«Информационные системы в экономике»  
Семестр 2

**Лабораторные работы +Курсовой проект**

**«Работа с базами данных в MS Access 2007»**

**Подготовила А.А. Коробецкая**

***kornast@yandex.ru***

Лабораторный практикум содержит 5 частей, охватывающих базовые аспекты создания и обработки баз данных MS Access 2007.

**К сдаче** курсового предоставляется **база данных** (файл accdb) и **отчет** по курсовому проекту (файл docx). После проверки отчет необходимо распечатать. Не переводите бумагу, печатайте отчеты после проверки электронного варианта.

Для оформления использовать шаблон из предыдущего семестра. По каждому выполненному заданию необходимы скриншоты или текстовые копии таблиц с пояснением (1-2 предложения), что на них показано. В тексте методички указывается, что вставлять в отчет. Скриншоты должны быть достаточно крупными, читаемыми.

Помимо этой методички, пользуйтесь справкой и примерами Access (очень подробные) и ресурсами Интернет (их масса).

**Варианты** заданий выдаются преподавателем в индивидуальном порядке по электронной почте.

**Оценка** **за курсовой** выставляется по баллам.

Баллы:

Задание 1 – 0-15 баллов

Задание 2 – 0-5 баллов

Задание 3 – 0-20 баллов

Задание 4 – 0-10 баллов

Задание 5 – 0-10 баллов

*Всего*: 60 баллов.

Оценка:

25-39 баллов – удовл.

40-54 баллов – хор.

55-60 баллов – отл.

К **экзамену** необходимо также подготовить доклад-конспект на индивидуальную тему (устно или письменно 3-5 страниц). Тема выдается преподавателем и не связана с темой БД.

Содержание

[Задание на лабораторный практикум 3](#_Toc417878840)

[Пример индивидуального задания 3](#_Toc417878841)

[1 Создание и заполенение базы данных 4](#_Toc417878842)

[1.1 Разработка концепции БД 4](#_Toc417878843)

[1.2 Создание БД в Access 6](#_Toc417878844)

[1.3 Создание схемы данных 8](#_Toc417878845)

[1.4 Настройка полей таблиц 9](#_Toc417878846)

[1.5 Добавление описания таблиц в отчет 12](#_Toc417878847)

[1.6 Заполнение таблиц. 13](#_Toc417878848)

[2 Сортировка, фильтрация и поиск в таблице 13](#_Toc417878849)

[2.1 Сортировка 14](#_Toc417878850)

[2.2 Фильтрация 15](#_Toc417878851)

[2.3 Поиск данных 15](#_Toc417878852)

[3 Создание запросов 16](#_Toc417878853)

[3.1 Запрос на выборку полей из одной таблицы 17](#_Toc417878854)

[3.2 Запрос на выборку полей двух таблиц 18](#_Toc417878855)

[3.3 Запрос на выборку с различными критериями отбора 19](#_Toc417878856)

[3.4 Запрос с вычисляемыми полями 21](#_Toc417878857)

[3.5 Запрос с итогами (запрос с группировкой) 23](#_Toc417878858)

[3.6 Запрос с параметром 24](#_Toc417878859)

[3.7 Перекрестный запрос 26](#_Toc417878860)

[4 Создание форм 29](#_Toc417878861)

[4.1 Добавим автоформа в один столбец 29](#_Toc417878862)

[4.2 Ленточная автоформа 31](#_Toc417878863)

[4.3 Табличная автоформа 33](#_Toc417878864)

[4.4 Подчиненные формы 33](#_Toc417878865)

[4.5 Связанные формы 36](#_Toc417878866)

[5 Создание отчетов 37](#_Toc417878867)

[5.1 Автоотчет в один столбец 37](#_Toc417878868)

[5.2 Отчет из нескольких полей 41](#_Toc417878869)

[5.3 Отчет с группировкой и подведением итогов 44](#_Toc417878870)

# Задание на лабораторный практикум

Вариант индивидуального задания выдается преподавателем.

1. Разработать структуру базы данных (схема БД и описание таблиц).
2. Заполнить базу данных.
3. Осуществить сортировку, фильтрацию и поиск по таблицам.
4. Выполнить в базе данных запросы на выборку:

* запрос на выборку полей одной таблицы;
* запрос на выборку полей двух таблиц;
* запрос на выборку с различными критериями отбора;
* запрос с вычисляемыми полями;
* запрос с итогами;
* запрос с параметром;
* перекрестный запрос.

1. Создать в базе данных формы:

* автоформы в один столбец, табличную и ленточную;
* составные формы подчиненную и связанную.

1. Создать в базе данных отчеты:

* автоотчет в один столбец;
* отчет по нескольким полям одной таблицы;
* отчет с группировкой и итогами.

Результаты отражаются в отчете. Как добавить в отчет те или иные результаты, указано в методичке.

## Пример индивидуального задания

Создать базу данных «Продажи товаров». В базе данных хранятся сведения о товарах (артикул, наименование, цена, количество, единицы измерения), о магазинах (название, город, адрес, телефон, ФИО директора), о продажах (№ чека, дата, товар, кол-во, магазин), о наличии товаров в магазинах (товар, остаток на складе). Один товар может присутствовать в нескольких магазинах, в одном магазине продается много товаров. В одном чеке может быть много товаров, каждый товар может продаваться по разным чекам.

# Создание и заполенение базы данных

## Разработка концепции БД

Реляционная БД состоит из нескольких таблиц, связанных между собой. Разработка БД всегда начинается с составления перечня таблиц и определения связей между ними (связь по-английски *relation*, поэтому такие БД называются реляционными).

Каждая **таблица** описывает какой-то цельный, независимый объект. Например, «Товар», «Студент», «Клиент», «Продажа». Названия таблиц обычно являются существительными в единственном числе.

Строки таблиц в БД называются **записями**, а столбцы – **полями**.

Каждое **поле** показывает свойство (атрибут) объекта, который хранится в таблице. Например, свойствами товара являются «Наименование», «Цена», «Артикул». Свойствами студента могут быть «ФИО», «Пол», «Возраст» или «Дата рождения». Свойствами продажи товара могут быть «Дата продажи», «Товар», «Количество», «Стоимость». Обратите внимание, количество проданного товара не является свойством самого товара, а определяется в момент продажи. Ведь каждый раз мы может продавать разное количество товара. А вот «Единица измерения» (кг, л, шт. и т.д.) может быть свойством самого товара, ведь она у него всегда одинаковая.

Каждая **запись** соответствует одному конкретному объекту из таблицы и хранит значения его полей. Например, запись из таблицы «Товар» с полями «Артикул», «Наименование» , «Единица измерения», «Цена»:

|  |  |  |  |
| --- | --- | --- | --- |
| Артикул | Наименование | Ед. измерения | Цена |
| 00143502 | Перчатки детские | пара | 212,00 |

Изучите описание БД в своем индивидуальном задании. Описание таблиц в задании не является абсолютной истиной и может потребовать доработки. Попробуйте изобразить **примерную схему данных** (на бумаге). При необходимости уточните задание у преподавателя. Предложите варианты справочных таблиц.

На схеме необходимо указать таблицы и связи между ними.

**Связи** между таблицами на самом деле являются связями между полями двух разных таблиц. Связь означает, что в поле одной таблицы нужно подставлять значения из другой таблицы.

Например, в таблицу «Продажа» в поле «Товар» нужно подставить «Артикул» из таблицы «Товар». Значит, между этими полями есть связь.

Каждой связи нужно назначить **кратность** связей. В общем случае, кратность связи может быть:

|  |  |  |  |
| --- | --- | --- | --- |
| Обозн. | Название | Пояснение | Аналогия |
| 1-1 | один-к-одному | Одной записи одной таблицы соответствует одна запись другой таблицы (или ни одной) | Моногамия – у одного мужа одна жена |
| 1-n,  1-∞ | Один-ко-многим | Одной записи одной таблицы может соответствовать много записей другой таблицы, но у второй таблицы каждая запись соответствует только одной записи первой таблицы | Полигамия – у одного мужа может быть много жен |
| n-m,  ∞-∞ | Многие-ко-многим | Одной записи одной таблицы может соответствовать много записей другой таблицы, и у второй таблицы каждой записи соответствует много записей другой таблицы | У каждого мужчины может быть много жен и у каждой женщины много мужей |

Однако в реальных БД связь «многие-ко-многим» обычно не используется. Если на схеме есть связи m-n (многие-ко-многим), их необходимо преобразовать к 1-n (один-ко-многим), введя вспомогательные таблицы (см. пример ниже).

Проверка кратности связи в примере:

*Товар-Магазин (продажа товара)*: Один товар может продаваться во многих магазинах. И в одном магазине может продаваться много разных товаров.

Вывод: связь m-n. Ее необходимо «развязать», введя промежуточную таблицу Продажа. Каждая «Продажа» подразумевает факт реализации одного товара в одном магазине (а не целой покупки-чека с разными товарами). Т.е. один товар может встречаться во многих продажах, но в одной продаже может быть только один товар (1-n). В одном магазине может быть сделано много продаж, но каждая продажа осуществляется только в одном магазине (1-n).

Аналогично развязывается связь *Товар-Магазин* (*наличие*).

Дополнительно можно выделить ***справочники*** – дополнительные таблицы со стандартным содержанием, которое редко изменяется, но часто приходится вводить пользователю. Тогда удобнее не вводить эти данные каждый раз вручную, а выбирать из готового списка. Этот список и хранится в таблице-справочнике. В нашем примере в такую таблицу можно вынести Город. Тогда в БД нужно будет сразу заполнить список доступных городов, а пользователи смогут просто выбирать нужный город из выпадающего списка, а не вводить название вручную. Это упростит заполнение БД и поможет избежать ошибок и опечаток.

*Магазин-Город*: Каждый конкретный магазин (не торговая сеть, а отдельный) может находиться только в одном городе, но в каждом городе есть много магазинов (1-n).

Окончательный вариант схемы для примера:

**Товар**

**Магазин**

**Город**

**Продажа**

**Наличие**

1

n

n

1

1

1

1

n

n

n

В отчет схему вставлять не нужно.

В данном примере нет связи один-к-одному (1-1). Рассмотрим, каким мог бы быть пример такой связи:

Допустим, среди обычных товаров мы выделяем продовольственные продукты, которым нужно дополнительно указать срок годности и состав. Тогда мы добавим таблицу «Продукт». В ней будут указаны только сведения о продукте (без остальных сведениях о товаре) и она будет связана с товаром связью 1-1.

**Товар**

**Продукт**

1

1

Один продукт соответствует ровно одному товару. Один товар может быть или не быть одним продуктом (но не несколькими сразу).

Для каждой таблицы необходимо определить **перечень полей**.

**Товар** (Артикул, Наименование, Цена, Количество, Единицы измерения)

**Магазин** (Название, Город, Адрес, Телефон, ФИО директора)

**Продажа** (№ чека, Дата, Товар, Кол-во, Магазин)

**Наличие** (Товар, Остаток на складе)

**Город** (Название)

Все поля и таблицы называются существительными в единственном числе с заглавной буквы. Можно использовать общепринятые сокращения: Кол-во, Ед. изм. Пробелы лучше заменять на знак подчеркивания \_.

При необходимости, большие поля можно разбить на несколько, например ФИО на Фамилия, Имя, Отчество, а Адрес на Индекс, Город, Улица, Дом, Квартира/Офис.

В разных таблицах могут быть поля с одинаковым именем (у нас есть поле «Название» и у магазина, и у города). Чтобы их различать, имя поля употребляется после названия таблицы через точку:

Таблица.Поле

Магазин.Название

Город.Название

Товар.Цена

Если в названии столбца/таблицы есть пробелы, то оно берется в []:

Магазин.[ФИО директора]

Наличие.[Остаток на складе]

Далее, необходимо выбрать ключ для каждой таблицы.

**Ключ** – это такое поле таблицы, в котором значения никогда не повторяются, и по которому можно однозначно найти строку (запись).

Например, товар можно однозначно найти по его артикулу (не бывает двух товаров с одинаковым артикулом), но не всегда можно найти по наименованию (у нас может быть 2 товара с названием «Яблоки» или «Свитер детский», но разной ценой, цветом и т.п.).

Человека можно однозначно найти по номеру паспорта, отпечаткам пальцев, но нельзя по ФИО (бывают полные тезки).

Если мы рассматриваем студентов одного вуза, то ключом будет номер студ. билета. Если студенты из разных вузов, то у них могут повторяться номера студ.билетов, и использовать их как ключ нельзя.

Иногда в таблице может быть два или более ключа одновременно. Тогда выбирают более короткий (по значениям). Например, в паспорте 10 цифр, в номере сотового телефона 10 цифр, а в номере студенческого билета 6 цифр. Лучше выбрать студенческий билет.

Может возникнуть ситуация, когда среди столбцов таблицы нет ни одного ключа, или ключ слишком длинный. Тогда в таблицу добавляют столбец «Код». Это просто номер строки, добавленной в таблицу, без какого-либо смыслового содержания. Даже если какую-то строку удалили, значения Кода все равно увеличивается (работает как счетчик). Таким образом, его значения никогда не повторяются.

В перечне полей ключ нужно подчеркнуть.

Для нашего примера:

**Товар** (Артикул, Наименование, Цена, Количество, Единицы измерения)

**Магазин** (Код, Название, Город, Адрес, Телефон, ФИО директора)

**Продажа** (Код, № чека, Дата, Товар, Кол-во, Магазин)

**Наличие** (Код, Товар, Остаток на складе)

**Город** (Название)

Пояснения к выбору ключа:

Название магазина не может быть ключом, т.к. в разных городах могут быть магазины с одинаковым названием. Адрес тоже может совпадать. И директор у нескольких магазинов может быть один. Телефон – маловероятно, но возможно (например, у нескольких магазинов вообще не указан телефон).

Поэтому добавляем магазину код.

Аналогично для Продажи. Номер чека – это не ключ, в одном чеке может быть несколько продаж разных товаров.

В таблице Наличие нет даже претендентов на ключ. Оба поля могут повторяться.

В таблице Город будем считать, что все названия уникальны и возьмем их за ключ.

Последний шаг – нужно создать **связи** между таблицами. Связь идет между парой полей в двух таблицах: со стороны 1 (один) идет ключ, а со стороны n (много) – неключевое поле другой таблицы.

Например, Товар.Артикул – Наличие.Товар. Это означает, что в таблице «Наличие» нужно указать только код товара, а остальная информация, если понадобится, может быть найдена в таблице «Товар» по этому коду.

Аналогично:

Товар.Артикул – Продажа.Товар

Продажа.Магазин – Магазин.Код

Магазин.Город – Город.Название

Но для одной связи пара отсутствует. Это связь между Магазином и Наличием. Поэтому в таблицу Наличие нужно добавить столбец Магазин для обеспечения связи.

Для нашего примера:

**Товар** (Артикул, Наименование, Цена, Количество, Единицы измерения)

**Магазин** (Код, Название, Город, Адрес, Телефон, ФИО директора)

**Продажа** (Код\_продажи, № чека, Дата, Товар, Кол-во, Магазин)

**Наличие** (Код\_наличия, Товар, Магазин, Остаток на складе)

**Город** (Название)

Проверьте, чтобы у вас была пара полей для каждой связи.

## Создание БД в Access

MS Access является локальной СУБД. Вся база данных хранится в одном файле. Создайте в MS Access **новую базу данных** (пустую). Сохраните ее на локальный диск в формате accdb (в старых версиях mdb).

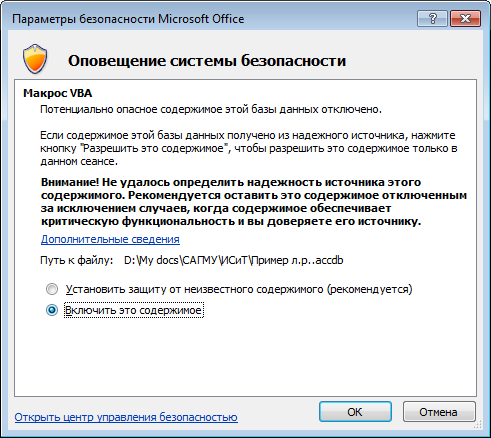
Не рекомендуется работать с БД, расположенной на съемном носителе или в сети. Это замедляет работу и увеличивает риск повреждения БД.

Важно

При загрузке ранее созданной БД, Access может выводить такое сообщение:



Нажмите кнопку «Параметры», выберите пункт «Включить это содержимое» и нажмите ОК.

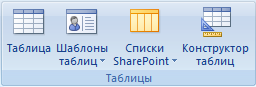


Пока вы не выполните это действия, возможности по работе с БД будут сильно ограничены.

**Создание таблиц** осуществляется через «Создание», группа кнопок «Таблица». С таблицами можно работать в двух основных режимах:

«*Таблица*» – ввод и редактирование данных.

«*Конструктор*» – формирование структуры таблице, определение полей, их типов и свойств.



По умолчанию база данных содержит одну таблицу «Таблица1». Переименуйте ее в соответствии со своим вариантом, или удалите и создайте новую таблицу.

По умолчанию новая таблица содержит 1 поле-счетчик для указания Кода.

Введите названия полей, укажите их типы. Подберите подходящий размер поля.

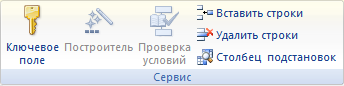
1. Названия таблицы и полей всегда пишутся в ед. числе.
2. В названиях допускаются буквы, цифры, знаки - и \_.
3. Если у нескольких таблиц есть поля с одинаковыми названиями, их желательно называть по-разному, например, *Код\_магазина*, *Код\_города*.

Каждому полю нужно выбрать подходящий **тип данных**. Тип данных определяет, что именно будет храниться в данном поле.

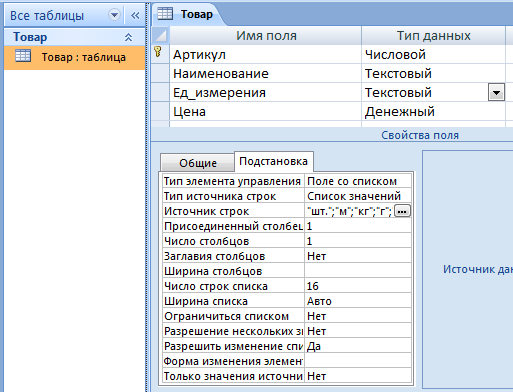
Основные типы данных Access:

* *числовой* – могут быть целыми или дробными (с плавающей точкой);
* *текстовый* – произвольная последовательность символов заданной максимальной длины (по умолчанию 255 символов);
* *поле МЕМО* – длинный многострочный текст;
* дата/время – позволяет хранить дату и/или время (в зависимости от вида);
* *денежный* – число с 2 цифрами после запятой и указанием валюты;
* *логический* – имеет ровно 2 значения (да/нет, вкл./выкл., муж./жен.);
* *поле объекта OLE* – внешний объект (файл), хранящийся в БД, например, рисунок или DOC-файл.

Укажите **ключевое поле** или поля.

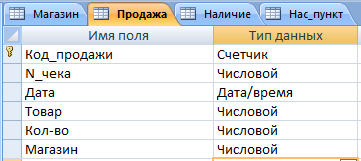
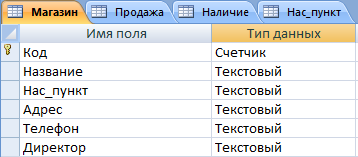


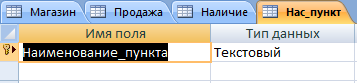
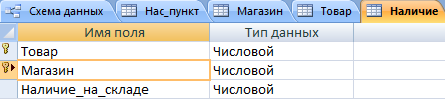
Для ключей-кодов используется тип «Счетчик». Это целочисленное поле, в которое при добавлении новой записи автоматически вводится следующее по порядку число. Таким образом, значения счетчика никогда не повторяются. Редактировать их вручную нельзя. Никакого смысла счетчик обычно не несет, и служит только для связывания таблиц, чтобы не использовать в качестве ключа поля с длинными значениями.



Аналогично создайте остальные таблицы.

В рассматриваемом примере мы добавили ключи «Код» типа «Счетчик» в таблицы «Продажа» и «Наличие». С другой стороны, ключи этих таблиц не участвуют в связях, поэтому отсутствие ключа или создание *ключа из нескольких полей* *не является ошибкой*.

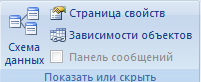




## Создание схемы данных

На схеме данных показываются связи между таблицами.

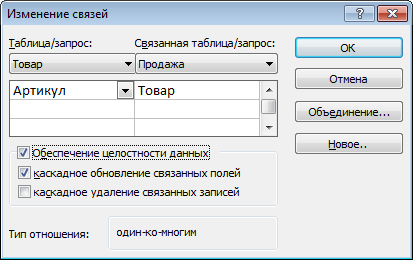
**Схема данных** создается через вкладку «Работа с базами данных» – «Показать или скрыть» – «Схема данных». Сами таблицы должны быть закрыты.



Добавьте на схему все таблицы БД. Разместите их на поле и добавьте связи.

Связь добавляется перетаскиванием поля одной таблицы к полю другой. Кратность связи задается автоматически. Типы данных связываемых полей должны совпадать (счетчик можно связать с длинным целым).

В свойствах связи установите флажки «Поддерживать целостность данных», «Каскадное обновление» и снимите флажок «Каскадное удаление».

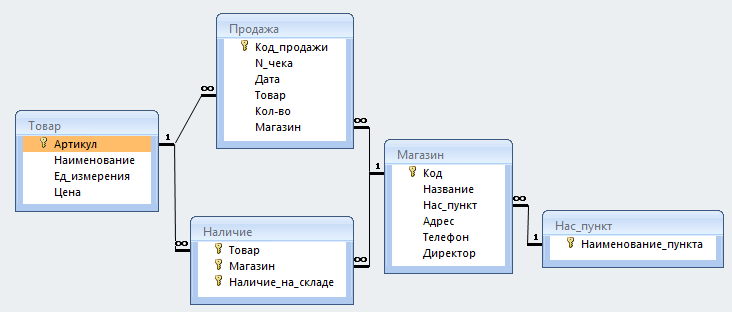


Это означает, что Access будет контролировать связываемые значения. Нельзя будет, например, продать несуществующий товар или создать магазин в несуществующем городе (которого нет в БД). А если поменять название города в таблице «Город», то оно поменяется и в таблице «Магазин».

Каскадное удаление может привести к серьезному повреждению базы при неосторожном удалении связанных записей. Например, если нечаянно удалить какой-то товар, то удалятся связанные с ним сведения о продажах и наличии в магазинах. Даже если восстановить строчку в таблице Товар, остальные данные сами не восстановятся.

Добавьте в отчет скриншот схемы данных. Он должен быть хорошо читаемым.

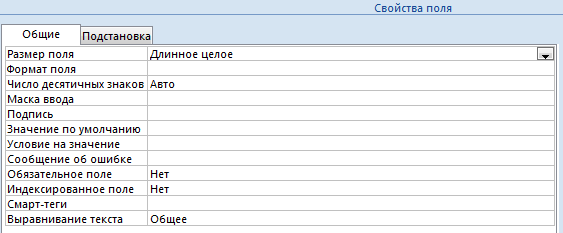
Схема данных для примера.



## Настройка полей таблиц

Кроме указания типа, каждое поле можно настроить, чтобы облегчить ввод данных и избежать ошибок.

Настройки полей находятся внизу в Конструкторе таблицы:



Там, где это возможно, задайте в свойствах полей:

* уникальность значений (могут ли значения поля повторяться в разных записях);
* значение по умолчанию;
* шаблон ввода, например для даты \_\_.\_\_.20\_\_;
* условия на значения, например, количество >=0;
* выпадающий список для выбора одного из постоянных значений или подстановки значения из справочника;
* другие свойства.

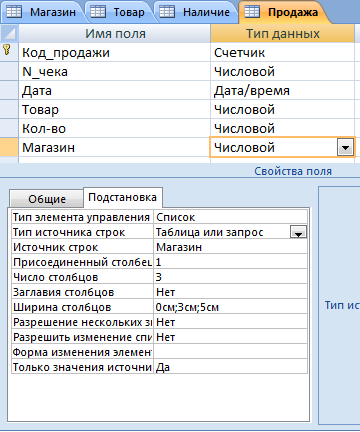
Для полей, которые участвуют в связях лучше использовать подстановку значений из списка. Это значительно упрощает ввод данных и позволяет избежать ошибок.

Список может содержать значения, введенные вручную, либо из другой таблицы или запроса.

Например, в таблицу «Наличие» код магазина и артикул товара желательно брать из соответствующих таблиц.

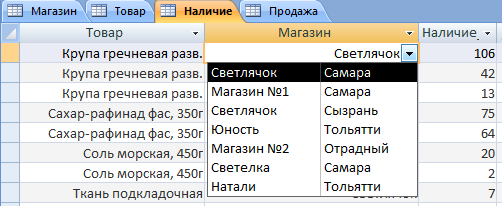
Удобнее всего то, что в списке может отображаться не только ничего не значащий столбец «Код», но и другие – «Наименование» и т.п.

Пример настройки подстановки значений в поле «Магазин» таблицы «Наличие»:

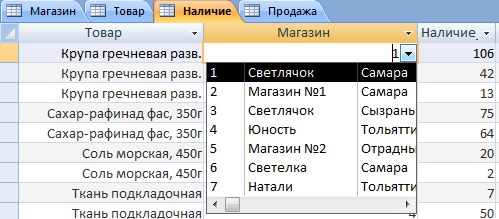


Обратите внимание на ширину столбцов: первый столбец (собственно, «Код\_магазина») имеет ширину 0. Он вообще не будет отображаться в списке, и даже в самом поле. Хотя реально в БД будет храниться именно код.

Результат в режиме таблицы:

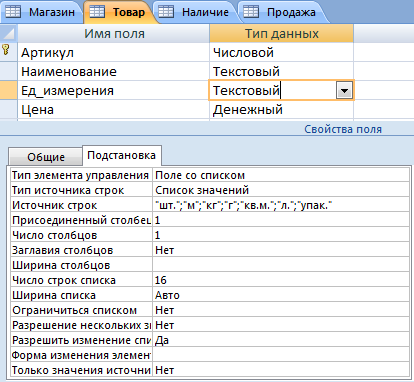


Если установить ширину первого столбца равной 1см, то таблица будет выглядеть так:



Можно построить и сложные запросы для выбора данных. Например, в таблице «Продажа» указывать не все товары, а только имеющиеся в наличии в данном магазине. Построение запросов рассматривается в л.р.3.

Пример подстановки из списка, введенного вручную, для поля «Ед\_измерения» таблицы «Товар»:

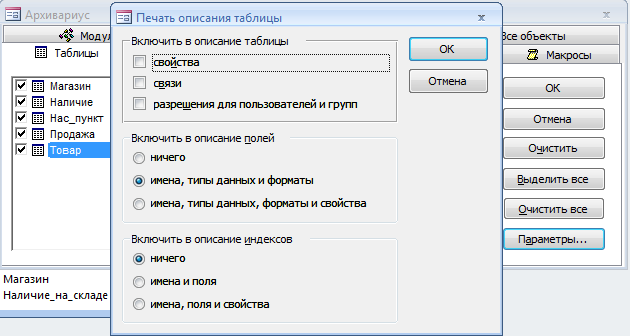


Здесь выбран тип элемента управления «Поле со списком», а не «Список», а свойство «Только значения источника» выставлено в «Нет». Это значит, что можно использовать и единицы измерения, не указанные в списке.

## Добавление описания таблиц в отчет

Выполняется полуавтоматически, через вкладку на ленте «Работа с базой данных» – «Архивариус».

Установите «Параметры», как показано на скриншоте. Отметьте флажками все таблицы.



Откроется окно с перечнем таблиц и их полей. Выполните экспорт в MS Word (формат rtf). Удалите лишние строки, разрывы, настройте оформление шрифта и абзаца, и добавьте результат в отчет. Выделите **жирным ключевые поля**.

Результат для примера:

**Таблица: Магазин**

*Имя Тип Размер*

**Код Длинное целое 4**

Название Текстовый 20

Нас\_пункт Текстовый 15

Адрес Текстовый 255

Телефон Текстовый 15

Директор Текстовый 255

**Таблица: Наличие**

*Имя Тип Размер*

**Товар Длинное целое 4**

**Магазин Длинное целое 4**

**Наличие\_на\_складе Длинное целое 4**

**Таблица: Нас\_пункт**

Имя Тип Размер

**Наименование\_пункта Текстовый 255**

**Таблица: Продажа**

*Имя Тип Размер*

**Код\_продажи Длинное целое 4**

N\_чека Длинное целое 4

Дата Дата/время 8

Товар Длинное целое 4

Кол-во Длинное целое 4

Магазин Длинное целое 4

**Таблица: Товар**

*Имя Тип Размер*

**Артикул Длинное целое 4**

Наименование Текстовый 255

Ед\_измерения Текстовый 8

Цена Денежный 8

## Заполнение таблиц.

Убедитесь, что схема данных, таблицы и их поля сформированы верно. Покажите их преподавателю. Только после этого приступайте к заполнению таблиц. Заполнение таблиц – самая долгая часть работы, т.к. вам придется придумать их содержание. Если вы ошиблись в схеме данных, то масса работы пройдет впустую.

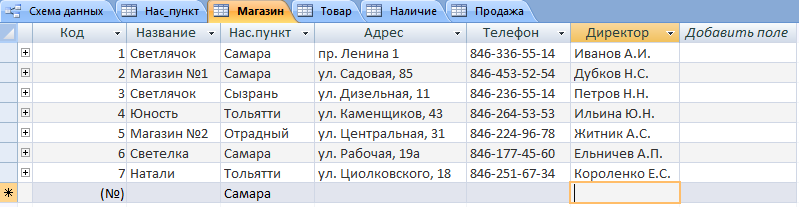
Последовательность заполнения: сначала заполняются таблицы на стороне связи 1, а потом – на стороне связи «много».

Необходимо заполнить:

* в таблицах на стороне связи «много» – не менее 12 записей,
* во таблицах на стороне связи «один» – не менее 5 записей,
* в справочниках – по необходимости.

При заполнении обязательно добавьте 2-3 повторяющихся значения на стороне «много» там, где это возможно.

Пример заполненной таблицы:



1. Экспорт заполненных таблиц в отчет выполняется через «Внешние данные» – «Экспорт» – «Экспорт в файл rtf».

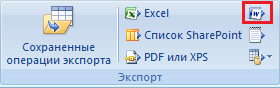


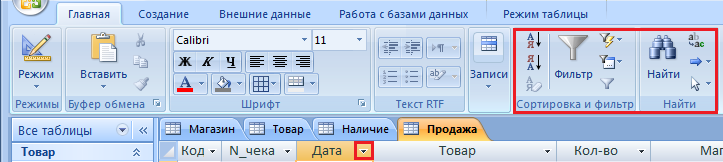
Таблица должна быть выделена или открыта.

# Сортировка, фильтрация и поиск в таблице

Реальные БД содержат десятки, сотни тысяч и даже миллионы записей. Просмотреть все эти записи вручную, чтобы найти нужную информацию, для человека почти невозможно.

Поэтому основными и одними из наиболее важных операций являются сортировка, фильтрация и поиск нужной записи. Они практически аналогичны сортировке и фильтрации в Excel.

Сортировка и фильтрация выполняется либо через вкладку «Главная», либо по клику непосредственно по наименованию поля в режиме «Таблица». Поиск также находится на вкладке «Главная», либо вызывается комбинацией клавиш Ctrl+F.



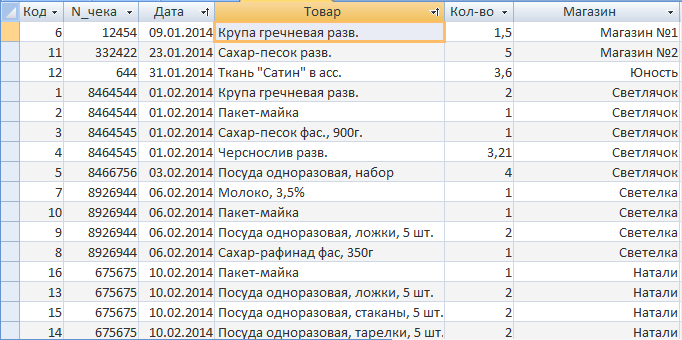
## Сортировка

Откройте одну из основных таблиц для сортировки.

В нашем примере выполним сортировку продаж сначала по дате (по возрастанию), затем по наименованию товара (по алфавиту).

Многоуровневая сортировка в Access отсутствует (точнее, она выполняется через запросы). Для сортировки по нескольким столбцам, она выполняется для каждого столбца в обратном порядке.

Т.е. в нашем примере необходимо сначала отсортировать продажи по наименованию товара, а потом по дате продажи. У отсортированных столбцов в заголовке появляется значок стрелки, показывающий направление сортировки:

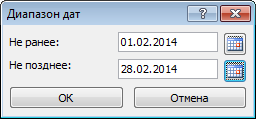


В отчете необходимо указать по каким полям выполнялась сортировка (не менее 2-х) и в каком направлении; вставить скриншот результата.

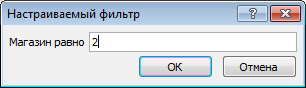
## Фильтрация

Выберите и откройте таблицу для фильтрации.

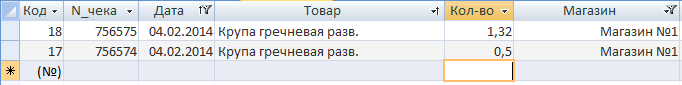
В нашем примере найдем все продажи в магазине №1 в феврале. Фильтр даты в столбце «Дата» можно указать диапазон дат с 01.02.14 по 31.02.14 или выбрать «В текущем месяце» (методичка составлялась в феврале 2014г.).



Фильтр поля «Магазин»:



Указывать надо фильтр для реально содержащихся в таблице данных, т.е. для кода магазина.



В отчет необходимо вставить скриншоты условий фильтрации и полученной таблицы. Фильтр желательно выбрать таким образом, чтобы в результате получилось несколько строк.

При закрытии таблицы фильтрация сбрасывается, а сортировка сохраняется.

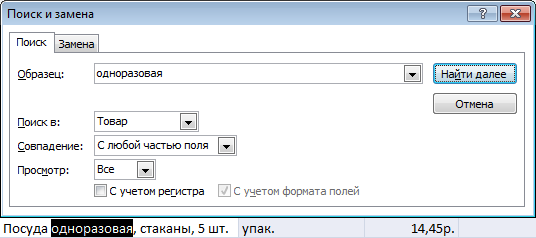
## Поиск данных

Отдельную запись в таблице помогает найти «Поиск». Во многом его функции пересекаются с фильтрацией. Основные отличия:

* + переход к найденной записи (можно сразу редактировать);
  + возможность поиска не в одном поле, а во всей таблице.

Также возможна автоматическая замена значений.

Пример: найти в таблице «Товар» записи, содержащие слово «одноразовая».



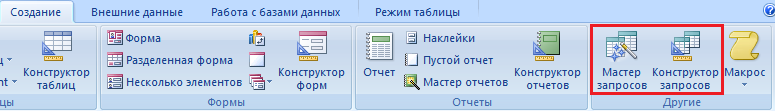
В отчет вставить скриншот окна поиска и найденной записи.

# Создание запросов

Запросы являются одним из основных инструментов работы с БД. Они позволяют выбирать из таблиц нужные данные, добавлять и удалять записи, редактировать содержимое таблиц, объединять таблицы, выполнять расчеты и многое другое.

На самом деле, почти любое действие с БД является запросом по сути. Просто в Access многие запросы автоматизированы, и пользователь их не видит. В предыдущих заданиях, когда вы просматривали таблицы, вводили в них данные, выполняли сортировку и фильтрацию, Access каждый раз формировал нужные запросы к БД.

Теперь составим запросы самостоятельно. Создание запроса осуществляется через вкладку «Создание» и возможно двумя способами: через «Мастер запросов» или «Конструктор». Мастер позволяет создавать несколько типовых запросов. Затем их всегда можно отредактировать в Конструкторе.



Существует большое число различных видов запросов. В данном практикуме мы выполним следующие:

* запрос на выборку полей одной таблицы;
* запрос на выборку полей двух таблиц;
* запрос на выборку с различными критериями отбора;
* запрос с вычисляемыми полями;
* запрос с итогами;
* запросы с параметром;
* перекрестный запрос.

Замечание: вам необходимо создать один запрос каждого типа.

По каждому запросу в отчет необходимо добавить два скриншота: вид запроса в Конструкторе и результат выполнения запроса.

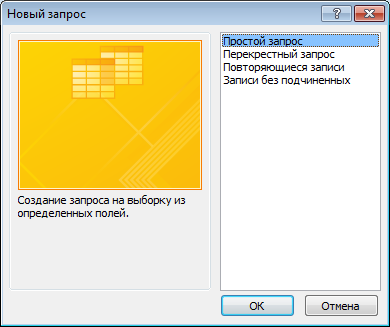
## Запрос на выборку полей из одной таблицы

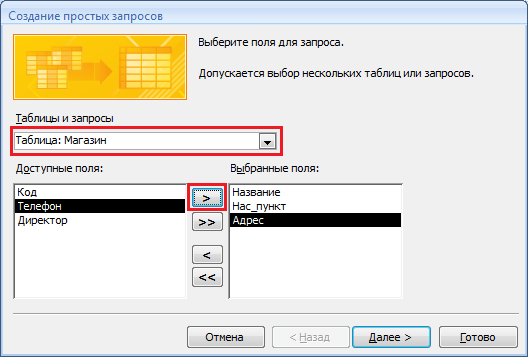
Это самый простой вид запросов. Он позволяет отобразить только те столбцы таблицы, которые нам нужны. Чаще всего они используются для выборок из больших таблиц, которые содержат 10 столбцов и более.

В нашем примере выполним выборку из таблицы «Магазин». Отобразим столбцы «Название», «Нас.пункт» и «Адрес».

Данный запрос можно выполнить как с помощью мастера, так и с помощью Конструктора.

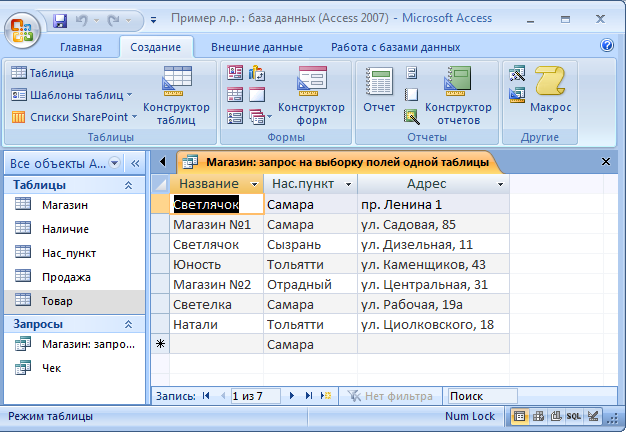
Мастер запросов:



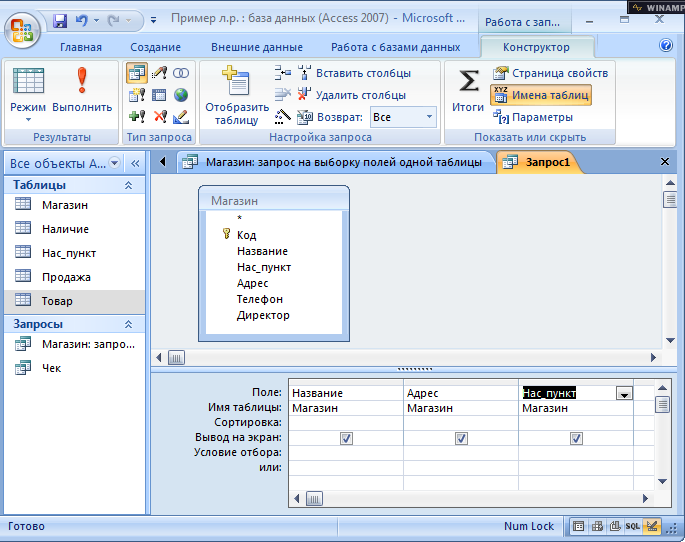


В последнем окне задайте имя запроса и нажмите «Готово». Название запроса должно отражать сущность содержащихся в нем данных: «Адреса магазинов».

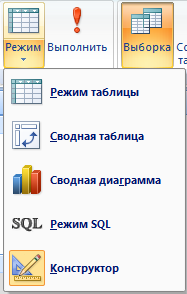
Результат:



Этот же запрос можно создать или изменить с помощью Конструктора. Запустите конструктор и добавьте таблицу, из которой будете делать запрос. Нужные столбцы добавляются двойным кликом или перетаскиванием.



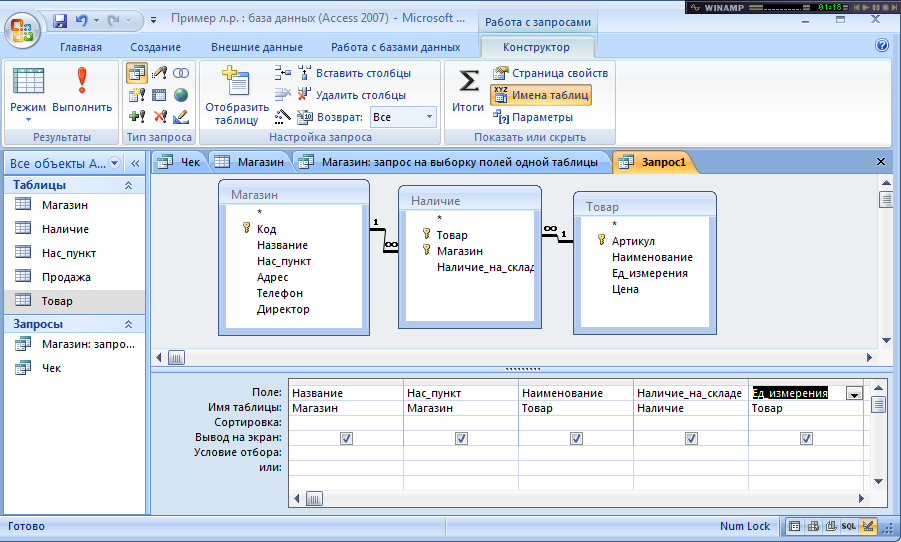
Для просмотра результата или возврата в Конструктор используйте кнопку «Режим» или «Выполнить» на ленте:



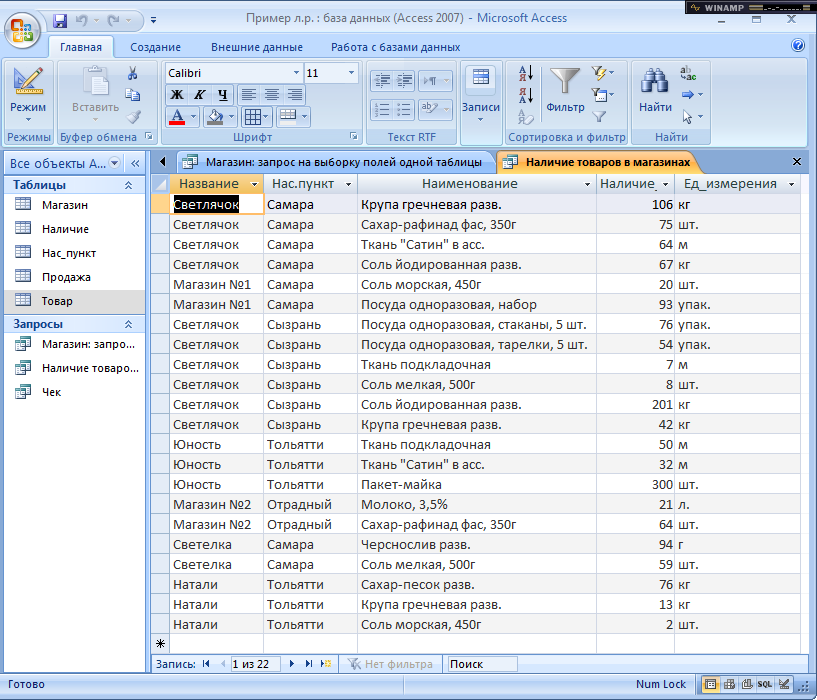
## Запрос на выборку полей двух таблиц

Выполняется полностью аналогично, за исключением того, что в запрос добавляются поля из двух или более связанных таблиц.

В нашем примере создадим запрос «Наличие товаров в магазинах», отобразим в запросе название магазина и населенный пункт, наименование и ед. измерения товара, количество на складе:



В результате получим:



## Запрос на выборку с различными критериями отбора

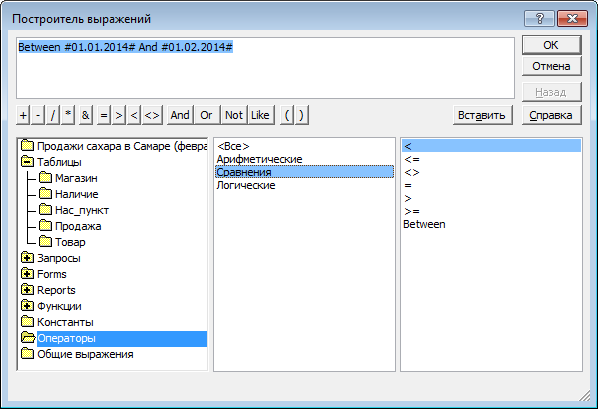
Напоминает фильтрацию данных: нам нужно отобразить не все записи, а только удовлетворяющие какому-то критерию.

Выберем магазины в Самаре, в которых в феврале продавался сахар (любой). Также отсортируем запрос по магазинам и количеству проданного сахара. Для этого запроса нам потребуются таблицы «Магазин», «Продажа», «Товар».

Требование: запрос должен состоять минимум из двух условий и возвращать не менее 2 записей! Сортировка обязательно минимум по 1 полю.

Критерии отбора указываются в Конструкторе в строке «Условие отбора». Например, для поля «Нас\_пункт» напишем "Самара" (в кавычках, это текстовая константа).

Для построения сложных условий удобно пользоваться мастером **** во всплывающем меню. В нем можно найти все доступные в запросах данные (из таблиц, других запросов, форм), функции и операторы:



Диапазон дат или значений можно задать с помощью оператора Between. Даты нужно указывать между символами ##.

Between #01.02.2014# And #28.02.2014#

А, например, даты за последние 30 дней можно получить с помощью функции Date() – текущая дата. Напомним, что дата представляется в памяти как число ней, прошедших с 01.01.1990 (=0).

>=Date()-30

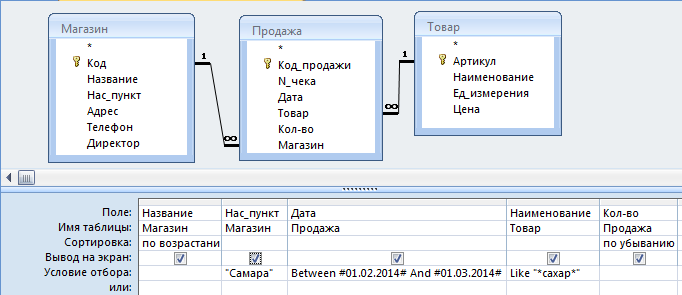
Для выбора текстовых значений используется оператор Like. В образце отбора можно использовать маски: \* обозначает любую последовательность символов, ? один любой символ. Регистр символов не учитывается.

Like "\*сахар\*"

Если бы мы написали в условии отбора просто "Сахар", то ничего бы не нашлось, т.к. у нас нет товара в точности с таким именем.

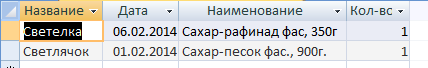
Примеры других запросов вы можете найти в справке  по Access (F1) в разделе «Примеры условий запроса».

Сортировка столбцов указывается в соответствующей строке. Порядок сортировки определяется порядком столбцов в запросе.



Но нужно ли нам отображать город? Мы и так знаем, что в запросе отображаются только магазины из Самары. Поэтому столбец «Нас\_пункт» можно скрыть, сняв флажок «Вывод на экран».

В нашей маленькой БД нашлось только 2 таких записи:



Если условие слишком длинное и не умещается на скриншоте, выпишите его в отчет в виде текста.

## Запрос с вычисляемыми полями

Вычисления – одна из наиболее распространенных функций запросов. Благодаря им нам не нужно хранить в базе, например, сумму продажи, мы всегда можем вычислить ее как произведение количества и цены.

Построим запрос «Стоимость продаж» из таблиц «Продажа» и «Товар». Отобразим дату продажи, номер чека, название товара, количество, единицы измерения, цену и сумму.

Первые столбцы добавляются в запрос как обычно.

Формула для вычисления суммы (цена\*кол-во) записывается прямо в строке «Поле». Основные правила:

* сначала указывается название будущего столбца, ставится двоеточие, потом пишется формула:

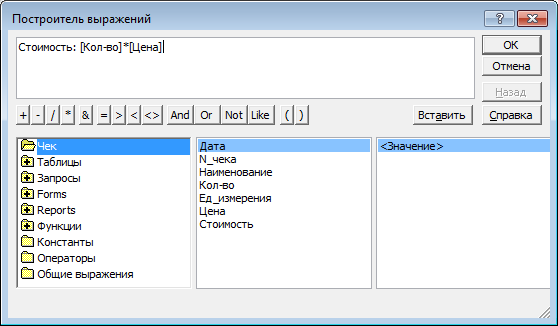
Название: формула

* названия полей таблиц, с которыми выполняются действия, и самих таблиц, пишутся в [], между таблицей и полем ставится восклицательный знак. Название таблицы можно не писать, если поле с таким именем встречается только в одной таблице

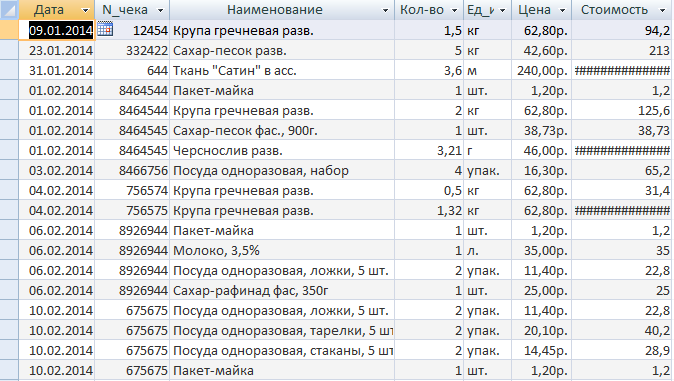
[Таблица]![Поле]

[Поле]

Как и для условий отбора, можно воспользоваться мастером «Построить...».

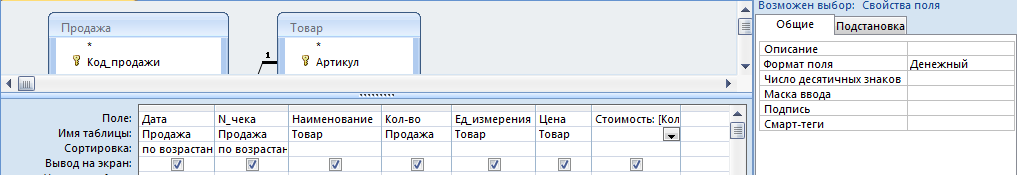


В итоге получим:

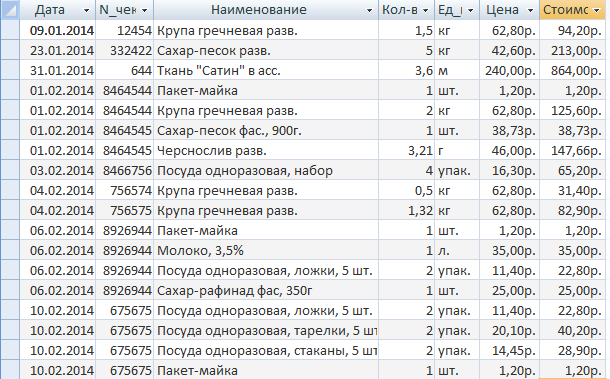


Но в поле сумма должны отображаться денежные единицы! Решетки ##### означают, что число слишком длинное и не умещается в столбец.

Чтобы это исправить, вернемся в Конструктор. Правый клик по полю «Стоимость», пункт меню «Свойства». Справа откроется окно свойств. Зададим формат поля – денежный. Кроме того, не помешает отсортировать запрос по дате продажи и номеру чека.



Теперь наш запрос немного похож на настоящие чеки, идущие без перерыва на чековой ленте:

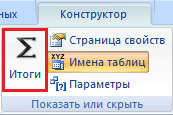


## Запрос с итогами (запрос с группировкой)

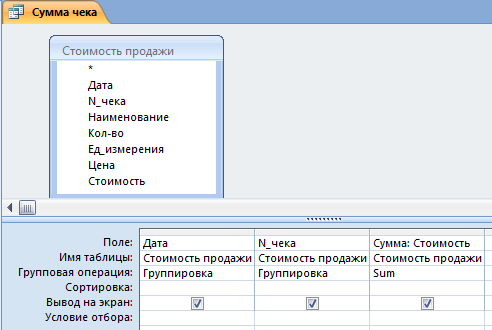
В предыдущем примере явно не хватает общей суммы по чеку. К сожалению, вывести ее в том же запросе не получится, нужно создать отдельный. Позже эти запросы можно будет вывести на экран или на печать вместе.

Добавим запрос «Сумма по чеку». Чтобы не повторять предыдущие действия, его мы будем строить не на основе таблиц, а на основе уже созданого запроса «Стоимость продажи».

Добавим в новый запрос поля «Дата», «№ чека» и «Стоимость». Далее необходимо включить подведение итогов в запросе.



В Конструкторе появится строчка «Группировка». Группировка объединяет совпадающие значения в одну запись. Нам нужно сгруппировать чеки по номеру и дате, а стоимость – просуммировать (**Sum**). Будет лучше переименовать поле «Стоимость» в поле «Сумма», это делается также как и для формул – через двоеточие. Сумме чека также нужно присвоить тип «Денежная».



Доступны и другие действия группировки:

**Avg** – среднее значение;

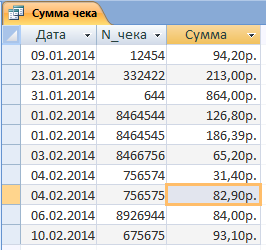
**Min, Max** – минимальное и максимальное значения;

**Count** – подсчет количества сгруппированных записей;

**StDev, Var** – стандартное отклонение и дисперсия;

**First, Last** – первое и последнее по порядку значение.

Результат выполнения запроса:



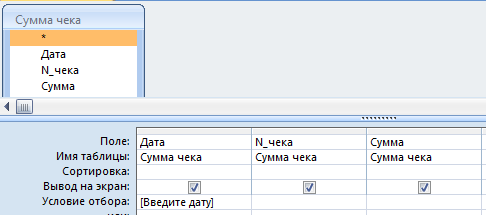
## Запрос с параметром

Не во всех случаях мы заранее знаем параметр отбора строк в запросе. Например, мы хотим выводить чеки за разные даты. Конечно, можно каждый раз открывать Конструктор и исправлять запрос. Но обычные пользователи БД этого не умеют. Они должны просто ввести дату и получить чеки за это число. А составить запрос – задача администратора БД или программиста.

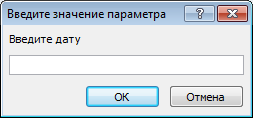
**Параметр запроса** – это как раз то условие, которое мы будем запрашивать у пользователя.

Параметры указываются в условии запроса в квадратных скобках [] (не путать с именами полей в формулах!). В скобках следует написать пояснение для пользователя – что ему нужно ввести.

[Введите дату]



В результате при выполнении запроса выведется окно:



Введем 01.02.14 и получим:



Параметр может быть частью условия отбора или формулы. Например, найдем все чеки на сумму более или равную указанной пользователем:

>=[Введите минимальную сумму чека:]

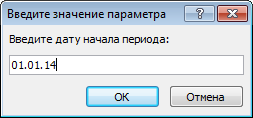
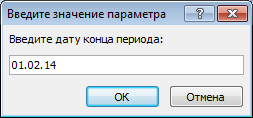
Отбор через оператор Like может выглядеть так:

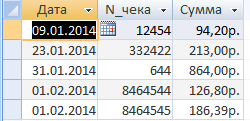
Like "\*"+[Введите подстроку для поиска товара:]+"\*"

Отбор по диапазону дат:

Between [Введите дату начала периода:] And [Введите дату конца периода:]

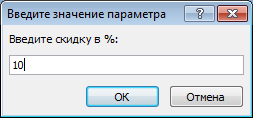
Здесь будет выдано два запроса подряд.

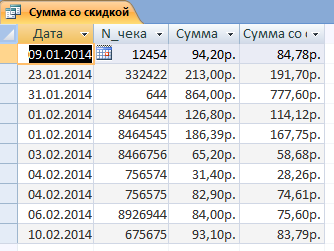
 



Пример использования параметра в формуле – вычисление суммы с произвольной скидкой:

Сумма со скидкой: [Сумма]\*(100-[Введите скидку в %:])/100





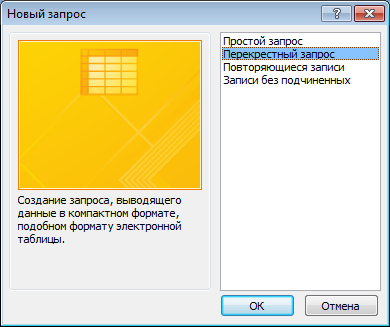
Замечание: вы можете создать запрос с параметром как в условии отбора, так и в формуле.

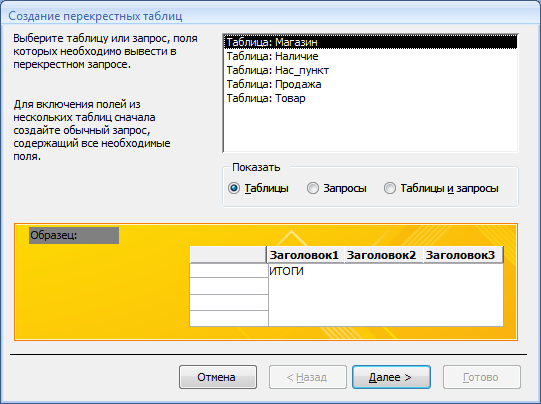
## Перекрестный запрос

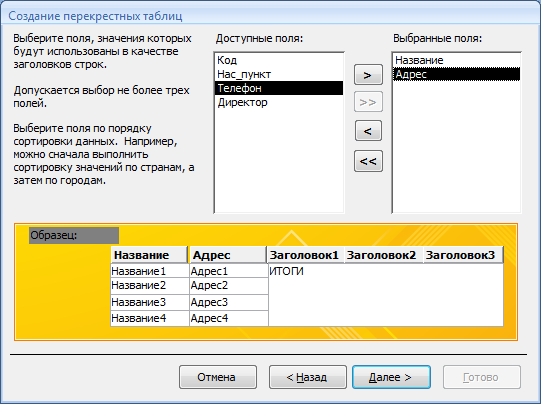
Подобен сводной таблице в Excel. Он позволяет превратить значения из записей в заголовки столбцов.

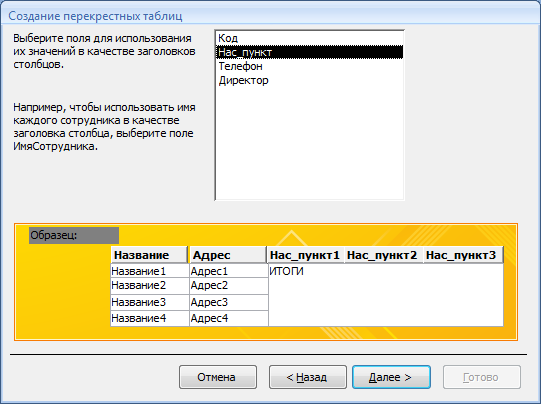
Простые перекрестные запросы – из одной таблицы или запроса – создаются через Мастер.

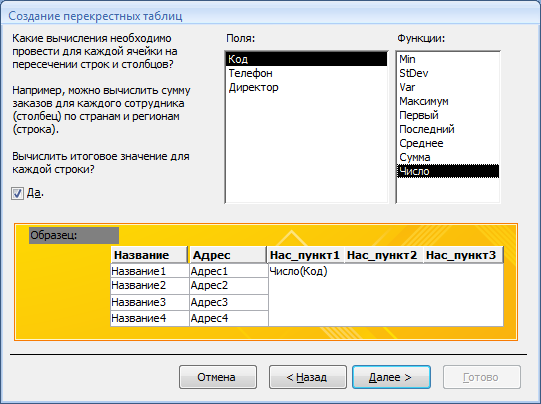
Например, покажем наличие магазинов по адресам в различных населенных пунктах:



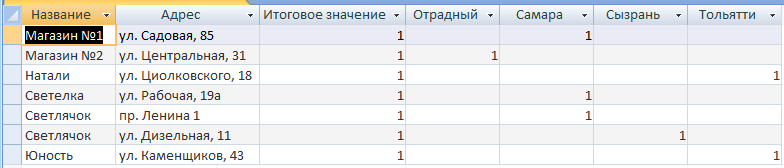








Результат:

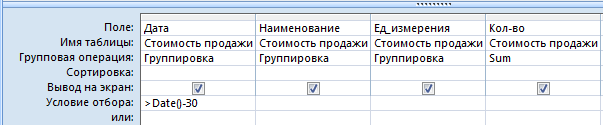


Если бы у нас было несколько магазинов по одному адресу, то в соответствующих ячейках отобразились бы 2, 3 и т.д. Кроме того, Access добавил столбец «Итоговое значение» с указанием общего числа таких магазинов.

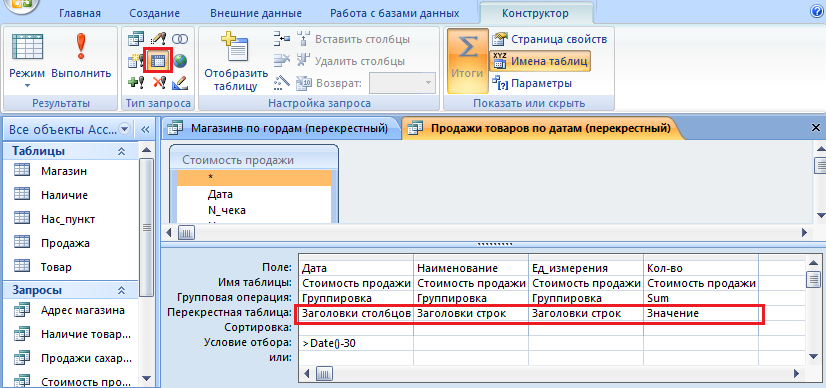
Более сложные запросы (из нескольких таблиц, с вычислениями, условиями отбора) создаются в конструкторе.

Создадим перекрестный запрос, в котором покажем продажи товаров за указанный диапазон дат.

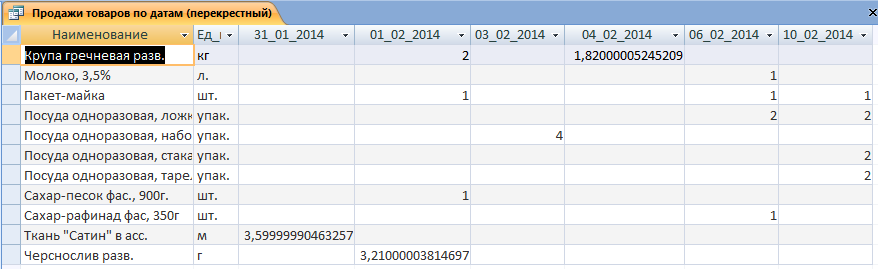
Сначала создадим обычный запрос с группировкой:



Теперь изменим его тип на «перекрестный». Вместо строки «Вывод на экран» появится «Перекрестная таблица». Там нужно указать, какие поля станут заголовками строк, какие – заголовками столбцов, какие – значениями:



Результат:



К сожалению, в перекрестных запросах имеются серьезные проблемы с округлением. Формат поля также не влияет на вывод значений.

В отчет необходимо вставить представление каждого запроса в Конструкторе и результат его выполнения (скриншотом или через экспорт, как для таблиц). Снабдите каждый запрос кратким пояснением – какие именно данные выбраны и из каких таблиц. Если условия запроса или формулы не умещаются на скриншоте, выпишите их отдельно.

# Создание форм

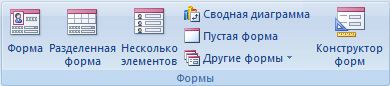
Формы необходимы для более удобного редактирования таблиц и просмотра запросов на экране. Соответственно, различают формы, ориентированные на ввод и редактирование данных, и формы, ориентированные на просмотр.

В данном практикуме мы создадим только формы для просмотра:

* автоформы в один столбец, табличную и ленточную;
* составные формы (подчиненную и связанную).

Формы можно создавать на основе таблиц или запросов. Автоформы в один столбец, табличная и ленточная создаются на основе одной таблицы, а составные формы – на основе нескольких связанных таблиц.

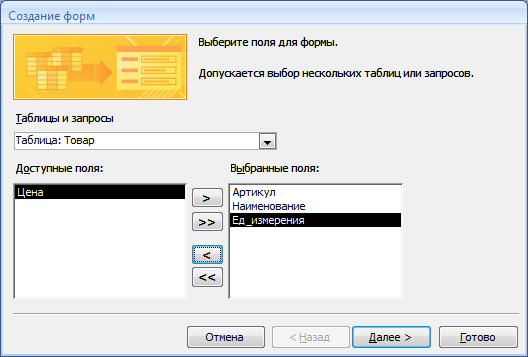
Как и запросы, формы можно создавать через Мастер (автоформы) или Конструктор. Некоторые формы вынесены непосредственно на Ленту.



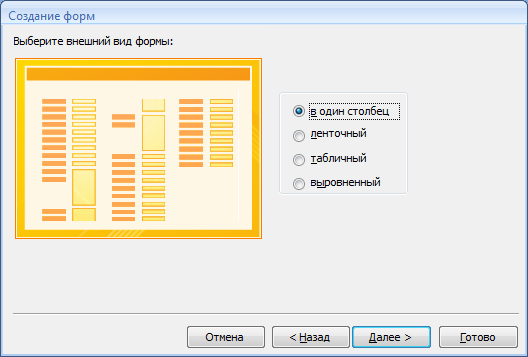
## Автоформа в один столбец

Это самый простой вид форм, где все поля располагаются одно под другим. Отобразим на ней содержимое таблицы «Товар», за исключением цены.

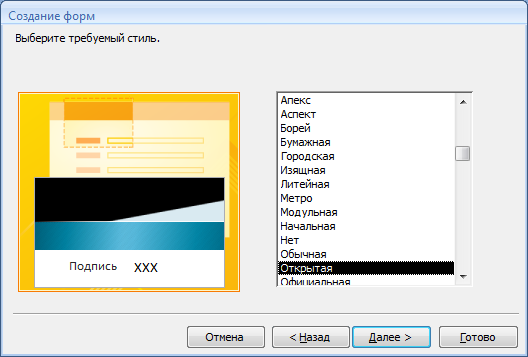
В раскрывающемся списке «Другие формы» выберите «Мастер форм». Выберите нужную таблицу или однотабличный запрос. С помощью кнопок со стрелками переместите выбранные поля в правый столбец.



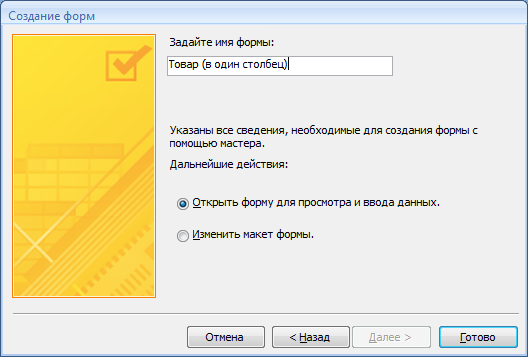
Вид формы – «в один столбец».



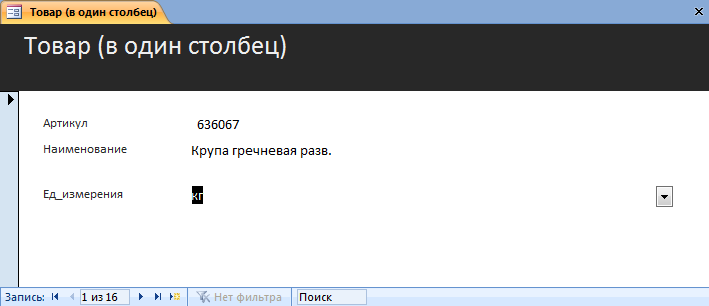
Выберите любой стиль, который Вам нравится:



Задайте имя формы:



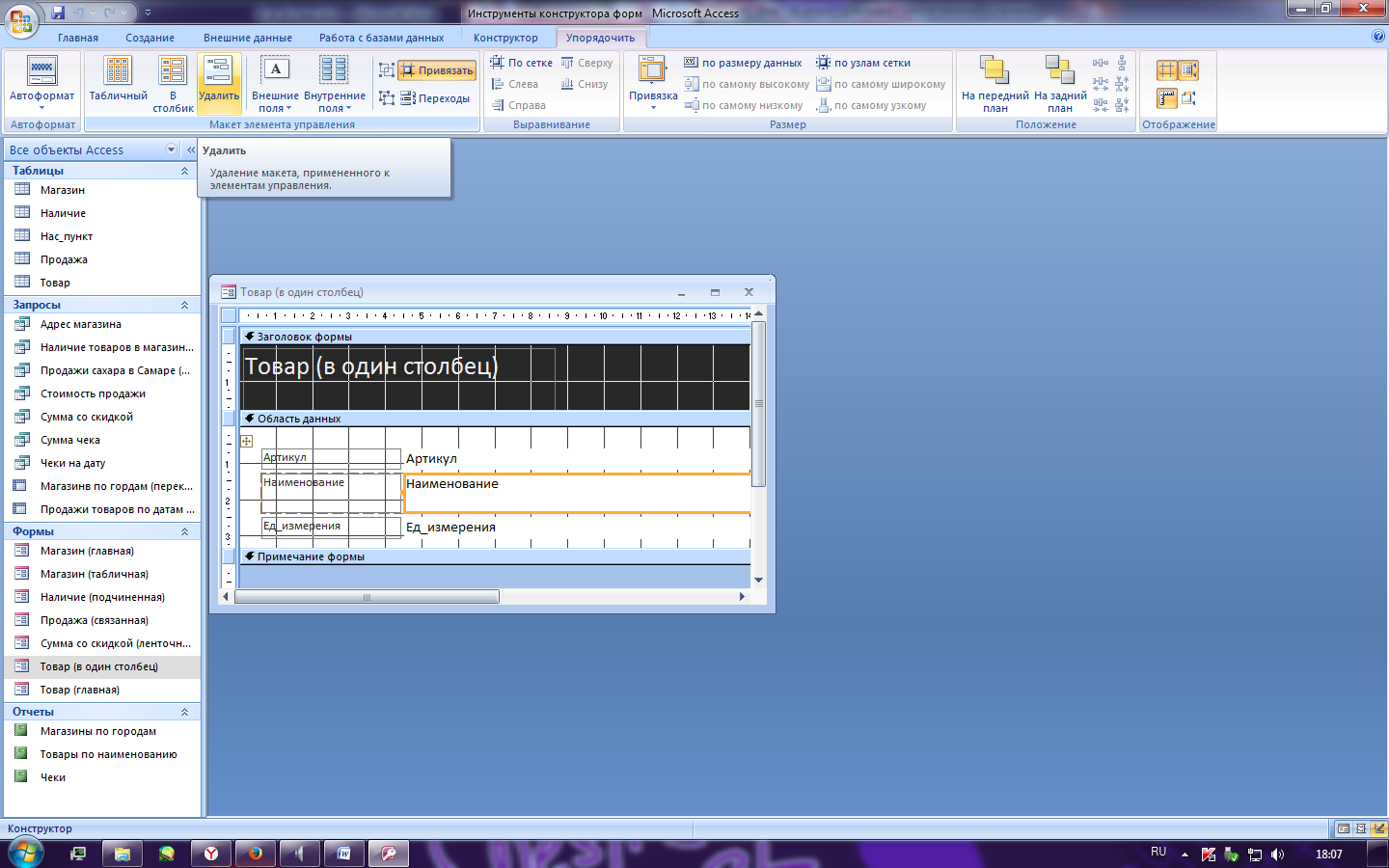
Результат:



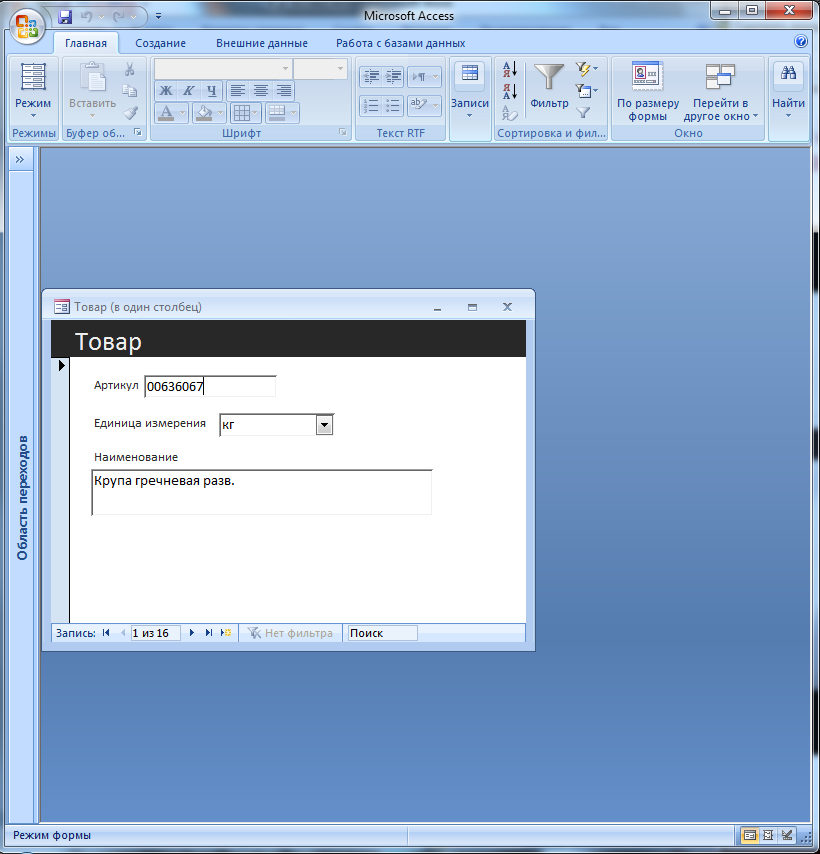
На такой форме отображается одна запись «за раз». Переход между записями осуществляется с помощью кнопок внизу. Если форма создана на основе таблицы, ее можно использовать как для просмотра, так и для ввода и редактирования данных.

Внешний вид формы можно настроить в режиме Конструктора. Переключение между режимами выполняется аналогично запросам.

Чтобы отключить автоматическое выравнивание у выделенного поля, на ленте на вкладке «Упорядочить» нажмите кнопку «Удалить» в группе «Макет элемента управления».



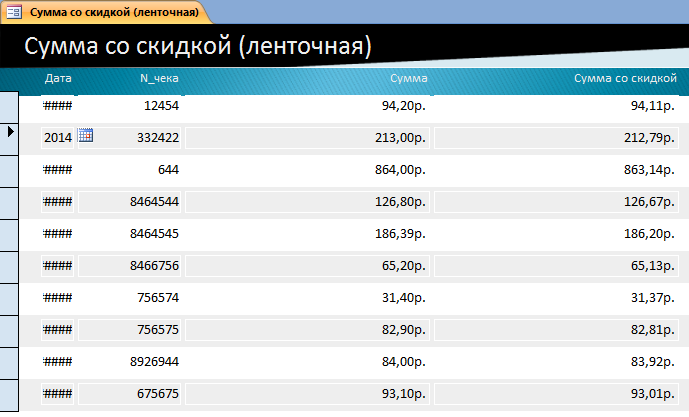
После этого данное поле можно будет редактировать отдельно, настраивать внешний вид, размер и положение. Также можно настроить размер формы, чтобы она умещалась даже на небольшом экране.



Подробные настройки каждого объекта на форме можно увидеть через правый клик – «Свойства».

## Ленточная автоформа

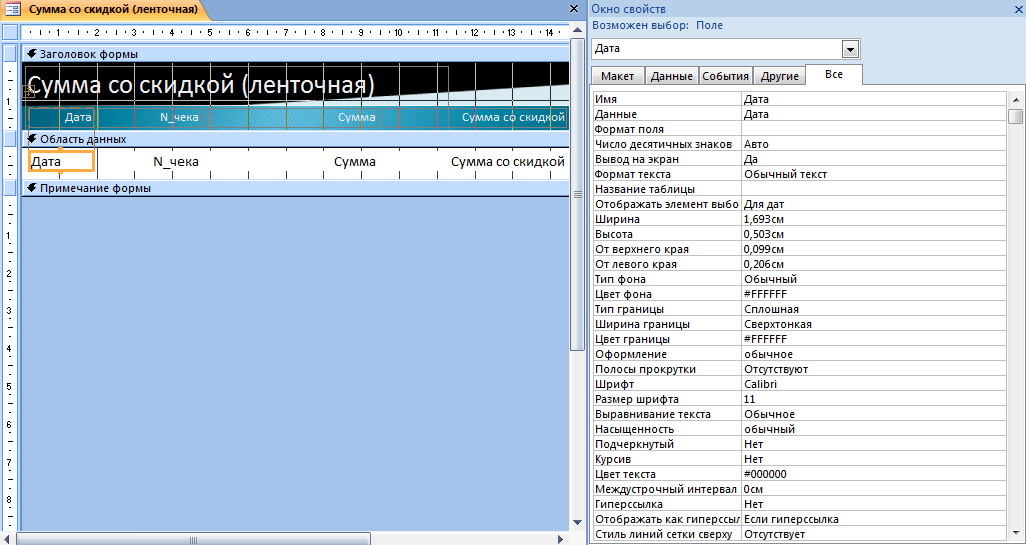
Создается полностью аналогично. Отличается лишь выбор вида формы на втором шаге и результат. На ленточной форме все записи отображаются вместе, как в таблице:



Данная форма создана на основе запроса «Сумма со скидкой».

Решетки в столбце «Дата» отображаются из-за того, что автоматической ширины столбца недостаточно для отображения даты. В то же время, столбцы «Сумма» и «Сумма со скидкой» слишком широкие.

Исправим это в Конструкторе:

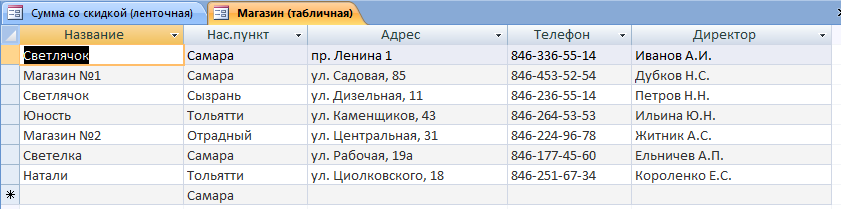


*Замечание*: используйте Конструктор по мере необходимости, если автоматически созданная форма неудовлетворительна.



## Табличная автоформа

Создается также и выглядит полностью как обычная таблица или запрос:



Такие формы используются редко, обычно, как вспомогательные или часть составных форм (см. ниже).

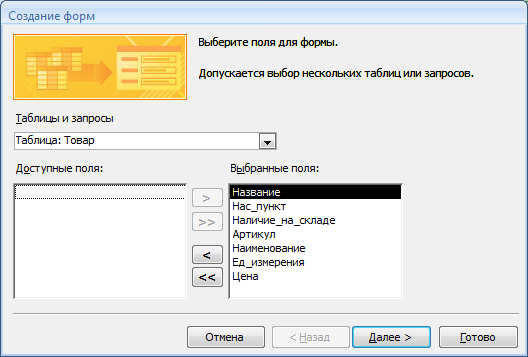
В Конструкторе табличная форма отображается как форма в один столбец.

## Подчиненные формы

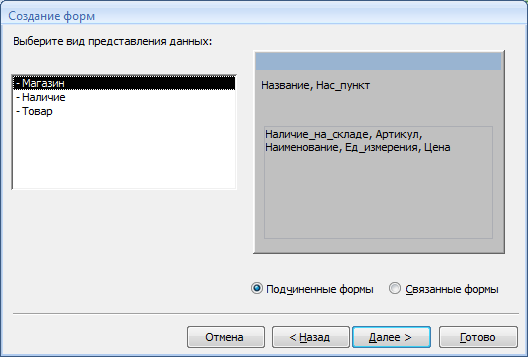
Отображают одновременно несколько связанных таблиц. Их также можно создать с помощью Мастера.

В примере покажем на одной форме товары, имеющиеся в каждом магазине.

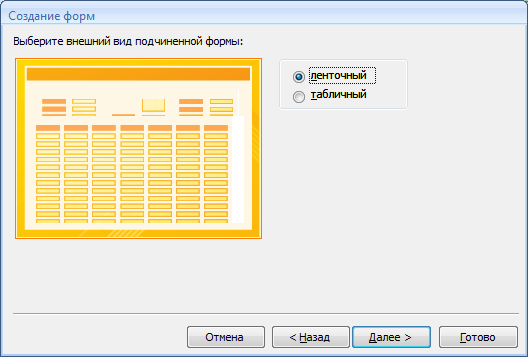
На первом шаге необходимо выбрать поля из нескольких связанных таблиц:



На втором шаге выбираем подчиненную форму и вид ее представления (какая таблица будет основной, какая – подчиненной). Основная форма отображается по одной записи, и одновременно можно увидеть все связанные с данной записи в подчиненной форме.

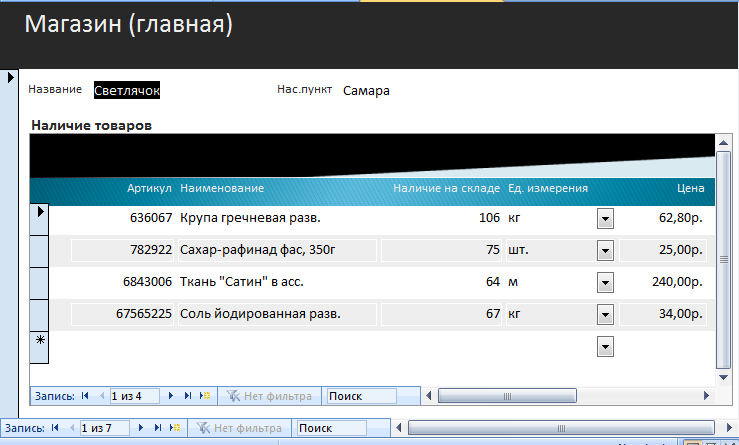


Внешний вид подчиненной формы можно выбрать любой:



Дальнейшие шаги пояснений не требуют.

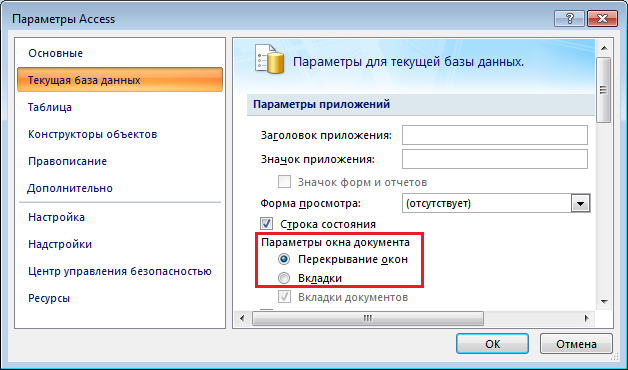
В результате будет создано две формы. Но подчиненная отображается как еще одно поле на главной форме. Размещение элементов на форме потребовалось отредактировать в Конструкторе.



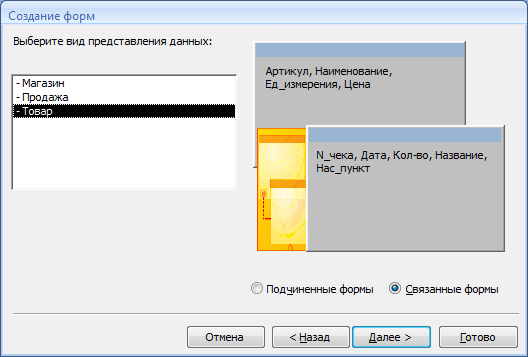
## Связанные формы

Также создаются из нескольких таблиц или подчиненных запросов. При этом на главной форме размещается кнопка, нажатие на которую вызывает подчиненную форму.

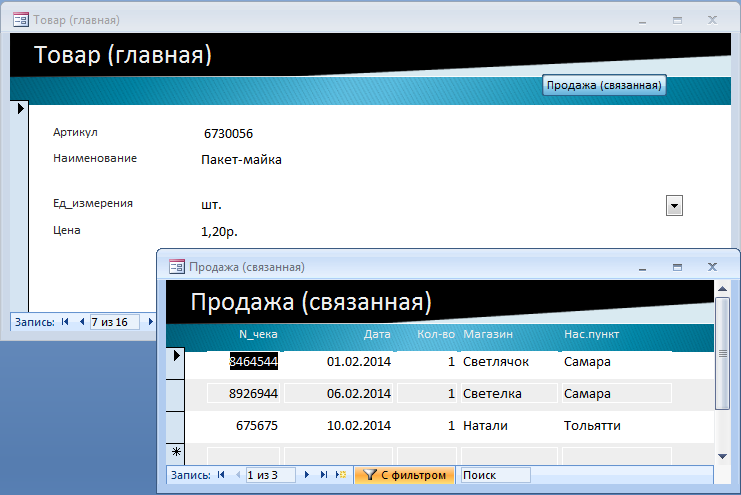
Связанные формы удобнее использовать, изменив настройки Access по умолчанию: формы лучше отображать не на вкладках, а как обычно. Чтобы настройки вступили в силу, нужно перезапустить БД.



Создаются связанные формы аналогично подчиненным, только на втором шаге нужно указать тип формы – связанная.



Связанные формы в примере созданы на основе таблиц «Продажа», «Товар», «Магазин». Вид форм отредактирован в Конструкторе.



Таком образом, мы можем увидеть, где и когда был продан выбранный товар. При перелистывании записей на главной форме, содержимое подчиненной или связанной формы сразу меняется.

В отчет по работе необходимо вставить скриншоты готовых форм. Связанные формы можно отдельными скриншотами.

# Создание отчетов

Отчеты необходимы для вывода сведений из БД на печать. Обычно отчеты содержат итоговые сведения из запросов.

Отчеты, как и формы, и запросы, можно создавать с помощью Мастера отчетов (автоотчеты) или через Конструктор. Любые отчеты можно отредактировать в Конструкторе.

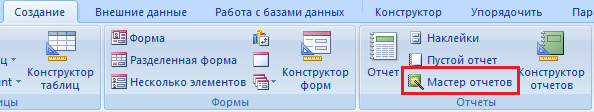
Создадим следующие типы отчетов:

* автоотчет в один столбец;
* отчет по нескольким полям одной таблицы;
* отчет с группировкой и итогами.

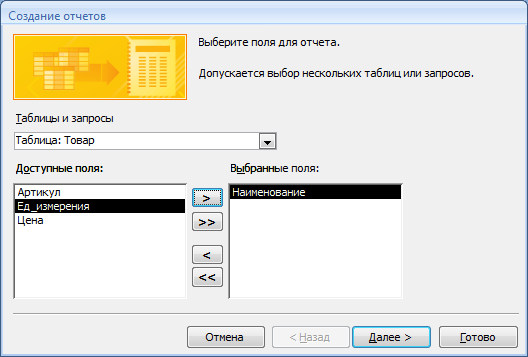
## Автоотчет в один столбец

Создается с помощью Мастера и позволяет вывести список всех значений какого-либо поля. Например, отобразим список всех указанных в БД товаров (наименований) в алфавитном порядке.

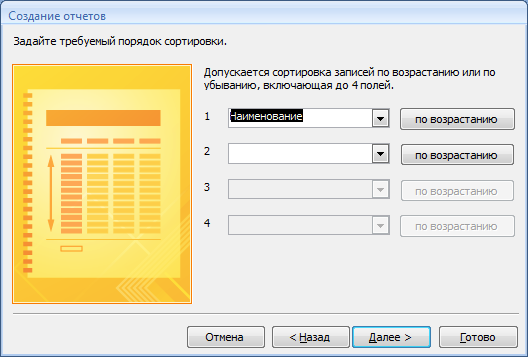
Создадим отчет с помощью Мастера.



В первую очередь, необходимо выбрать поля из таблиц и/или запросов, которые будут отображаться в БД.



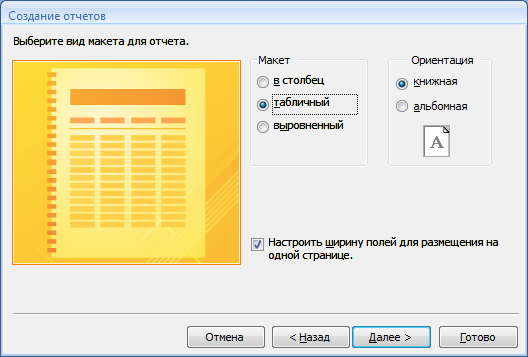
Затем указываем порядок и направление сортировки (если требуется).



Макет отчета можно выбрать любой:

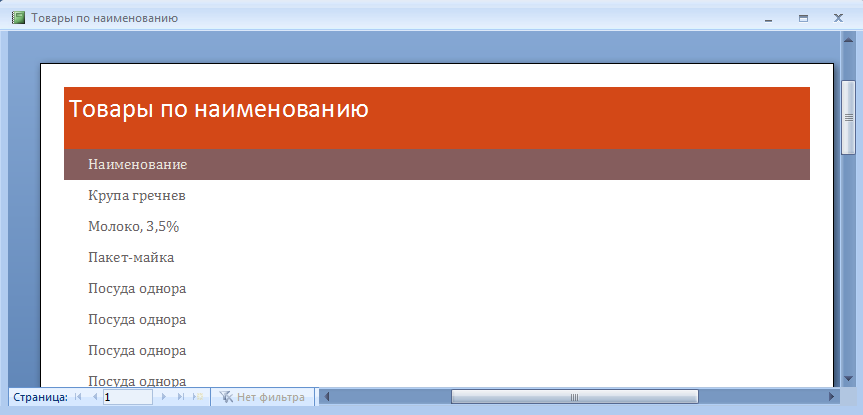
* *табличный* подходит для большинства отчетов;
* *в столбец* выводит данные наподобие формы, подходит для отображения полей с большим содержимым (длинный текст, рисунки);
* *выровненный* стоит использовать в отчетах с большим количеством столбцов, где они имеют разный размер или просто не умещаются в одну строчку таблицы.

Желательно использовать флажок «Настроить ширину полей для размещения на одной странице», иначе часть столбцов может «уехать» на другие страницы.



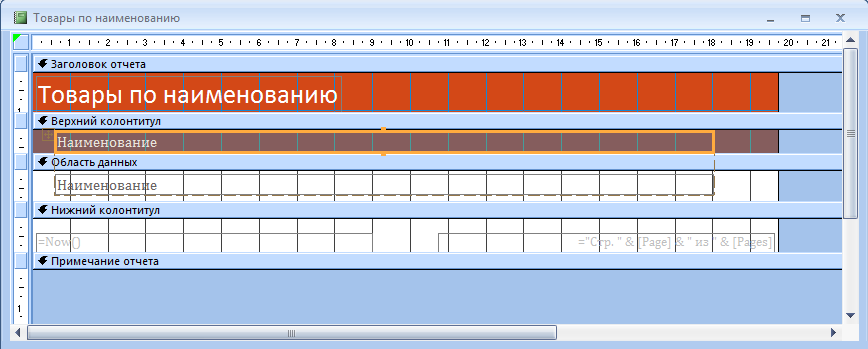
Далее выбирается стиль оформления (в примере выбран стиль «Справедливость») и задается имя отчета (Товары по наименованию). Имя лучше изменить – таблицы БД называются в единственном числе, а отчеты не обязательно.

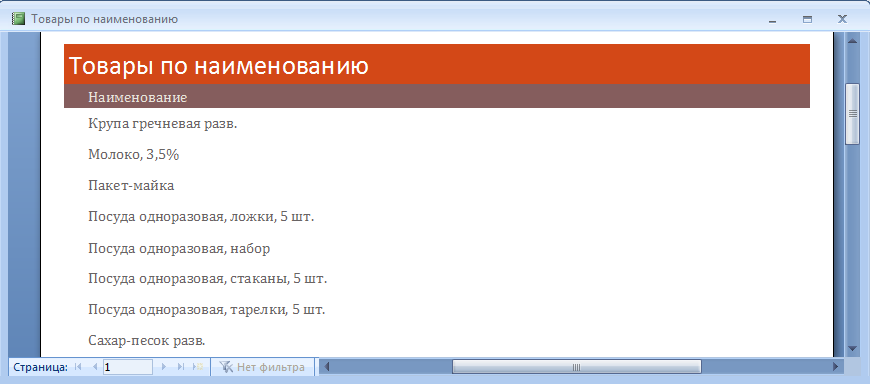
Результат:



Ширина столбца, очевидно, недостаточна – названия товаров обрезаны. Исправим структуру отчета через Конструктор (кнопка «Закрыть окно предварительного просмотра»).

Поскольку столбец всего один, его можно растянуть на всю ширину страницы. Обратите внимание на ширину самого отчета (сверху, на линейке) – он не должен превышать ширину листа A4 (21см минус поля).





Полученный отчет можно распечатать из Access или экспортировать в Word или другую программу.

Для добавления в отчет по лабораторной работе, выполните экспорт в Word. В конечном документе исправим шрифт (заголовок напечатан белым цветом), уберем колонтитулы. Чтобы отделить отчеты друг от друга добавьте вокруг них границы (как для таблиц: выделить отчет, кнопка  на вкладке «Главная»).

Результат:

Товары по наименованию

**Наименование**

Крупа гречневая разв.

Молоко, 3,5%

Пакет-майка

Посуда одноразовая, ложки, 5 шт.

Посуда одноразовая, набор

Посуда одноразовая, стаканы, 5 шт.

Посуда одноразовая, тарелки, 5 шт.

Сахар-песок разв.

Сахар-песок фас., 900г.

Сахар-рафинад фас, 350г

Соль йодированная разв.

Соль мелкая, 500г

Соль морская, 450г

Ткань "Сатин" в асс.

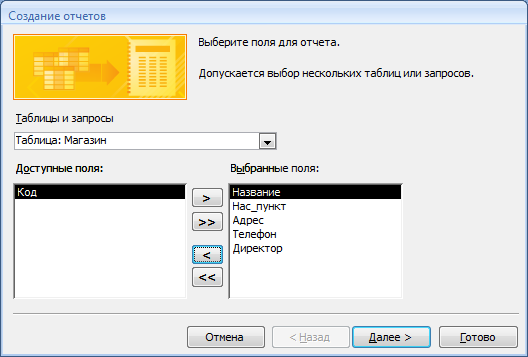
Ткань подкладочная

Черснослив разв.

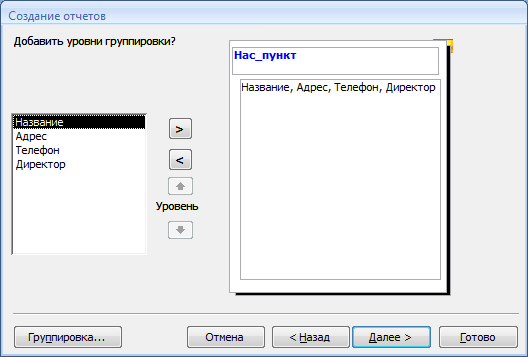
## Отчет из нескольких полей

В отчете из нескольких полей одной таблицы покажем сведения о магазинах.

В целом, процесс создания отчета аналогичен.



Однако на втором шаге предлагается выбрать группировку записей. Название группы будет выводиться в виде подзаголовка.



В данном случае достаточно логично объединить магазины по населенному пункту, в котором они находятся. В других вариантах можно обойтись без группировки.

Сортировку выполним по адресу.

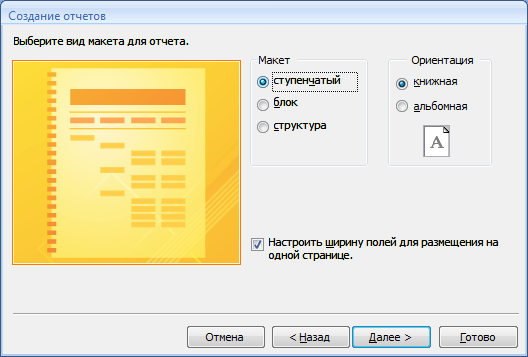
При наличии группировки типы макетов будут другими:

* ступенчатый;
* блок;
* структура.

«Ступенчатый» и «блок» похожи, они представляют собой таблицу, в которой не будут повторяться одинаковые значения записей.

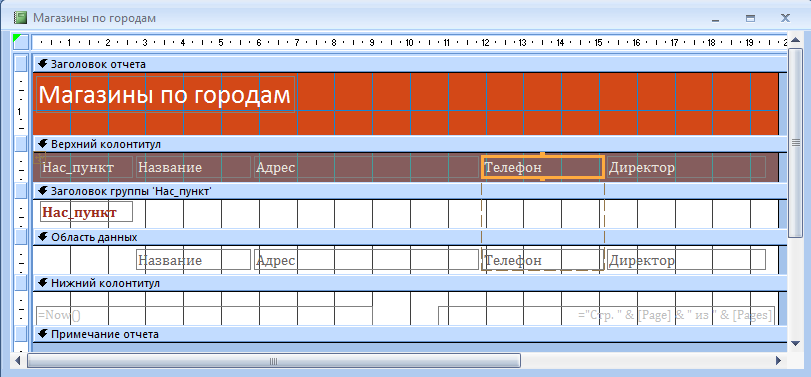
В «структуре» сгруппированные поля отображаются как подзаголовки.

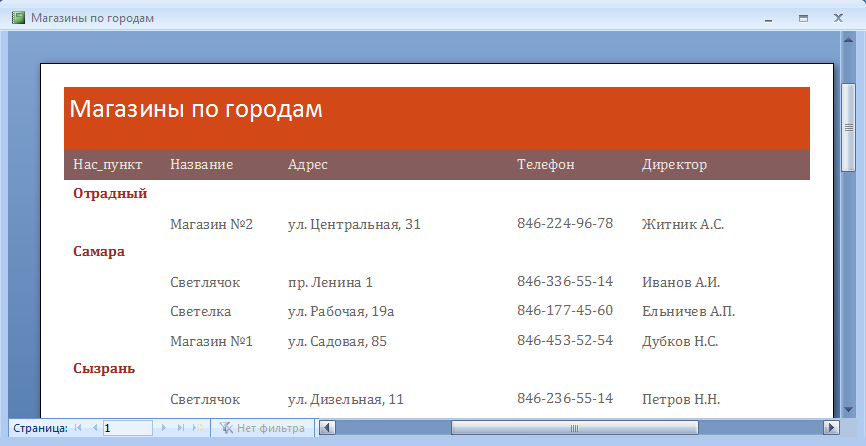
В примере выберем ступенчатый макет.



Стиль – «Справедливость», название отчета «Магазины по городам».

Ширину столбцов также потребовалось отредактировать в Конструкторе, а также поменять порядок столбцов: хоть мы и сортируем магазины по адресу, логично первым столбцом показать название.





Результат экспорта в Word:

Магазины по городам

**Нас\_пункт Название Адрес Телефон Директор**

**Отрадный**

Магазин №2 ул. Центральная, 31 846-224-96-78 Житник А.С.

**Самара**

Светлячок пр. Ленина 1 846-336-55-14 Иванов А.И.

Светелка ул. Рабочая, 19а 846-177-45-60 Ельничев А.П.

Магазин №1 ул. Садовая, 85 846-453-52-54 Дубков Н.С.

**Сызрань**

Светлячок ул. Дизельная, 11 846-236-55-14 Петров Н.Н.

**Тольятти**

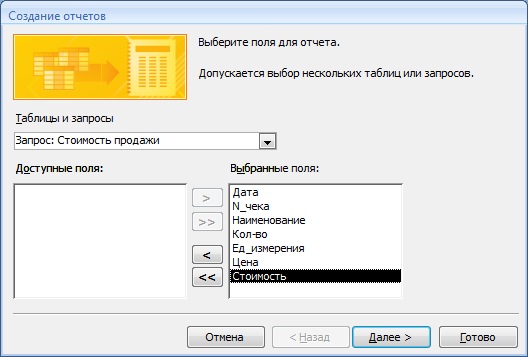
Юность ул. Каменщиков, 43 846-264-53-53 Ильина Ю.Н.

Натали ул. Циолковского, 18 846-251-67-34 Короленко Е.С.

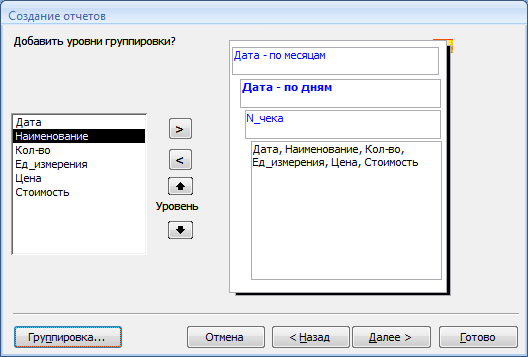
## Отчет с группировкой и подведением итогов

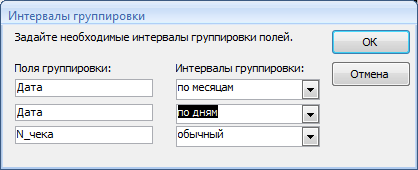
Можно сделать на основе нескольких таблиц или запроса. В пример воспользуемся ранее созданным запросом «Стоимость продаж».

В отчет добавим все столбцы запроса.



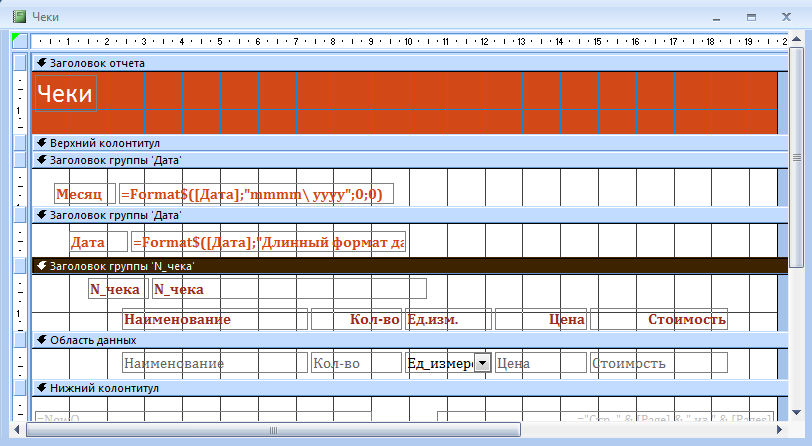
Группировку желательно делать многоуровневой. Для дат может потребоваться дополнительная настройка через кнопку «Груупирвока...». В данном случае по датам сгруппируем дважды: по месяцам и по дням.



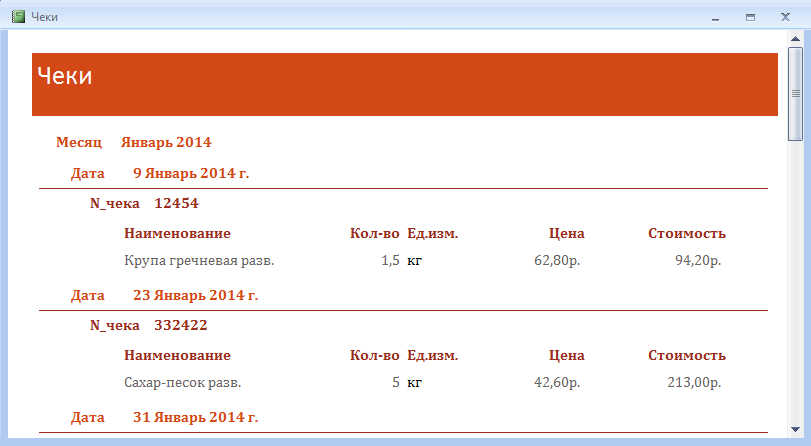


Сортировку в данном случае выполнять не будем. Макет выберем «Структура», стиль также «Справедливость». Название отчета – «Чеки».

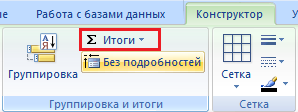
Отображение отчета также потребовалось исправить через Конструктор: убрать лишнее поле «Дата», настроить ширину и расположение столбцов, горизонтальные разделительные линии и т.п.



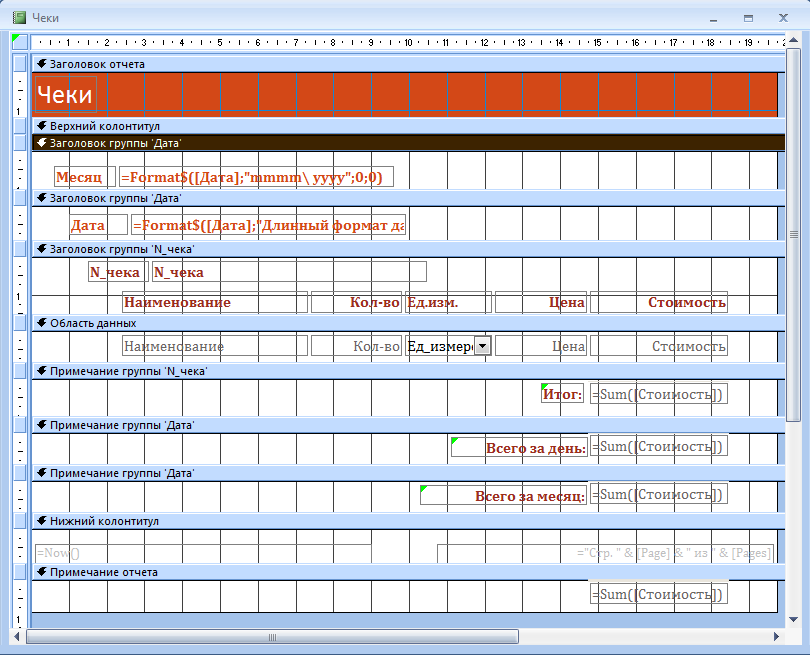
В результате отчет выглядит так:



Теперь необходимо добавить итоги. Это можно выполнить через Конструктор. Итоги будет подводить по сумме чека. Сначала необходимо выделить столбец «Стоимость», затем нажать на кнопку «Итоги».



В результате в макет отчета были добавлены разделы «Примечание группы...», а также «Примечание отчета». Вручную добавим текстовые подписи к каждому итогу: сумма по чеку, за день, за месяц.



Результат имеет вид:



Чеки

**Месяц** **Январь 2014**

**Дата** **9 Январь 2014 г.**

**N\_чека** **12454**

**Наименование** **Кол-во** **Ед.изм.** **Цена** **Стоимость**

Крупа гречневая разв. 1,5 кг 62,80р. 94,20р.

**Итог:** 94,20р.

**Всего за день:** 94,20р.

**Дата** **23 Январь 2014 г.**

**N\_чека** **332422**

**Наименование** **Кол-во** **Ед.изм.** **Цена** **Стоимость**

Сахар-песок разв. 5 кг 42,60р. 213,00р.

**Итог:** 213,00р.

**Всего за день:** 213,00р.

**Дата** **31 Январь 2014 г.**

**N\_чека** **644**

**Наименование** **Кол-во** **Ед.изм.** **Цена** **Стоимость**

Ткань "Сатин" в асс. 3,6 м 240,00р. 864,00р.

**Итог:** 864,00р.

**Всего за день:** 864,00р.

**Всего за месяц:** 1 171,20р.

**Месяц** **Февраль 2014**

**Дата** **1 Февраль 2014 г.**

**N\_чека** **8464544**

**Наименование** **Кол-во** **Ед.изм.** **Цена** **Стоимость**

Пакет-майка 1 шт. 1,20р. 1,20р.

Крупа гречневая разв. 2 кг 62,80р. 125,60р.

**Итог:** 126,80р.

**N\_чека** **8464545**

**Наименование** **Кол-во** **Ед.изм.** **Цена** **Стоимость**

Сахар-песок фас., 900г. 1 шт. 38,73р. 38,73р.

Черснослив разв. 3,21 г 46,00р. 147,66р.

**Итог:** 186,39р.

**Всего за день:** 313,19р.

**Дата** **3 Февраль 2014 г.**

**N\_чека** **8466756**

**Наименование** **Кол-во** **Ед.изм.** **Цена** **Стоимость**

Посуда одноразовая, набор 4 упак. 16,30р. 65,20р.

**Итог:** 65,20р.

**Всего за день:** 65,20р.

**Дата** **4 Февраль 2014 г.**

**N\_чека** **756574**

**Наименование** **Кол-во** **Ед.изм.** **Цена** **Стоимость**

Крупа гречневая разв. 0,5 кг 62,80р. 31,40р.

**Итог:** 31,40р.

**N\_чека** **756575**

**Наименование** **Кол-во** **Ед.изм.** **Цена** **Стоимость**

Крупа гречневая разв. 1,32 кг 62,80р. 82,90р.

**Итог:** 82,90р.

**Всего за день:** 114,30р.

**Дата** **6 Февраль 2014 г.**

**N\_чека** **8926944**

**Наименование** **Кол-во** **Ед.изм.** **Цена** **Стоимость**

Пакет-майка 1 шт. 1,20р. 1,20р.

Молоко, 3,5% 1 л. 35,00р. 35,00р.

Посуда одноразовая, 2 упак. 11,40р. 22,80р.

Сахар-рафинад фас, 350г 1 шт. 25,00р. 25,00р.

**Итог:** 84,00р.

**Всего за день:** 84,00р.

**Дата** **10 Февраль 2014 г.**

**N\_чека** **675675**

**Наименование** **Кол-во** **Ед.изм.** **Цена** **Стоимость**

Посуда одноразовая, 2 упак. 11,40р. 22,80р.

Посуда одноразовая, 2 упак. 20,10р. 40,20р.

Посуда одноразовая, 2 упак. 14,45р. 28,90р.

Пакет-майка 1 шт. 1,20р. 1,20р.

**Итог:** 93,10р.

**Всего за день:** 93,10р.

**Всего за месяц:** 669,79р.

1 840,99р.

В отчет по лабораторной работе необходимо добавить экспортированные отчеты из Access. Скриншоты необязательны.

Допускается отчеты выносить в отдельный раздел с более узкими полями, а также уменьшать межабзацные отступы для экономии места.