

Программа предлагает сохранить создаваемую таблицу

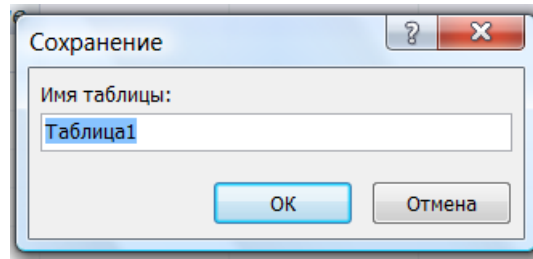


Рис.5. Сохранение таблицы

Введем в поле **Имя таблицы** вместо предлагаемого системой имени **Таблица1** имя **Клиенты**.

На экране появится бланк для ввода данных о структуре таблицы.

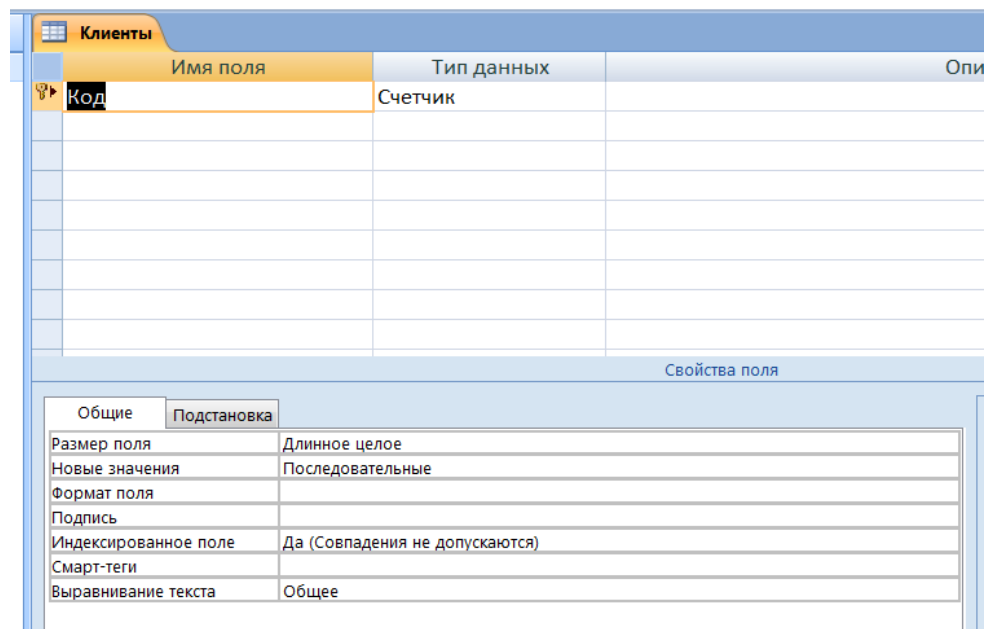


Рис.6. Бланк для ввода данных о структуре таблицы.

В системе ACCESS столбцы таблицы называются *полями*, а строки – *записями*. В таблице **КЛИЕНТЫ** 4 столбца, поэтому каждая запись данной таблицы будет состоять из 4-х полей.

Вместо предлагаемого программой имени первого поля **Код** вводим имя первого поля "Код клиента". Переходим в колонку **Тип данных** и, щелкнув на кнопке раскрытия списка типов полей, выбираем тип *Числовой*. Код клиента был введен в таблицу. Каждый клиент имеет уникальный код в базе данных. Программа автоматически делает это поле ключевым, о чем говорит изображение ключика.

**Ключевое поле** содержит значения, которые не повторяются, поэтому могут использоваться для однозначной идентификации записей о клиентах.

Переходим к следующей строке бланка конструктора и вводим имя поля **Фамилия Имя Отчество**. Программа предлагает текстовый тип данных. Это то, что нам нужно.

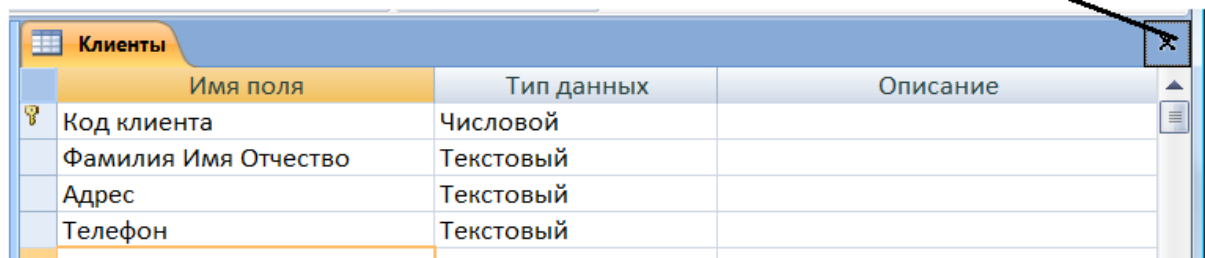
Остается ввести в следующие строки бланка имена полей таблицы **Адрес** и **Телефон**, для которых также подойдет текстовый тип данных.

	Имя поля	Тип данных
🔑	Код клиента	Числовой
	Фамилия Имя Отчество	Текстовый
	Адрес	Текстовый
	Телефон	Текстовый

Рис.7. Структура записи таблицы **КЛИЕНТЫ**

Мы ввели данные обо всех полях таблицы, поэтому заканчиваем создание таблицы, закрывая окно конструктора таблиц.

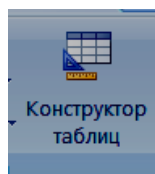
Кнопка, закрывающая окно конструктора таблиц



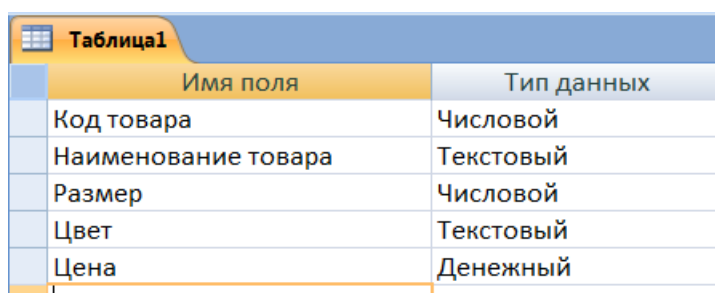
Имя поля	Тип данных	Описание
Код клиента	Числовой	
Фамилия Имя Отчество	Текстовый	
Адрес	Текстовый	
Телефон	Текстовый	

В появившемся диалоговом окне нажимаем кнопку «Да», чтобы сохранить созданную структуру таблицы **Клиенты**

Переходим к созданию таблицы **Товары**. Для этого надо выбрать на ленте вкладку **Создание**, а затем выбрать команду **Конструктор таблиц**



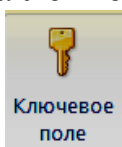
В появившийся бланк вводим сведения об именах полей и типах данных таблицы **Товары**:



Имя поля	Тип данных
Код товара	Числовой
Наименование товара	Текстовый
Размер	Числовой
Цвет	Текстовый
Цена	Денежный

Рис.8. Структура записи таблицы **Товары**

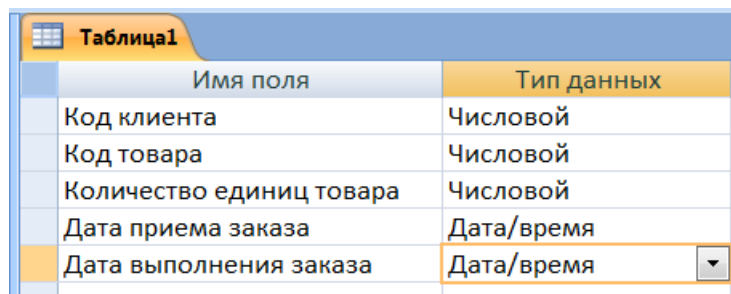
Поле **Код товара** надо назначить **ключевым** для данной таблицы. Для этого надо щелкнуть мышью слева от имени поля **Код товара**, а затем выбрать на ленте команду



Изображение ключика появится рядом с именем поля **Код товара**.

Закрываем окно конструктора таблиц и сохраняем таблицу с именем **Товары**.

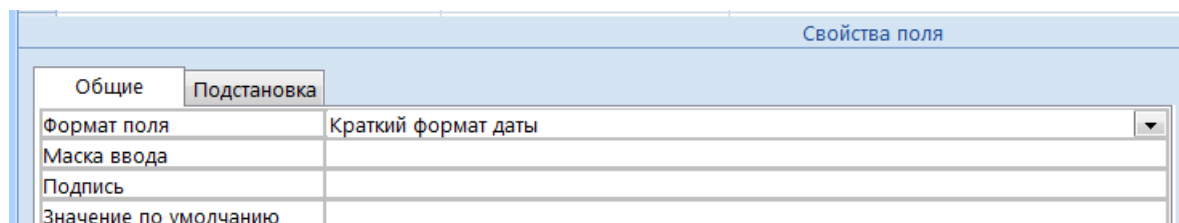
Нам осталось создать таблицу **Заказы**. Повторяем описанные выше действия, чтобы с помощью конструктора создать следующую структуру таблицы:



Имя поля	Тип данных
Код клиента	Числовой
Код товара	Числовой
Количество единиц товара	Числовой
Дата приема заказа	Дата/время
Дата выполнения заказа	Дата/время

Рис.9. Структура записи таблицы **Заказы**

Для полей **Дата приема заказа** и **Дата выполнения заказа** в разделе **Свойства поля** надо в строке **Формат** выбрать **Краткий формат даты**



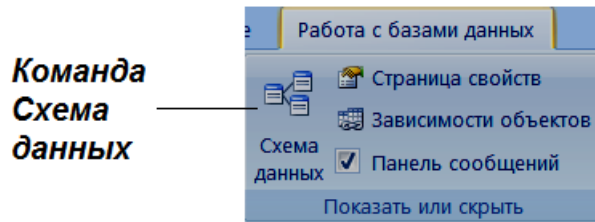
Свойства поля	
Общие	Подстановка
Формат поля	Краткий формат даты
Маска ввода	
Подпись	
Значение по умолчанию	

Закрываем окно конструктора и вводим в окне сохранения имя таблицы **Заказы**. У этой таблицы не будет ключевых полей, так как любой клиент может оформить множество заказов и заказать любые товары. Отказываемся от предложения системы создать ключевое поле и завершаем создание этой таблицы.

## Создание связей между таблицами

Для обеспечения целостности данных (см. «ВВЕДЕНИЕ») надо определить связи между таблицами нашей базы данных.

На ленте находим вкладку **Работа с базами данных** и выбираем пункт **Схема данных**.



Открывается диалоговое окно **Добавить таблицу**.

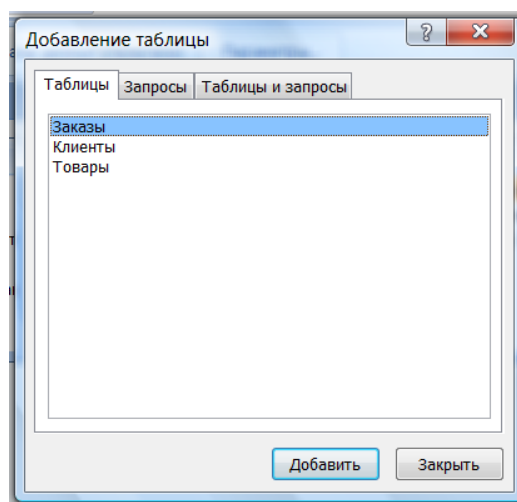
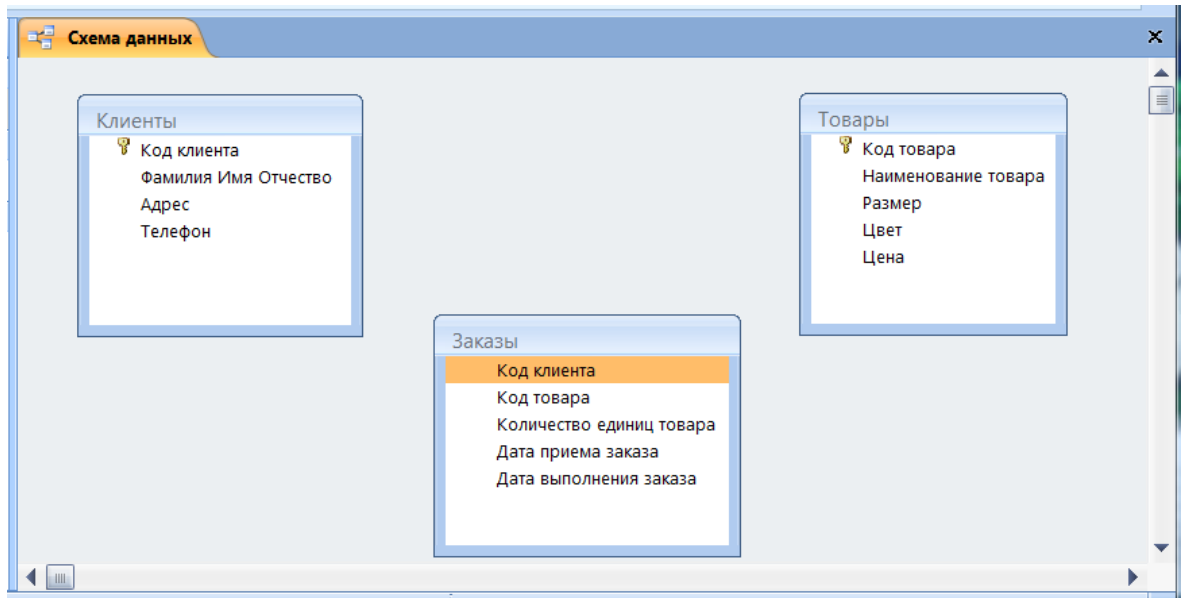


Рис.10. Добавление таблиц в схему данных

Выбираем первую по списку таблицу и нажимаем кнопку **Добавить**. Аналогично добавляем в схему данных 2 другие таблицы и нажимаем кнопку **Заккрыть**.



Ри.11. Размещение таблиц в окне «Схема данных»

Ставим курсор на ключевое поле **Код клиента** в таблице **Клиенты** и перетаскиваем его мышкой на поле с таким же именем в таблице **Заказы**. В появившемся диалоговом окне **Изменение связей** отмечаем галочками **Обеспечение целостности**, **Каскадное удаление** и **каскадное обновление** и нажимаем кнопку **Создать**.

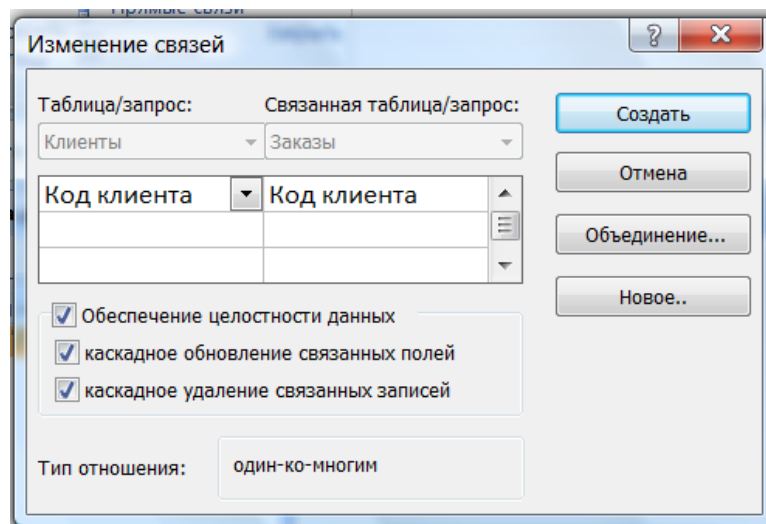


Рис.12. Окно Изменение связей



На схеме данных появляется графическое изображение связи типа «один-ко-многим» между таблицами **Клиенты** и **Заказы**.

Из таблицы **Товары** перетаскиваем ключевое поле **Код товара** на поле с таким же именем в таблице **Заказы** и подтверждаем условия обеспечения целостности.

В результате наших действий схема данных принимает следующий вид:

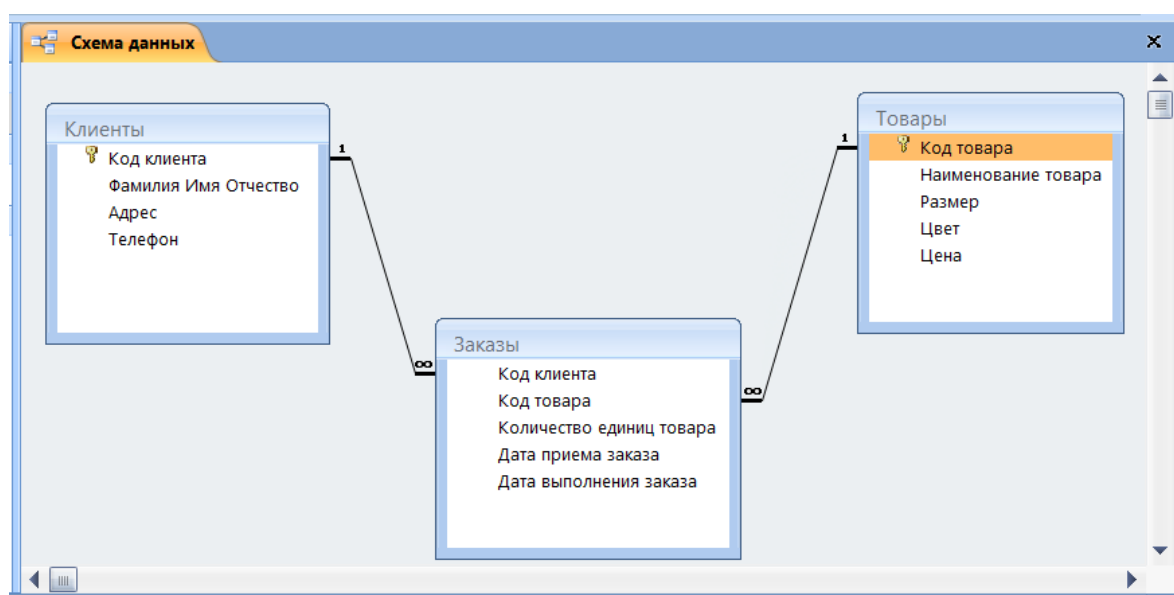


Рис.13. Схема данных


Закрываем окно **Схема данных** и подтверждаем ее сохранение.

### **Ввод данных в режиме таблицы**

С данными в системе ACCESS можно работать в двух основных режимах – в *режиме таблицы* и в *режиме формы*. В данной работе познакомимся с работой в *режиме таблицы*, который чем-то напоминает работу в электронных таблицах. Знакомство с формами запланировано в следующей лабораторной работе.

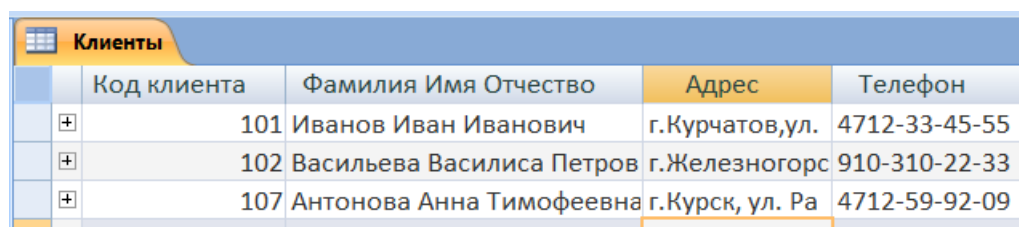
Мы задали связи между таблицами с автоматическим контролем целостности данных, поэтому в таблицу **Заказы** можно вводить только коды существующих в базе клиентов и

товаров. Следовательно, сначала надо заполнить таблицы **Клиенты** и **Товары**, а затем – в таблицу **Заказы**.

Мы видим на экране слева имена созданных нами таблиц. Для того, чтобы получить возможность ввести в таблицу данные, надо выбрать таблицу и дважды щелкнуть мышью по значку  .

На экране появится пустая таблица, состоящая из одной записи. Новые записи будут появляться по мере заполнения бланка.

Откроем таблицу **Клиенты** и введем несколько записей



	Код клиента	Фамилия Имя Отчество	Адрес	Телефон
+	101	Иванов Иван Иванович	г.Курчатов, ул.	4712-33-45-55
+	102	Васильева Василиса Петров	г.Железногорск	910-310-22-33
+	107	Антонова Анна Тимофеевна	г.Курск, ул. Ра	4712-59-92-09

Рис.14. Ввод записей в таблицу **Клиенты**

Обратите внимание на то, что система контролирует уникальность значений в ключевых полях и не допускает ввод повторяющихся значений. Система контролирует также и соответствие вводимых значений данных типам полей таблицы.

Заканчивается ввод данных закрытием окна таблицы и подтверждением сохранения данных.

## **Проверка обеспечения целостности данных**

### **Сущностная целостность**

Попробуем ввести повторяющиеся значения в ключевое поле таблицы. Система выдаст сообщение о недопустимости ввода повторяющихся значений в ключевое поле.

Попробуем ввести текстовые данные в числовое поле. Система выдаст сообщение о несоответствии типов вводимого значения и поля таблицы.

## Ссылочная целостность

Попробуем удалить одну из записей в таблице о любом объекте, информация о котором содержится также и в таблице связи. Например, нам потребовалось удалить информацию о клиенте с кодом 110. Открываем таблицу КЛИЕНТЫ, выделяем первую запись, нажимаем правую кнопку мыши и в появившемся контекстном меню выбираем команду. **Удалить запись**. Система выдает сообщение о невозможности такого удаления

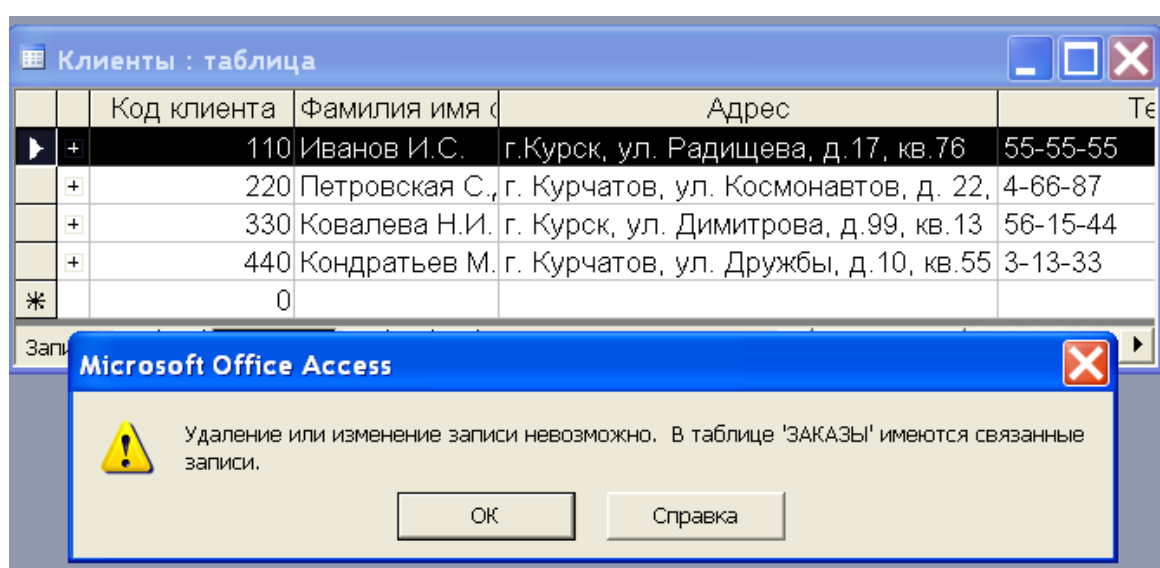


Рис.15. Сообщение о нарушении целостности при попытке удаления записи

Для того чтобы действительно удалить из базы информацию о данном клиенте, надо сначала удалить все связанные с ним записи из таблицы ЗАКАЗЫ.

## Лабораторная работа №2

### РАБОТА С ФОРМАМИ

**Форма** — это объект базы данных, который можно использовать для ввода, изменения или отображения данных из таблицы или запроса. Формы могут применяться для управления доступом к данным: с их помощью можно определять, какие поля или строки данных будут отображаться. Например, некоторым пользователям достаточно видеть лишь несколько полей большой таблицы. Если предоставить им форму, содержащую только нужные им поля, это облегчит для них использование базы данных. Для автоматизации часто выполняемых действий в форму можно добавить кнопки и другие функциональные элементы.

Формы можно рассматривать как окна, через которые пользователи могут просматривать и изменять базу данных. Рационально построенная форма ускоряет работу с базой данных, поскольку пользователям не требуется искать то, что им нужно. Внешне привлекательная форма делает работу с базой данных более приятной и эффективной, кроме того, она может помочь в предотвращении неверного ввода данных.

### Задание

1. Создайте для каждой из таблиц вашей базы форму.

**Примечание.** Для таблицы, отображающей связь между данными об объектах (аналогично таблице ЗАКАЗЫ) необходимо создать **форму с элементами управления типа «поле со списком»**.

2. Введите в каждую из ваших таблиц по 5 записей в режиме формы.

## ТЕХНОЛОГИЯ РАБОТЫ

Начнем с создания формы для работы с таблицей Клиенты. На ленте выбираем вкладку Создание. В группе Формы из раскрывающегося списка Другие формы выбираем Мастер форм

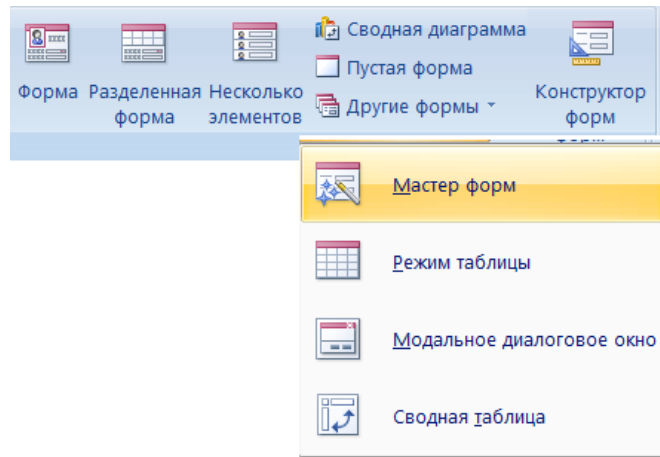


Рис.16. Вызов Мастера форм

В открывшемся окне Создание форм из списка Таблицы и запросы выбираем *Таблица: Клиенты*

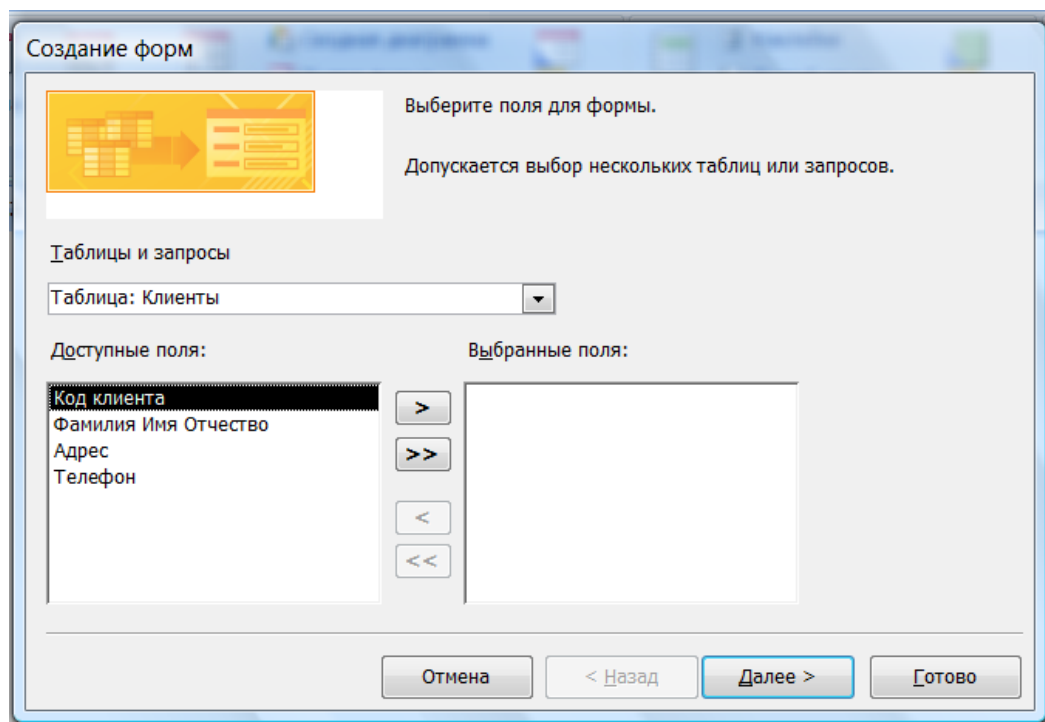

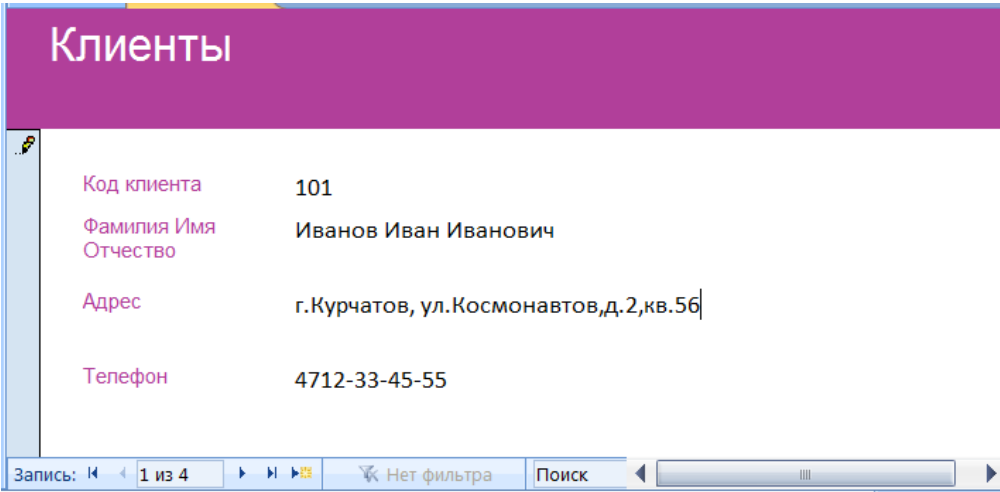


Рис.17. Выбор таблицы, для которой создается форма

Для того, чтобы поместить в форму все поля нашей таблицы, в этом же окне с помощью кнопки  переносим все поля из списка **Доступные поля** в список **Выбранные поля**. Нажимаем кнопку **Далее**.

На следующем шаге выбираем внешний вид в **один столбец** и нажимаем кнопку **Далее**. Выбираем любой понравившийся стиль (внешний вид), например, **Изящная** и переходим к следующему шагу. Остается задать имя формы (согласимся с предлагаемым программой именем **Клиенты**) и нажать кнопку **Готово**. Перед нами открывается готовая для работы форма:



The screenshot shows a software form titled "Клиенты" (Clients). The form contains the following fields:

Код клиента	101
Фамилия Имя Отчество	Иванов Иван Иванович
Адрес	г. Курчатов, ул. Космонавтов, д. 2, кв. 56
Телефон	4712-33-45-55

At the bottom of the form, there is a navigation bar with the text "Запись: 1 из 4", "Нет фильтра", and "Поиск".

Рис.18. Форма для таблицы Клиенты

Аналогичным образом создаем форму для таблицы **Товары**.

**Включение в форму элементов управления «поле со списком»**

При вводе данных в формы обычно легче выбрать значение из списка, чем вводить его по памяти. Кроме того, список вариантов гарантирует, что в поле будут вводиться допустимые значения. Записи таблицы **Заказы** включают в себя коды клиентов и коды товаров, которые предварительно должны быть введены в соответствующие таблицы. Было бы

удобно дать возможность работающему с базой данных человеку просто выбирать ранее введенные коды клиентов и товаров из раскрывающихся списков. Покажем, как это можно сделать.

Сначала надо создать для таблицы **Заказы** форму, аналогичную уже созданным нами для таблиц **Клиенты** и **Товары**. Особенностью формы является то, что она пока будет включать не все поля. Не будем пока включать в список отображаемых полей поля **Код клиента** и **Код товара**.

Рис.19. Предварительный вид формы

Добавим в созданную форму поле **Код клиента** в виде *поля со списком*. Для этого открытую на экране форму **Заказы** надо закрыть.

Наводим курсор на значок формы **Заказы** в области переходов, нажимаем правую кнопку мыши и в появившемся контекстном меню выбираем пункт **Конструктор**.

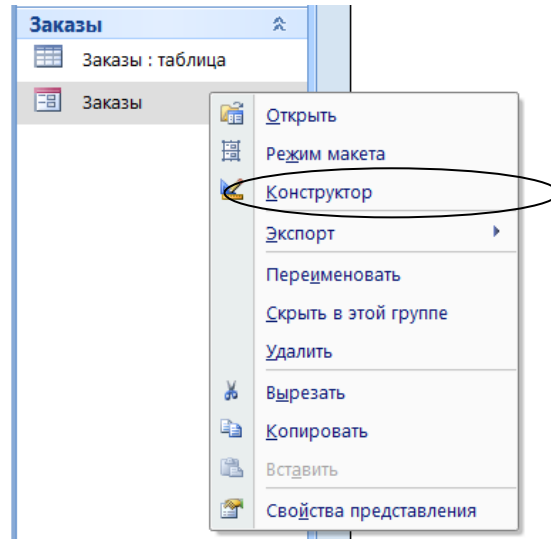


Рис.20. Вызов Конструктора форм

Открывается окно Конструктора форм с формой Заказы.

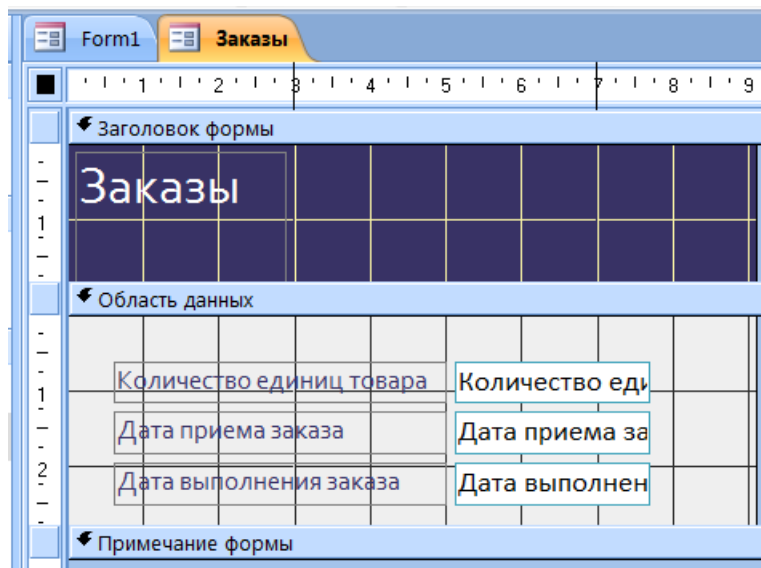


Рис.21. Окно Конструктора форм с формой Заказы

Надо приготовить на этой форме свободное место для размещения двух раскрывающихся списков. Для этого с помощью мыши надо несколько сдвинуть вниз все отображаемые поля.



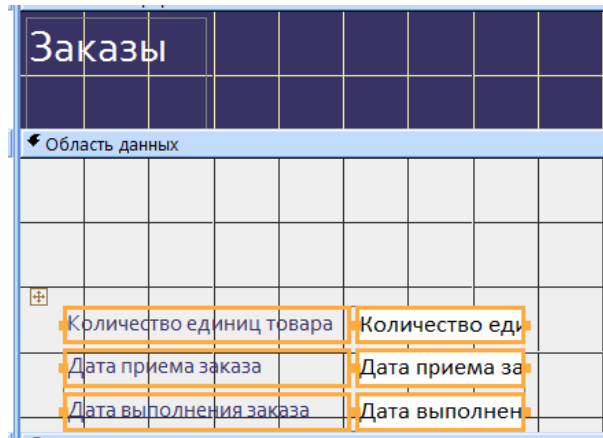


Рис.22. Подготовка места для раскрывающихся списков

На вкладке ленты **Конструктор** в группе **Элементы управления** находим элемент **Поле со списком**.

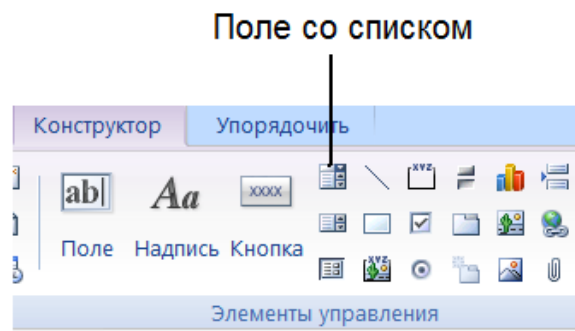


Рис.23. Элемент **Поле со списком** в группе **Элементы управления**

Щелкаем мышью по его изображению, а потом щелкаем по форме в месте, где мы хотим его расположить. Запускается **Мастер создания поля со списком**:

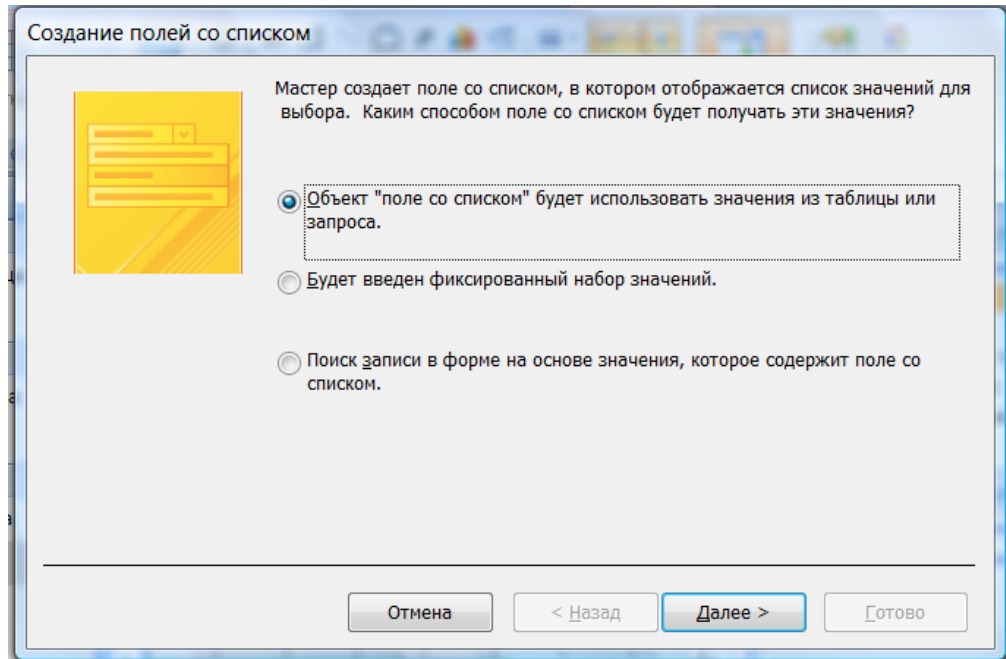


Рис.24. Окно Мастера создания поля со списком:

В нашем случае список будет получать значения из таблицы, поэтому соглашаемся с предложением программы, нажав кнопку **Далее**.

В следующем окне выбираем из списка таблиц таблицу **Клиенты** и нажимаем кнопку **Далее**.

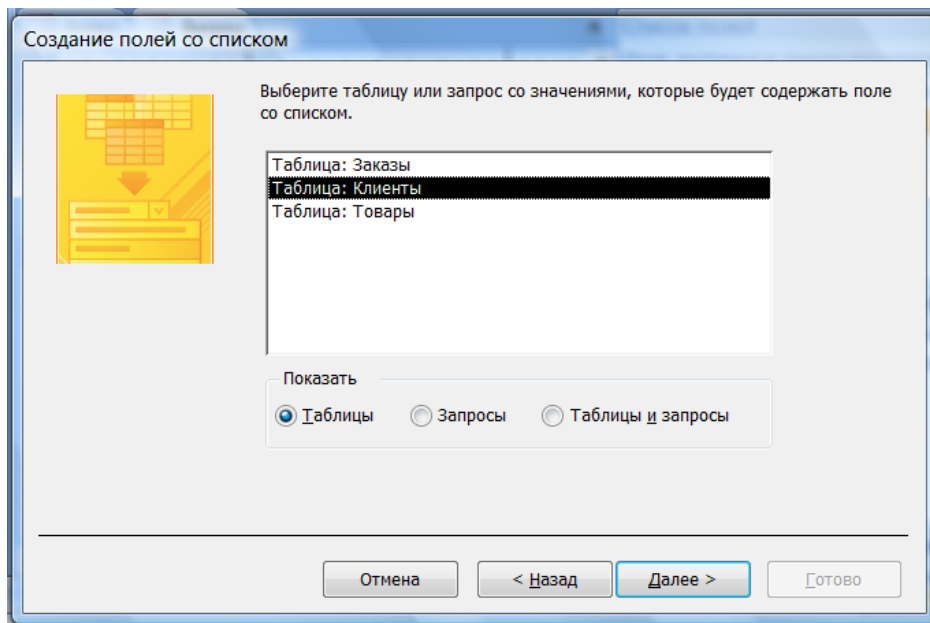


Рис.25. Выбор таблицы Клиенты

Из списка полей таблицы **Клиенты** выбираем поле **Код клиента**.

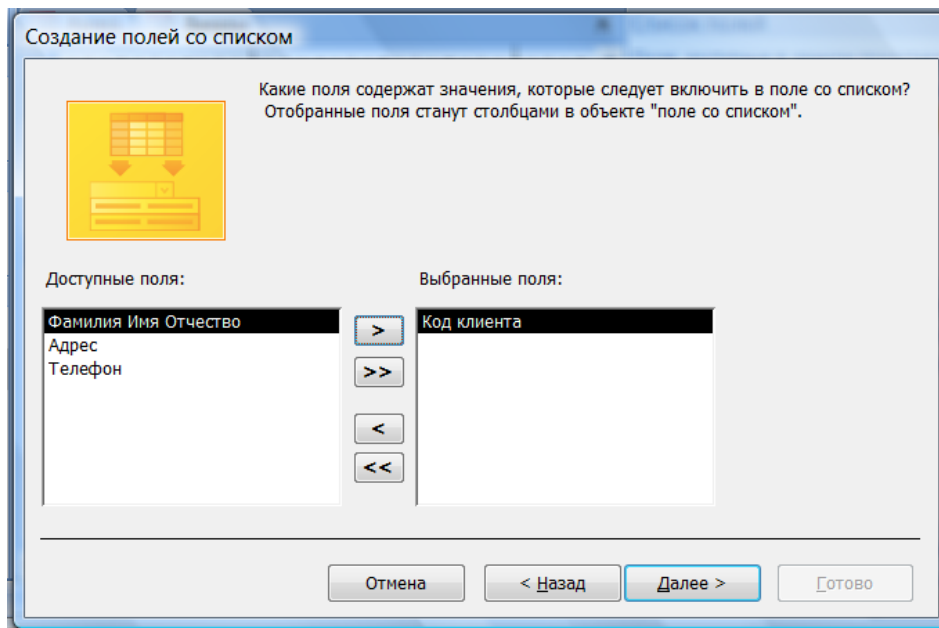


Рис.26. Выбор поля Код клиента

Значения кодов, удобнее выбирать из списка, если они упорядочены, например, по возрастанию. Надо выбрать из раскрывшегося в следующем окне списка поле **Код клиента** и порядок **По возрастанию**:



Рис.27.Задание порядка По возрастанию.

На следующем шаге Мастер предлагает установить ширину отображаемого поля данных

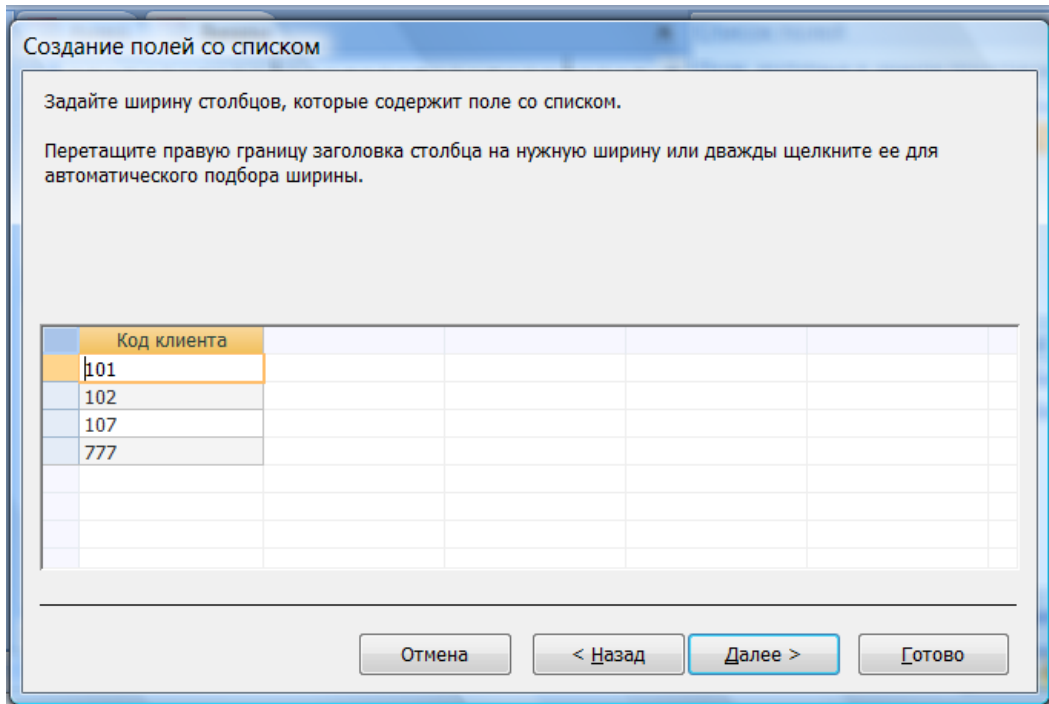


Рис.28. Установка ширины отображаемого поля данных

Выбираемые из списка значения должны сохраняться в поле **Код клиента** (см. рис.29).

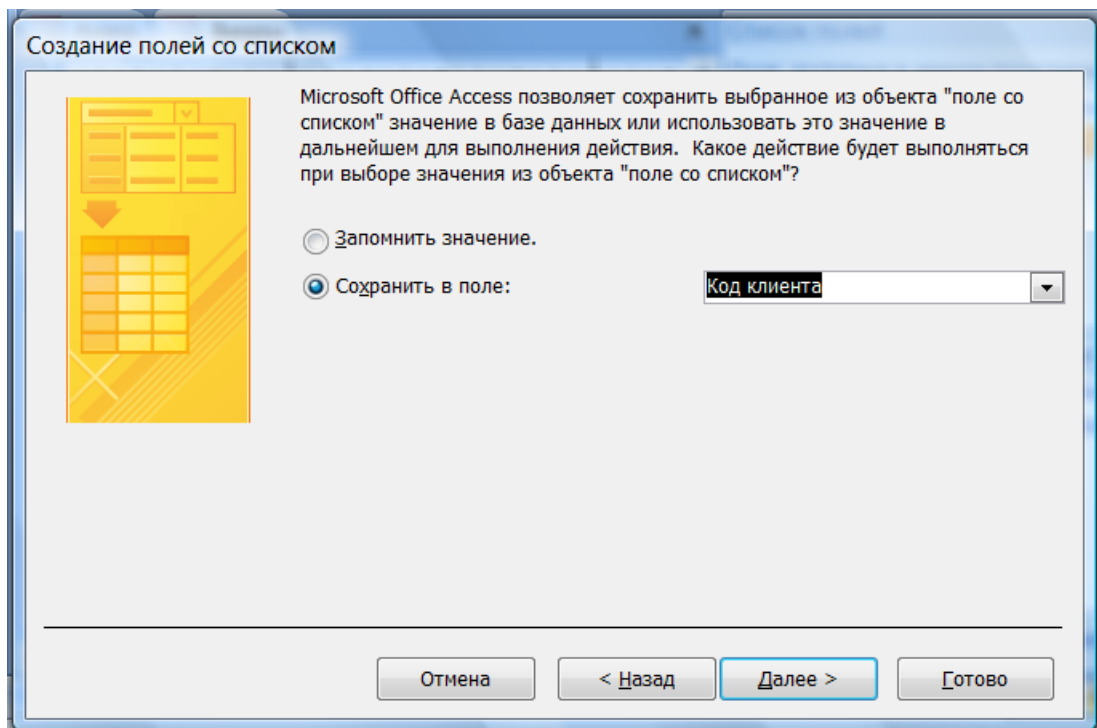


Рис.29. Задание имени поля для сохранения выбранного значения

Пользователю должно быть понятно, с каким полем он работает, поэтому в качестве подписи вводим имя поля **Код клиента**:

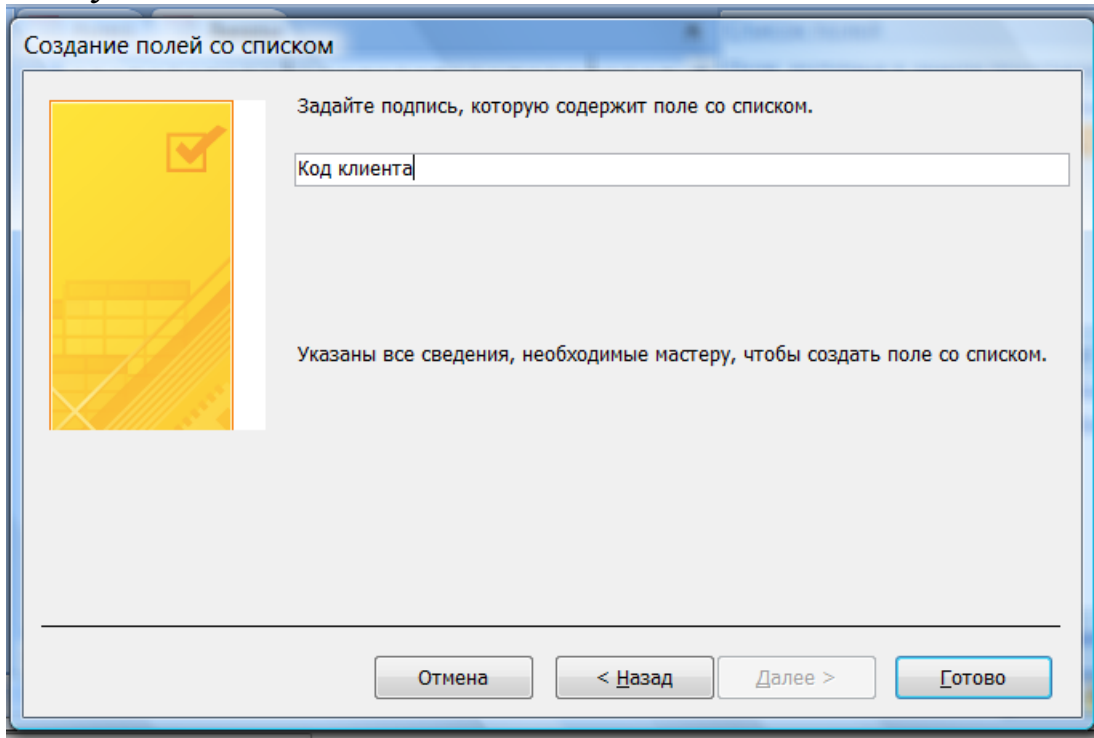


Рис.30. Ввод подписи для поля со списком

Нажимаем кнопку **Готово**. На форме в режиме конструктора отобразится добавленный нами элемент. Обычно требуется немного подвинуть вставленный список, чтобы была видна относящаяся к нему надпись.

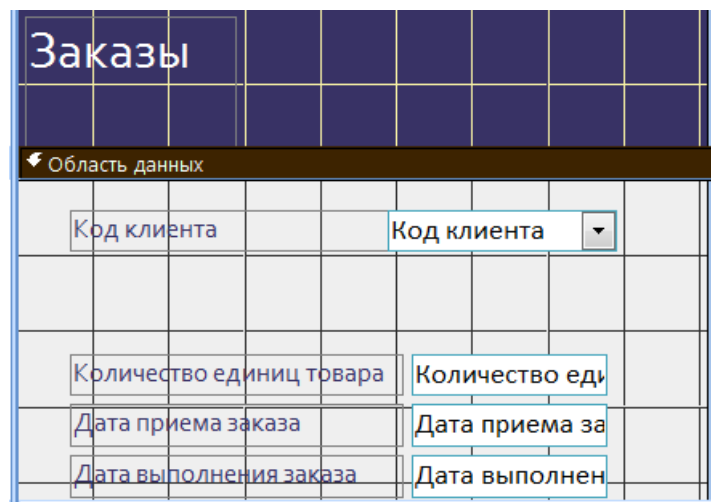


Рис.31. Готовая форма в режиме конструктора

По описанной технологии добавим на форму ещё одно поле со списком, связанное с полем **Код товара** из таблицы **Товары**.

Закрываем окно **Конструктора**. Двойным щелчком по значку формы **Заказы** открываем форму для работы с данными.

Заказы

Код клиента

Код товара

Количество единиц товара

Дата приема заказа

Дата выполнения заказа

Запись: 3 из 3 Нет фильтра Поиск

Рис. 32. Таблица **Заказы** в режиме формы

Теперь можно при добавлении новой записи о заказе выбирать из раскрывающихся списков **Код товара** и **Код клиента**. Кроме этого, для ввода дат можно воспользоваться встроенным календарем, который вызывается при щелчке мышью по полю типа **Дата**.

## Заказы

Код клиента

Код товара

Количество единиц товара

Дата приема заказа

Дата выполнения за

Декабрь 2011

Пн	Вт	Ср	Чт	Пт	Сб	Вс
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Сегодня

Запись: 3 из 3

Рис.33. Ввод даты с помощью встроенного календаря

## Лабораторная работа №3

### РАБОТА С ЗАПРОСАМИ

*Запросы* являются основным рабочим инструментом базы данных и могут выполнять множество различных функций. Самая распространенная функция запросов — извлечение определенных данных из таблиц. Данные, которые необходимо просмотреть, как правило, находятся в нескольких таблицах; запросы позволяют представить их в одной таблице. Кроме того, поскольку обычно не требуется просматривать все записи сразу, с помощью запросов можно, задав ряд условий, «отфильтровать» только нужные записи. Часто запросы служат источником записей для форм и отчетов.

Некоторые запросы предусматривают возможность обновления: это означает, что данные в основных таблицах можно изменять через таблицу запроса. Изменения фактически вносятся не только в таблицу запросов, но и в соответствующие таблицы базы данных.

Существует *два основных вида запросов*: запросы на выборку и запросы на изменение.

*Запрос на выборку* просто извлекает данные и дает возможность пользоваться ими. Результаты такого запроса можно просмотреть на экране, распечатать или скопировать в буфер обмена. Кроме того, их можно использовать в качестве источника записей для формы или отчета.

*Запрос на изменение* выполняет действия с данными. Запросы на изменение можно использовать для создания новых таблиц, добавления данных в существующие таблицы, обновления или удаления данных.

### **Задание**

Спроектировать и реализовать 9 запросов к своей базе данных. *Запросы по сложности должны быть аналогичными тем, которые приведены в качестве примеров в данных методических указаниях. Организовать поиск во всех таблицах вашей базы данных.*



Оформить в тетради отчет по данной лабораторной работе. Отчет должен содержать для каждого запроса 1) формулировку запроса на естественном языке, 2) зарисованный с экрана заполненный бланк запроса.

### ***Технология работы***

#### ***Создание запроса на выборку на основе одной таблицы***

***Пример 1***. Клиента фирмы интересует, какие товары 48-го размера можно заказать. Тогда запрос на выборку можно сформулировать так «**Вывести наименования товаров 48-го размера**».

#### ***Построения запроса с помощью конструктора***

На вкладке **Создание** в группе **Другие** выбираем **Конструктор запросов**.

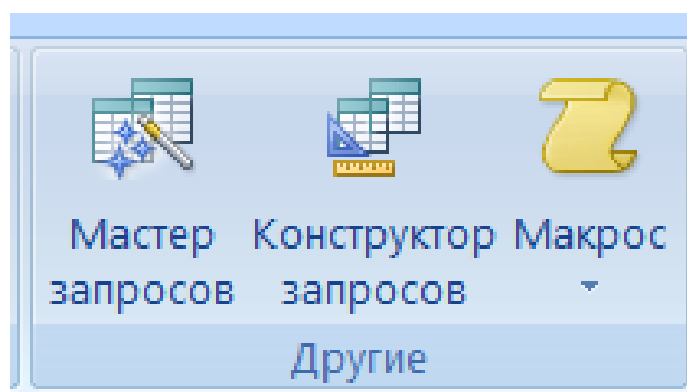


Рис. 34. Вызов Конструктора запросов

На экране появляется диалоговое окно **Добавление таблицы**. Информация о товарах хранится в таблице **Товары**,

поэтому выбираем таблицу и нажимаем кнопку **Добавить**, а затем – кнопку **Закреть**.

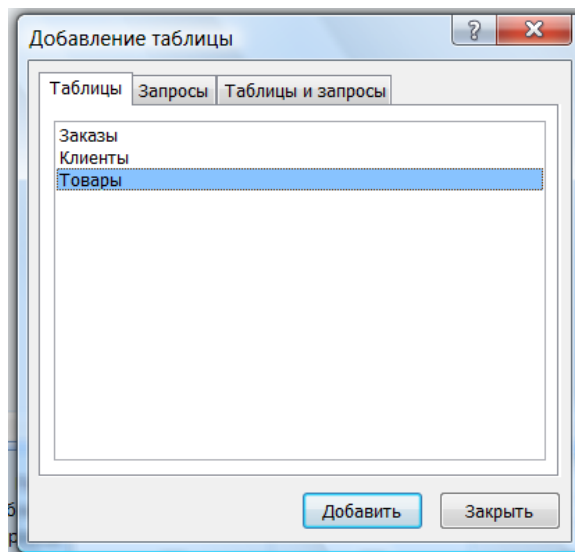


Рис.35. Добавление таблицы в запрос

Открывается окно конструктора запроса. Нижняя часть окна - **бланк запроса** – содержит описание запроса в табличной форме.

В первой колонке бланка запроса в строке **Поле** из раскрывающегося списка выбираем поле **Размер**. Нам надо выбрать только те записи, в которых значение поля **Размер** равно **48**. Для этого надо задать соответствующее условие отбора.

**Условие отбора** — это правило, определяющее, какие записи требуется включить в результаты запроса.

В строке **Условие отбора** вводим значение **48**. В таблице **ТОВАРЫ** содержатся разные данные о товарах. Нас интересуют только наименования товаров, поэтому во второй колонке в строке **Поле** из раскрывающегося списка выбираем поле **Наименование**

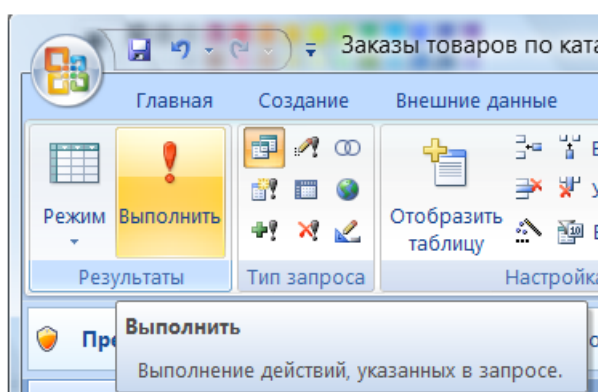
В результате наших действий бланк запроса принимает вид, показанный на рис.36.

Поле:	Размер	Наименование товара	
Имя таблицы:	Товары	Товары	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:	48		
или:			

Рис.36. Бланк запроса для примера 1

## Просмотр результатов выполнения запроса

Просмотреть результаты выборки данных можно, выбрав пункт **Выполнить** (изображение большого красного восклицательного знака) в группе **Результаты**.

Рис.37. Команда **Выполнить** запрос

Если в таблице **Товары** есть интересующие нас данные, то они будут выведены в виде таблицы, а если нет – результатом выполнения запроса будет пустая таблица.

## Сохранение запроса

Запрос можно использовать многократно, поэтому лучше его сохранить. Закрываем окно с результатами выборки и отвечаем утвердительно на вопрос о сохранении запроса. В поле

«имя» надо заменить предлагаемое системой имя **Запрос1** на любое другое, например, **Товары 48-го размера**.

**Пример 2. Вывести наименования товаров, цены на которые не превышают 1000 руб.**

Интересующая нас информация содержится в той же таблице **Товары**, но теперь условие отбора надо связать с полем **Цена**. Следуя описанной в первом примере технологии, создаем с помощью конструктора бланк запроса следующего вида.

Поле:	Цвет	Наименование товара	
Имя таблицы:	Товары	Товары	
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Условие отбора:	<=1000		
или:			

Рис.38. Бланк запроса для примера 2

В первом примере было использовано простейшее условие отбора – совпадение значения в указанном поле числового типа с заданной константой. Во втором примере использовано условие с использованием операции сравнения. Программа Access позволяет задавать разнообразные условия отбора с использованием символов операций сравнения **<**, **<=**, **>**, **>=**.

Примеры записи возможных условий отбора для полей числового и денежного типа приведены в Таблице 1.

Таблица 1

**Примеры записи условий отбора для полей типа  
Числовой и Денежный**

Записи	Условие	Результат запроса
Точно соответствуют определенному значению, например 1500	1500	Записи, в которых цена единицы товара составляет 1500 рублей.
Не соответствуют значению, например, 1000	Not 1000	Записи, в которых цена единицы товара не равна 1000 рублей.
Содержат значение, которое меньше заданного, например 2000	< 2000 <= 2000	Записи, в которых указана цена единицы меньше 2000 рублей (<2000). Второе выражение (<=2000) позволяет отобразить записи, в которых цена единицы меньше либо равна 2000.
Содержат значение, которое больше заданного, например 990,99	>990,99 >=990,99	Записи, в которых указана цена единицы больше 990,99 рублей (>990,99). Второе выражение позволяет отобразить записи, в которых цена единицы больше либо равна 990,99.
Содержат значение, которое входит в определенный диапазон	>490,99 and <990,99 -или- Between 490,99 and 990,99	Записи, в которых указана цена единицы в диапазоне между 490,99 и 990,99 рублей (сами эти значения не включаются в результаты).

**Пример 3. Вывести коды и адреса клиентов, фамилии которых, начинаются на букву «А».**

Нужная нам информация содержится в таблице Клиенты, а условие отбора необходимо связать с полем Фамилия. В первых двух запросах выборка данных производилась в полях числового типа. Теперь нам надо организовать выборку из поля текстового типа. Примеры записи условий для текстовых полей приведены в таблице 2. Бланк запроса для третьего примера показан на рисунке 39.

**Таблица 2**

**Примеры условий отбора для полей типа Текстовый**

Записи	Условие	Результат запроса
Точно соответствуют определенному	"Иванов"	Записи, в которых поле «Фамилия» содержит значение «Иванов».
Начинаются с заданной строки символов	Like С*	Записи, в которых фамилия клиента начинается с буквы «С», например, Семёнов, Савин и т.д. Используемая в выражении звездочка (*) является подстановочным знаком и заменяет любое количество любых символов. .
Заканчиваются заданной строкой,	Like "*ина"	Записи, в которых фамилии заканчиваются на «ина», например «Савина» или «Кирилина».
Входят в определенный диапазон,	Like "[А-Г]*"	Записи, в которых фамилия начинается с одной из букв от «А» до «Г».

Поле:	Фамилия	Имя	Отчество	Адрес	Телефон
Имя таблицы:	Клиенты	Клиенты	Клиенты	Клиенты	Клиенты
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Условие отбора:	Like A*				
или:					

Рис.39. Бланк запроса для примера 3

**Пример 4.** *Вывести коды товаров, заказы на которые должны быть выполнены в марте 2012 года.*

Условие отбора для поля Дата выполнения заказа включает диапазон значений дат от 01.03.07 до 31.03.07. Диапазон значений задается с помощью оператора BETWEEN (между). Признаком константы типа дата является разделитель "#". Бланк запроса имеет следующий вид:

Поле:	Дата выполнения заказа	Код товара
Имя таблицы:	Заказы	Заказы
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Between #01.03.2007# And #31.03.2007#	
или:		

Рис.40. Бланк запроса для примера 4

**Таблица 3**

**Примеры условий для полей типа «Дата/Время»**

Записи	Условие	Результат запроса
Точно соответствуют определенному значению, например 02.02.2006	#02.02.2006#	Значения даты должны быть окружены знаками #, чтобы Access мог отличить значения даты от текстовых строк.
Содержат значения, которые предшествуют определенной дате, например 02.02.2006	< #02.02.2006#	Записи об операциях, совершенных до 2 февраля 2006 г.
Содержат значения, которые следуют за определенной датой, например 02.02.2006	> #02.02.2006#	Записи об операциях, совершенных после 2 февраля 2006 г.
Содержат значения, которые входят в определенный диапазон дат	>#02.02.2006# and <#04.02.2006# ИЛИ Between #02.02.2006# and #04.02.2006#	Записи об операциях, совершенных между 2 и 4 февраля 2006 г.
Содержат текущую дату	Date()	Записи об операциях, совершенных на текущую дату.
Содержат вчерашнюю дату	Date()-1	Записи об операциях, совершенных за один день до текущей даты.
Содержат прошедшую дату	< Date()	Записи об операциях, совершенных до наступления текущей даты.



## Поиск по значениям двух полей

### Пример 5. Вывести коды и цены юбок 46-го размера.

В данном примере условия должны быть заданы для двух полей: Наименование товара и Размер, а выводиться на экран должны коды и цены соответствующих товаров. Бланк запроса показан на рисунке 41.

Поле:	Наименование товара	Размер	Код товара	Цена
Имя таблицы:	Товары	Товары	Товары	Товары
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	"юбка"	46		
или:				

Рис.41. Бланк запроса для примера 5.

Условия, заданные в соседних колонках бланка запроса, объединяются с использованием логической операции "И", то есть результатом выборки будут те записи, для которых все эти условия выполняются одновременно.

## Поиск по двум таблицам

### Пример 6. Вывести коды и наименования товаров, которые заказал клиент с кодом 100.

Сведения о кодах товаров, которые заказал клиент с определенным кодом, содержатся в таблице ЗАКАЗЫ, а наименования товаров находятся в таблице ТОВАРЫ. Следовательно, для ответа на вопрос надо организовать поиск по двум таблицам.

На вкладке Создание в группе Другие выбираем Конструктор запросов.

В окне Добавление таблицы добавляем 2 таблицы ЗАКАЗЫ и ТОВАРЫ. Между этими таблицами нами была

установлена связь, которая отображается в окне бланка запроса. Заполняем бланк запроса так, как показано на . рис.42:

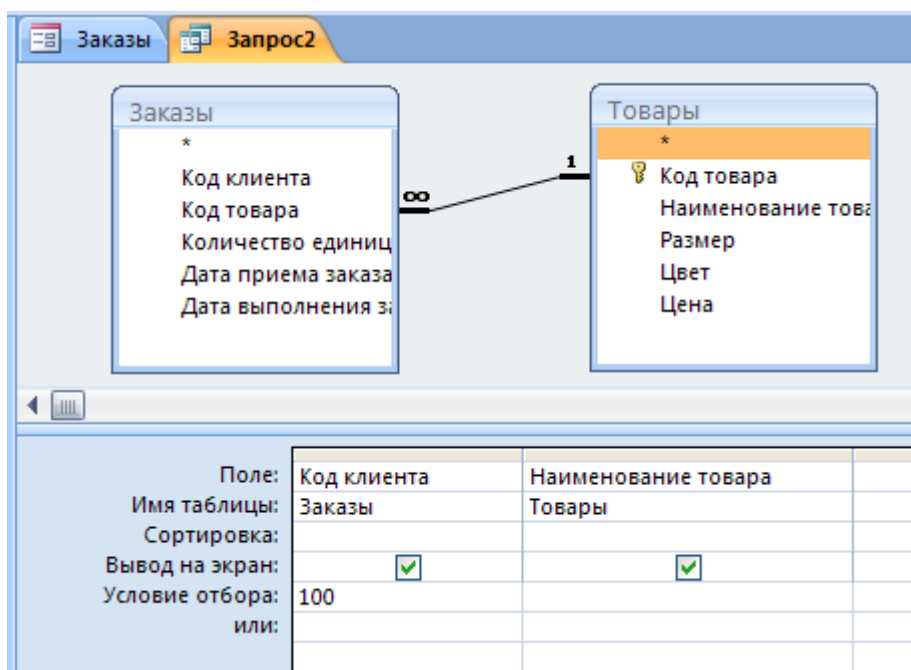


Рис.42. Бланк запроса для примера 6

## Запрос с параметром

**Пример 7. Вывести список наименований товаров, заказанных клиентом, номер которого вводится с клавиатуры.**

Запрос, в котором в условии отбора используется не константа, а введенное с клавиатуры значение, реализуется с помощью запроса с параметром. Запрос с параметром отличается от обычного запроса тем, что в поле УСЛОВИЕ ОТБОРА записывается не константа, по которой идет поиск по данному полю, а приглашение пользователю *ввести* значение для поиска. Это значит, что с помощью одного и того же запроса можно получать сведения о разных клиентах, вводя с клавиатуры соответствующие коды. Приглашение вводится в бланк запроса в виде текста, заключенного в квадратные скобки.

Таблицы включаются в запрос так же, как и в предыдущем примере. Бланк с приглашением для ввода показан на рисунке 43.

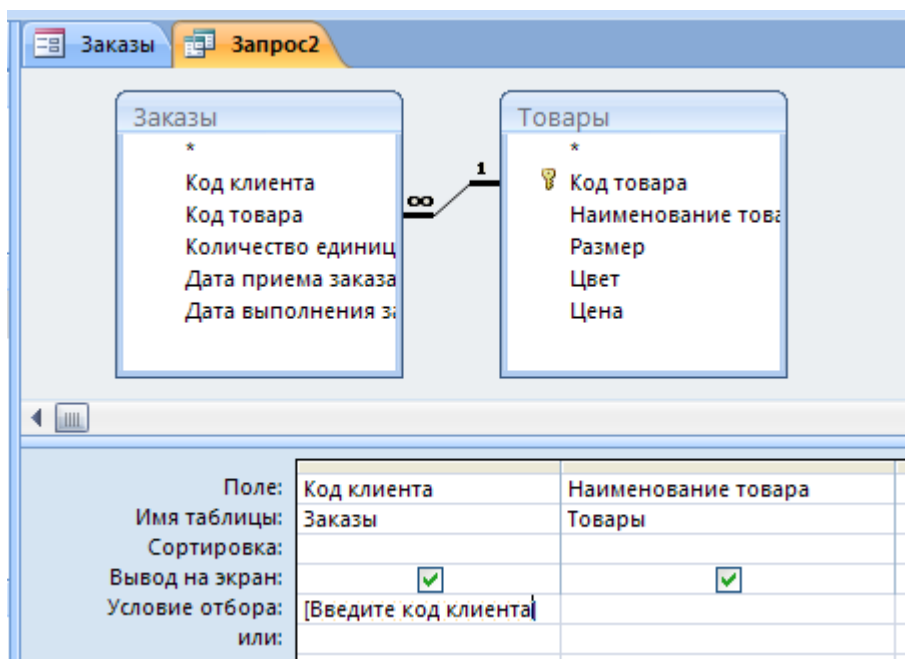


Рис.43. Бланк запроса для примера 7.

## Запрос с вычислениями

**Пример 8. *Определить стоимость каждого заказа.***

В таблице ЗАКАЗЫ есть сведения о количестве единиц заказанных товаров, но нет информации о стоимости заказа. Стоимость заказа можно вычислить, умножив число единиц товара на его цену. Вычисления можно произвести с помощью запроса с вычислением. Результат вычисления помещается в вычисляемое поле, которое будет существовать только в запросе (в базовой таблице никаких изменений не происходит).

На вкладке **Создание** в группе **Другие** выбираем **Конструктор запросов**.

В окне **Добавление таблицы** добавляем 2 таблицы **ЗАКАЗЫ** и **ТОВАРЫ**. Между этими таблицами нами была установлена связь, которая отображается в окне бланка запроса.

В первых 3-х колонках бланка запроса устанавливаем имена полей Код клиента, Код товара, Количество единиц товара.

В четвертую колонку бланка вводим формулу

Стоимость заказа: [Количество единиц товара]\*[Цена].

Получаем бланк как на рисунке 44.

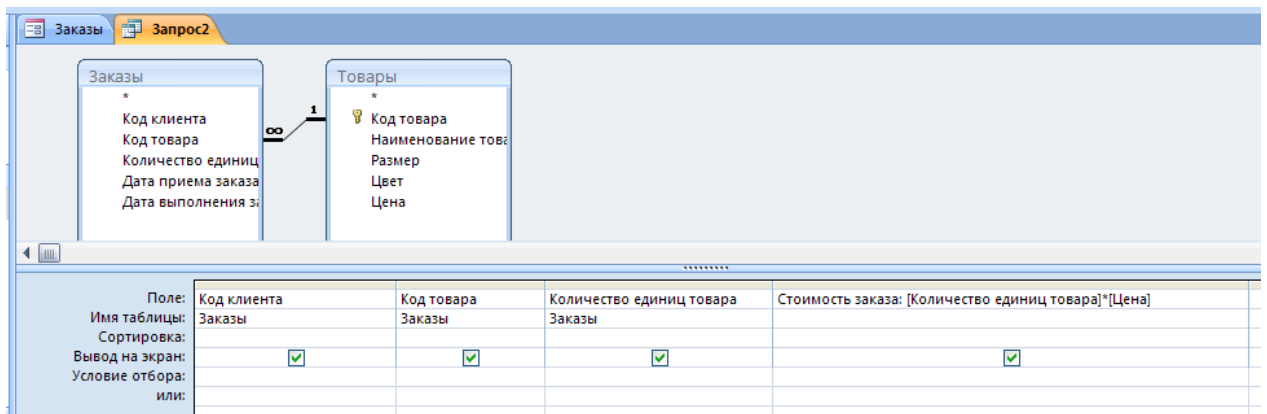


Рис.44. Бланк запроса для примера 8

## Итоговый запрос

**Пример 9.** *Определить суммарную стоимость заказов клиента с кодом 110.*

Перед нами стоит задача: найти суммарную стоимость заказов конкретного клиента. Сведения о стоимости заказов содержатся не в базовой таблице, а в *запросе* **Стоимость заказов**. Поэтому входим в режим создания запроса с помощью конструктора и в окне «Добавление таблицы» выбираем вкладку **Запросы**. Из появившегося списка запросов выбираем **Стоимость заказов** и закрываем данное окно.

В первой колонке бланка запроса в строке **Поле** из списка полей выбираем **Код клиента**.

В строке **Условие отбора** для данного поля вводим код клиента 110.

Во второй колонке бланка запроса в строке **Поле** из списка полей выбираем **Стоимость заказа**.

Просуммировать значения в поле **Стоимость заказа** можно путем добавления в запрос **строки итогов**. Для этого сначала надо выполнить запрос (нажав кнопку с изображением красного восклицательного знака). Запрос становится доступным в режиме таблицы.

На вкладке **Начальная страница** в группе **Записи** выберите команду **Итоги**.

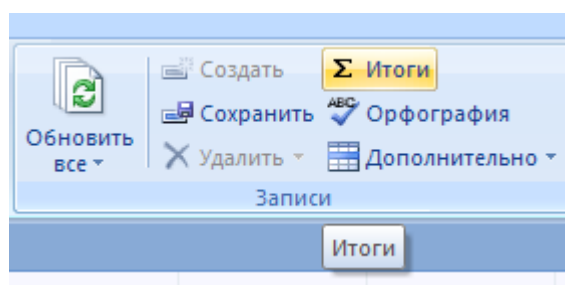


Рис.45. Команда **Итоги**.

В таблице появится новая строка **Итог**.

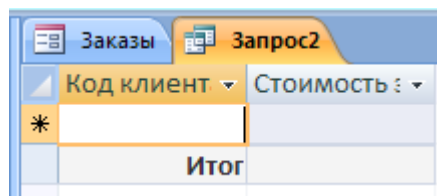


Рис.46. Добавление строки **ИТОГ** в бланк запроса

В строке **Итог** щелкните ячейку в поле, по которому необходимо провести суммирование, и выберите в списке функцию **Сумма**.

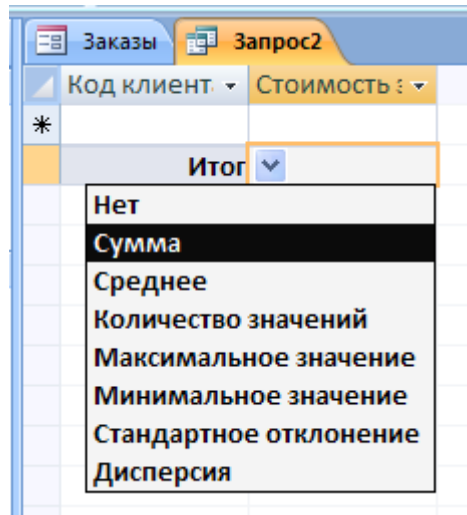


Рис.47. Выбор функции Сумма

На экране появится результат вычисления суммарной стоимости заказов клиента с кодом 100.