

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Куижева Саида Казбековна
Должность: Ректор
Дата подписания: 03.08.2023 23:17:07
Уникальный программный ключ:
71183e1134ef9c6028618671b310f54f

МИНОБРНАУКИ РОССИИ

**Политехнический колледж филиала федерального государственного бюджетного
образовательного учреждения высшего образования «Майкопский государственный
технологический университет» в поселке Яблоновском**

МАТЕРИАЛЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

по дисциплине

ОП.03 Информационные технологии

специальность

09.02.07 Информационные системы и программирование

по разделу «Базы данных»

форма обучения

очная

квалификация выпускника

программист

Яблоновский, 2020

УДК 004.65 (07)
ББК 32.81
М-54

Одобрено предметной (цикловой) комиссией
информационных и математических дисциплин

Разработчик: Хуаде Р.А. - преподаватель первой категории политехнического колледжа филиала федерального государственного бюджетного образовательного учреждения высшего образования «Майкопский государственный технологический университет» в поселке Яблоновском

Создание базы данных в MS Access

Цель

Научиться создавать новый файл базы данных Access, создавать таблицы в этом файле с помощью средств пользовательского интерфейса Access.

В качестве предметной области рассмотрим склад: учет, заказ и оптовую реализацию товаров. Приведем описание предметной области.

- Товары поступают на склад отдельными партиями от разных поставщиков.
- Заказ товаров поставщикам выполняется при достижении минимального запаса товара на складе.
- Для каждого типа товара на складе имеются полки. Партии товаров размещаются на полках склада в соответствии с типом товара.
- Партии товаров могут включать разное количество единиц товара, но имеют стандартные упаковки, под размеры которых спроектированы полки склада.
- Полки склада имеют разные размеры, соответственно возможно размещение различного количества партий определенного типа товара (характеристика – объем полки). Например, полка для размещения 5 упаковок по 1000 штук цветных карандашей.
- Товар, хранящийся на складе, поступает в оптовую продажу.
- Покупатели оформляют бланк заказа на поставку определенного количества партий различных товаров.
- Партии товара, имеющие одинаковую стоимость, могут быть вписаны в бланке заказа одной позицией с указанием количества партий. Если стоимость партий различна, они вписываются в различные позиции заказа.

На этапе анализа и моделирования предметной области была построена логическая модель данных «сущность-связь» (рис. 1). Построение логических моделей данных подробно рассмотрено в материалах лекции к теме «Базы данных».

Для модели на рис. 1 выполнены этапы нормализации до 3 нормальной формы включительно. Устранены связи типа «М:М».

Перейдем к этапу проектирования физической структуры базы данных.

Самостоятельно

Выполните преобразование логической модели (рис.1) в физическую модель в соответствии с правилами, рассмотренными в лекции по теме «Базы данных».

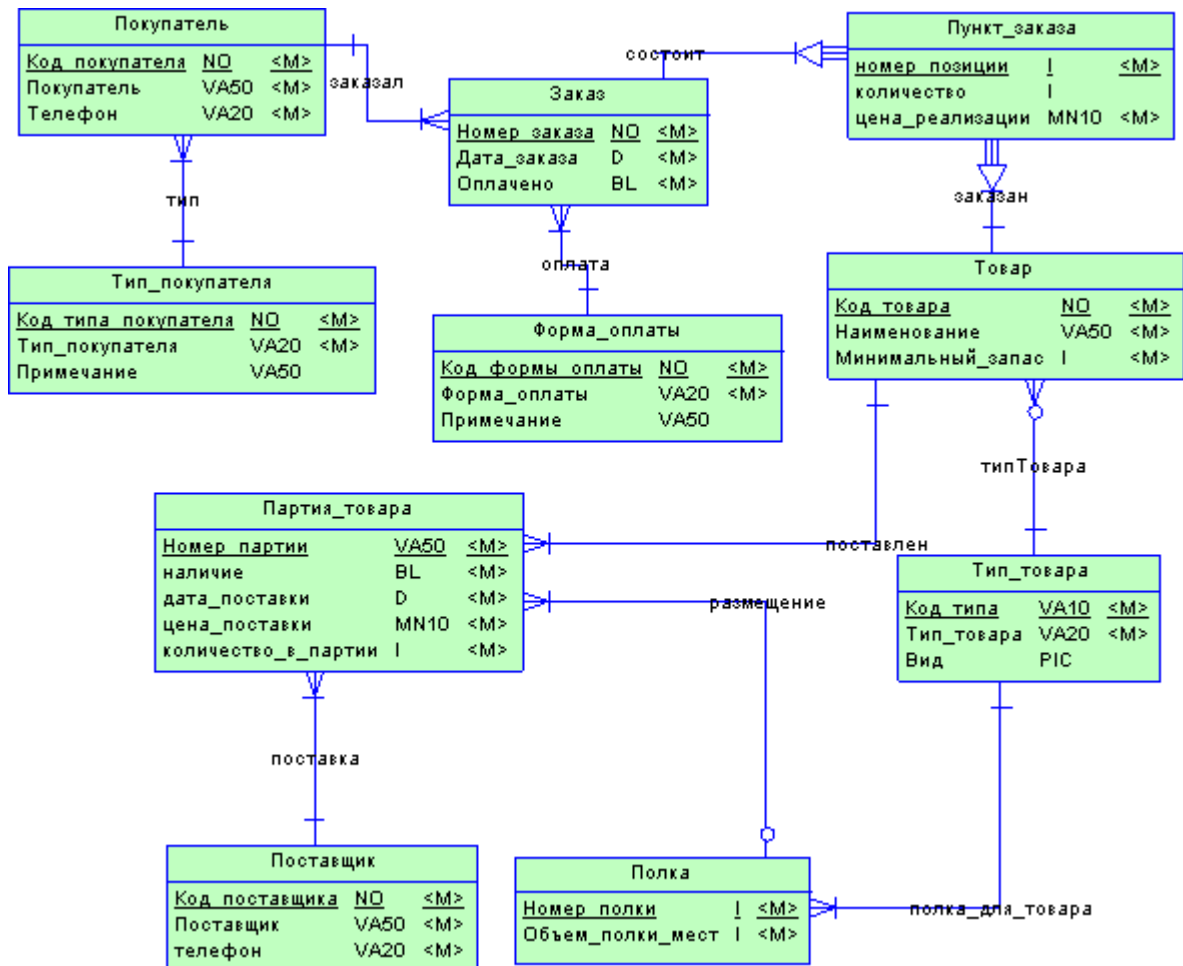


Рис.1. Логическая модель «Склад»

Создание нового файла базы данных Access

1. Запустите программу Microsoft Access.
2. В диалоге создания базы данных выберите шаблон **Новая база данных**, укажите папку, в которой будет сохранена ваша база данных, дайте имя файлу. Нажмите кнопку **Создать**.
3. Структуру таблицы следует разрабатывать в режиме **Конструктор** (рис.2). Для начала создадим простые таблицы, так называемые **справочники** (это таблицы, находящиеся в связи на стороне «1»).

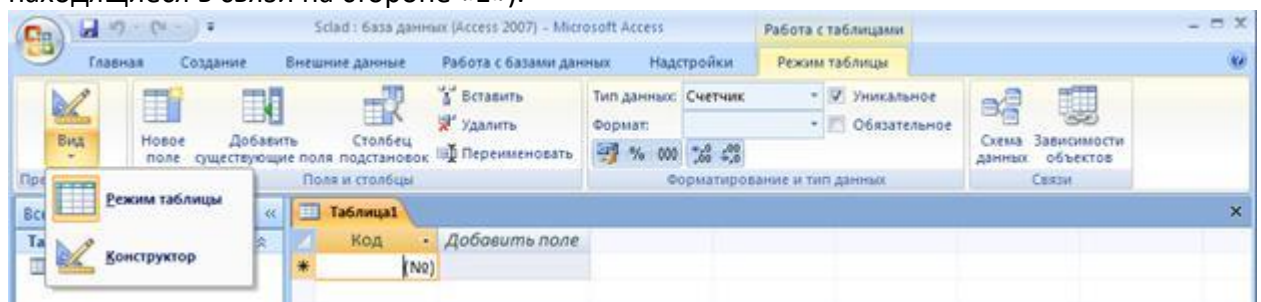


Рис.2. Изменение режима редактирования таблицы


В режиме Конструктор таблица создается путем задания имен полей, их типов и свойств. На рис. 3. показана структура таблицы ТИП_ПОКУПАТЕЛЯ.

4. В столбец **Имя поля** введите имя поля.
5. В столбце **Тип данных** выберите из раскрывающегося списка соответствующий тип поля.
6. В столбце **Описание** можно ввести описание данных этого поля (не обязательно).

7. В нижней части **Конструктора таблиц** на вкладках **Общие** можно установить свойства каждого из полей таблицы (рис.3).

Для всех полей создаваемых таблицы обязательно устанавливайте следующие свойства:

- **Размер поля,**
- **Обязательное поле,**
- **Пустые строки.**

8. Укажите ключевые поля. Для указания ключевого поля выделите соответствующую строку (установите в этой строке курсор) и нажмите на кнопку **Ключевое поле**  на панели инструментов (рис.3).

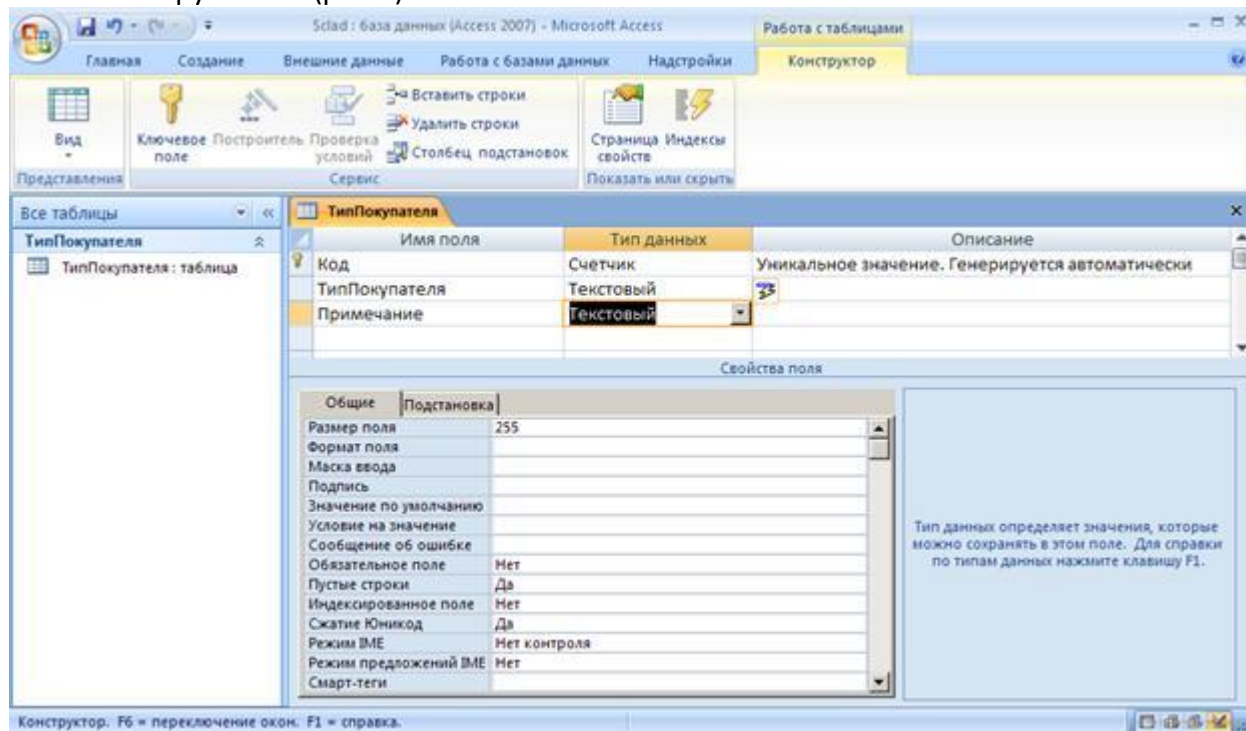



Рис.3. Создание таблицы в режиме КОНСТРУКТОРА

9. После описания всех полей нажмите кнопку  **Сохранить**. Таблицу можно закрыть. Правил именования объектов в MS Access:

- Имена полей должны быть уникальны в рамках таблицы.
- Имена таблиц должны быть уникальны в рамках базы данных.
- Имена могут содержать не более 64 символов, включая пробелы
- Желательно избегать употребления имен, совпадающих с именами встроенных функций или свойств Microsoft Access, например, Name, Property, Count и так далее.
- Имена полей и таблиц не должны начинаться с пробела или управляющего символа (коды ASCII 00-31).
- Имена полей и таблиц могут содержать любые символы, включая буквы, цифры, пробелы. Специальные символы за исключением точки (.), восклицательного знака (!), прямых скобок ([]).

Рассмотрим также описание типов данных, предлагаемых MS Access (табл.1).

Таблица 1. Типы данных Microsoft Access

Тип данных	Описание
Текстовый	Символьные или числовые данные, не требующие вычислений. Поле данного типа может содержать до 255 символов. Размер поля задается с помощью свойства Размер поля в нижней части окна Конструктора таблицы .

Поле MEMO	Поле MEMO предназначено для ввода текстовой информации, по объему превышающей 255 символов. Это поле может содержать до 65535 символов. Этот тип данных отличается от типа ТЕКСТОВЫЙ тем, что в таблице хранятся не сами данные, а ссылки на блоки текста , которые хранятся отдельно. За счет этого ускоряется обработка таблиц (сортировка, поиск и другие операции). Поле MEMO не может быть ключевым полем или проиндексированным полем.
Числовой	Числовой тип используется для хранения числовых данных, используемых в математических расчетах. Имеет много подтипов , от выбора которых зависит размер поля .
Дата/Время	Тип для представления даты и времени. Размер поля 8 байт.
Денежный	Тип данных, предназначенный для хранения данных, точность представления которых колеблется от 1 до 4 знаков после запятой. Целая часть данного типа может содержать до 15 десятичных знаков.
Счетчик	Поле содержит 4-байтный уникальный номер, определяемый MS Access для каждой новой записи автоматически путем увеличения предыдущего значения на 1 или случайным образом. Значение поля типа счетчик обновлять нельзя. Максимальное значение числа записей в таблице с полем счетчика не должно превышать 2 миллиарда.
Логический	Логическое поле, которое может содержать только два значения, интерпретируемых как Да/Нет, Истина/Ложь, Включено/Выключено. Поле логического типа не может быть ключевым, но это поле можно проиндексировать.
Поле объекта OLE	Содержит ссылку на OLE-объект (лист Excel, документ Word, звук, изображение и так далее). Объем объекта ограничивается только имеющимся в наличие на диске пространством.
Гиперссылка	Позволяет вставлять в поле гиперссылку, с помощью которой можно сослаться на произвольный фрагмент данных внутри файла или страницы на том же компьютере, в локальной сети или в сети Internet. Поле типа ГИПЕРССЫЛКА не может быть ключевым или индексированным.
Вложение	Любой совместимый по формату объект: картинка, файл..

В таблице ТИП_ПОКУПАТЕЛЯ мы использовали поле СЧЕТЧИК для автоматической генерации ключевого значения, ТЕКСТОВОЕ поле размером в 50 символов и поле типа MEMO для хранения дополнительной символьной информации.

Самостоятельно

1. По аналогии с таблицей ТИП_ПОКУПАТЕЛЯ создайте остальные таблицы справочники: ПОСТАВЩИК, ТИП_ТОВАРА, ФОРМА_ОПЛАТЫ. Для создания новой таблицы выберите пункт меню **Создание – Таблица** (рис.4).

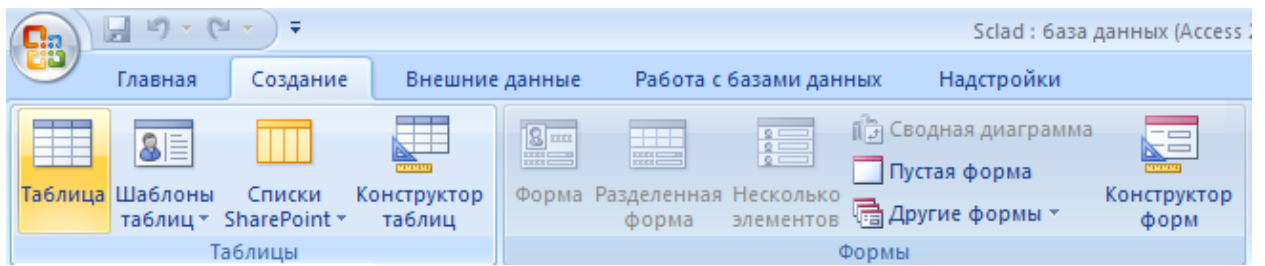


Рис.4. Создание новой таблицы

Каждому типу данных соответствует список свойств. Рассмотрим свойство **Маска ввода**. Маска ввода обеспечивает различные виды управления вводом данных. Маска ввода применяется в следующих случаях:

- для проверки значения поля на соответствие определенному шаблону ввода, например, шаблон (000)000-00-00 задает ввод номера телефона;
- для автоматического ввода постоянных символов-разделителей, например. дефиса (-) или круглых скобок;
- для обозначения позиции ввода с тем, чтобы облегчить ввод данных;
- для сохранения данных вместе с константами или без констант, в зависимости от вашего предпочтения.

Создание маски ввода для поля облегчено тем, что существует специальный Мастер, помогающий настраивать существующие маски и создавать новые маски. В табл.2. приведены символы маски ввода. В табл.3 показаны примеры масок ввода.

Таблица 2. Символы маски ввода

Символ	Вводимое в поле значение
0	Цифра (от 0 до 9), ввод обязателен, символы плюс и минус не допускаются.
9	Цифра или пробел, ввод не обязателен, символы плюс и минус не допускаются.
#	Цифра или пробел, ввод не обязателен, допускаются символы плюс и минус.
L	Буква (от A до Z или от A до Я), ввод обязателен.
?	Буква, ввод не обязателен.
A	Буква или цифра, ввод обязателен.
a	Буква или цифра, ввод не обязателен.
&	Любой символ или пробел, ввод обязателен.
C	Любой символ или пробел, ввод не обязателен.
.,;:-/	Десятичный разделитель, а также разделитель тысяч, дат и времени.
<	Перевод всех следующих символов в нижний регистр.
>	Перевод всех следующих символов в верхний регистр.
!	Заполнение маски ввода справа налево, а не слева направо (этот символ в маске ввода можно помещать в любую позицию).
	Ввод любого следующего символа в качестве текстовой константы (используется для ввода в маску перечисленных в данной таблице символов как текстовых констант, а не как управляющих символов).

Таблица 3. Примеры масок ввода

Маска ввода	Допустимые значения
(999)000-0000	(658)555-9999 ()555-9999
>LL0000	AB0932
Aaa	123 12 AB1 AB 1a
00000-9999	94000-0000 94000-
000-00-0000	999-99-9999
>L<??????????	Петров Сидоров

1. Самостоятельно создайте остальные таблицы базы данных: ПОКУПАТЕЛЬ, ЗАКАЗ, ПУНКТ_ЗАКАЗА, ТОВАР, ПОЛКА, ПАРТИЯ.

2. Самостоятельно разработайте маски ввода для полей Телефон, Покупатель, Поставщик, КоличествоВПартии, ТипТовара, МинимальныйЗапас, Количество.

Свойства полей и схема данных

Цель

Научиться устанавливать свойства полей таблицы и создавать схему данных.

Задание 1. Включение таблиц в схему данных и определение связей между ними

1. С помощью команд меню Работа с базами данных – Схема данных откройте диалог выбора таблиц.

2. Выделите все таблицы и нажмите кнопку Добавить.

3. Закройте диалоговое окно.

4. Визуальными методами (drag and drop) выполните создание связей между таблицами в соответствии с моделью данных (рис.1).

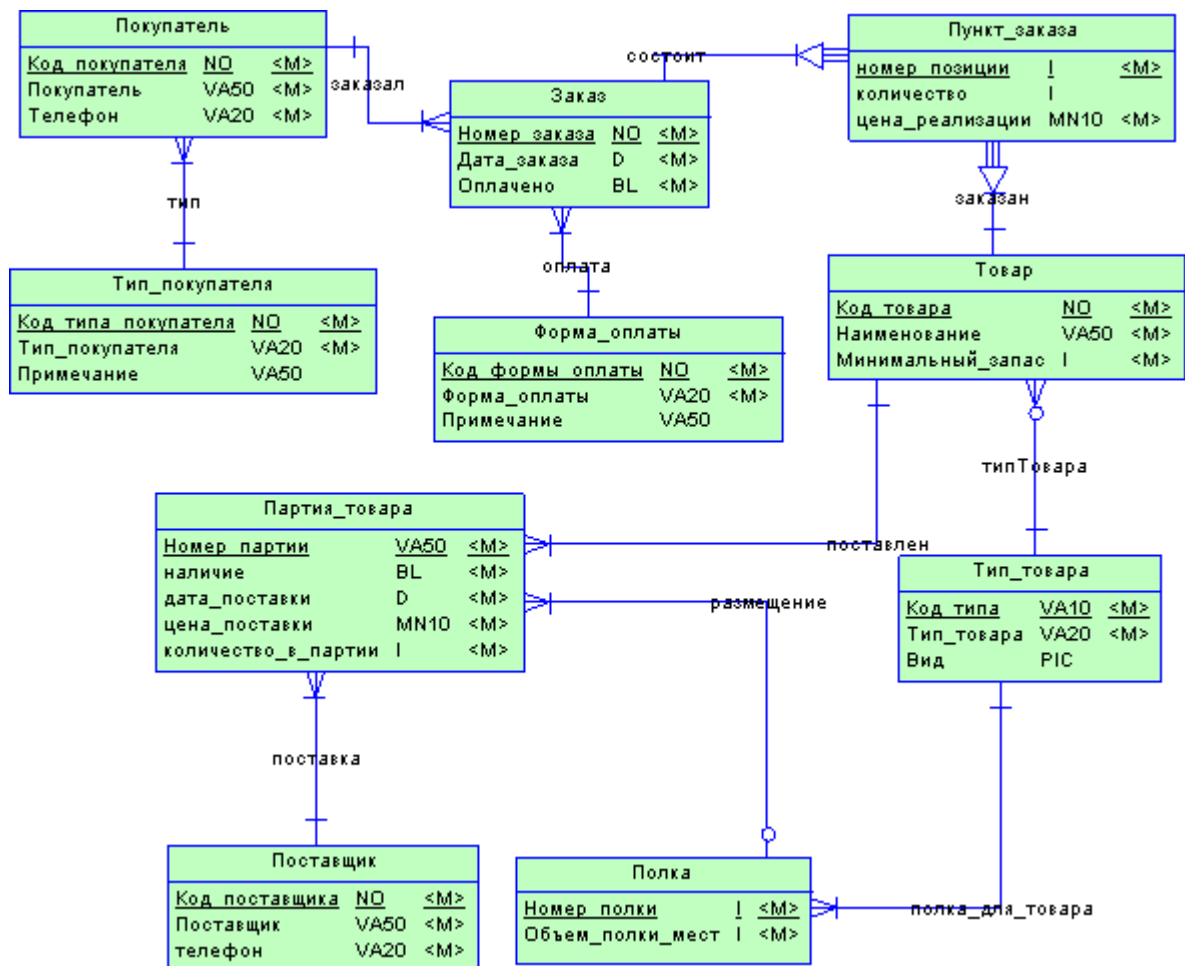


Рис.1. Логическая модель данных Склад

5. При установлении связей желательно включать поддержку целостности данных (рис.2).
6. Полученную схему данных нужно сохранить. Можно закрыть окно, но можно оставить схему доступной для просмотра.

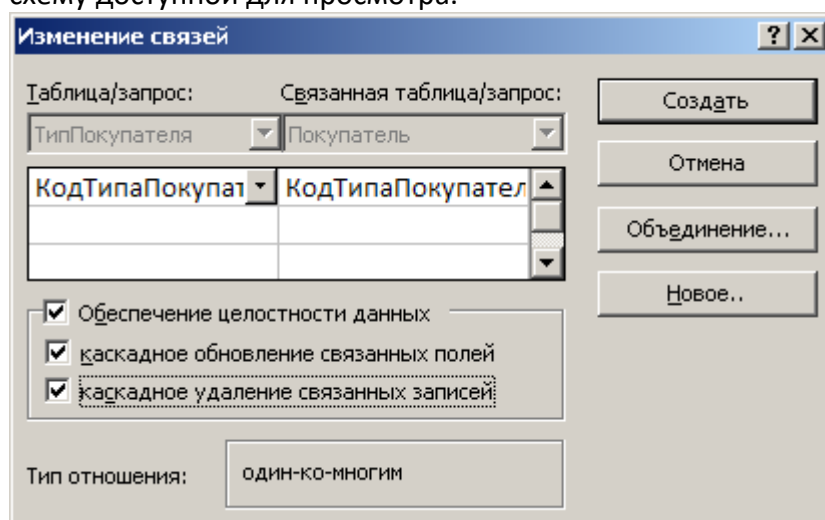


Рис.2. Целостность данных

Задание 2. Собственный список значений для поля подстановки

В СУБД MS Access существует возможность разработки списка значений для подстановки в столбец таблицы. Такой прием оправдан, если список значений представляет собой конечное множество, которое можно ввести на стадии проектирования таблиц.

1. Для таблицы ТипПокупателя можно организовать подобный список значений поля ТипПокупателя.

2. Раскройте таблицу ТипПокупателя в режиме Конструктор и откройте выпадающий список Типов данных для поля ТипПокупателя (рис.3).
3. Выберите Мастер подстановок.

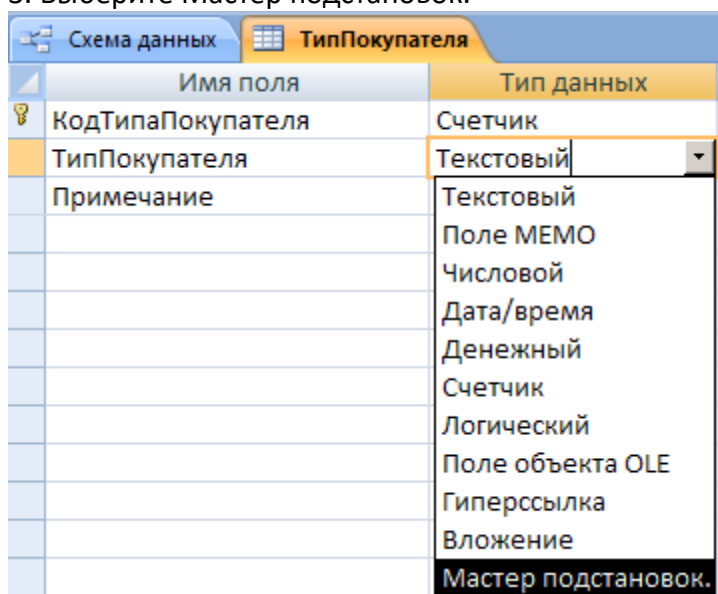


Рис.3. Вызов Мастера подстановок для поля ТипПокупателя

4. Выберите режим Фиксированный набор значений. Введите несколько значений различных типов покупателей.
5. Перейдите в режим ввода данных (Вид – Режим таблицы) и внесите несколько записей в таблицу ТипПокупателя.

В поле КОД_ТИПА_ПОКУПАТЕЛЯ значения вводить не надо, значения должны формироваться автоматически, пропускайте это поле, например, нажатием клавиши TAB. В поле ТИП_ПОКУПАТЕЛЯ следует выбирать значения из встроенного списка. При вводе данных в поле ПРИМЕЧАНИЕ (тип поля МЕМО) нажмите клавиши SHIFT + F2 (рис.4).

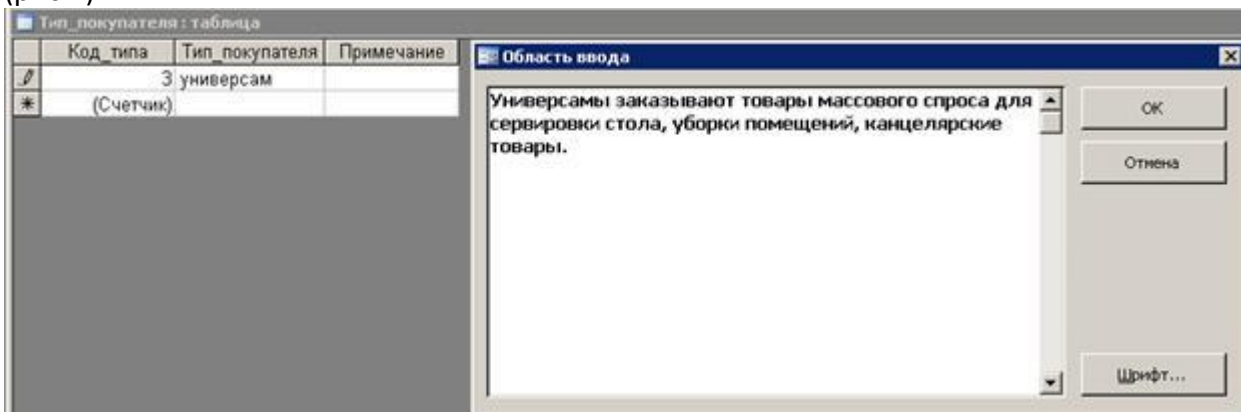


Рис.4. Ввод значений в поле МЕМО

Задание 3. Внесение изменений в список подстановки

После создания списка с помощью МАСТЕРА ПОДСТАНОВОК можно вносить в него изменения. Это можно сделать двумя способами:

- Открыть таблицу в режиме КОНСТРУКТОР и перезапустить МАСТЕР ПОДСТАНОВОК.
- Отредактировать свойства поля подстановки на вкладке ПОДСТАНОВКА (рис.5) в нижней части окна КОНСТРУКТОРА.

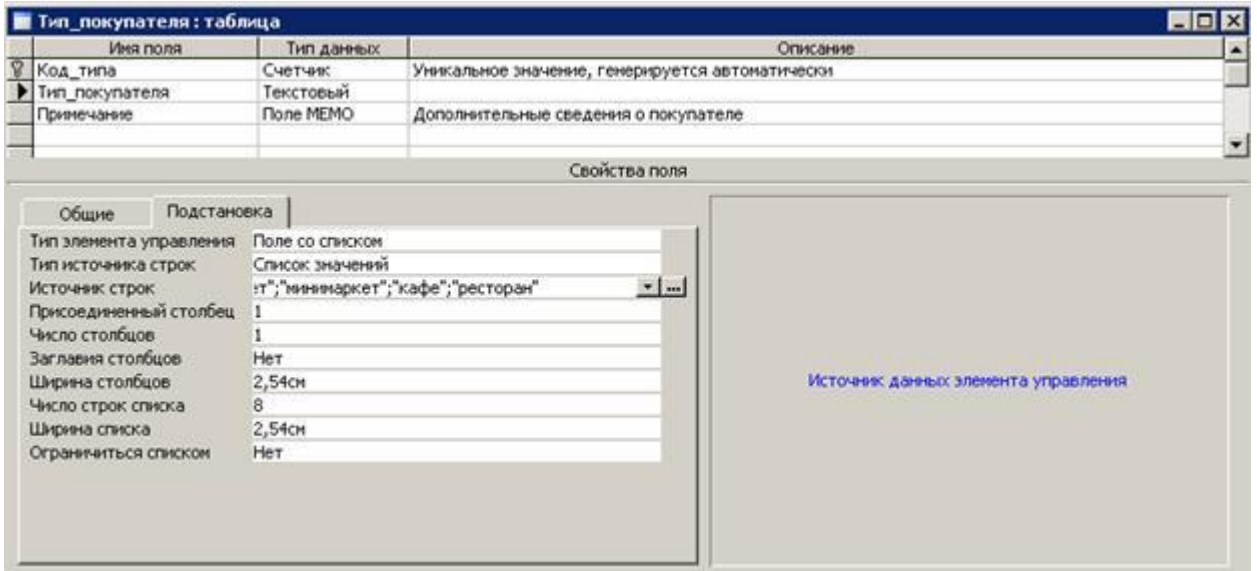


Рис.5. Внесение изменений в список подстановки

1. Внесите изменения в список, созданный для поля с подстановкой.

Задание 4. Определение для поля значения по умолчанию

Если в таблице есть поле, которое всегда, или в большинстве случаев, будет содержать одно и то же значение, можно задать для него **значение по умолчанию**.

Значение по умолчанию задается в свойстве поля ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ в нижней части КОНСТРУКТОРА ТАБЛИЦЫ (рис.6).

Для назначения свойства ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ можно также использовать функции Visual Basic. **Функция** – это фрагмент программного кода, который возвращает значение определенного типа. Например, функция Date() возвращает текущую дату, функция Now() возвращает дату и время.

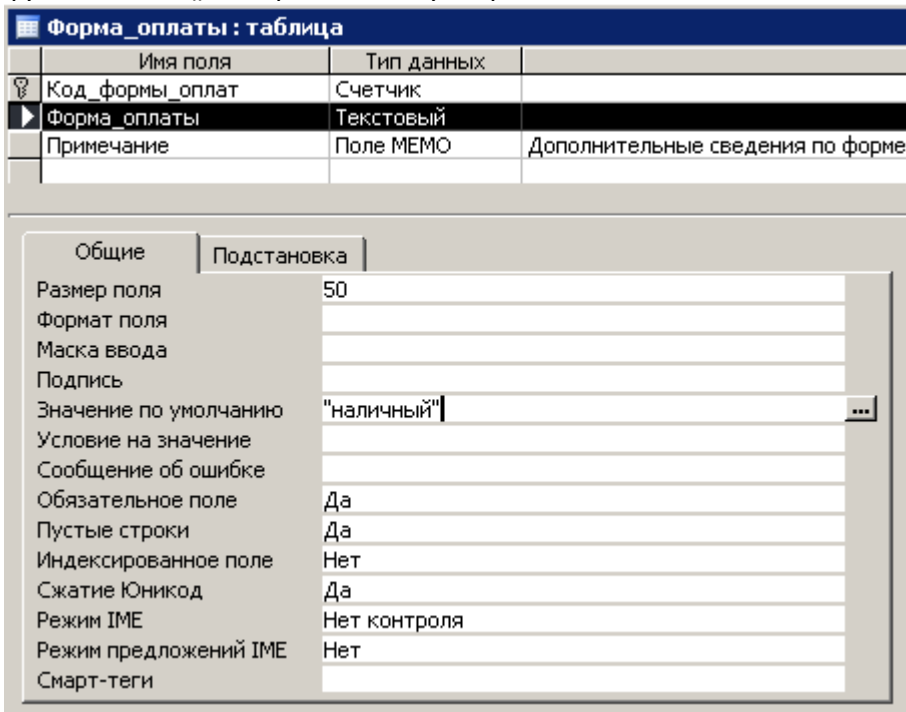


Рис.6. Использование константы в качестве значения по умолчанию

Для вставки функции в качестве значения по умолчанию выполните следующие действия:

1. Установите курсор в поле (например, поле ДАТА_ЗАКАЗА[1] таблицы ЗАКАЗ) и щелкните по кнопке ПОСТРОИТЕЛЯ ВЫРАЖЕНИЙ (рис.7).

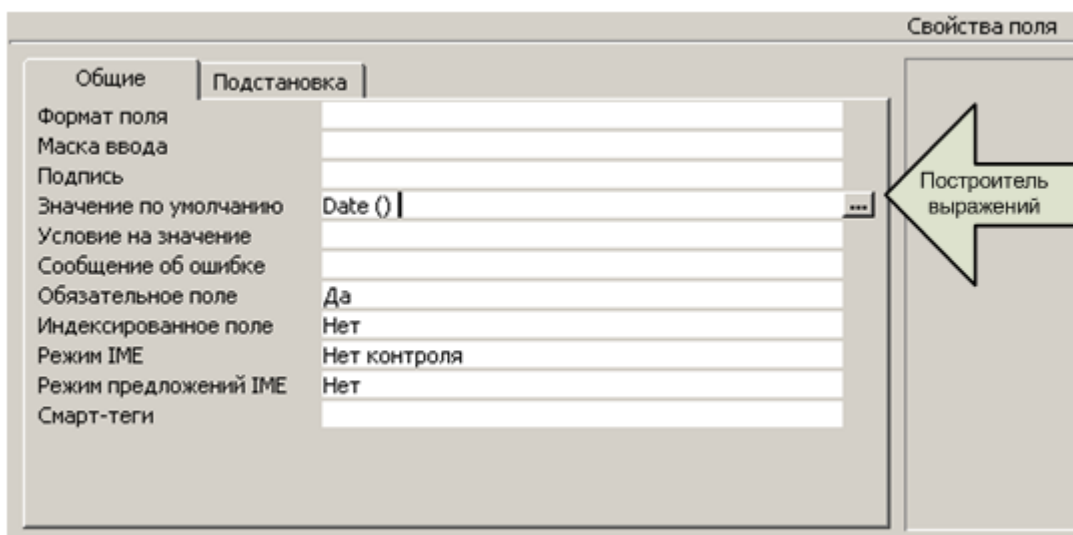


Рис.7. Вызов ПОСТРОИТЕЛЯ ВЫРАЖЕНИЙ

2. В появившемся диалоге (рис.8) откройте папку ФУНКЦИИ, затем ВСТРОЕННЫЕ ФУНКЦИИ. Выберите категорию Дата/Время и, наконец, функцию Date().

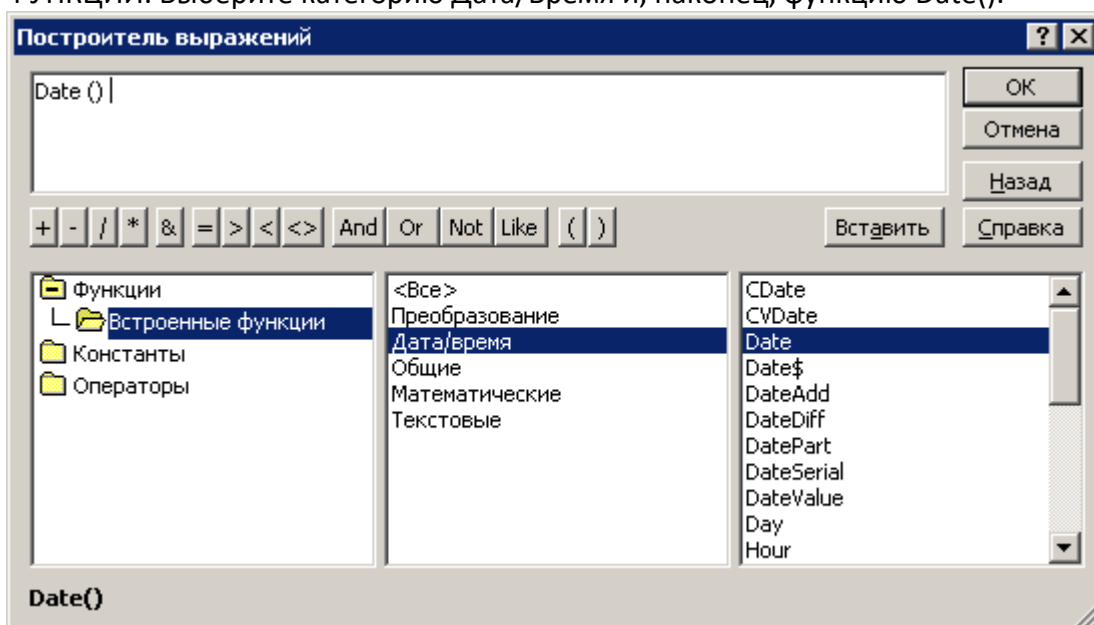


Рис.8. Работа с ПОСТРОИТЕЛЕМ ВЫРАЖЕНИЙ


3. Для всех полей, имеющих тип Дата/Время, добавьте значение по умолчанию с помощью встроенных функций.

Задание 5. Индексирование полей в таблице

Индексы в MS Access используются для ускорения поиска данных по одному или нескольким полям. Любая таблица, содержащая ключевое поле (поля) автоматически индексируется по это эму полю (полям). Такой индекс называют **первичным индексом**. Вам может потребоваться создавать собственные индексы по другим полям, для которых часто выполняется операция поиска, например, ДАТА_ЗАКАЗА, ОБЪЕМ_ПОЛКИ и так далее.

Индексы также полезны для неключевых полей, по которым выполняется объединение таблиц, а также для полей, по которым выполняется сортировка.

Для создания **дополнительного индекса** в таблице раскройте таблицу в режиме Конструктора:

1. Щелкните на кнопке ИНДЕКСЫ  на панели инструментов. Появится диалоговое окно (рис.9).

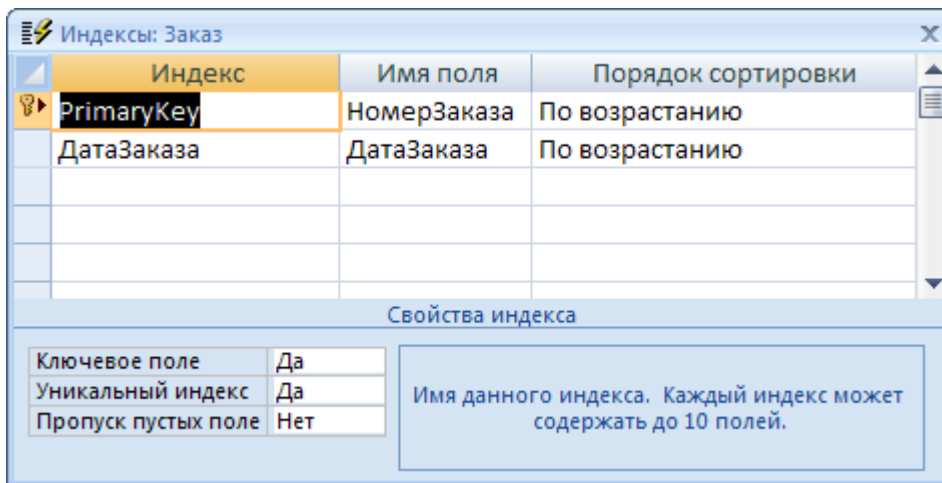


Рис.9. Диалог ИНДЕКСЫ

2. Поместите курсор в пустую строку. Поле ИНДЕКС не заполняйте, а в поле ИМЯ_ПОЛЯ выберите поле таблицы, по которому еще нет индексирования.
3. Установите значения свойств УНИКАЛЬНЫЙ ИНДЕКС и ПРОПУСК ПУСТЫХ ЗНАЧЕНИЙ.
4. Научитесь удалять созданные вами индексы.

Задание 6. Форматы полей

При изучении MS EXCEL мы научились создавать собственные форматы для представления данных (пользовательские форматы).

Приемы по созданию форматов полей таблиц MS Access аналогичны. На рис.10 показан пример формата для поля ОБЪЕМ_ПОЛКИ.

1. Создайте собственные форматы представления данных для большинства полей таблиц базы данных.
2. Примените имеющиеся стандартные форматы (рис.10).
3. Внесите данные в таблицы и убедитесь, что ваши форматы «работают» (рис.11).

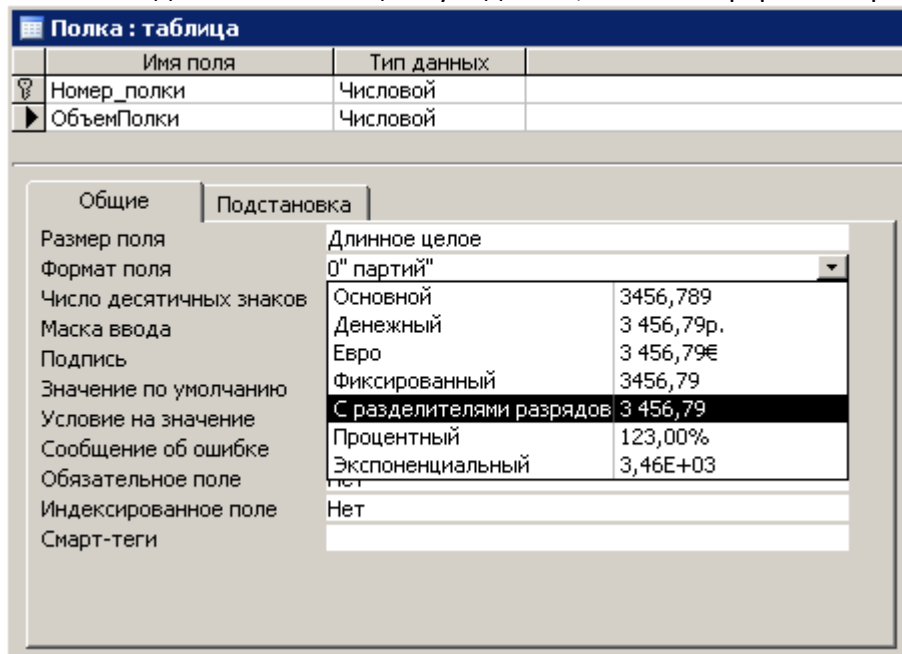


Рис.10. Создание собственных форматов для полей таблицы

Номер_полки	ОбъемПолки	Тип_товара
1	2 партий	
0	0 партий	

Запись: 1 из 1

Рис.11. Проверка «работы» созданных форматов

Задание 7. Проверка поддержания связной целостности данных

Проверим, как влияют установленные ранее параметры поддержки целостности при внесении изменений в таблицы Покупатель и ТипПокупателя.

1. Откройте таблицы Покупатель и ТипПокупателя в режиме таблицы.
2. Внесите несколько связанных записей в эти таблицы.
3. Удалите запись, имеющую связанные записи, из таблицы ТипПокупателя.
4. Вставьте запись в таблицу Покупатель с кодом ТипаПокупателя, который отсутствует в таблице ТипПокупателя.
5. Внесите изменения в содержание записи в таблице ТипПокупателя и убедитесь, что для связанных записей эти изменения выполнились автоматически.

Задание 8. Ввод и корректировка записей во взаимосвязанных таблицах

Если для связей между таблицами установлена поддержка целостности данных, то пользователь может при удалении, изменении и добавлении записей сам отслеживать непротиворечивость данных и целостность связей (рис.12).

КодПокупат	Покупатель	Телефон	КодТипаПо	Добавить поле
2	Федоров Е.Л.	234-56-54	3	

НомерЗака	ДатаЗаказа	Оплачено	КодФормы	Добавить по
(№)	24.02.2009	<input checked="" type="checkbox"/>		
(№)				

Рис.12. Отображение подчиненных записей в главной таблице

1. Организуйте ввод связанных непротиворечивых записей в таблицы ТипТовара и Товар (рис.13).

КодТипа	ТипТовара	Вид	Добавить поле
КанцБум	бумажная кан		

КодТовара	Наименова	Минималь	Добавить по
1	блокнот в тве	100	
2	блокнот в мяг	100	
(№)			

Рис.13. Ввод связанных записей

2. Продемонстрируйте ввод и корректировку данных в другие взаимосвязанные таблицы.

Задание 9. Поиск и замена значений в полях таблиц

Иногда требуется найти определенное значение в полях таблиц и заменить его на другое (или не изменять).

Для выполнения поиска и замены следует открыть таблицу и в меню Главная воспользоваться режимами поиска и замены (рис.14).

1. Примените все параметры из списка Совпадение. Научитесь менять направление поиска.

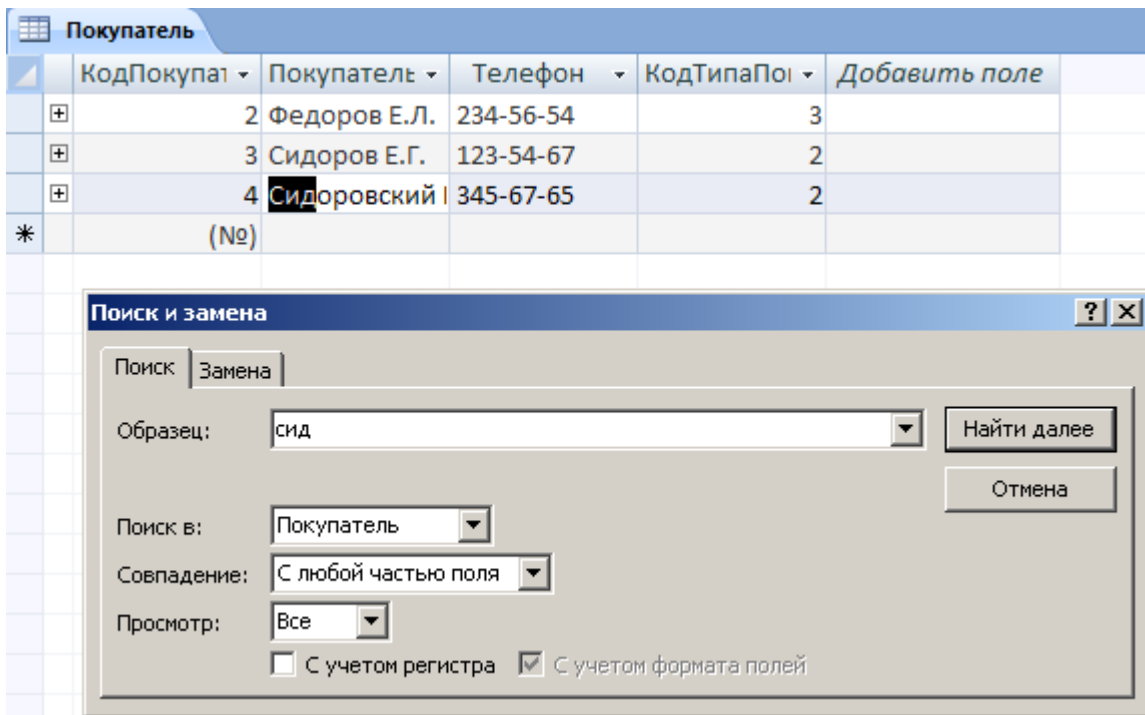


Рис.14. Режим поиска и замены

2. Примените режимы сортировки и фильтрации для записей таблицы (меню Главная – Сортировка и фильтр).

Конструирование форм

Цель

Научиться конструировать и использовать формы как объекты отображения и ввода данных в таблицы базы данных.

После того как созданы все необходимые таблицы базы данных, установлены связи между ними, мы должны создать объекты, которые обеспечат работу пользователя с данными в этих таблицах.

Microsoft Access имеет специальные объекты для организации взаимодействия пользователя с данными в таблицах: **форм и отчеты**.

Формы предназначены как для отображения, так и для ввода данных в таблицы. Формы позволяют отобразить данные в виде, удобном для пользователя, обеспечить корректировку этих данных, ввод новых записей и удаление ненужных.

Формы, как правило, не применяются для вывода данных на печать, для этого используются **отчеты**.

Задание 1. Окно конструктора форма

Начнем разработку форм для таблиц-справочников. В нашей модели это таблицы: ТИП_ПОКУПАТЕЛЯ, ФОРМА_ОПЛАТЫ, ПОСТАВЩИК и ТИП_ТОВАРА. Конструирование форм, как и всех других объектов, будем выполнять в режиме КОНСТРУКТОРА.

1. В меню Создание – Конструктор форм нажмите кнопку Добавить существующие поля и раскройте список полей соответствующей таблицы (рис.1).

Макет формы первоначально содержит только один раздел – область данных. Это основной раздел формы, в нем будут располагаться поля для заполнения таблицы. Но форма имеет и другие разделы.

ЗАГОЛОВОК формы определяет верхнюю часть формы. Этот раздел добавляется в форму вместе с разделом ПРИМЕЧАНИЕ формы.

2. В области ЗАГОЛОВОК поместите текст с названием формы с помощью элемента НАДПИСЬ из панели инструментов (рис.2).

3. Щелкните правой кнопкой мыши в левый верхний угол рамки надписи (то есть вызовите контекстное меню объекта).
4. Выберите из контекстного меню команду СВОЙСТВА. Настройте внешний вид надписи (рис.2).

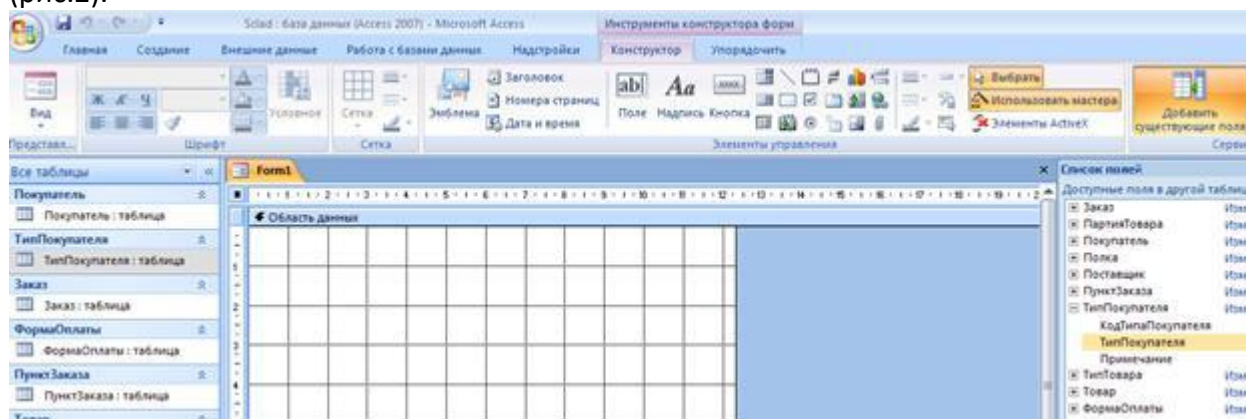


Рис.1. Вызов Конструктора форм для таблицы



Рис.2. Оформление ЗАГОЛОВКА формы

5. Ширина и высота каждого раздела регулируется мышью. Измените высоту области ЗАГОЛОВКА.
6. В область ДАННЫХ перенесите мышью все поля таблицы из списка, расположенного в окне КОНСТРУКТОРА формы (рис.3).
7. Внешний вид и размеры полей можно также изменить (команда СВОЙСТВА каждого из полей).
8. Для просмотра внешнего вида конструируемой формы периодически следует переключаться в режим формы.
9. Поместите в области ПРИМЕЧАНИЙ любую надпись, например, как показано на рис. 3.

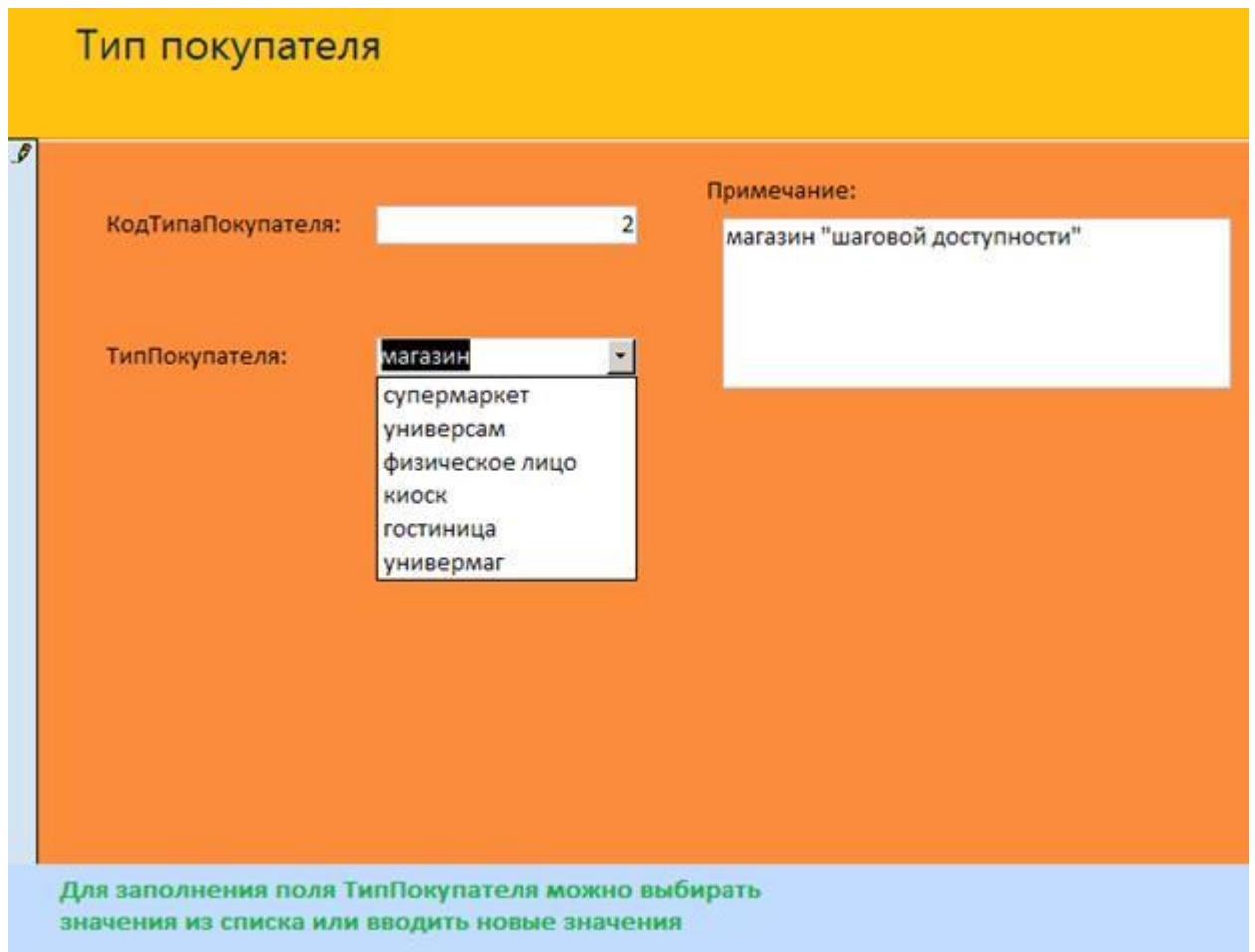


Рис.3. Изменение размеров и оформления элементов формы

10. Добавьте в форму графическое изображение в область ЗАГОЛОВКА с помощью элемента РИСУНОК из панели инструментов (рис.4).

11. Выполните его форматирование командой СВОЙСТВА. Измените УСТАНОВКУ РАЗМЕРОВ и ВЫРАВНИВАНИЕ рисунка (рис.4).

12. Можно добавить фоновый рисунок для всей формы. Вызовите список свойств ФОРМЫ как показано на рис.5 (щелкните в кнопку в левом верхнем углу формы).

13. В СВОЙСТВАХ формы укажите название файла рисунка в команде РИСУНОК и включите МОЗАИЧНОЕ ЗАПОЛНЕНИЕ, если это необходимо.

14. Чтобы удалить фоновый рисунок выделите команду РИСУНОК в свойствах формы и нажмите клавишу DELETE.

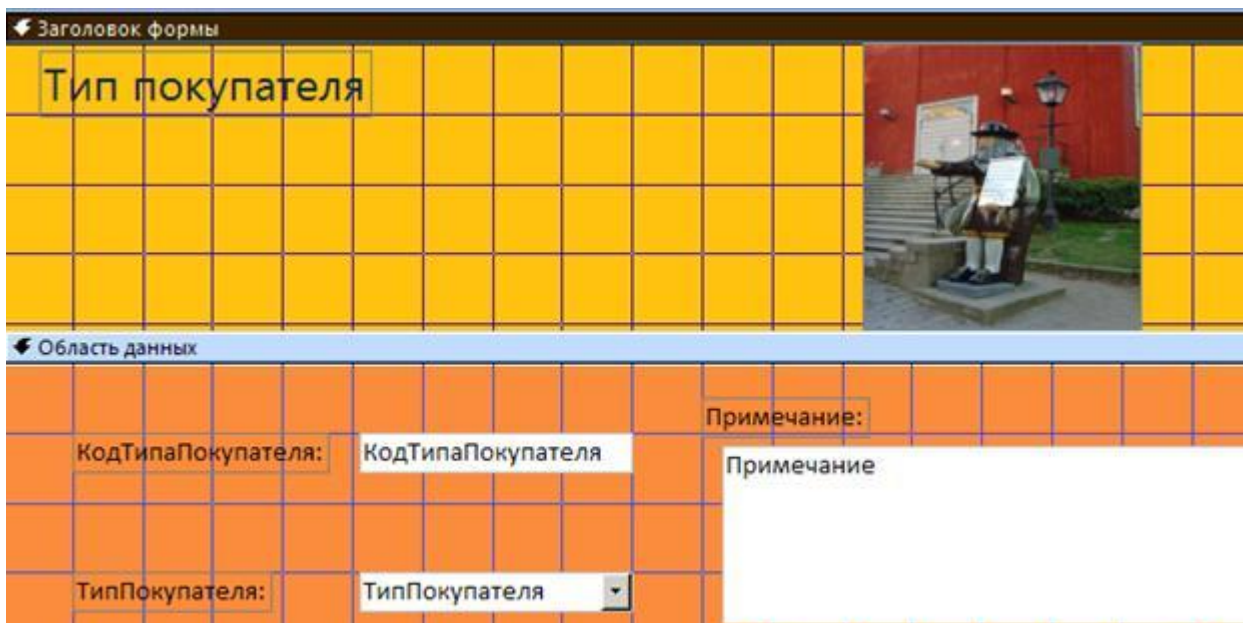


Рис.4. Форматирование рисунка в форме

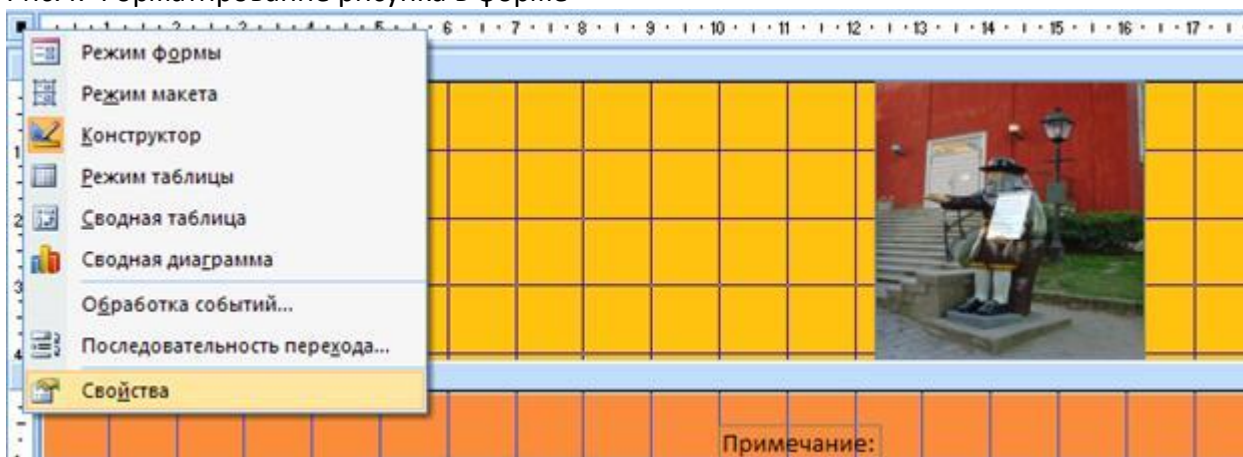


Рис.5. Свойства объекта ФОРМА

15. Сохраните сконструированную форму и дайте форме имя – ТИП_ПОКУПАТЕЛЯ.
16. Перейдите в режим формы и внесите несколько записей в таблицу ТИП_ПОКУПАТЕЛЯ с помощью созданной формы.
17. Устраните возникшие неудобства работы с формой в режиме КОНСТРУКТОРА формы. Например, можно удалить поле КОД_ТИПА_ПОКУПАТЕЛЯ, так как значение в этом поле формируются автоматически при создании новой записи (рис.6).
18. Значения в поле ТИП_ПОКУПАТЕЛЯ можно выбирать из встроенного списка или непосредственно вводить. Проверьте эту возможность.
19. Просмотрите наличие новых записей в таблице ТИП_ПОКУПАТЕЛЯ.

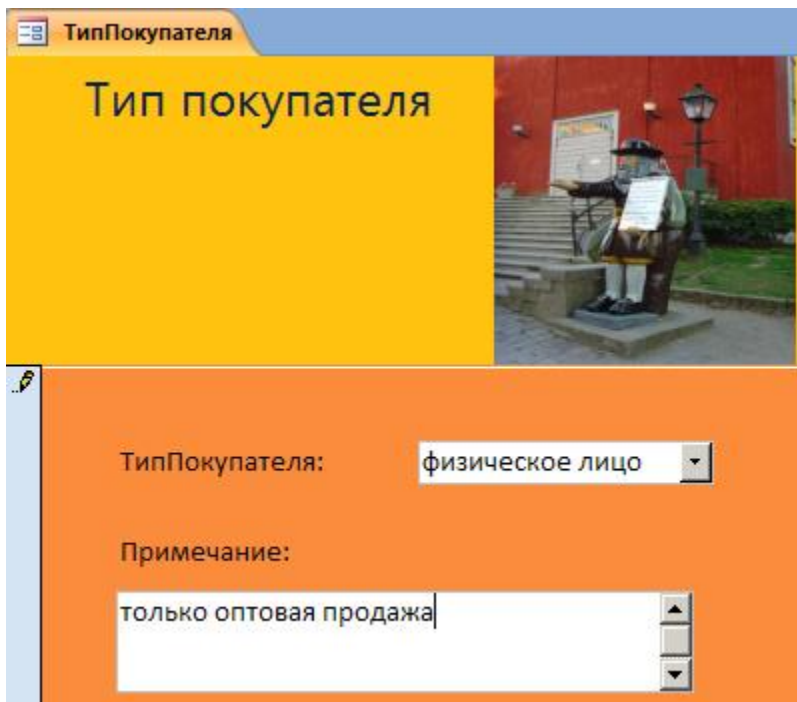
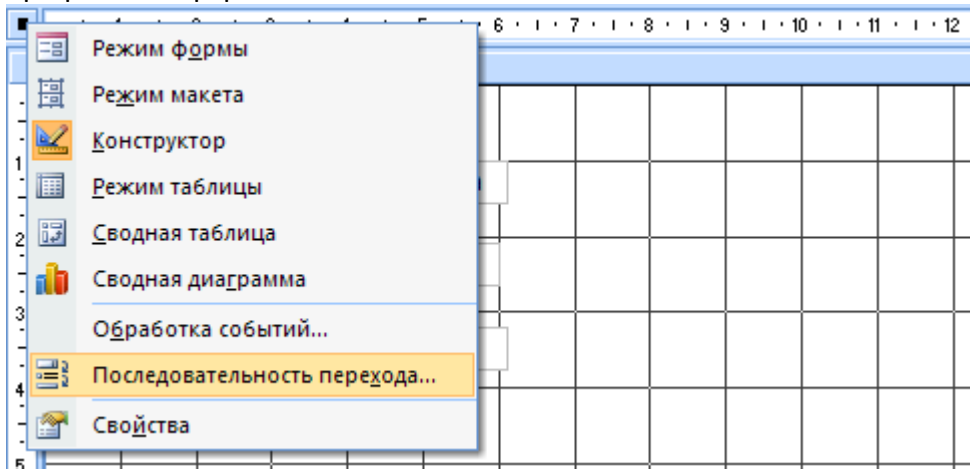


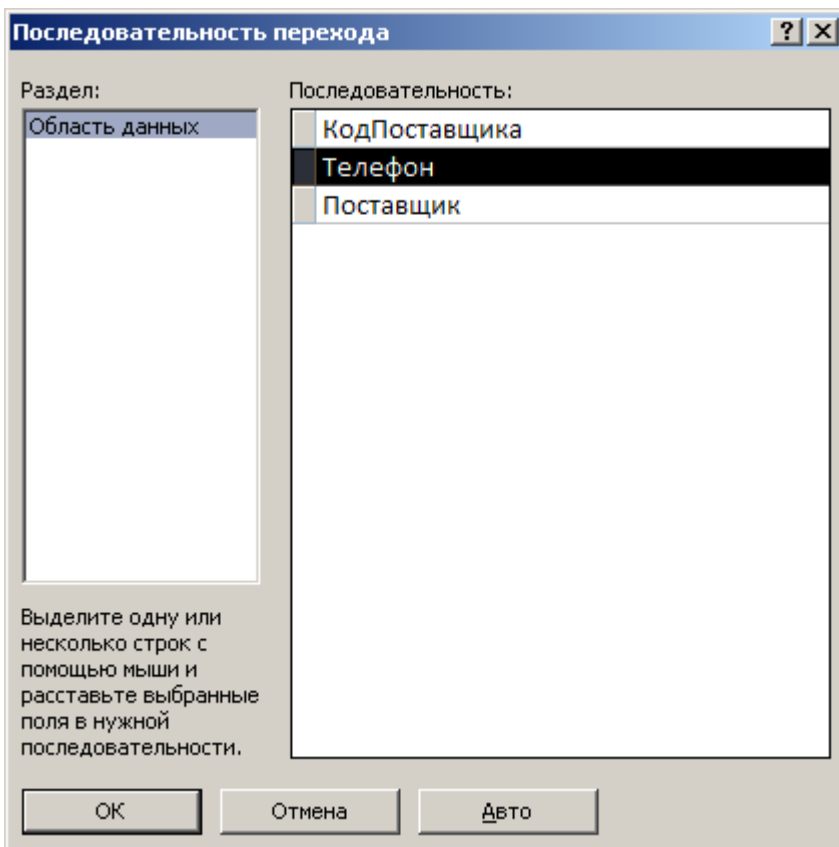
Рис.6. Удаление полей, заполнение которых выполняется автоматически

Самостоятельно

1. Сконструируйте форму для таблицы-справочника ПОСТАВЩИК.
2. Внесите записи в таблицу ПОСТАВЩИК с помощью формы.
3. Проверьте их наличие в таблице ПОСТАВЩИК.
4. На рис. 7 показана технология изменения последовательности перехода между полями при работе с формой.



а). Выбор команды из контекстного меню формы



б). Изменение порядка следования полей

Рис.7 а, б. Изменение последовательности перехода по полям формы

5. Внесите изменения в порядок следования полей для формы ПОСТАВЩИК.

На рис.8. показана форма в Режиме Таблицы.

6. Поработайте с формой в режиме таблицы: вставьте и удалите записи (рис.9). Убедитесь в том, что записи вставляются и удаляются в таблице ПОСТАВЩИК.

КодПоставки	Поставщик	Телефон
1	Универсам Окей №12	123-34-56
2	Универсам Перекресток №	154-98-90
*	(№)	

Рис.8. Изменение режима отображения формы

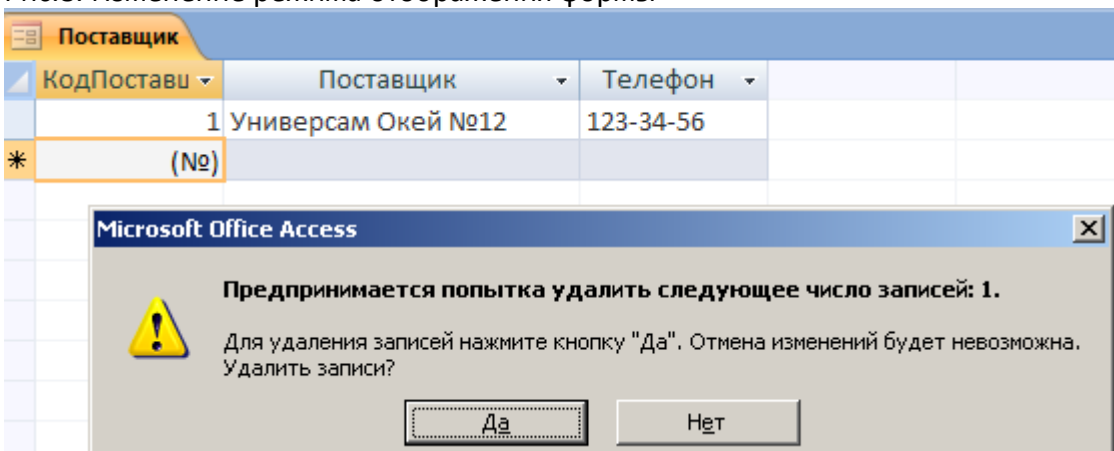


Рис. 9. Удаление записей из формы

Задание 2. Использование элементов управления Конструктора форм

На рис.10 приведены элементы панели КОНСТРУКТОРА форм. Мы знакомы со следующими элементами: НАДПИСЬ, ПОЛЕ, РИСУНОК.

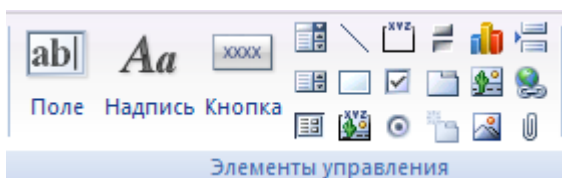


Рис.10. Элементы панели управления КОНСТРУКТОРА форм

Продолжим знакомство с другими элементами этой панели.

Элементы управления СВОБОДНАЯ РАМКА ОБЪЕКТА и ПРИСОЕДИНЕННАЯ РАМКА ОБЪЕКТА используются для того, чтобы вставить в форму объекты, созданные в других приложениях, например, документ Word, таблицу Excel или рисунок.

Эти элементы различаются тем, будут ли объекты, вставленные в эти рамки, связаны с записями в таблицах Access, или не будут.

СВОБОДНАЯ РАМКА ОБЪЕКТА содержит объекты, которые хранятся либо в самой форме, либо во внешнем файле, и независимо от того, какая запись отображается в форме.

Объект будет отображаться один и тот же. Этот объект может быть изменен прямо из формы с помощью того приложения, в котором он был создан.

ПРИСОЕДИНЕННАЯ РАМКА ОБЪЕКТА содержит объекты, которые либо сами хранятся в таблице Access в поле, имеющем тип ПОЛЕ ОБЪЕКТА OLE, либо в таблице хранится ссылка на эти объекты. При переходе от записи к записи объект в присоединенной рамке объекта будет меняться.

1. Создайте форму для таблицы ТИП_ТОВАРА. Переместите мышью поля КОД_ТИПА и ТИП_ТОВАРА из списка полей в область данных.

2. Переместите поле ВИД из списка полей в область данных, правой кнопкой мыши вызовите контекстное меню объекта и убедитесь, что был создан объект ПРИСОЕДИНЕННАЯ РАМКА ОБЪЕКТА (рис.11).

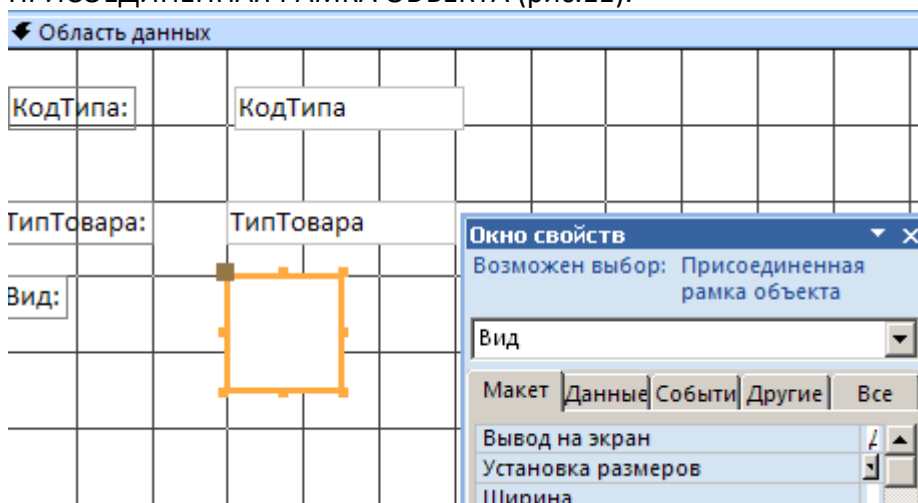


Рис.11. Создание присоединенной рамки объекта

3. Настройте свойства рамки по своему усмотрению. Измените размер, положение, тип границы.

4. Перейдите в режим формы и добавьте новую запись с помощью созданной формы.

Вставка объекта в рамку для графического изображения выполните, например, с помощью команды ВСТАВИТЬ ОБЪЕКТ из контекстного меню поля ВИД. На рис.12. показаны шаги для вставки графического файла.

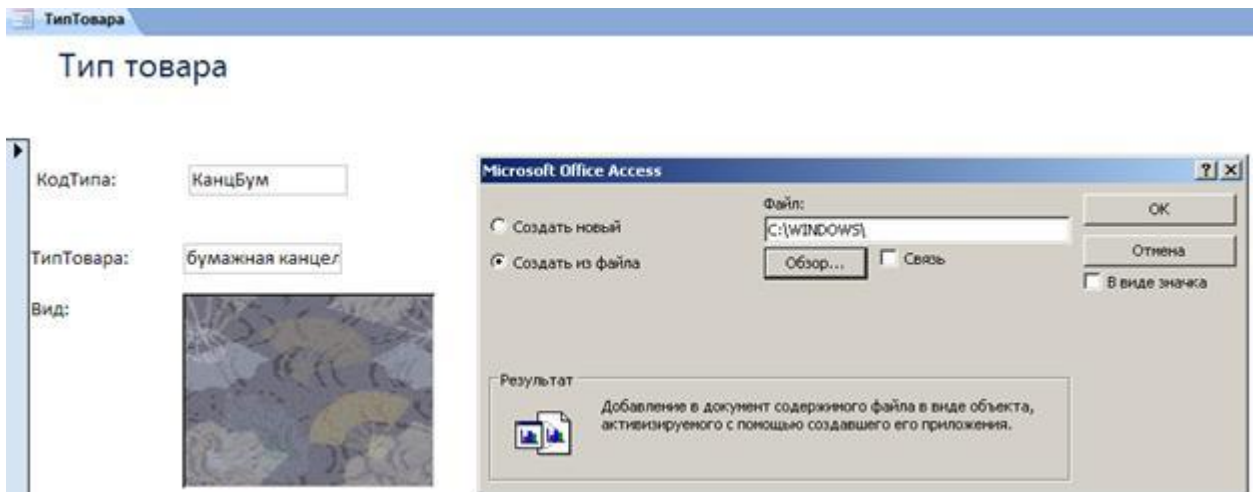


Рис.12. Вставка файла с изображением в поле присоединенной рамки

5. Добавьте несколько записей с графическими изображениями. Убедитесь, что ссылки на вставленные объекты записываются в поле Вид таблицы ТИП_ТОВАРА (рис.13) и вы можете открыть эти объекты для возможного редактирования.

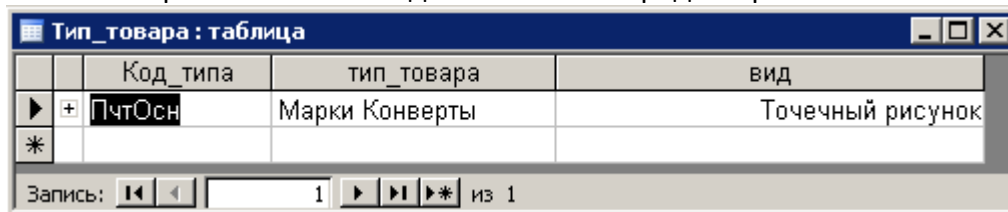


Рис.13. Вид записи в таблице с полем типа OLE

Элемент управления НАБОР ВКЛАДОК используется для создания многостраничных форм.

6. Откройте форму ТИП_ТОВАРА в режиме КОНСТРУКТОРА.

7. Удалите все поля в форме ТИП_ТОВАРА, тем самым освободите на бланке формы место под набор вкладок. Щелкните на кнопке ВКЛАДКА на панели инструментов, затем щелкните на бланке формы в том месте, где должен появиться набор вкладок.

8. Переименуйте вкладки (введите ПОДПИСИ в режиме СВОЙСТВА для каждой вкладки), например, как показано на рис.14.

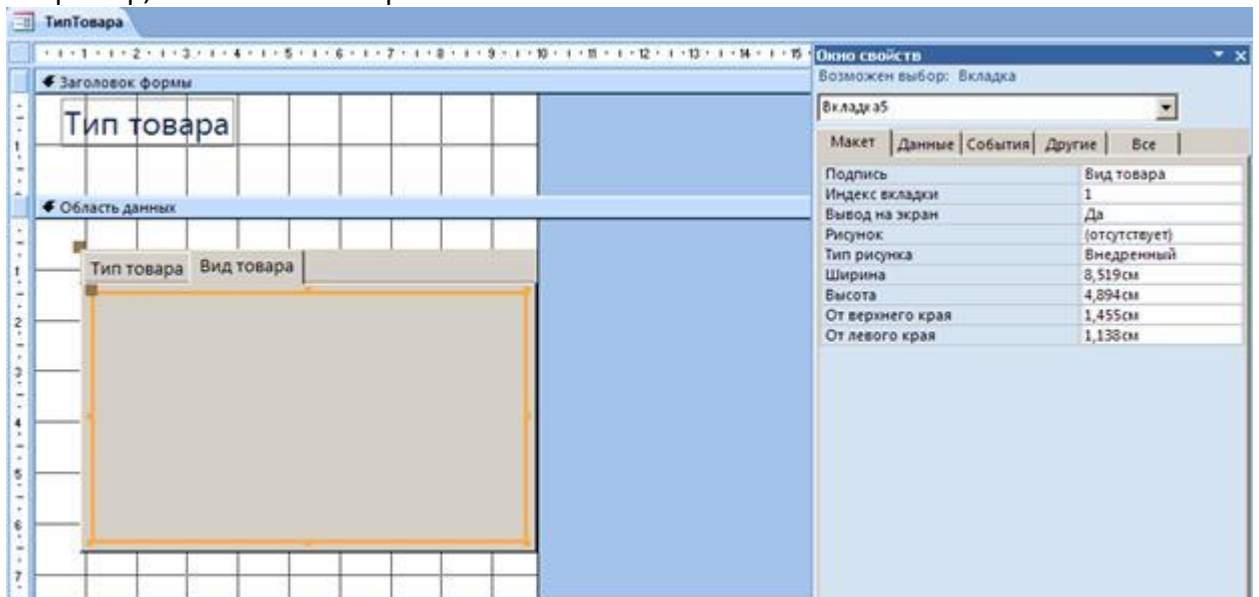


Рис.14. Конструирование вкладок в форме

9. Переместите поля (мы их удалили) из списка полей на каждую из вкладок.

10. Внесите несколько записей в таблицу ТИП_ТОВАРА с помощью измененной формы.

Элемент управления КНОПКА является очень важным элементом формы, поскольку именно с кнопками связаны различные действия. Выполняемые пользователем в приложении: сохранение введенных данных, вызов другой формы, вывод на печать и так далее.

Для создания КНОПКИ удобно использовать МАСТЕР КНОПОК, поскольку существует коллекция стандартных процедур, код которых уже разработан.

В форме ТИП_ТОВАРА создадим кнопку для удаления текущей записи.

11. Откройте форму ТИП_ТОВАРА в режиме КОНСТРУКТОРА.

12. Щелкните по элементу КНОПКА на панели инструментов.

13. Указателем мыши обведите размер будущей кнопки в форме ТИП_ТОВАРА.

14. В диалоговом окне выберите команды в соответствии с рис. 15.

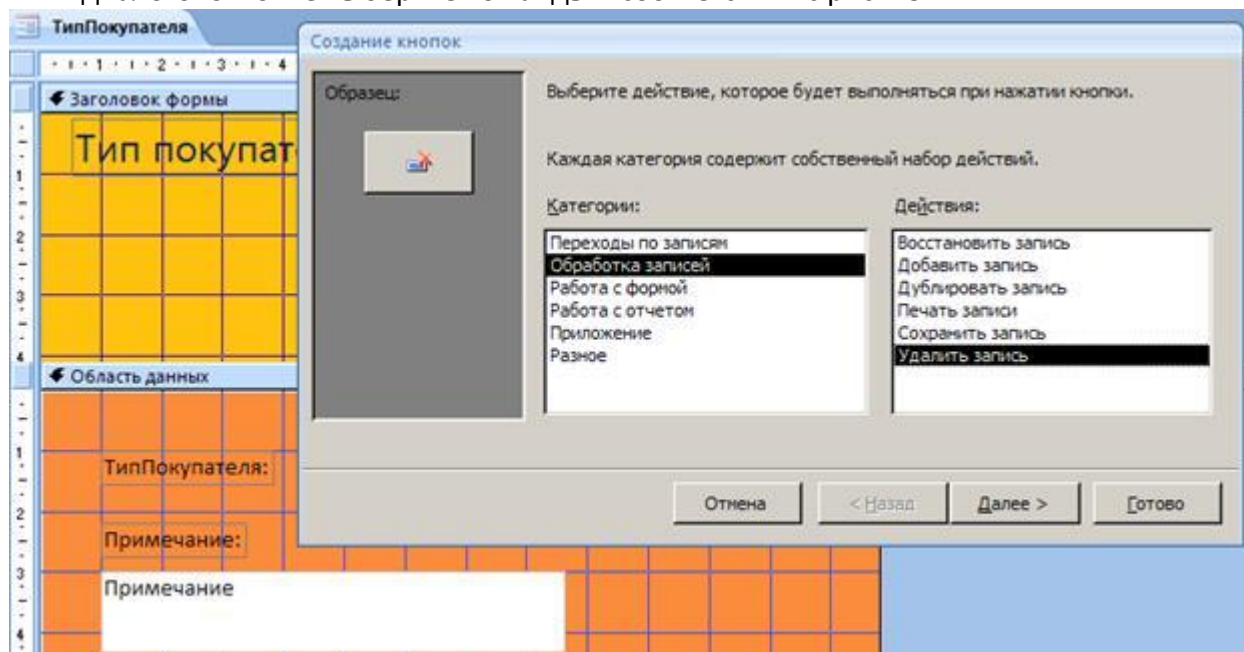


Рис.15. МАСТЕР КНОПОК

15. Пройдите остальные шаги МАСТЕРА КНОПОК. Найдите подходящий рисунок для кнопки.

16. Убедитесь в том, что установленная в форме кнопка выполняет удаление текущей записи не только в форме, но и таблице.

Самостоятельно

Примените элементы управления ЛИНИЯ и ПРЯМОУГОЛЬНИК для оформления созданных форм.

Создайте кнопки для закрытия форм и перехода по записям формы.

Примените все освоенные приемы для оформления формы ФормаОплаты.

Проектирование форм для работы с данными связанных таблиц

Цель

Научиться конструировать и использовать формы для работы с данными нескольких взаимосвязанных таблиц – многотабличные формы.

Проектирование формы для работы со связанными данными выполняется на основе схемы данных, состоящей из нормализованных таблиц, связанных одно/многочисленными отношениями.

В процессе проектирования формы целесообразно выполнить приведенную ниже последовательность действий:

1. Определить таблицу, которая будет источником записей основной части формы.

2. Выбрать таблицы, которые будут источниками записей, загружаемых в поля основной таблицы.

3. Выбрать способ реализации для источников записей.

Задание 1. Проектирование формы для работы с данными двух связанных таблиц

Разработаем форму для таблицы ПОКУПАТЕЛЬ. Таблица ПОКУПАТЕЛЬ будет источником записей основной части формы.

1. На основе схемы данных определим данные для конструирования формы (рис.1).



Рис.1. Подсхема данных для конструирования формы на базе таблицы ПОКУПАТЕЛЬ
Источник записей для поля КОД_ТИПА_ПОКУПАТЕЛЯ – таблица ТИП_ПОКУПАТЕЛЯ.
Реализацию источника записей удобно выполнить с помощью элементов управления СПИСОК или ПОЛЕ СО СПИСОКОМ.

2. Создадим форму в режиме КОНСТРУКТОРА для таблицы ПОКУПАТЕЛЬ (рис.2).

3. Переместим из списка полей в область данных формы поля ПОКУПАТЕЛЬ и ТЕЛЕФОН.



Рис.2. Создание формы ПОКУПАТЕЛЬ

4. Щелкните на кнопке ПОЛЕ СО СПИСОКОМ на панели инструментов.

5. Перетащите поле ТИП_ПОКУПАТЕЛЯ из списка в область данных формы.

6. Появится диалоговое окно Мастера. На первом шаге выберите режим «Объект будет использовать значения из таблицы или запроса» (рис.3).

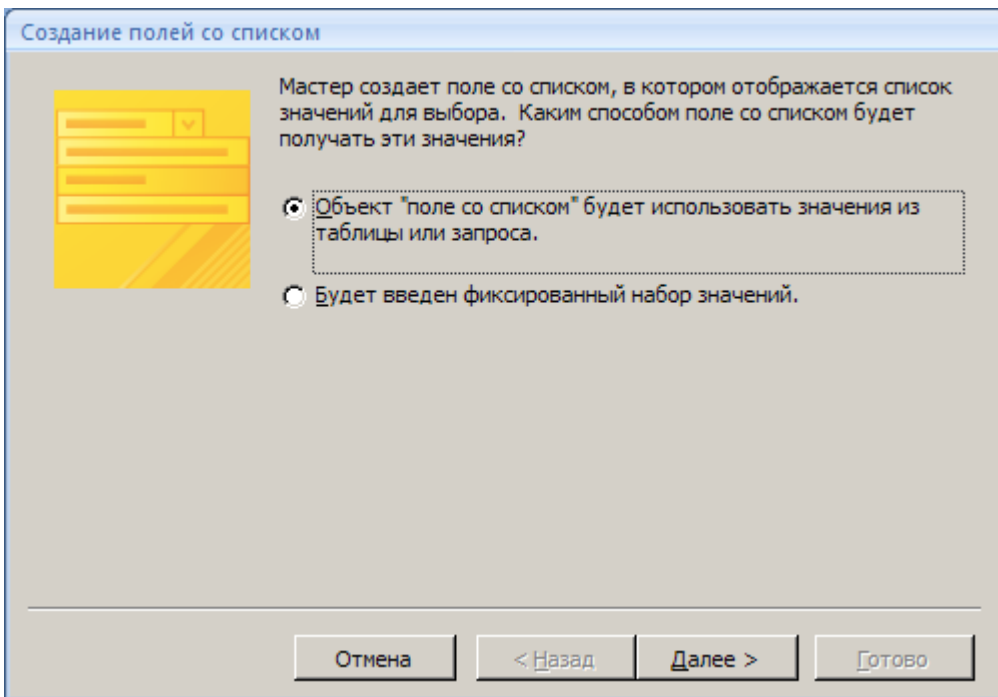


Рис.3. Первый шаг Мастера создания списков

7. На следующем шаге Мастера вам необходимо указать название таблицы-источника записей. В нашем случае укажите таблицу ТИП_ПОКУПАТЕЛЯ.

8. В следующем диалоге вам предлагают определиться с составом будущего списка. В нашем случае удобнее перенести оба поля: КОД_ТИПА является полем для связи таблиц, а поле ТИП_ПОКУПАТЕЛЯ содержит смысловую информацию для пользователя (рис. 4).

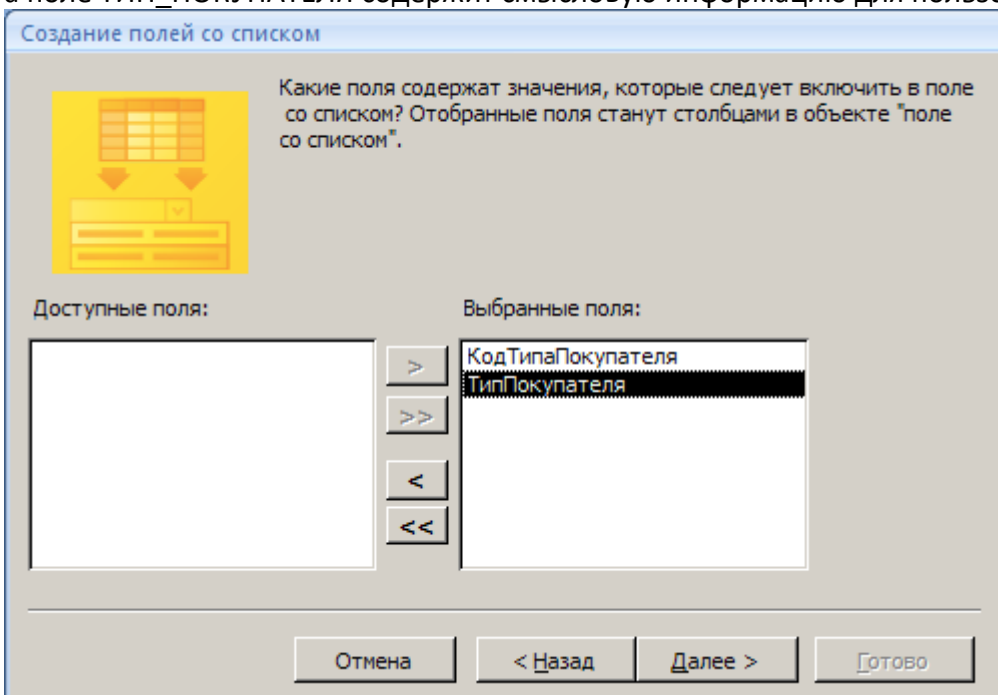


Рис.4. Выбор полей для списка

9. На следующем шаге МАСТЕР предполагает вам скрыть ключевой столбец. Можно согласиться с предложением, это удобно.

10. И, наконец, последний шаг МАСТЕРА, – в каком поле таблицы ПОКУПАТЕЛЬ следует сохранить значения из таблицы ТИП_ПОКУПАТЕЛЬ. Если названия полей совпадают (у нас это не так), то обычно, МАСТЕР делает верное предположение, иначе вам необходимо выбрать из списка нужное поле (ТИП_ПОКУПАТЕЛЯ).

11. Результат должен быть примерно таким, какой показан на рис.5.

12. Внесите несколько записей с помощью новой формы.
13. Убедитесь, что записи попадают в таблицу ПОКУПАТЕЛЬ.
14. Добавьте кнопку для удаления текущей записи (рис.5).

Рис.5. Окончательный вид формы для 2-х связанных таблиц

Самостоятельно

1. Разработайте и выполните реализацию форм для таблиц ПОЛКА (рис.6) и ТОВАР.
2. При конструировании форм используйте элементы управления СПИСОК и ПОЛЕ СО СПИСОКОМ.

Рис.6. Использование элемента управления СПИСОК

Упражнение 2. Проектирование многотабличных форм

Многотабличная форма создается для работы с данными нескольких взаимосвязанных таблиц. Многотабличная форма может состоять из **основной части** и одной или нескольких **подчиненных** включаемых форм, то есть быть составной.

Проектирование многотабличных форм выполняется на основе схемы данных, состоящей из нормализованных таблиц, связанных одно/многочисленными отношениями.

В процессе проектирования формы целесообразно выполнить приведенную ниже последовательность действий:

- Определить таблицы, которые будут источником записей основной части формы.
- Выбрать таблицы, которые будут источниками записей подчиненных форм, включаемых в составную форму.
- Определить состав полей каждой из форм.
- Выбрать способы реализации подчиненных форм и источников записей.

Многотабличная форма может быть создана в режиме КОНСТРУКТОРА или с помощью МАСТЕРА форм. Наиболее технологичным является использование КОНСТРУКТОРА.

1. На основе схемы данных определим данные для конструирования формы (рис.7).



Рис.7. Подсхема данных для конструирования формы на базе таблицы ЗАКАЗ
 Источник записей для основной части формы ЗАКАЗ – поля таблицы ЗАКАЗ. Поля таблицы ПУНКТ_ЗАКАЗА являются источниками записей подчиненной формы. Для поля основной части ПОКУПАТЕЛЬ источником записей будет таблица ПОКУПАТЕЛЬ. Для поля ФОРМА_ОПЛАТЫ источником записей – поля таблицы ФОРМА_ОПЛАТЫ. Реализацию списков записей для полей ПОКУПАТЕЛЬ и ФОРМА_ОПЛАТЫ в основной части формы удобно выполнить с помощью элементов управления ПОЛЕ СО СПИСКОМ. Реализацию полей подчиненной формы удобно выполнить с помощью вложенной формы. Следовательно, необходимо предварительно разработать подчиненную форму ПУНКТ_ЗАКАЗА.

2. В режиме КОНСТРУКТОРА создайте форму для таблицы ПУНКТ_ЗАКАЗА (рис.8).
3. Для поля ТОВАР источником записей является таблица ТОВАР. Реализацию удобно выполнить с помощью элемента ПОЛЕ СО СПИСКОМ (рис.8).



Рис.8. Форма ПУНКТ_ЗАКАЗА в режиме КОНСТРУКТОРА

4. Вносить записи с помощью созданной формы неудобно, так как нужно помнить номер заказа. Эта форма должна быть вложенной формой для формы Заказ. Форму ПунктЗаказа можно просмотреть в режиме ВИД (рис.9).

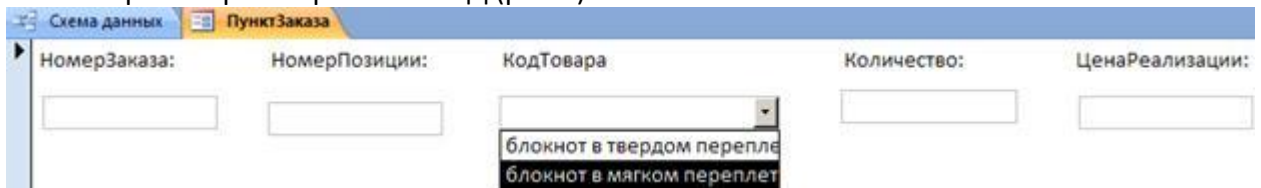


Рис.9. Окончательный вид подчиненной формы ПУНКТ_ЗАКАЗА

Прежде, чем начать создавать основную часть формы ЗАКАЗ, внесем некоторые изменения в нашу базу данных. Иногда используемые элементы интерфейса и способы реализации полей таблиц в форме Access позволяют сократить количество таблиц в схеме базы данных.

- В таблице ЗАКАЗ поле ФОРМА_ОПЛАТЫ имеет всего два значения: наличная оплата и безналичная оплата. В КОНСТРУКТОРЕ форм на панели инструментов есть элемент ПЕРЕКЛЮЧАТЕЛЬ, имеющий два состояния: 1/0, ДА/НЕТ, ВКЛ/ВЫКЛ. Использование элемента ПЕРЕКЛЮЧАТЕЛЬ для реализации поля ФОРМА_ОПЛАТЫ позволит:
- Хранить значения поля ФОРМА_ОПЛАТЫ в числовом формате, а не в текстовом формате (например, 1 – наличная, 0 – безналичная), что является экономией памяти.
 - Удалить из схемы таблицу ФОРМА_ОПЛАТЫ (комментарии к значениям поля можно вынести непосредственно в форму).
 - Реализовать понятный и привлекательный интерфейс формы.
5. Удалите из базы данных таблицу ФОРМА_ОПЛАТЫ.
 6. В режиме КОНСТРУКТОРА создайте форму для таблицы ЗАКАЗ (рис.13).
 7. Перенесите поля НОМЕР_ЗАКАЗА, ДАТА_ЗАКАЗА, ОПЛАЧЕНО в область данных формы из списка полей.
 8. С помощью свойств поля НОМЕР_ЗАКАЗА это поле можно сделать недоступным для ввода (поле типа счетчик), как показано на рис.15.
 9. Сконструируйте поле ПОКУПАТЕЛЬ с помощью элемента ПОЛЕ СО СПИСКОМ (источник записей – таблица ПОКУПАТЕЛЬ).
 10. Для создания поля ФОРМА_ОПЛАТЫ щелкните мышью по элементу ГРУППА ПЕРЕКЛЮЧАТЕЛЕЙ на панели инструментов КОНСТРУКТОРА и затем покажите в форме место для помещения этого элемента.
 11. Откроется первое диалоговое окно МАСТЕРА создания переключателей. Введите подписи для переключателей (рис.10).

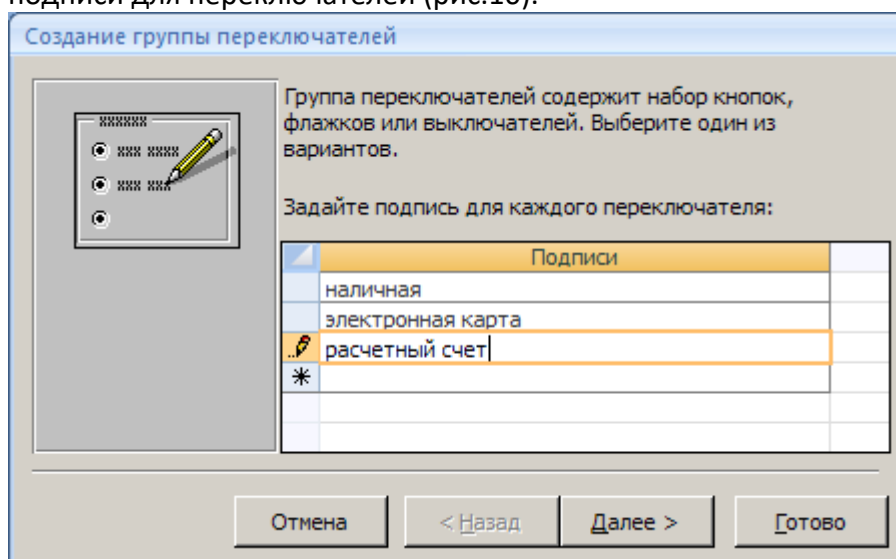


Рис.10. Мастер создания элементов ПЕРЕКЛЮЧАТЕЛИ

12. В следующем диалоге можно выбрать один из переключателей в качестве переключателя, выбранного по умолчанию.
13. В следующем окне диалога вам будет предложено установить числовые значения, которые будут сохранять в поле ФОРМА_ОПЛАТЫ в таблице ЗАКАЗ. Предположим, что наличная форма оплаты – это число 1, безналичная – число 0 (рис.11).
14. Далее надо указать в каком поле будут сохранены значения 1 или 0 (рис.12).

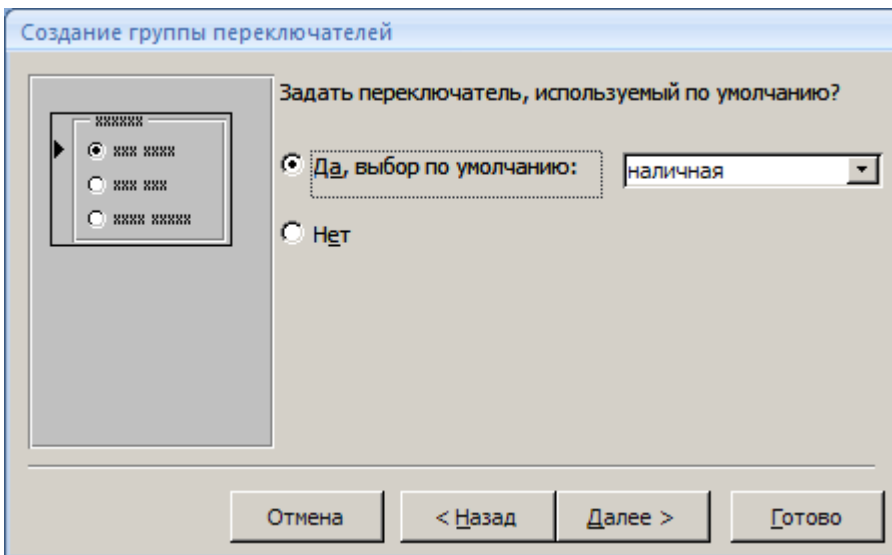


Рис.11. Создание переключателей в форме

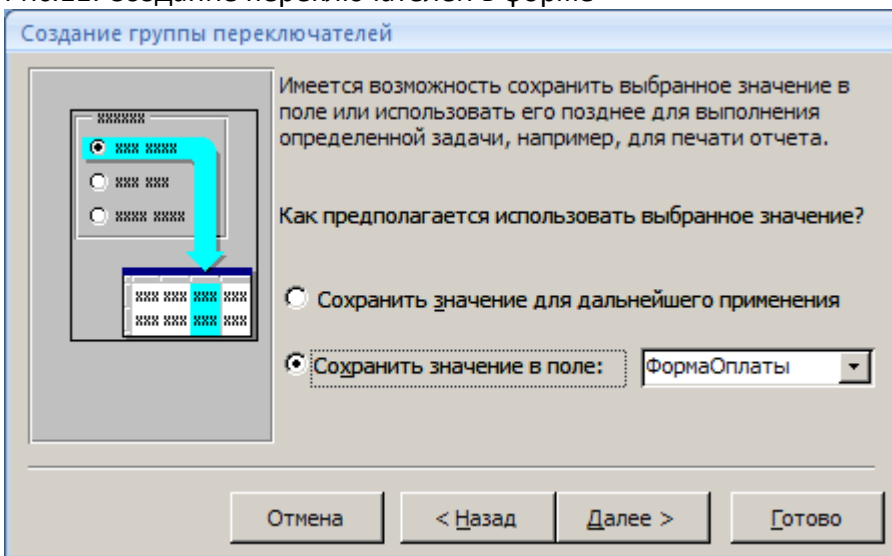


Рис.12. Сохранение значений переключателей

15. Следующий шаг определит дизайн вашей группы переключателей.

16. Последний шаг диалога. Введите название вашей группы переключателей, например – форма оплаты.

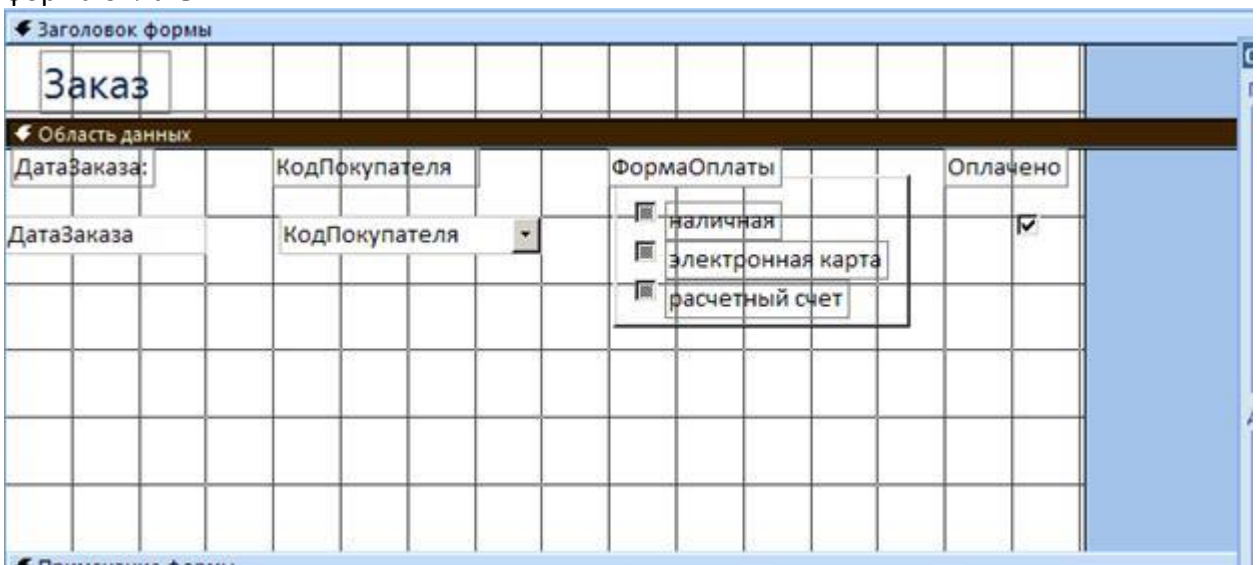


Рис.13. Форма ЗАКАЗ в режиме КОНСТРУКТОРА

Добавление подчиненной формы в основную форму может быть с помощью кнопки ПОДЧИНЕННАЯ ФОРМА/ОТЧЕТ на панели элементов режима КОНСТРУКТОР (рис.14).

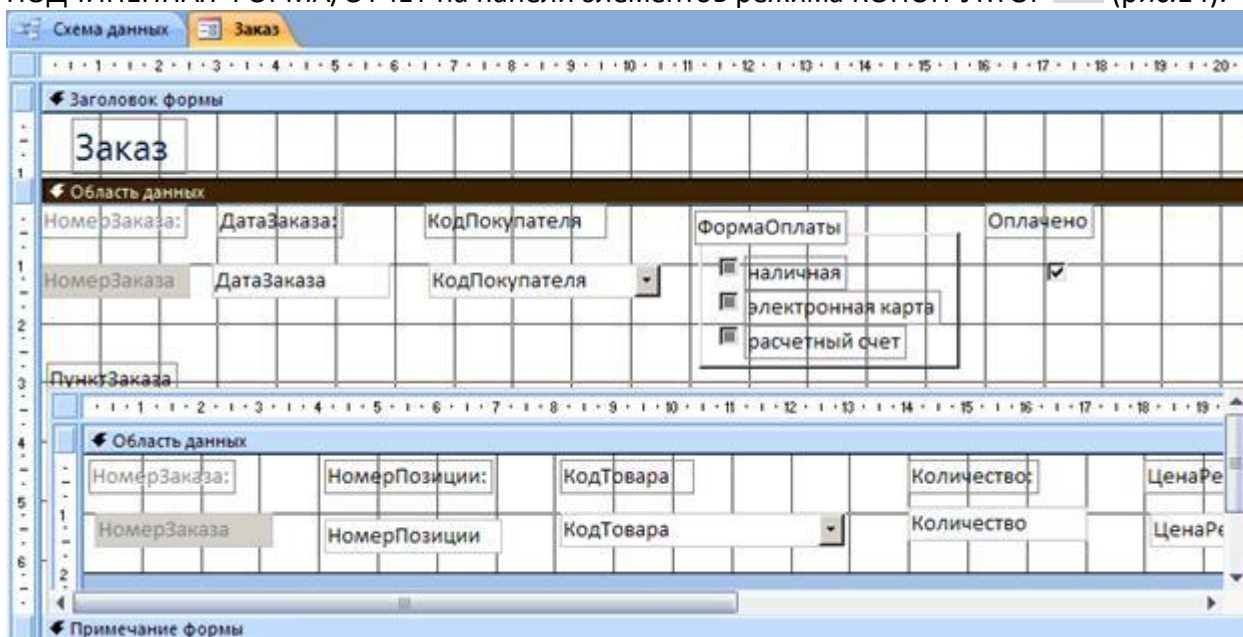


Рис.14. Размещение подчиненной формы в основной форме

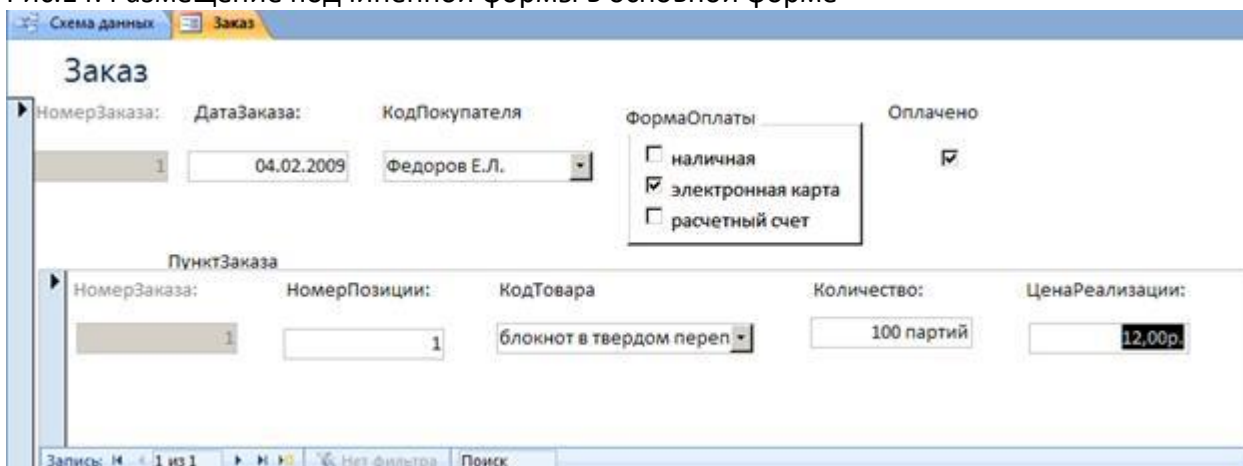


Рис.15. Окончательный вид многотабличной формы ЗАКАЗ

17. Проверьте работоспособность многотабличной формы (рис.15). Введите несколько заказов, каждый из которых состоит из нескольких позиций[2].

18. Проверьте, что введенные в форму данные помещаются в таблицы ЗАКАЗ и ПУНКТ_ЗАКАЗА.

Самостоятельно


1. Добавьте в форму ЗАКАЗ кнопки для выполнения следующих операций:

- Удаления текущей записи.
- Открытия формы ПОКУПАТЕЛЬ.
- Обновления состояния записей формы.

2. В свойствах поля ЦЕНА_РЕАЛИЗАЦИИ в подчиненной форме установите формат ДЕНЕЖНЫЙ (рис.15).

3. Проверьте возможность добавления нового покупателя непосредственно из формы ЗАКАЗ. Убедитесь, что эти записи попадают в таблицу ПОКУПАТЕЛЬ.

4. Добавьте в форму ТОВАР подчиненную формы, созданную на основе таблицы ПАРТИЯ_ТОВАРА.

5. В форме ПАРТИЯ_ТОВАРА создайте вычисляемое поле ЦЕНА_ПРОДАЖИ (рис. 16) с помощью ПОСТРОИТЕЛЯ ВЫРАЖЕНИЙ (кнопка  на панели инструментов КОНСТРУКТОРА формы) и элемента управления ПОЛЕ из панели элементов **abl**.

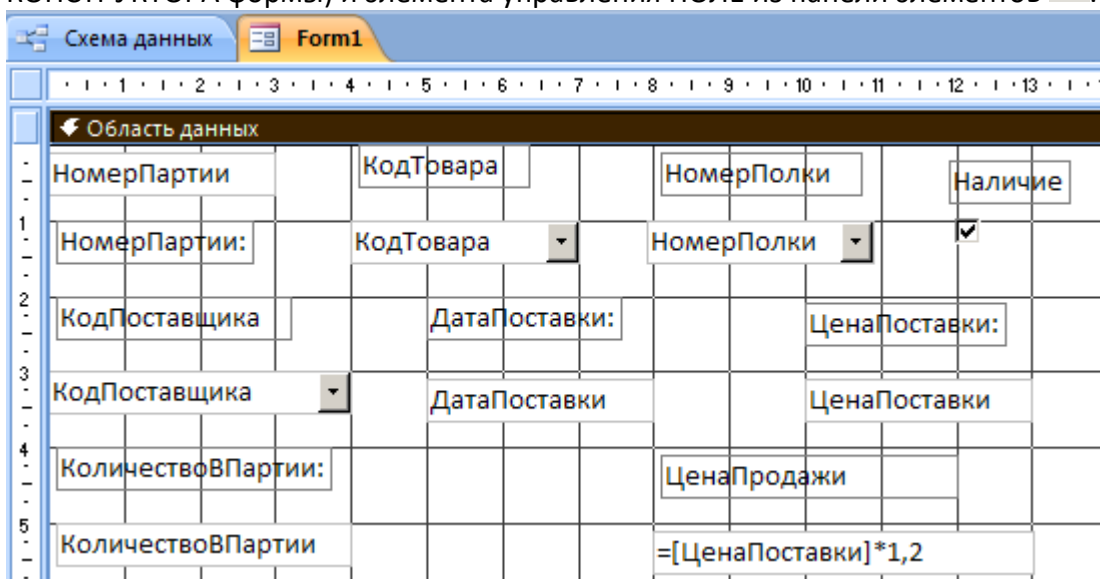


Рис.16. Создание вычисляемого поля в форме

На рис.17 показан примерный вид формы для таблицы ТОВАР в режиме КОНСТРУКТОРА формы.

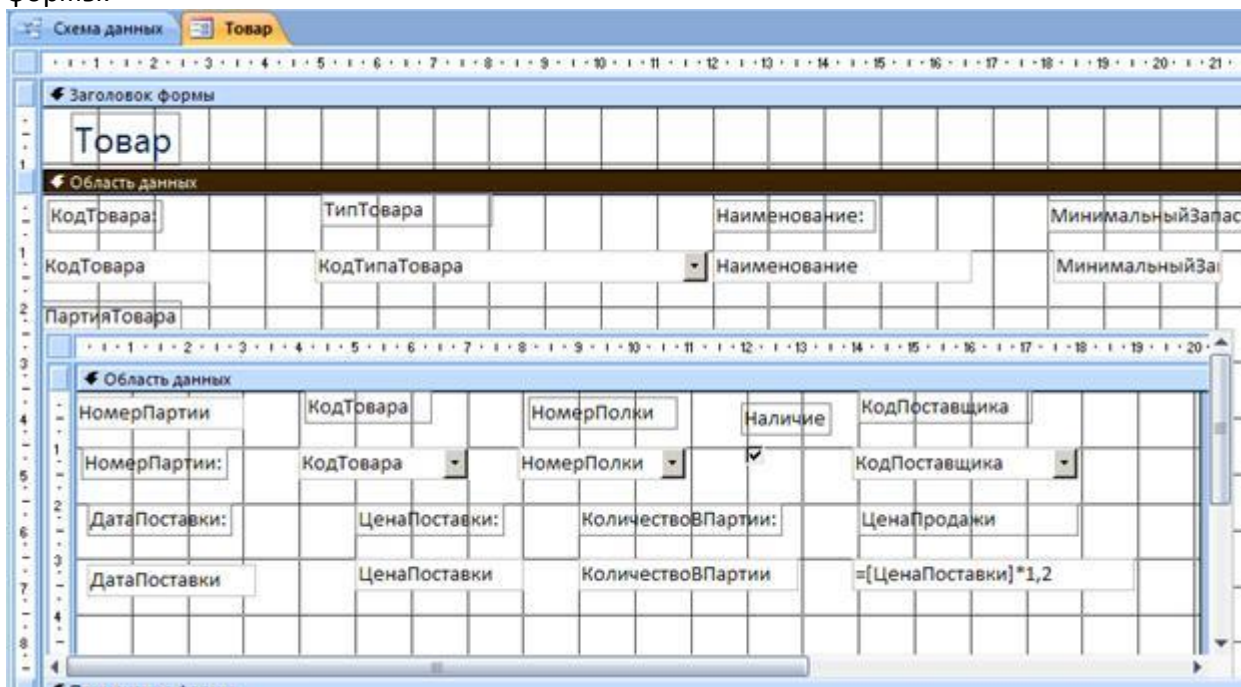


Рис.17. Вид формы ТОВАР в режиме КОНСТРУКТОРА

6. Заполните все таблицы вашей базы данных с помощью созданных форм.

7. Добавьте при необходимости кнопки для удаления записей, кнопки для открытия форм-справочников из многотабличных форм.

Запросы на выборку данных

Цель

Научиться создавать и использовать запросы к данным, хранящимся в связанных таблицах базы данных.

Графический инструментарий для разработки запросов. Ввод условий отбора записей.

Запросы являются одним из основных инструментов выборки и обработки данных в таблицах базы данных. Запросы используются для анализа, просмотра и изменения

данных. Запросы могут служить источниками записей для форм, отчетов и страниц доступа к данным.

Результат выполнения запроса – это новая, обычно **временная, таблица**, которая существует до закрытия запроса. Структура такой таблицы определяется полями, выбранными из одной или нескольких таблиц. Записи формируются путем **объединения записей таблиц**, участвующих в запросе.

Существует несколько видов запросов (рис.1). Основным является **запрос на выборку**. С его помощью выбираются данные из взаимосвязанных таблиц и других запросов. На основе запроса этого типа могут быть построены запросы других типов.

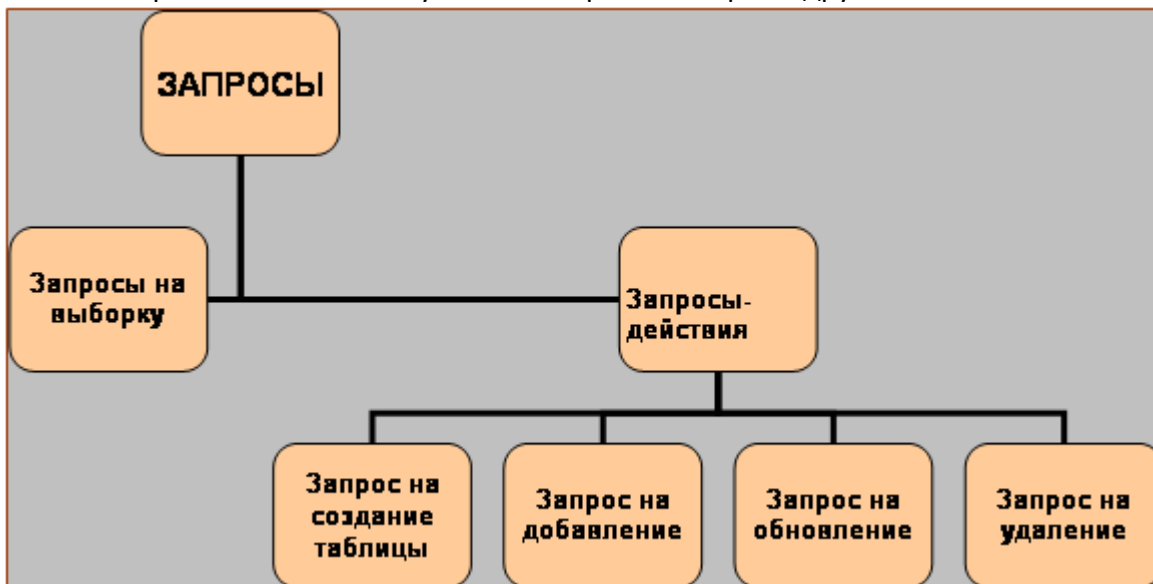


Рис.1. Классификация запросов в MS Access

Запрос строится на основе одной или нескольких взаимосвязанных таблиц. При этом могут быть использованы таблицы самой базы данных и таблицы, полученные в результате выполнения запросов и сохраненные для дальнейшего применения.

Кроме того, запросы могут формироваться на основе других запросов с использованием временных таблиц, получаемых сразу при выполнении этих запросов.

Задание 1. Графический инструмент для разработки запросов

Запросы достаточно сложного уровня можно создавать в режиме КОНСТРУКТОРА. СУБД Microsoft Access предоставляет удобное для пользователя диалоговое графическое средство формирования запросов по образцу – QBE (Query by Example). Инструментарий QBE располагает встроенными интеллектуальными компонентами, с помощью которых легко может быть построен достаточно сложный запрос.

Окно КОНСТРУКТОРА запросов QBE имеет следующую организацию (рис.2).

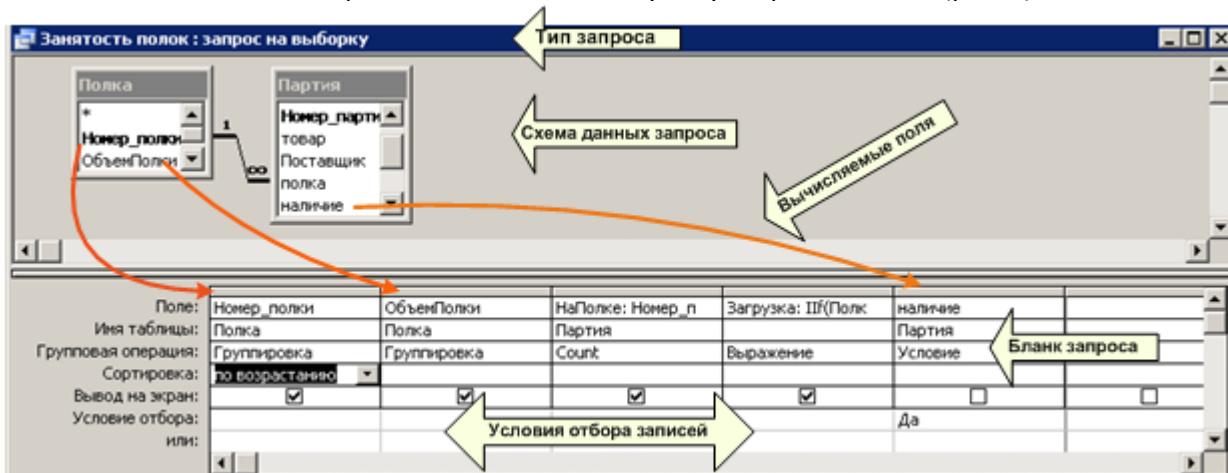



Рис.2. Окно КОНСТРУКТОРА запросов QBE

С отдельными компонентами КОНСТРУКТОРА запросов мы познакомимся, создавая запросы к базе данных.

Создадим запрос на выборку заказов, которые были сделаны после 01/01/2008 года.

1. Выберите в главном меню базы данных СОЗДАНИЕ – КОНСТРУКТОР ЗАПРОСОВ.
2. Для выполнения запроса нам достаточно данных из одной таблицы – ЗАКАЗ, поэтому в открывшемся списке таблиц щелкните дважды только по таблице ЗАКАЗ. Закройте окно ДОБАВЛЕНИЕ ТАБЛИЦ.

Если вам потребуется добавить таблицу уже после того, как вы закрыли список таблиц, нажмите на панели инструментов на кнопку  ОТОБРАЗИТЬ ТАБЛИЦУ.

3. Переносить поля в бланк запроса можно двумя способами: перетаскивать поля непосредственно из таблицы или выбирать поля из раскрывающегося списка в заголовке пустого поля (рис.3).

Если перетащить символ * из таблицы в поле запроса, то все поля таблицы будут активными полями для запроса.

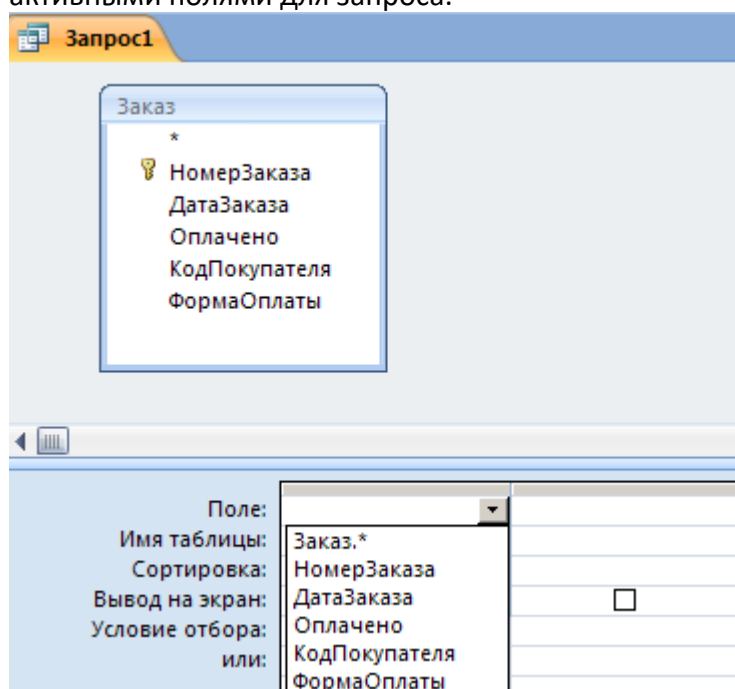


Рис.3. Выбор поля для запроса

4. Перетащите поля НОМЕР_ЗАКАЗА и ДАТА_ЗАКАЗА в пустые поля бланка запроса.
5. Введите условие отбора записей как показано на рис.4.

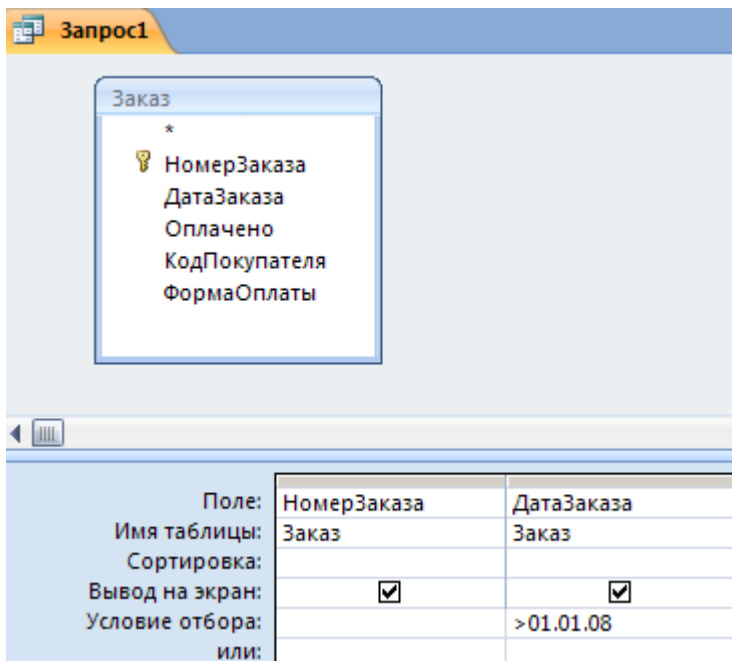




Рис.4. Ввод критериев отбора записей базы данных

6. Для просмотра временной таблицы с найденными записями, соответствующими критерию отбора, нажмите кнопку  ВЫПОЛНИТЬ на панели инструментов или перейдите в режим ВИД (кнопка  на панели инструментов). Результаты поиска представлены во временной таблице (рис.5).

НомерЗаказа	ДатаЗаказа
3	08.01.2009
1	04.02.2009
2	08.03.2009
* (№)	08.03.2009

Рис.5. Временная таблица с результатами поиска

7. Сохраните запрос командой СОХРАНИТЬ из контекстного меню бланка запроса с именем ДАТЫ_ЗАКАЗОВ[3].

8. Вернитесь в режим КОНСТРУКТОРА запросов и внесите следующие изменения: снимите флажок вывода на экран для поля ДАТА_ЗАКАЗА и установите сортировку по убыванию для поля НОМЕР_ЗАКАЗА (рис.6).

9. Запустите запрос на исполнение и сравните результаты вывода с предыдущим случаем (рис. 5).

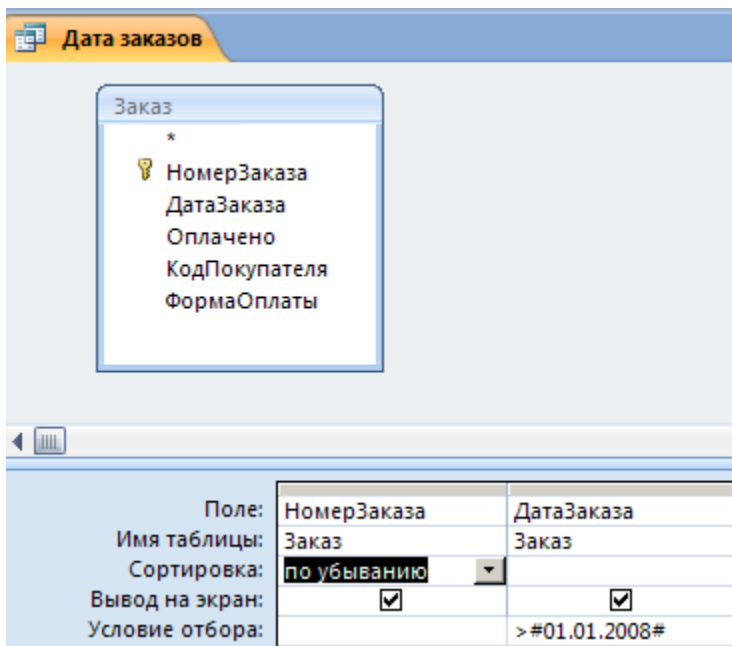


Рис.6. Режимы сортировки и вывода на экран

Обратите внимание, что автоматически изменился синтаксис записи даты в поле условия – появились знаки #.

Создайте запрос для поиска товаров, поставленных на склад в текущем году.

Отсортируйте список товаров по возрастанию наименований.

Задание 2. Ввод условий отбора записей

Условия, записанные в одной строке слева направо, объединены логической операцией

И. Условия, записанные в разных строках бланка запроса, объединены логической операцией **ИЛИ**.

1. Добавьте в бланк запроса Даты_Заказов еще одну таблицу – ПОКУПАТЕЛЬ.

2. Для значений своих данных сформируйте условие отбора на использование логики **ИЛИ**, например, как показано на рис.7.

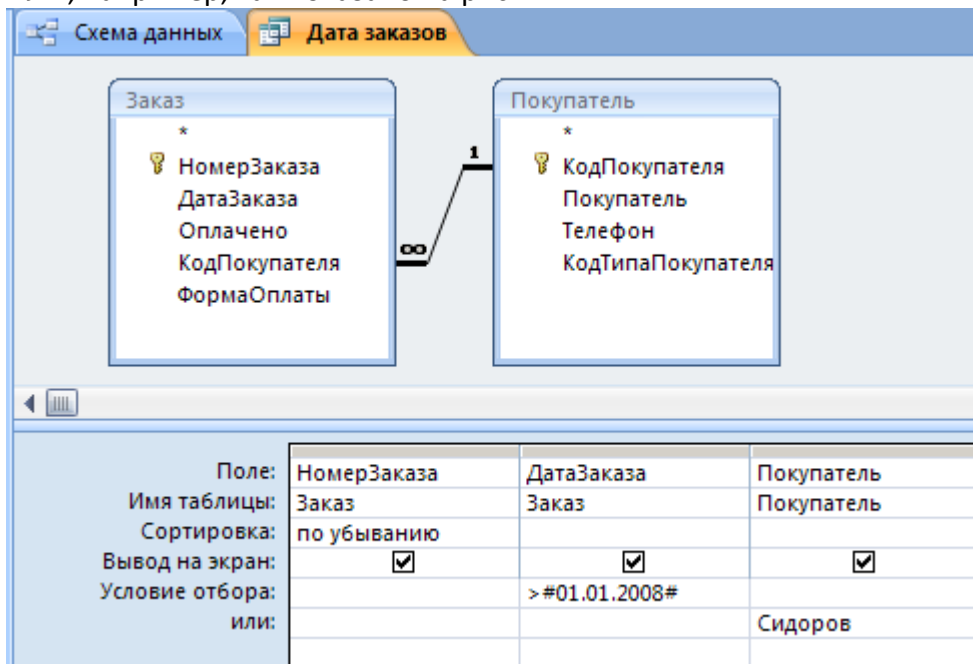


Рис.7. Условия отбора, объединенные логической операцией ИЛИ

3. Сформируйте условия отбора, объединенные логической операцией **И**.

4. Сформируйте условия отбора, объединенные логическими операциями **И** и **ИЛИ**.

Для полей, имеющих **логический тип данных**, условиями отбора являются значения «да/нет» (рис.8).

5. Найдите все оплаченные заказы.

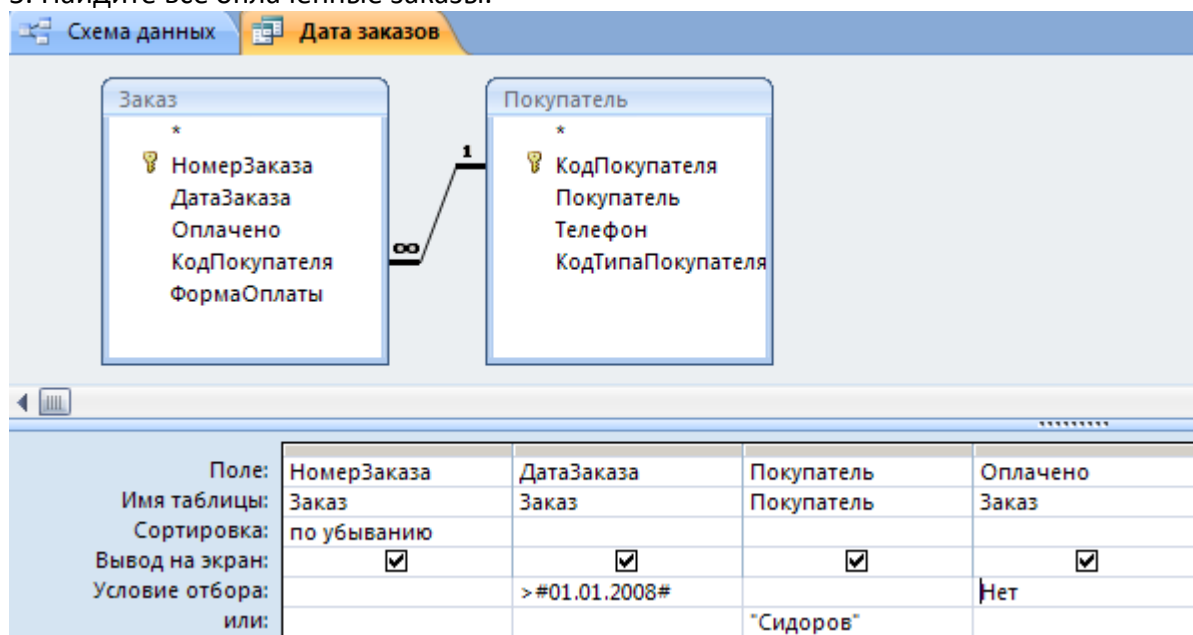


Рис.8. Отбор записей по логическому полю

Можно выбирать строки, принадлежащие указанному **диапазону значений** (рис.9). Эту же выборку можно создать, используя выражение Between #01.01.2007# And #20.01.2007#.

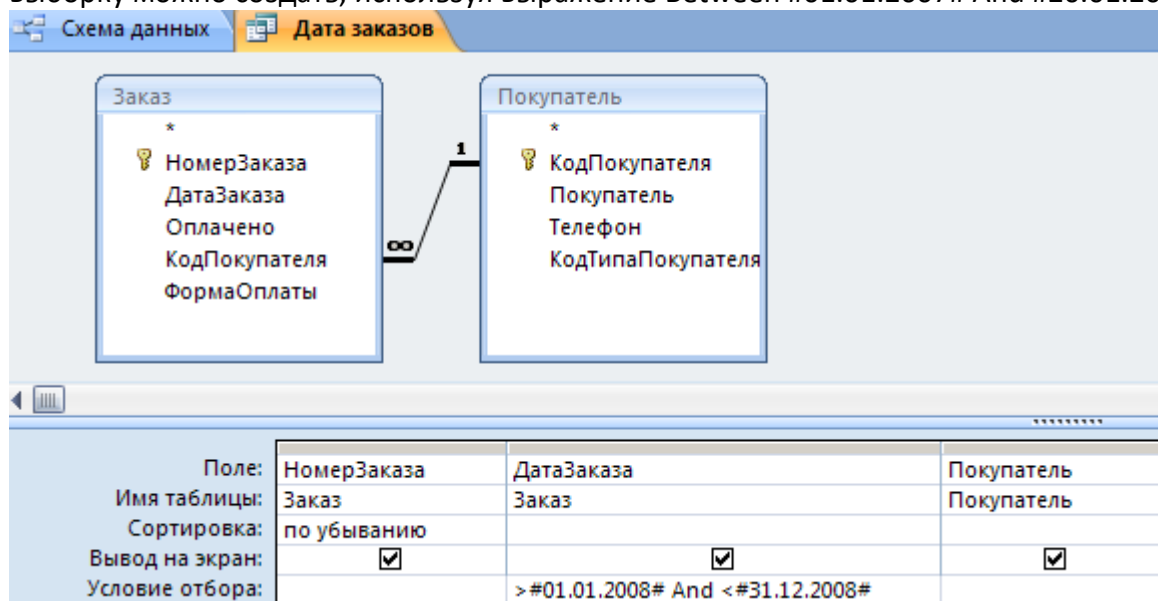


Рис.9. Выбор диапазона значений

6. Отберите записи, принадлежащие определенному вами диапазону. Диапазон может быть указан и для текстового поля, например, для поля ПОКУПАТЕЛЬ: >Г And <М. Такой критерий позволит отобразить все записи о покупателях, фамилия которых лежит в диапазоне букв Г и М.[4]

Чтобы выбрать записи с **пустыми значениями** в определенном поле[5], в бланке запроса введите в строке УСЛОВИЕ ОТБОРА значение **null**.


Перед запуском запроса Access изменит введенное значение на выражение Is Null.

7. Отберите записи о типах покупателей, для которых не сформулировано дополнительное описание (поле ПРИМЕЧАНИЕ).

8. Измените запрос на противоположный: найдите типы покупателей, для которых есть дополнительные сведения (рис.10).

MS Access позволяет для задания условий отбора использовать функции Visual Basic, например, как Date(). Более того, можно объединять в выражении функции, математические операторы, постоянные значения и ссылки на поля. На рис. 11 приведен пример использования функций для формирования условий отбора записей.

Вводить функции в поле условия отбора удобнее с помощью ПОСТРОИТЕЛЯ ВЫРАЖЕНИЙ (рис.11). Для вызова ПОСТРОИТЕЛЯ необходимо выполнить следующие действия:

- Установите курсор в поле для ввода условия отбора.
- Нажмите на панели инструментов кнопку  ПОСТРОИТЕЛЬ.
- Откройте списки: ФУНКЦИИ, ВСТРОЕННЫЕ ФУНКЦИИ, укажите требуемую КАТЕГОРИЮ, выберите нужную функцию в окна ПОСТРОИТЕЛЯ (рис.11).

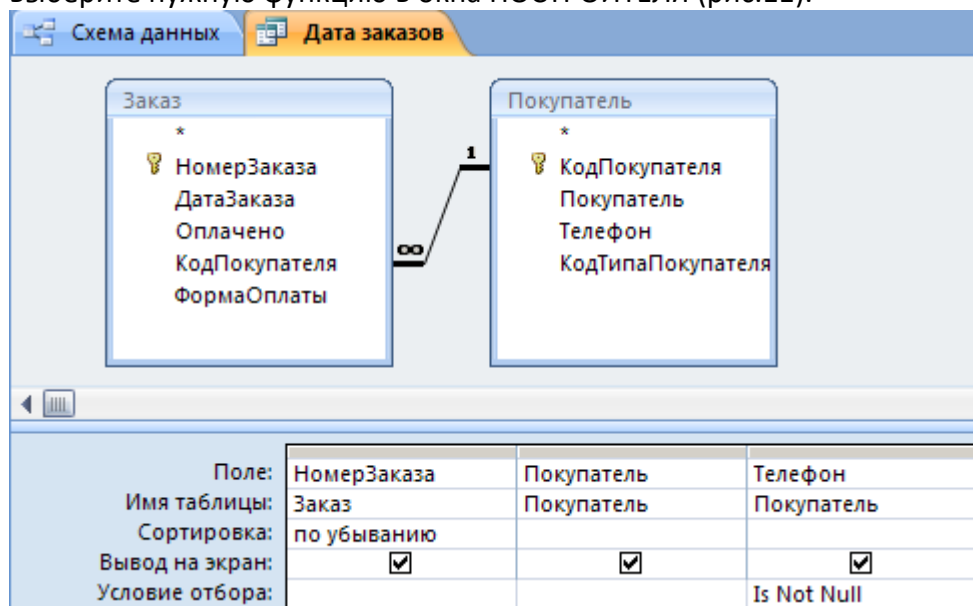


Рис.10. Выборка записей с непустыми значениями

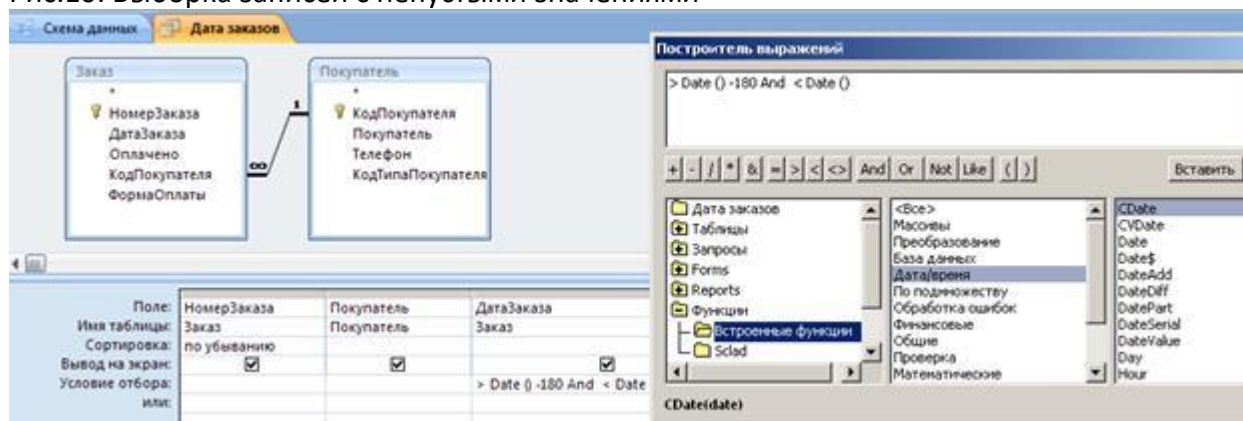


Рис.11. Использование ПОСТРОИТЕЛЯ ВЫРАЖЕНИЙ для ввода условий отбора

9. Выполните поиск заказов, сделанных в течение последних 3 месяцев.
10. Найдите покупателей, у которых указан номер городского телефона (условно, пусть это будет номер, имеющий определенное количество цифр).
Текстовая функция LEN(строка) возвращает число символов в указанной строке.
11. Найдите всех поставщиков, у которых указан телефон с кодом города Санкт-Петербурга (опять же, условно, пусть у телефона первые три цифры 812).
Текстовая функция LEFT(строка, число символов) возвращает указанное количество символов строки, взятых слева.
12. Найдите все заказа, сделанные в прошлом году. Функция YEAR(дата) возвращает год из указанной даты.

При формировании условий отбора можно использовать подстановочные знаки: * и ?. Например, вы можете использовать для поиска покупателей в качестве условия отбора выражение ??????, которое обеспечит выборку записей, в поле ПОКУПАТЕЛЬ которых присутствуют значения типа Иванов, Петров.

При использовании подстановочных символов следует помнить о том, что символ * соответствует любому одному (или нескольким) символам или цифре, а символ ? соответствует любому текстовому символу. Поэтому, обратите внимание на оператор LIKE, который необходимо использовать при поиске в числовом поле.

13. Создайте запросы с использованием подстановочных символов.

Задание 3

1. Найдите все заказы, в которых не указано количество заказанных партий определенного товара. Проведите анализ возможности подобной ситуации в нашей базе данных.

2. Найдите всех поставщиков, товары которых были заказаны в прошлом месяце.

3. Просмотрите список товаров на полке №1.

4. Выведите список товаров, поставки которых выполняются партиями по 100 единиц.

5. Определите перечень товаров, стоимость которых более 1000 рублей.

6. Определите типы товаров, стоимость которых менее 1000 рублей.

7. Определите поставщиков товаров, стоимость которых находится в диапазоне от 1000 до 2000 рублей.

8. Найдите товары, заказ которых оплачен по безналичному расчету.

9. Определите список покупателей, оплачивающих заказа наличными деньгами.

10. Определите заказы прошлого месяца, оплаченные по безналичному расчету.

11. Найдите заказы прошлого года, оплаченные наличными деньгами.

12. Выполните анализ, какие типы покупателей покупают товар определенной категории.

13. Сформируйте список товаров, минимальный запас для которых меньше 100 единиц.

14. Найдите типы товаров, минимальный запас для которых больше 100 единиц.

15. Определите список поставщиков товаров, для которых минимальный запас от 100 до 200 единиц.

16. Найдите полки, количество мест которых более 3.

17. Выведите список товаров, цена реализации которых менее 500 рублей.

18. Определите категории, цена реализации которых от 2000 до 3000 рублей.

19. Найдите поставщиков, у которых мобильные телефоны определенного провайдера (условно, любые первые цифры кода, например, 921).

20. Перечислите товары, партии которых имеются в наличии на складе.

Вычисления в запросах на выборку данных

Цель

Научиться создавать и использовать запросы к данным, хранящимся в связанных таблицах базы данных.

Создание запросов с вычисляемыми полями. Групповые операции в запросах.


Задание 1. Создание запросов с вычисляемыми полями

При составлении запроса на выборку можно создавать в бланке запроса собственное поле для вычисления тех или иных значений. Например, можно создать поле СТОИМОСТЬ, в котором вычислять выражение КОЛИЧЕСТВО * ЦЕНА_РЕАЛИЗАЦИИ для всех строк таблицы ПУНКТ_ЗАКАЗА.

Все, что необходимо сделать для создания вычисляемого поля – ввести в свободный столбец бланка запроса имя нового поля и после двоеточия – выражение для расчета.

Вычисляемое выражение удобно вводить с помощью ПОСТРОИТЕЛЯ.

Для таблицы ПУНКТ_ЗАКАЗА создадим запрос, в котором будем вычислять итоговую стоимость каждой позиции заказа.

1. Создайте запрос для таблицы ПУНКТ_ЗАКАЗА в режиме КОНСТРУКТОРА (рис.1).
2. Перетащите элемент * из таблицы ПУНКТ_ЗАКАЗА в пустое поле бланка запроса (рис.1).
3. Щелкните мышью в следующем пустом поле и нажмите кнопку  ПОСТРОИТЕЛЬ на панели инструментов (рис.1).

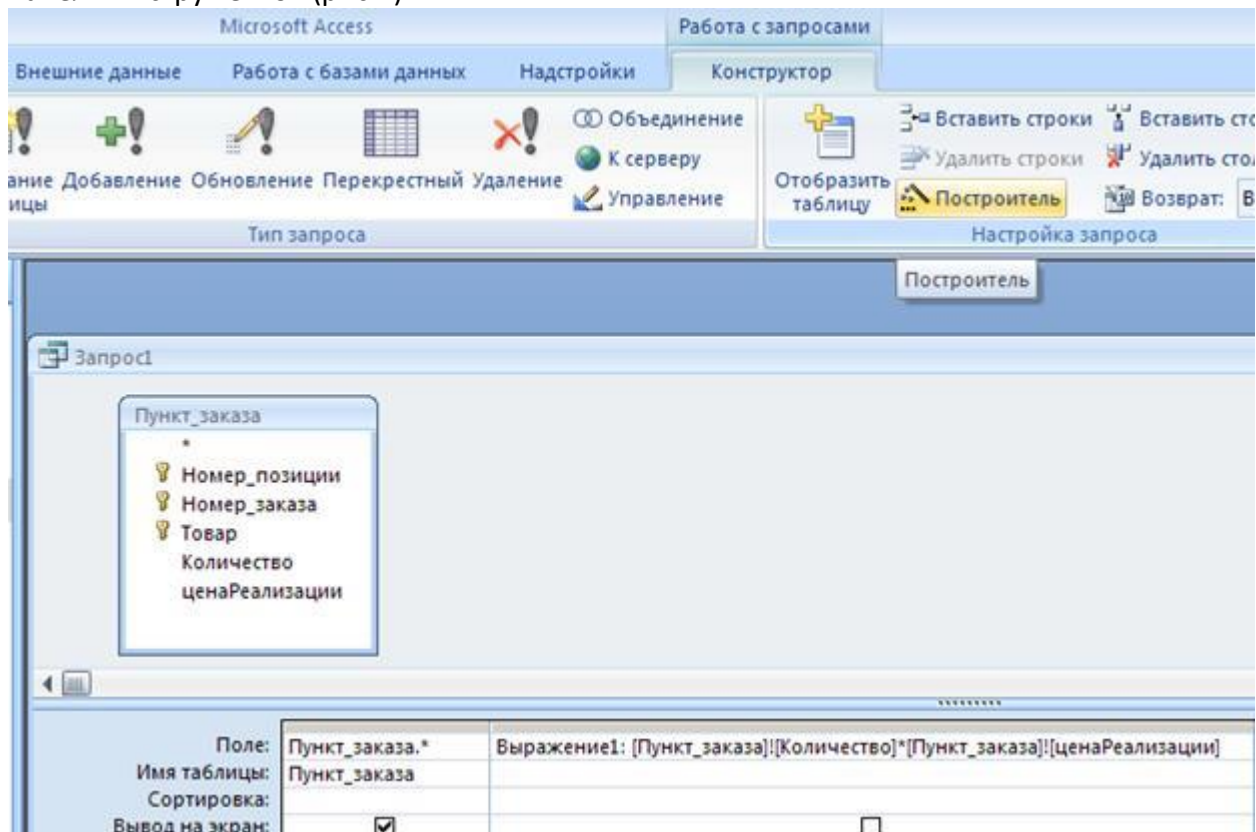


Рис.1. Создание вычисляемого поля в запросе

При создании вычисляемого выражения используйте списки полей, это удобнее и надежнее, чем непосредственно вводить названия полей (рис.2).

Можно сразу в ПОСТРОИТЕЛЕ выражений изменить название ВЫРАЖЕНИЕ1 вычисляемого поля, например, на ИТОГО (рис.2).

Осторожно, не удалите разделительный символ двоеточие (:), это служебный символ, он отделяет название поля от его значения.

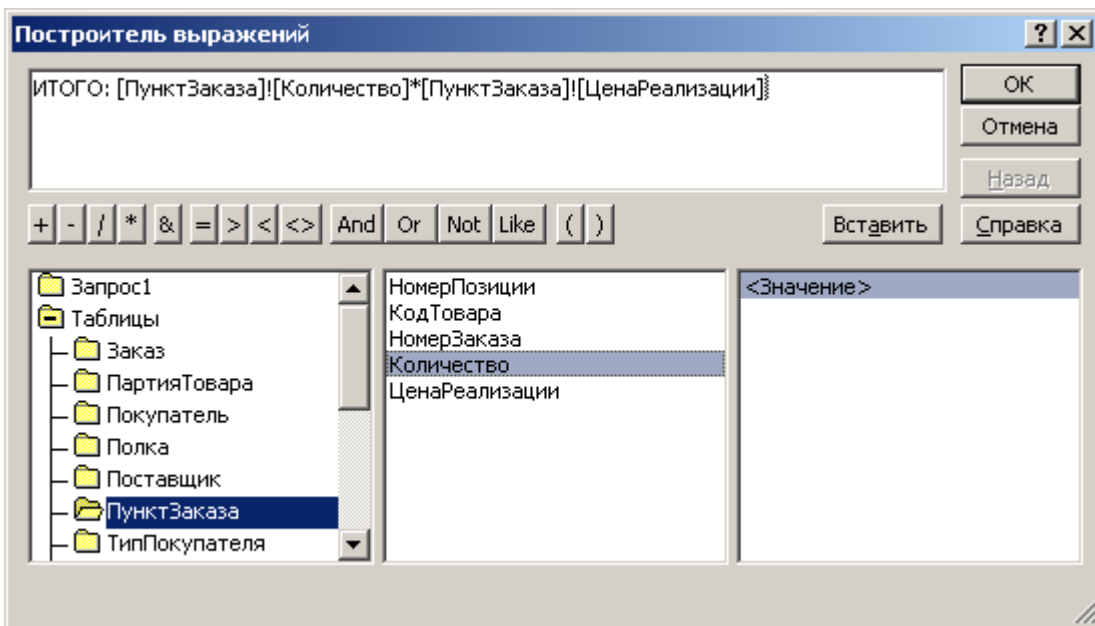


Рис.2. Окно ПОСТРОИТЕЛЯ выражений

4. Запустите запрос на исполнение и проверьте правильность вычислений в нем.

Задание 2. Групповые операции в запросах

При использовании запросов часто требуется видеть не каждую строку таблицы, а только итоговые значения по группам данных.

Групповые операции (**группировка**) позволяют выделить группы записей с одинаковыми значениями в указанных полях и использовать для некоторых полей этих групп одну из статистических функций.

В MS Access предусмотрено девять статистических функций:

- SUM – сумма значений некоторого поля для группы;
- AVG – среднее от всех значений поля в группе;
- MAX, MIN – максимальное, минимальное значение поля в группе;
- COUNT – число значений поля в группе без учета пустых значений;
- StDev – среднеквадратичное отклонение от среднего значения поля в группе;
- VAR – дисперсия значений поля в группе;
- First, Last – значение поля из первой или последней записи в группе.

Результат запроса с использованием групповых операций содержит по одной записи для каждой группы.

В запрос, как правило, включаются поля, по которым производится группировка, и поля, для которых выполняются групповые функции.


Рассмотрим технологию создания запроса с группировкой записей и применения статистической функции к сгруппированным записям.

Создадим запрос для вычисления количества (COUNT) различных товаров в каждой группе товаров (типе товаров).

Из формулировки запроса видно, что товары следует группировать по названию типа товара, а для каждой полученной группы следует применить статистическую функцию COUNT.

1. В режиме КОНСТРУКТОРА создайте запрос для таблиц ТОВАР и ТИП_ТОВАРА. Таблица ТИП_ТОВАРА нам потребуется для расшифровки значений поля КОД_ТИПА (рис.3).

2. Перенесите в бланк запроса поля ТИП_ТОВАРА из таблицы ТИП_ТОВАРА и НАИМЕНОВАНИЕ из таблицы ТОВАР (рис.3).

3. Нажмите кнопку на панели инструментов  ИТОГИ.

4. Для поля НАИМЕНОВАНИЕ в строке ГРУППОВАЯ ОПЕРАЦИЯ щелкните мышью в правом углу поля (должен появиться элемент управления список) и выберите из раскрывающегося списка операцию COUNT (рис.3).

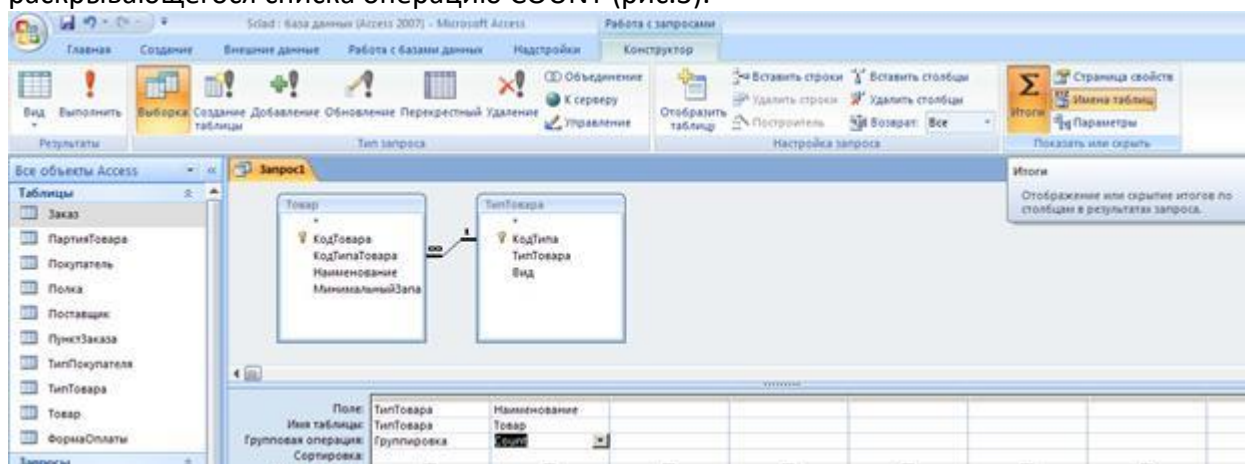


Рис.3. Групповые операции в запросах

Запустите запрос на исполнение и убедитесь в появлении групповых итогов (рис.4).

тип_товара	Count-НаименованиеТовара
Бумага	2
Ручки	2
Фломастеры	1

Запись: 1 из 3

Рис.4. Итоги групповой операции

Можно изменить название поля Count_НаименованиеТовара на любое другое, например, КОЛИЧЕСТВО_НАИМЕНОВАНИЙ.

5. Вернитесь в режим КОНСТРУКТОРА, щелкните мышью в названии поля НАИМЕНОВАНИЕ_ТОВАРА и введите КОЛИЧЕСТВО_НАИМЕНОВАНИЙ до названия поля, поставьте знак двоеточие (:) для разделения позиций (мы обсуждали с вами этот вопрос).

6. Запустите запрос на исполнение и убедитесь в появлении нового названия поля.

Можно применять несколько статистических функций одновременно.

7. Добавьте в бланк запроса еще одно поле – МИНИМАЛЬНЫЙ_ЗАПАС.

8. Примените (выберите из списка) для этого поля функцию AVG(), вычисляющую среднее от всех значений поля в группе.

9. Измените название поля и просмотрите результаты выполнения запроса (рис.5).

Поле:	ТипТовара	Количество наименс	Минимальный запас: МинимальныйЗапас
Имя таблицы:	ТипТовара	Товар	Товар
Групповая операция:	Группировка	Count	Min
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рис.5. Запрос с несколькими статистическими функциями

В запросах с групповыми операциями можно задавать условия отбора, также как, и в обычных запросах на выборку.

10. Измените для поля МИНИМАЛЬНЫЙ_ЗАПАС функцию AVG() на значение УСЛОВИЕ в том же раскрывающемся списке (рис.6).

11. Введите значение условия. Запустите запрос на исполнение.

Обратите внимание на отсутствие флажка ВЫВОД НА ЭКРАН, это особенность задания условий в групповых запросах.

ТипТовара	Количество наименов	Минимальный запас: МинимальныйЗапас
ТипТовара	Товар	Товар
Группировка	Count	Условие
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		> 50

Рис.6. Использование условий отбора в запросах с групповыми операциями
 В запросах с группировкой можно создавать вычисляемые поля, так же, как и в обычных запросах на выборку.

Создадим запрос, вычисляющий стоимость **всего заказа** и внедрим его в форму ЗАКАЗ.

12. В режиме КОНСТРУКТОР создайте запрос для таблицы ПУНКТ_ЗАКАЗА.

13. Переместите поле НОМЕР_ЗАКАЗА в бланк запроса и включите для этого поля операцию ГРУППИРОВКА.

14. В пустом поле щелкните указателем мыши и вызовите ПОСТРОИТЕЛЬ.

15. Используйте встроенные функции для ввода вычисляемого выражения (рис.7).

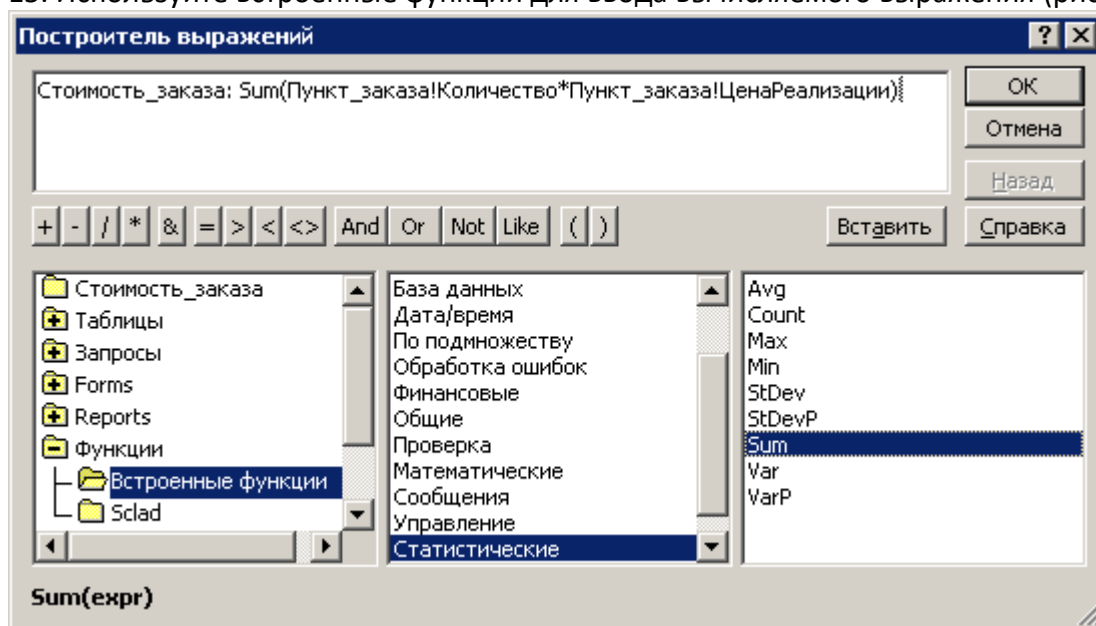


Рис.7. Создание вычисляемого поля

16. Переименуйте Выражение1, например, в СТОИМОСТЬ_ЗАКАЗА.

17. Нажмите ОК в ПОСТРОИТЕЛЕ выражений.

18. Выберите в поле ГРУППОВАЯ ОПЕРАЦИЯ для нового поля строку ВЫРАЖЕНИЕ из раскрывающегося списка (рис.8).

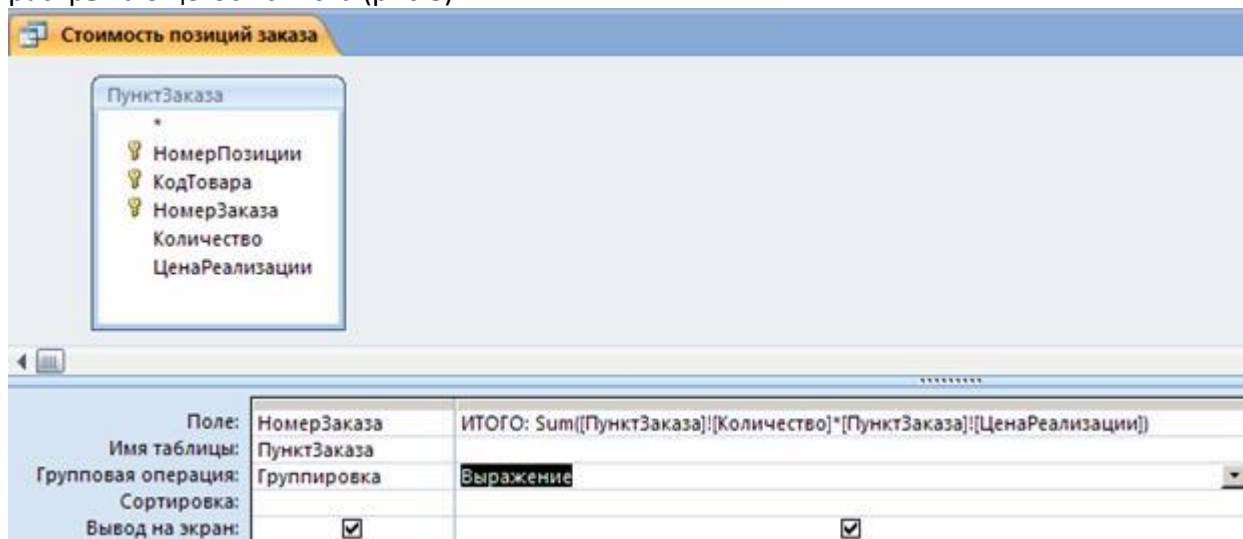


Рис.8. Создание вычисляемого поля в запросе с группировкой

19. Запустите запрос на исполнение и убедитесь в выполнении группировки и вычислений (рис.9).

Номер_заказа	Стоимость_заказа
11	638,00р.
12	75,00р.

Запись: 2 из 2

Рис.9. Результат выполнения запроса

20. Сохраните запрос, назовите его, например, Стоимость_заказа. Закройте созданный запрос.

21. Раскройте форму ЗАКАЗ в режиме КОНСТРУКТОРА, определите место расположения для внедренного запроса.

22. Возьмите указателем мыши запрос и перетащите его в подготовленное место в форме ЗАКАЗ.

23. В открывшемся диалоговом окне МАСТЕРА подчиненных форм (рис. 10) укажите режим САМОСТОЯТЕЛЬНОЕ ОПРЕДЕЛЕНИЕ для установления связи между формой и внедренным в нее запросом.

24. В раскрывающихся списках выберите поля для связи (рис.10).

25. Нажмите кнопку ДАЛЕЕ, дайте подчиненной форме (запрос становится подчиненной формой) осмысленное название и нажмите ГОТОВО.

Мастер подчиненных форм

Самостоятельно определите поля связи между главной и подчиненной формами или выберите их из списка.

Выбор из списка Самостоятельное определение

'Стоимость_заказа' для каждой записи 'Заказ', поле связи 'Номер_заказа' отсутствует

Отмена < Назад Далее > Готово

Рис.10. Первый шаг МАСТЕРА подчиненных форм

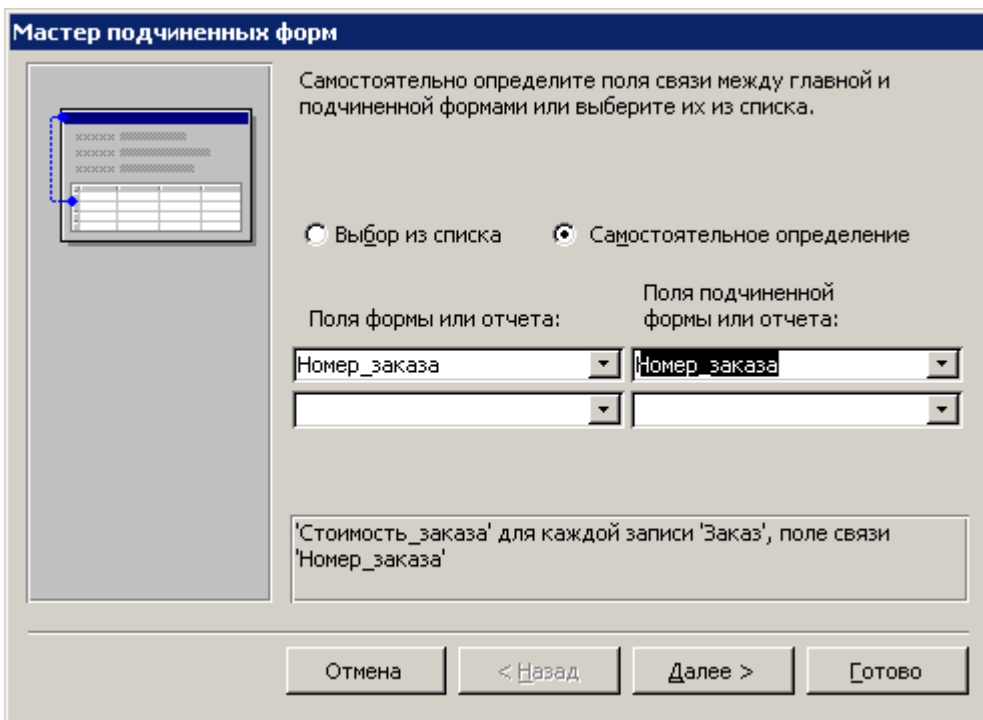


Рис.11. Второй шаг МАСТЕРА подчиненных форм

26. Откройте форму ЗАКАЗ в режиме ВИД (рис.12).

27. С помощью СВОЙСТВ измените внешний вид внедренной формы (запроса). Можно, например, выключить изображения линии прокрутки, элемента выбора и так далее (рис. 12).

28. Поле НОМЕР_ЗАКАЗА можно свернуть.

29. Проверьте работоспособность измененной формы: введите несколько новых позиций в имеющиеся заказы и создайте несколько новых заказов.

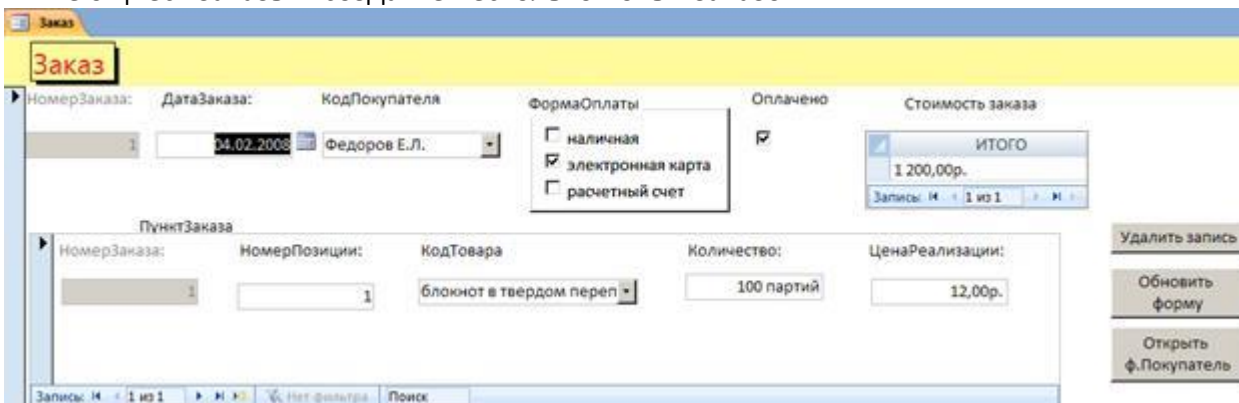


Рис.12. Окончательный вид формы ЗАКАЗ с внедренным запросом

Задание 3

Создать запрос, определяющий занятость полок и внедрить его в виде кнопки в форму ТОВАР.

В запросе необходимо создать вычисляемое поле, в котором сравнивается объем полки и количество партий, размещенных на этой полке. Если на полке есть свободные места, то можно выдать сообщение о количестве свободных мест.

Категория логических функций, с которыми мы познакомились в Excel, в Access называется категорией УПРАВЛЕНИЯ. В этой категории имеется три функции (рис. 13): Switch(), IIF() и Choose(). Для выполнения запроса необходимо рассмотреть синтаксис функции IIF().

IIF(логическое условие; операторы если условие true; операторы если условие false)

На рис.13 приведено окно ПОСТРОИТЕЛЯ выражений, в котором вычисляется занятость полок с использованием функции IIF().

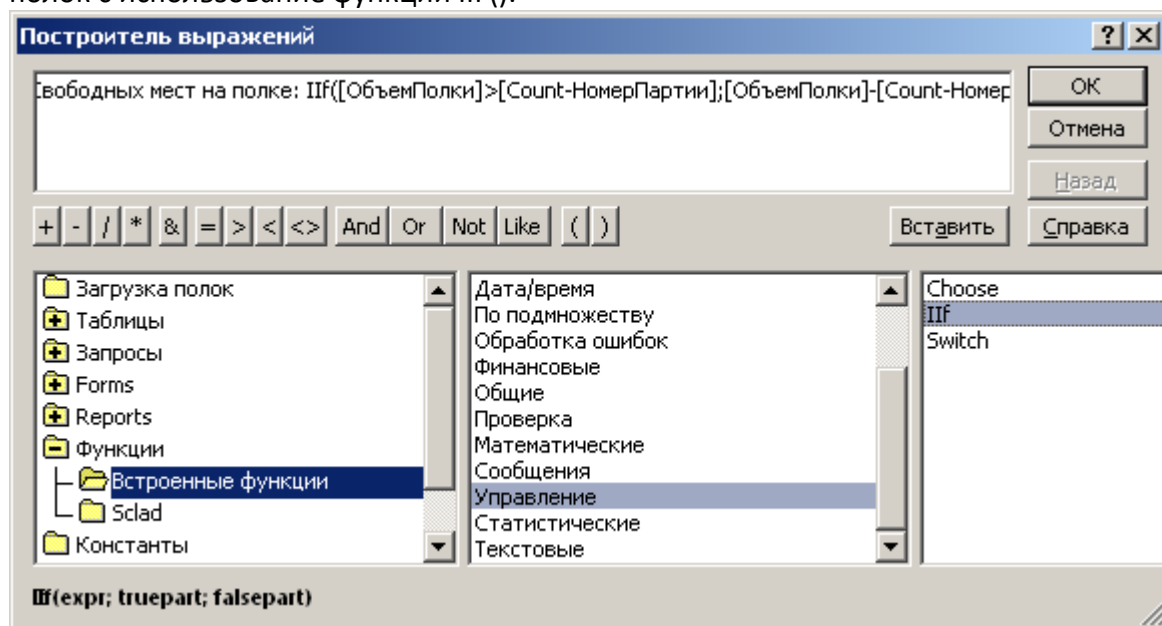


Рис.13. Построение вычисляемого поля для запроса о занятости полок

На рис.14 показан результат выполнения запроса. В случае если полка занята, в поле Свободных мест на полке появится сообщение «0 свободных мест». В случае неполной загрузки полки (на рис.14) вычисляется количество свободных мест.

НомерПолки	ОбъемПолки	Count-НомерПартии	Свободных мест на полке
1	10 партий	2	8 свободных мест
2	5 партий	1	4 свободных мест

Рис.14. Выполнение запроса на определение занятости полок

Запрос готов к работе. Сохраните запрос, например, как СВОБОДНЫЕ МЕСТА НА ПОЛКАХ и внедрите в структуру формы ТОВАР.

1. Откройте форму ТОВАР в режиме КОНСТРУКТОРА.
2. Нажмите на панели инструментов элемент КНОПКА и щелкните в области формы. Автоматически откроется диалоговое окно МАСТЕРА создания кнопок.
3. На первом шаге выберите категорию РАЗНОЕ и действие – ВЫПОЛНИТЬ ЗАПРОС.
4. На втором шаге МАСТЕРА укажите название запроса – СВОБОДНЫЕ МЕСТА НА ПОЛКАХ.
5. Следующий шаг МАСТЕРА требует указания или названия кнопки, или картинки на кнопке.
6. Проверьте работоспособность формы с кнопкой. Имейте в виду, что полки, которых нет в списке занятых полок (в списке, который выдает запрос) – свободны. Перечень всех полок вы видите в раскрывающемся списке поля НОМЕР_ПОЛКИ.

Задание 4

Создайте запросы, отвечающие на следующие вопросы к базе данных.

1. Сколько заказов было сделано в прошлом году?
2. Сколько заказов было оплачено по безналичному расчету в прошлом месяце?
3. На какую сумму в прошлом году были оплачены заказы наличными деньгами?
4. Сколько товаров каждого типа имеется в наличии на складе?
5. Выведите список покупателей, которые заказывали более 3-х партий одного наименования в одном заказе.
6. На какую сумму не оплачены заказы на сегодняшний день?
7. На какую сумму каждый поставщик поставил товаров в прошлом году?

8. Какую прибыль склад получил в прошлом месяце?

В случае отсутствия необходимых данных, добавьте эти данные с помощью форм в таблицы базы данных.

Все запросы должны быть не только сконструированы, но и выдавать конкретные результаты при выполнении.

Запросы-действия

Цель

Научиться создавать и использовать запросы к данным, хранящимся в связанных таблицах базы данных.

Создание новой таблицы с помощью запроса. Обновление группы записей. Обновление поля таблицы вычисляемым значением. Добавление данных из другой таблицы.

Задание 1. Создание новой таблицы с помощью запроса

Перейдем к рассмотрению **запросов-действий**, которые позволяют изменять, создавать, удалять или добавлять наборы данных в таблицы базы данных.

Запрос на создание таблицы используется для сохранения результатов выполненного запроса в виде таблицы базы данных.

Этот тип запроса основан на **запросе на выборку**, но в отличие от него сохраняет таблицу, содержащую результаты поиска.

Необходимость в сохранении результатов выполнения запроса возникает, например, в том случае, когда один и тот же запрос выполняется неоднократно, а исходные данные для него не меняются. Используя вместо запроса таблицу, можно значительно ускорить доступ к данным, особенно если запрос построен на нескольких таблицах большого объема.

Сохранение таблицы запроса полезно и в том случае, когда необходимо накапливать итоговые данные на протяжении длительного времени, а данные, на которых базируется запрос можно удалить.

Создадим запрос на создание таблицы, хранящей сведения о проданных партиях товаров; сохраним полученную выборку в виде архивной таблицы. В последующем, удалим эти записи из основной таблицы[6].

1. Сконструируйте запрос на выборку для поиска проданных партий товаров (рис.1).
2. Убедитесь в его работоспособности и сохраните.
3. В режиме КОНСТРУКТОРА преобразуйте этот запрос из запроса на выборку в запрос на создание таблицы. Нажмите на панели конструктора кнопку СОЗДАНИЕ ТАБЛИЦЫ.
4. В диалоговом окне укажите имя будущей таблицы (рис.2).
5. Чтобы просмотреть, какие записи будут помещены в новую таблицу, запустите запрос на выполнение еще раз.
6. Теперь осталось убедиться, что именно эти записи помещены в новую таблицу. Закройте запрос. В окне базы данных в разделе ТАБЛИЦЫ найдите и просмотрите новую таблицу (рис.3).

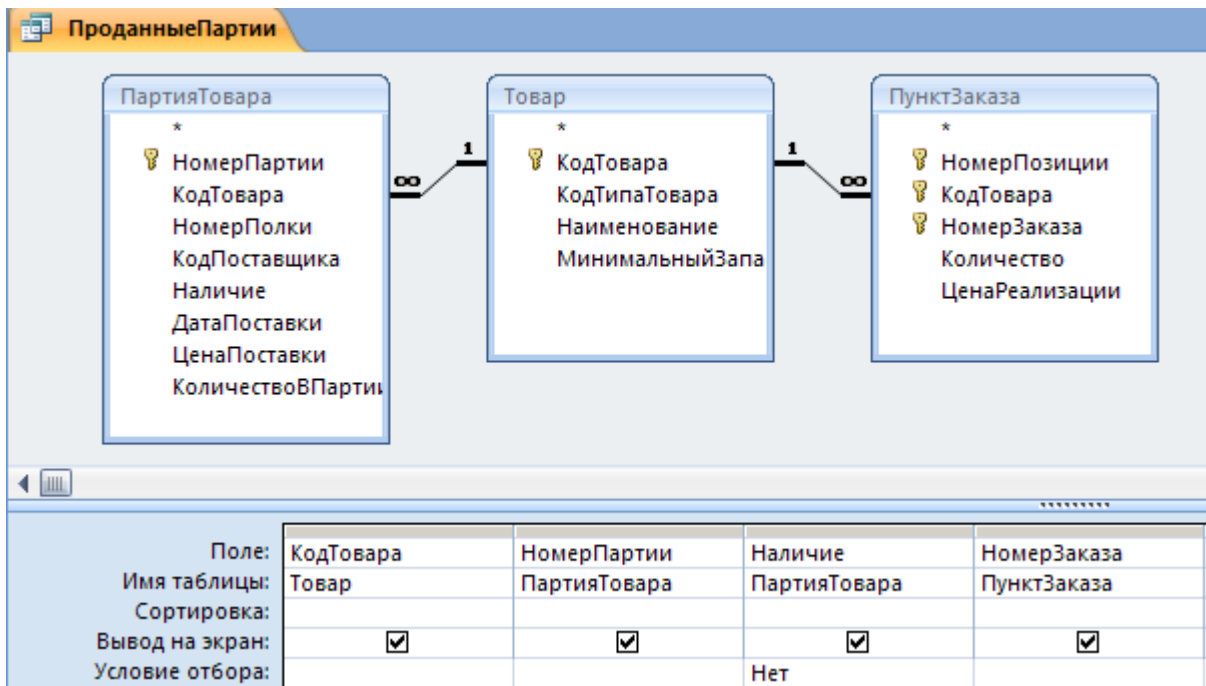


Рис.1. Запрос для поиска проданных партий товаров

Рис.2. Именованная таблица для сохранения найденных данных

КодТовара	НомерПарт	Наличие	НомерЗака
1	1	0	1
1	2	0	1
*	(No)		

Рис.3. Таблица, созданная в результате выполнения запроса

7. Убедитесь, что записи, помещенные в новую таблицу, не удалены из исходных таблиц.

Задание 2. Обновление группы записей

Запрос на обновление используется для обновления данных в полях таблицы базы данных.

Изменения вносятся в поля целой группы записей, отбираемых с помощью указанных пользователем условий отбора, а не отдельной записи. В результате одним запросом можно выполнить необходимые изменения во многих записях.

Создадим запрос, который будет изменять значение поля НАЛИЧИЕ для всех записей в таблице ПАРТИЯ_ТОВАРА, в случае заказа этих партий товаров.

1. В режиме КОНСТРУКТОРА создадим обычный запрос на выборку для поиска всех заказанных партий товаров, но для которых еще не отмечен факт их продажи (рис.4).

Поле:	КодТовара	НомерПартии	Наличие	НомерЗаказа
Имя таблицы:	Товар	ПартияТовара	ПартияТовара	ПунктЗаказа
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			Да	

Рис.4. Запрос на поиск всех проданных партий товаров, для которых еще не отмечен факт их продажи

- Преобразуйте этот запрос на ОБНОВЛЕНИЕ с помощью соответствующей кнопки на панели инструментов.
- Добавьте в условии ОБНОВЛЕНИЕ для поля НАЛИЧИЕ данные для изменения (рис.5).
- Запустите запрос на исполнение и убедитесь в выполненных им обновлениях.
- Обратите внимание на значок, который появляется у запроса в окне базы данных.

Поле:	КодТовара	НомерПартии	Наличие	НомерЗаказа
Имя таблицы:	Товар	ПартияТовара	ПартияТовара	ПунктЗаказа
Обновление:			нет	
Условие отбора:			Да	

Рис.5. Изменение типа запроса на ОБНОВЛЕНИЕ

Задание 3. Обновление поля таблицы вычисляемым значением

В процессе проектирования базы данных мы с вами уже решали проблему вычисления ЦЕНЫ_РЕАЛИЗАЦИИ (или ЦЕНЫ_ПРОДАЖИ), связывая эту величину с ЦЕНОЙ_ПОСТАВКИ.

Для этого мы в форме ПАРТИЯ_ТОВАРА использовали вычисляемое поле. Попробуем решить эту же проблему другим способом.

1. В режиме КОНСТРУКТОРА создайте все тот же запрос для поиска партий товаров, вписанных в бланк заказа (рис.6).
2. Измените тип запроса на ОБНОВЛЕНИЕ.

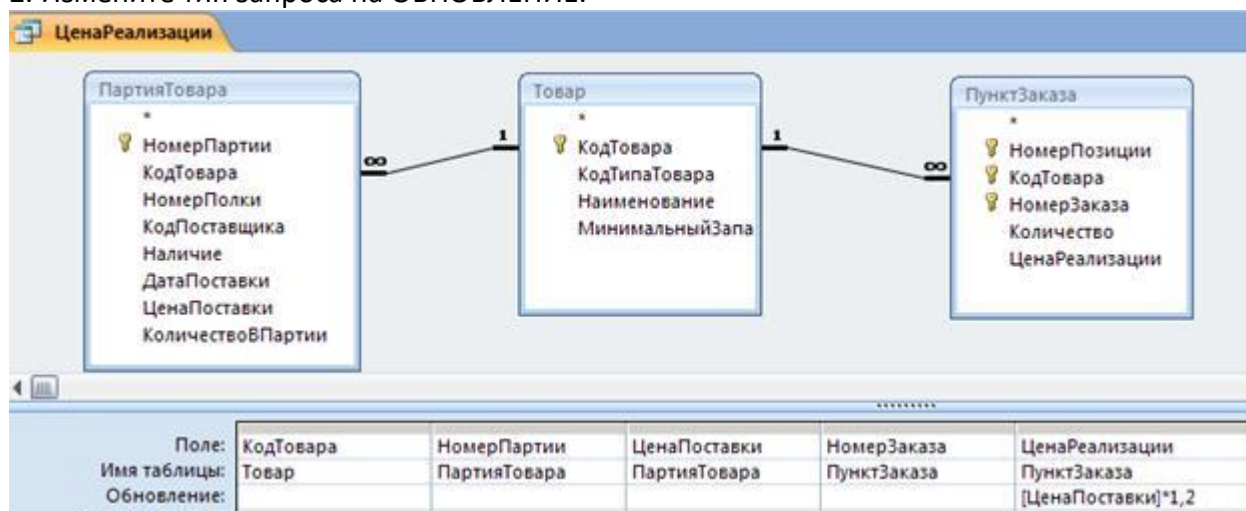


Рис.6. Запрос на обновление вычисляемым значением

3. Для поля ЦЕНА_РЕАЛИЗАЦИИ в строке ОБНОВЛЕНИЕ введите вычисляемое выражение (рис.6).
4. Запустите запрос на исполнение и убедитесь, что происходит изменение поля ЦЕНА_РЕАЛИЗАЦИИ в таблице ПУНКТ_ЗАКАЗА.
5. Сохраните созданный запрос с именем ЦЕНА_РЕАЛИЗАЦИИ.

Позже мы внесем все созданные запросы в соответствующие формы в виде командных кнопок.

Задание 4. Добавление данных из другой таблицы

С помощью **запроса на добавление** производится добавление записей из таблицы с результатами запроса в таблицу базы данных. Поэтому в запросе должны быть сформированы записи с полями, соответствующими полям в дополняемой записями таблице базы данных.

Структура записи таблицы запроса может не полностью совпадать со структурой записи таблицы, в которую добавляются записи. В записи запроса может быть меньше полей, если поля в таблице, куда добавляются записи, не наложено требование по обязательности их заполнения. Допускается несоответствие типов полей, если возможно преобразование типа данных одного поля в тип данных другого поля. Кроме того, в бланк запроса могут быть включены поля, по которым задаются условия отбора.

Мы с вами создали таблицу ПРОДАННЫЕ_ПАРТИИ_ТОВАРОВ для хранения архивных данных о проданных партиях. В эту таблицу необходимо добавлять записи по мере продажи других партий товара.

1. Создадим еще раз запрос на выборку для поиска проданных партий товаров (рис.7).
2. В окне КОНСТРУКТОРА преобразуйте его в запрос на ДОБАВЛЕНИЕ. В диалоговом окне МАСТЕРА укажите имя таблицы, в которую должны добавляться записи (ПРОДАННЫЕ_ПАРТИИ_ТОВАРОВ) (рис.8).
3. Для формирования добавляемых записей обязательно включите в бланк запроса поля, составляющие ключ таблицы, в которую будет производиться добавление.
4. Поскольку в нашем случае имена полей в таблице и запросе совпадают, можно запускать запрос на исполнение. Сохраните запрос с именем АРХИВ_ПРОДАННОГО.

- Обратите внимание на повторное занесение записей в таблицу ПРОДАННЫЕ_ПАРТИИ_ТОВАРОВ (рис.9). Конечно, так не должно быть, но мы еще не умеем удалять записи из таблицы ПАРТИЯ_ТОВАРА.
- Обратите внимание также на значок созданного запроса АРХИВ_ПРОДАННОГО в окне базы данных.

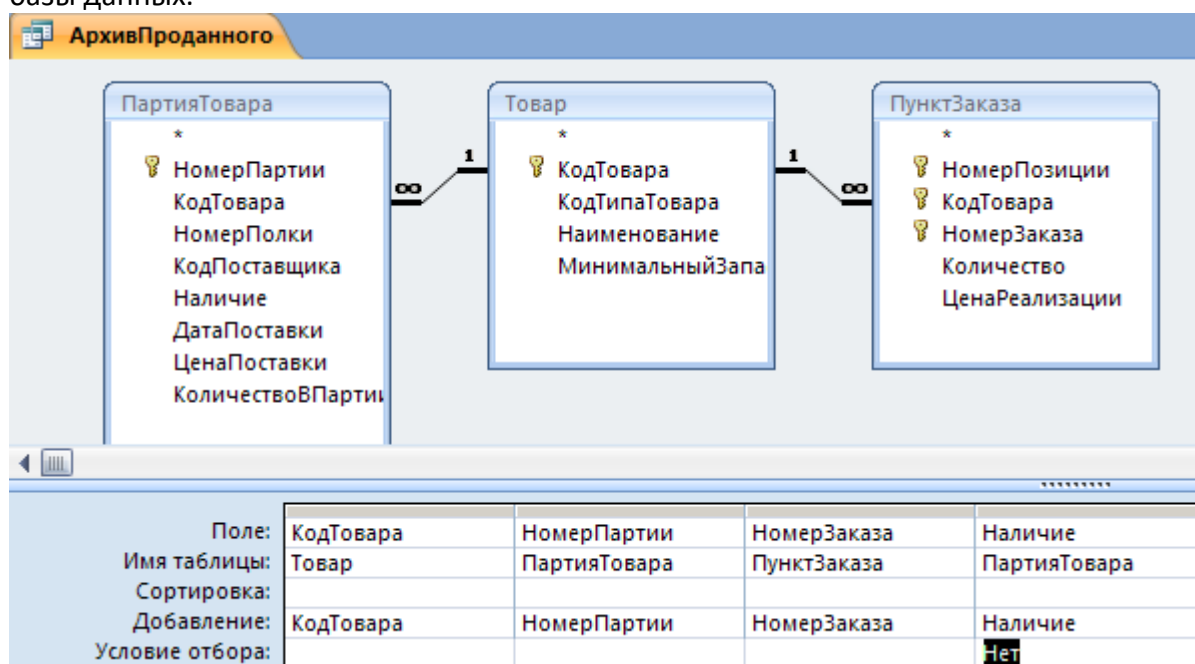


Рис.7. Запрос на ДОБАВЛЕНИЕ

Добавление записей в таблицу

имя таблицы:

в текущей базе данных

в другой базе данных:

имя файла:

Обзор...

OK

Отмена

Рис.8. Выбор таблицы для добавления записей

КодТовара	НомерПарт	Наличие	НомерЗака
1	1	0	1
1	2	0	1
1	1	0	1
1	2	0	1

Рис.9. Таблица для архивации записей о проданных товарах
Запросы обязательно сохраняйте, они потребуются нам для автоматизации форм.

Задание 5. Удаление группы записей

Запрос на удаление позволяет удалить записи из одной таблицы или из нескольких взаимосвязанных таблиц, для которых установлен параметр обеспечения целостности данных каскадное удаление связанных записей. Нам необходимо удалить записи о проданных партиях товаров из таблицы ПАРТИЯ_ТОВАРА.

- Сначала, как обычно, сконструируйте запрос на выборку (рис.10).
- В режиме КОНСТРУКТОРА измените тип запроса в запрос на УДАЛЕНИЕ (рис.10).

Обратите внимание, что перетаскивание полей в бланк запроса удобно реализовать с помощью элемента *. Поле НАЛИЧИЕ следует перенести дополнительно для указания условия отбора.

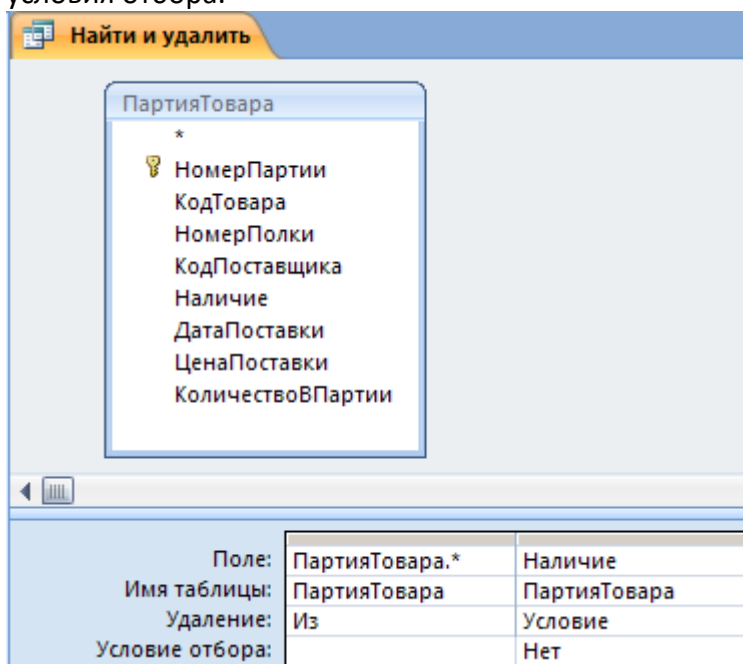


Рис.10. Построение запроса на удаление записей

3. Запустите запрос на исполнение. Сохраните запрос.

Если в запросе на удаление объединяются две или более таблиц, находящихся в отношении 1:М, то удалить можно только записи одной из таблиц на стороне «многие».

Задание 6. Автоматизация форм

Теперь, когда мы с вами имеем несколько запросов-действий, можно с их помощью автоматизировать формы.

1. Можно удалить вычисляемое поле ЦЕНА_РЕАЛИЗАЦИИ в подчиненной форме ПАРТИЯ и использовать запрос на обновление ЦЕНА_РЕАЛИЗАЦИИ.

2. Запрос на обновление ЦЕНА_РЕАЛИЗАЦИИ может быть добавлен в виде командной кнопки в подчиненную форму ПУНКТ_ЗАКАЗА. В этом случае у формы следует изменить режим изображения на ОБЫЧНЫЙ, так как в режиме ТАБЛИЦА командные кнопки не отображаются (рис. 11).

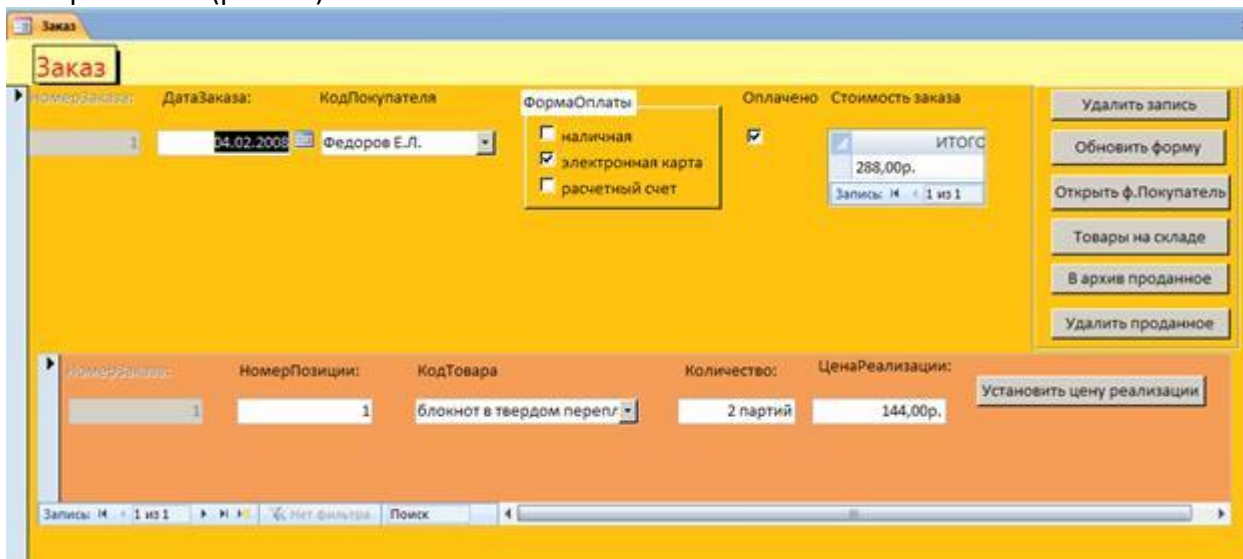


Рис.11. Примерный вид формы ЗАКАЗ с командными кнопками для запросов-действий

3. В форму ЗАКАЗ можно добавить еще одну командную кнопку для выполнения запроса на выборку НАЛИЧИЕ_ТОВАРА_НА_СКЛАДЕ (рис.12).

4. При наличии такой кнопки пользователь может вызывать запрос на исполнение, выбирать нужную партию товара и снимать флажок в поле НАЛИЧИЕ в случае продажи партии.

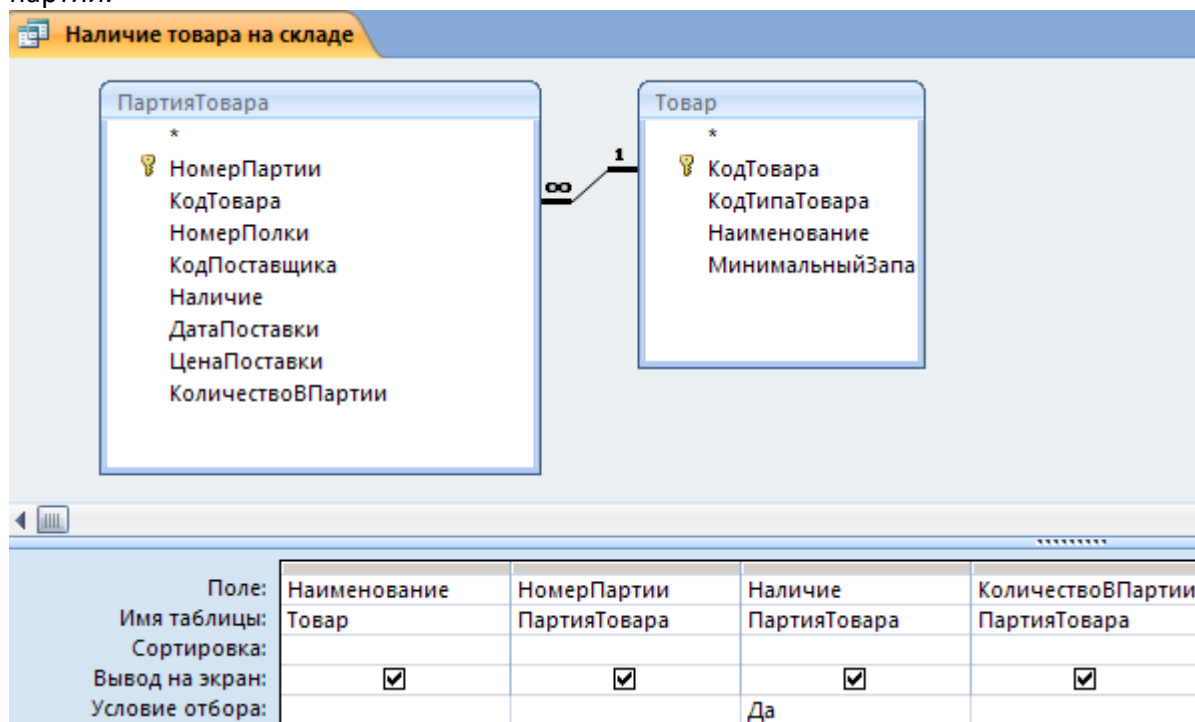


Рис.12. Запрос о наличии партий товара на складе

5. В форму ЗАКАЗ также можно добавить командную кнопку для выполнения запроса АРХИВ_ПРОДАННОГО. Напомним, этот запрос должен добавить в заранее созданную таблицу ПРОДАННЫЕ_ПАРТИИ_ТОВАРОВ записи о проданных партиях товаров. Создание таблицы ПРОДАННЫЕ_ПАРТИИ_ТОВАРОВ следует выполнить один раз, до того, как пользователь будет формировать заказы. Мы предусмотрим это позже, при создании главной формы приложения.

6. И, наконец, в форму ЗАКАЗ можно добавить командную кнопку для выполнения запроса на удаление записей о партиях проданных товаров из таблицы ПАРТИЯ_ТОВАРА (рис.11).

Создание командной кнопки для выполнения запроса на удаление выполняется по другой технологии, не так, как вы привыкли создавать кнопки для всех других запросов-действий.

Рассмотрим эту технологию.

1. В окне базы данных щелкните мышью по меню СОЗДАНИЕ - МАКРОС.
2. В открывшемся окне МАКРОСА из выпадающего списка выберите команду ОТКРЫТЬ ЗАПРОС (рис. 13).
3. В списке свойств в нижней части окна для строки ИМЯ ЗАПРОСА в раскрывающемся списке выберите имя запроса на удаление, в нашем случае – «Найти и удалить».
4. Сохраните макрокоманду, дайте ей название – УДАЛИТЬ. Закройте окно КОНСТРУКТОРА макрокоманд.

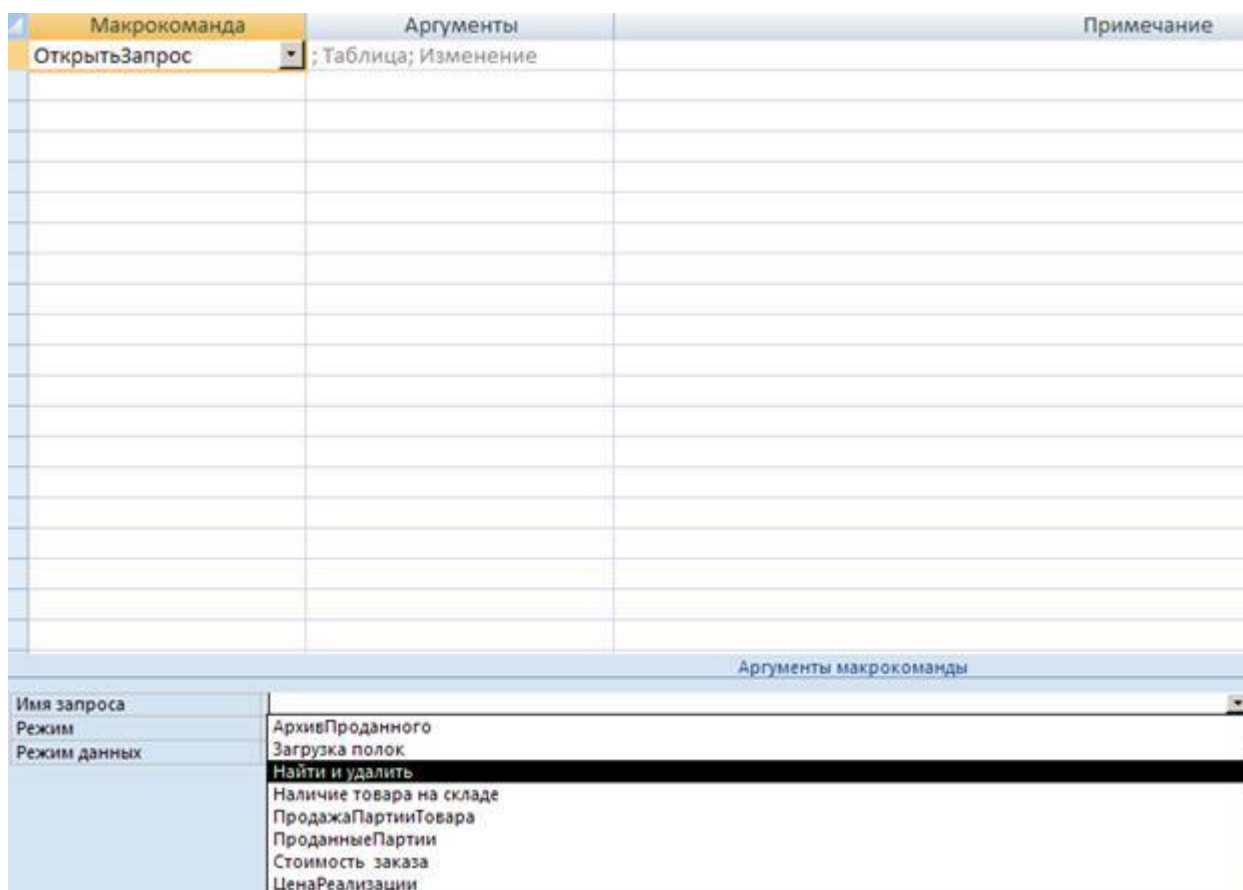


Рис.13. Конструктор макроккоманд

5. Раскройте форму ЗАКАЗ в режиме КОНСТРУКТОРА.
6. Щелкните по элементу КНОПКА на панели инструментов и покажите место расположения кнопки в области формы.
7. В окне МАСТЕРА создания кнопок укажите действие Выполнить макрос – «Удалить».
8. Перейдите в РЕЖИМ ФОРМЫ и протестируйте работу с формой.
9. Необходимо написать инструкцию пользователю о последовательности нажатия командных кнопок, чтобы не произошло удаления записей из таблицы ПАРТИЯ_ТОВАРА, до их копирования в архивную таблицу.
10. В области ПРИМЕЧАНИЕ формы напишите для пользователя краткое руководство к действиям.

Параметрические и перекрестные запросы

Цель

Научиться создавать и использовать запросы к данным, хранящимся в связанных таблицах базы данных.

Ввод параметров в запрос. Перекрестные запросы.

Задание 1. Ввод параметров в запрос

Мы научились конструировать запросы на выборку, а также запросы-действия, в которых можно задавать определенные условия их выполнения. Пока мы научились задавать условия отбора записей с использованием конкретных текстовых или числовых значений, например, >12. between 10 and 20, LIKE A* и так далее. Совершенно очевидно, что если значение 10 изменится и станет 50, то необходимо оторвать бланк запроса в режиме КОНСТРУКТОРА и вписать измененное значение параметра, что, конечно же, неудобно. Эта проблема решается при использовании параметрических запросов или запросов с параметрами.

Запрос с параметрами — это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести данные, например, условие для поиска

покупателей или значение, которое требуется вставить в поле. Можно разработать запрос, выводящий приглашение на ввод нескольких единиц данных, например, двух дат. Затем, Access может вернуть все записи, приходящиеся на интервал времени между этими датами.

Другими словами, вместо ввода в условие отбора для поля конкретного значения, для этого поля вводится вызов стандартного диалогового окна, в которое пользователь может ввести значение параметра для поиска по этому полю.

Пусть необходимо просматривать список партий товаров для товара, наименование которого пользователь может указать (ввести в качестве входного параметра).

1. Создайте обычный запрос на выборку для таблицы ПАРТИЯ_ТОВАРА (рис.1).
2. В условии отбора для поля НАИМЕНОВАНИЕ вместо конкретного наименования товара введите скобки [], в которых поместите текст, поясняющий: значение какого параметра пользователь должен ввести (рис.1).
3. Выполните запрос. При этом Access должен вывести диалоговое окно с вашими пояснениями (рис.2).
4. Введите в диалоговом окне название любого товара из вашей базы данных. Результат выполнения запроса приведен на рис.3.

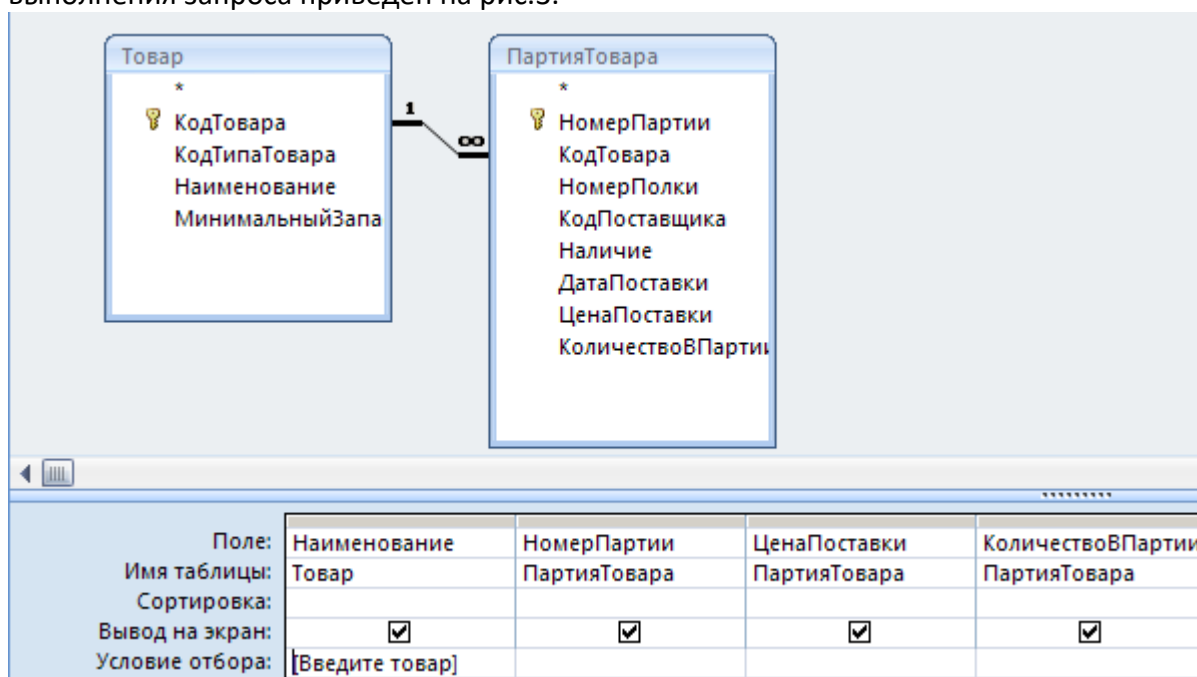


Рис.1. Формирование запроса с параметрами

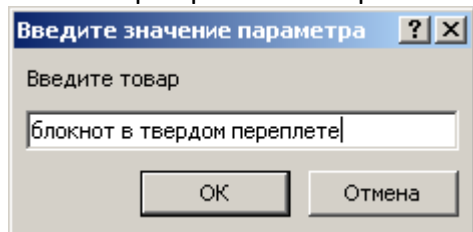


Рис.2. Диалоговое окно в предложение ввести значение параметра

	Наименование	НомерПарт	ЦенаПостаг	Количество
	блокнот в твердом переплете	1	100,00р.	50
	блокнот в твердом переплете	2	120,00р.	50
	блокнот в твердом переплете	5	90,00р.	40

Рис.3. Результат выполнения запроса с параметрами

5. Запустите запрос на исполнение еще раз. Введите другое наименование товара. Вы должны убедиться в том, что условия отбора записей изменяются.

Можно использовать неограниченное количество параметров.

6. Измените запрос на поиск товаров определенного типа и определенного наименования (рис.4).

7. Диалоговые окна должны появляться в зависимости от порядка их вызова в бланке запроса (приоритет в расположении: слева направо и сверху вниз).

8. На рис.4 показан порядок вызова диалоговых окон для запроса, имеющего более одного параметра.

9. Ориентируясь на рис. 5. с примером размещения параметров, изучите последовательность вызова диалоговых окон в зависимости от расположения параметров в бланке запроса.

Поле:	ТипТовара	Наименование	НомерПартии	ЦенаПоставки	КоличествоВПартии
Имя таблицы:	ТипТовара	Товар	ПартияТовара	ПартияТовара	ПартияТовара
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	[Введите тип товара]		[Введите товар]		

а). Бланк запроса в режиме КОНСТРУКТОРА

б) Порядок следования диалоговых окон

ТипТовара	Наименование	НомерПарт	ЦенаПостаг	Количество
бумажная канцеляр	блокнот в твердом переплете	1	100,00р.	50
бумажная канцеляр	блокнот в твердом переплете	2	120,00р.	50
бумажная канцеляр	блокнот в твердом переплете	5	90,00р.	40

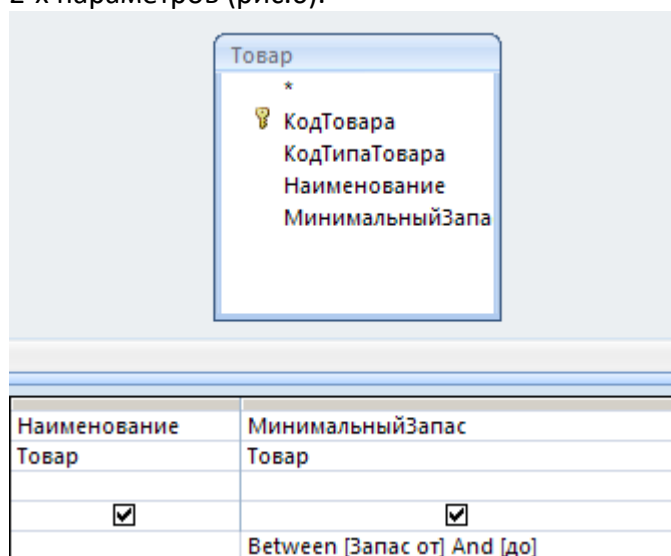
в). Результат поиска запроса с двумя параметрами

Рис.4. Использование нескольких параметров при конструировании запроса

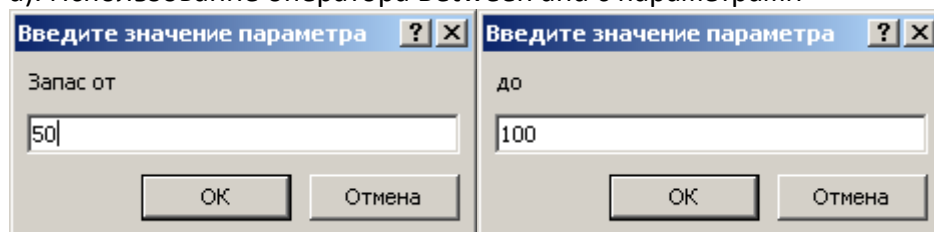
Поле:	ТипТовара	Наименование	НомерПартии	ЦенаПоставки	КоличествоВПартии
Имя таблицы:	ТипТовара	Товар	ПартияТовара	ПартияТовара	ПартияТовара
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	[Введите тип товара]		[Введите товар1]		
или:			[Введите товар2]		

Рис.5. Несколько параметров в запросе

Для одного поля может быть сконструирован диапазон условий поиска с использованием 2-х параметров (рис.6).



а). Использование оператора Between and с параметрами



б). Последовательно ввода параметров

Наименование	Минималы
блокнот в твердом переплете	70
блокнот в мягком переплете	100

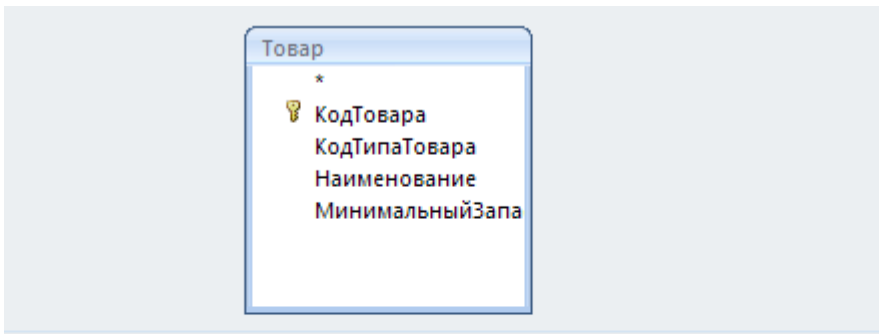
в). Результат выбора значений

Рис.6. Использование нескольких параметров для формирования условия для одного поля

10. Используя пример на рис.6, создайте собственные запросы с использованием нескольких параметров для задания условия по одному полю.

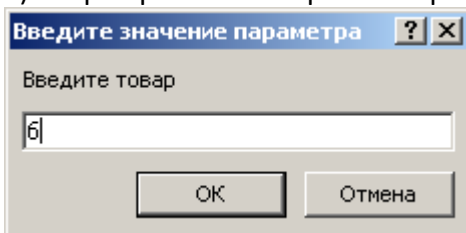
При формировании ввода значений входных параметров можно использовать символы подстановки * и ?. На рис.7. показан пример использования этих символов.

11. Используя пример на рис.7, создайте собственные запросы с символом подстановки *.



Наименование	МинимальныйЗапас
Товар	Товар
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Like [Введите товар] & "*"	

а). Формирование запроса с параметрами и символом подстановки *



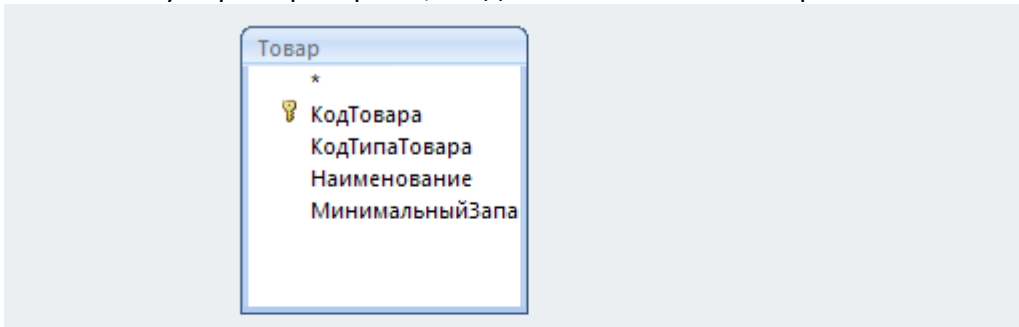
б). Ввод параметров

Наименование	Минималы
блокнот в твердом переплете	70
блокнот в мягком переплете	100

в). Результат выполнения запроса с символом подстановки *

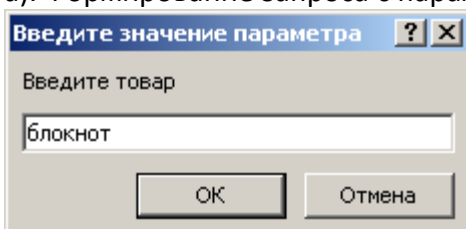
Рис.7. Использование символа подстановки *

12. Используя пример на рис.8, создайте собственные запросы с символом подстановки ?.



Наименование	МинимальныйЗапас
Товар	Товар
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Like [Введите товар] & " в ???????? переплете"	

а). Формирование запроса с параметрами и символом подстановки ?



б). Ввод параметров

Наименование	Минималь
блокнот в твердом переплете	70

в). Результат выполнения запроса с символом подстановки ?

Рис.8. Использование символа подстановки ?

Задание 2. Перекрестные запросы

Перекрестный запрос является специальным запросом итогового типа.

Перекрестный запрос позволяет увидеть вычисляемые значения в виде перекрестной таблицы, напоминающей электронную таблицу в MS Excel.

В перекрестном запросе левый столбец представляет собой **заголовки строк** из значений одного поля или нескольких полей. Заголовки строк могут быть вложенными.

Верхняя строка образует **заголовок столбцов**. Для заголовков столбцов вложенность не предусмотрена.

На пересечении строк и столбцов размещаются **итоговые значения**, вычисленные по значениям третьего поля.

Для получения итоговых значений записи группируются по полям, используемым в качестве заголовков столбцов и строк, и для значений третьего поля в полученных группах записей применяется одна из выбранных статистических функций.

Можно предусмотреть выполнение статистической функции и получение общего итогового значения для каждой строки в целом.

Рассмотрим создание перекрестного запроса на примере запроса о размещении партий товаров на полках склада.

Перекрестный запрос можно создать либо для полей одной таблицы, либо для полей из нескольких таблиц.

1. Создадим запрос для данных трех таблиц: ПОЛКА, ПАРТИЯ_ТОВАРА, ТОВАР (нам нужна полная информация о полях). Пример запроса показан на рис.9.

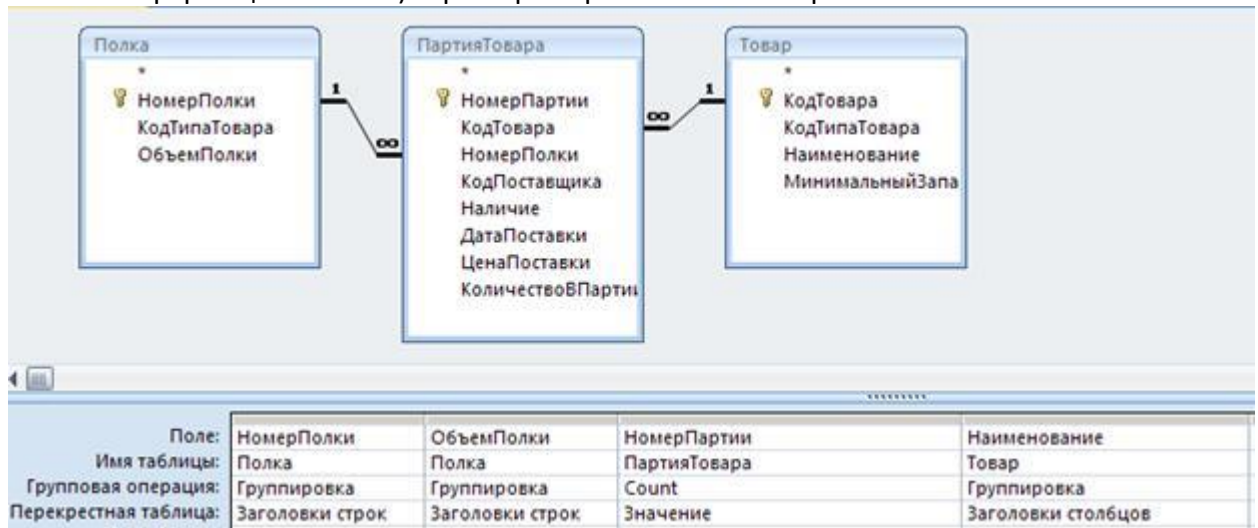


Рис. 9. Создание перекрестного запроса

2. Измените тип запроса на Перекрестный и укажите роли полей (заголовок строк, заголовок столбца, значение). Для поля, выбранного в качестве значения, установите функцию вычисления этого значения (рис.9).

3. Запустите запрос на исполнение. Результат выполнения запроса показан на рис.10.

НомерПолк	ОбъемПолки	блокнот в мягком	блокнот в твердом
1	10 партий	1	2
2	5 партий	1	1

Рис.10. Результат выполнения перекрестного запроса

Самостоятельно

1. Создайте запрос для вычисления количества партий, которое необходимо заказать у поставщиков, для всех наименований товаров. В качестве условия заказа необходимо учитывать величину минимального запаса каждого товара и количество партий товара на складе.
2. Создайте перекрестный запрос для вычисления прибыли, полученной от продажи товаров по всем заказам в прошлом месяце.
3. Создайте запрос для определения покупателей, у которых просрочена дата оплаты заказа. Пусть, например, срок оплаты после формирования заказа не должен превышать одной недели.
4. Измените созданный в п.3 запрос на запрос с параметрами, позволяющий изменять срок оплаты заказа.
5. Создайте перекрестный запрос для вычисления максимальной стоимости заказа.
6. Создайте запрос с параметрами для вычисления количество заказов, сделанных в определенном месяце.
7. Создайте запрос с параметрами для анализа загруженности определенной полки.
8. Создайте запрос для вычисления количество заказов, сделанных по указанному пользователем типу оплаты заказа.
9. Вычислите, на какую сумму не оплачены заказы за прошлый месяц.
10. Постройте перекрестный запрос о поставках каждого поставщика (схему запроса разработайте самостоятельно).

Создание отчетов в Access

Цель

Научиться создавать и использовать отчеты на основе данных, хранящихся в связанных таблицах базы данных.

Отчет для данных одной таблицы. Отчет на основе двух взаимосвязанных таблиц.

Разработка сложных отчетов с использованием запросов.

Задание 1. Отчет для данных одной таблицы

Средства разработки отчетов в Access предназначены для создания макета отчета, по которому может быть осуществлен вывод данных в виде выходного печатного документа. Эти средства позволяют создавать отчет сложной структуры, обеспечивающий вывод взаимосвязанных данных из многих таблиц, их группировку, вычисление итоговых значений.

Процесс конструирования отчетов выполняется с помощью панели инструментов КОНСТРУКТОРА и списка полей той таблицы (таблиц) или запроса, на основании которых отчет строится.

Приступим к созданию простого отчета для данных, хранящихся в одной таблице. Все действия, по-прежнему, будем выполнять в режиме КОНСТРУКТОРА.

Создадим отчет для всех заказов, оформленных в нашей базе данных. Определим требования к будущему отчету:

¼ Список заказов должен выводиться последовательно с сортировкой по номеру заказа.

¼ На каждой странице отчета должно выводиться название отчета, дата его создания.

¼ Страницы отчета должны быть пронумерованы.

1. В главном меню базы данных выберите СОЗДАНИЕ – КОНСТРУКТОР ОТЧЕТОВ.
2. Добавьте панель со списком доступных объектов, в ней раскройте список полей таблицы ЗАКАЗ и разместите нужные поля в область данных бланка отчета (рис.1).
3. Первоначально макет отчета не имеет разделов ЗАГОЛОВОК и ПРИМЕЧАНИЯ. Включите эти разделы с помощью соответствующей команды контекстного меню отчета.
4. В разделах ЗАГОЛОВОК и ПРИМЕЧАНИЕ разместите с помощью элемента управления НАДПИСЬ текст заголовка и примечаний для всего отчета (рис.1).

5. Для выполнения сортировки по номеру заказа нажмите на панели инструментов окна базы данных кнопку ГРУППИРОВКА И СОРТИРОВКА.
6. Укажите поле для группировки и порядок сортировки, например, выберите для группировки поле НОМЕР_ЗАКАЗА.
7. В нижней части окна макета в области задания дополнительных свойств (Больше...) укажите, что поле НОМЕР_ЗАКАЗА не будет полем заголовка группы и не требуется ПРИМЕЧАНИЕ для раздела ГРУППА.
8. Откорректируйте шрифты и расположение полей в бланке отчета по вашему усмотрению. Просмотрите готовый отчет.
9. Отчет можно сохранить для последующей печати, а можно сразу распечатать без сохранения (к сожалению, принтер не подключен).
10. В область верхнего колонтитула добавьте надпись с название отчета (рис.1).
11. В области нижнего колонтитула поместите поле с нумерацией страниц (рис.1).
12. Еще раз просмотрите готовый отчет в режиме ВИД – ПРЕДСТАВЛЕНИЕ ОТЧЕТА и внесите необходимые изменения.

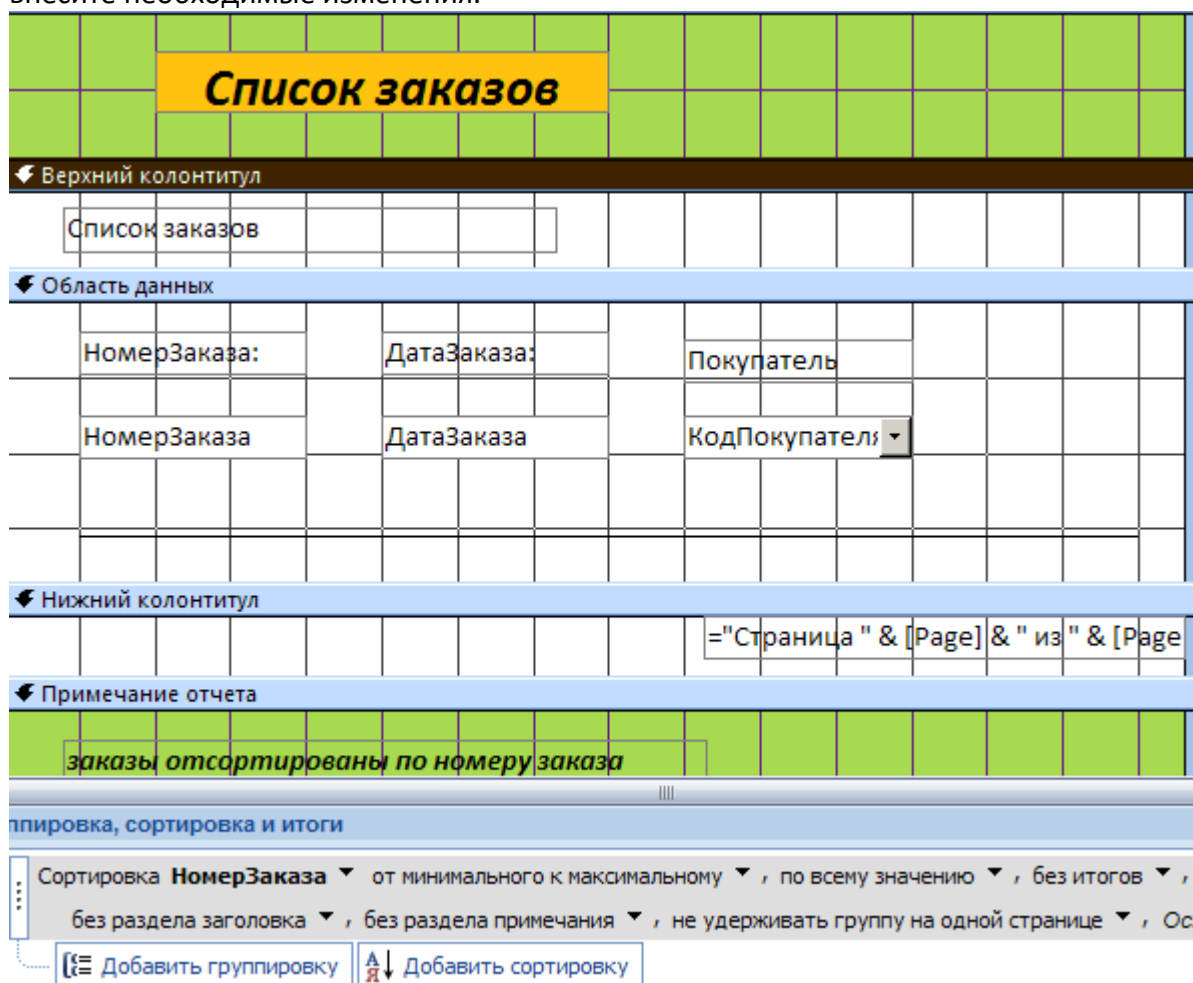


Рис.1. Макет отчета для данных одной таблицы в режиме КОНСТРУКТОРА отчетов

Задание 2. Отчет на основе двух взаимосвязанных таблиц

Многотабличный отчет можно сконструировать различными способами.

Один из способов – создать основную часть отчета для таблицы на стороне 1 (главной таблицы) и подчиненную часть отчета для таблицы на стороне М (подчиненной таблицы). Отчеты создать отдельно для каждой из таблиц, а затем, вложить подчиненный отчет в главный отчет так же, как мы это делали с вложенными формами.

1. Продолжим работу с отчетом для таблицы ЗАКАЗ.

2. Создайте в режиме КОНСТРУКТОРА еще один отчет (он будет подчиненным отчетом) для отчета ЗАКАЗЫ. Сконструируйте его по своему усмотрению, но расположение полей выполните компактно, например, как показано на рис.2.

Рис.2. Подчиненный отчет ПОЗИЦИИ_ЗАКАЗА в режиме КОНСТРУКТОРА

3. Сохраните и закройте отчет для таблицы ПОЗИЦИИ_ЗАКАЗА.

4. Откройте отчет ЗАКАЗЫ в режиме КОНСТРУКТОРА и разместите в области данных отчет ПОЗИЦИИ_ЗАКАЗА (рис.3).

Рис.3. Создание многотабличного отчета с помощью двух отчетов

5. Просмотрите результат совмещения отчетов в режиме ВИД. Внесите корректировки по размерам и расположению встроеного отчета.

Задание 3. Разработка сложных отчетов с использованием запросов

Рассмотрим другую технологию создания сложных отчетов.

Вы могли убедиться в том, что запрос является мощным и удобным средством выборки взаимосвязанных данных. Поэтому с помощью запроса можно подготовить данные для построения отчета любой сложности.

Построим другой отчет. Пусть требуется создать описание всех полок склада с указанием партий товаров, хранящихся на этих полках.

1. Создадим запрос для объединения полей из таблиц ПОЛКА, ПАРТИЯ_ТОВАРА, ТОВАР (рис.4). Сохраните и закройте запрос СКЛАД.

2. Начините создавать новый отчет с помощью МАСТЕРА ОТЧЕТОВ. В списке доступных объектов выберите запрос СКЛАД.

3. Создадим иерархию групп для отчета. Пусть верхний уровень группировки будет выполнен по полю НОМЕР_ПОЛКИ, второй уровень группировки будет по полю НАИМЕНОВАНИЕ_ТОВАРА (рис. 5).

4. В бланке отчета должны появиться дополнительные разделы для заголовков групп (рис. 5).

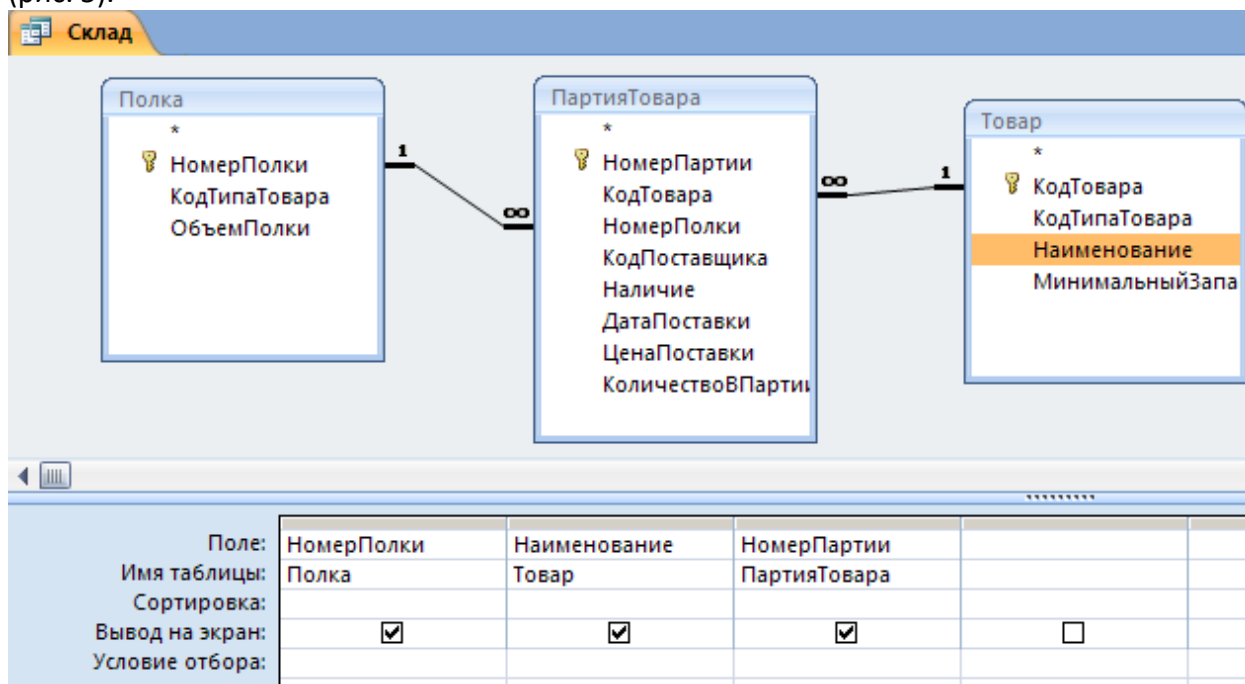


Рис.4. Запрос, объединяющий поля таблиц для отчета

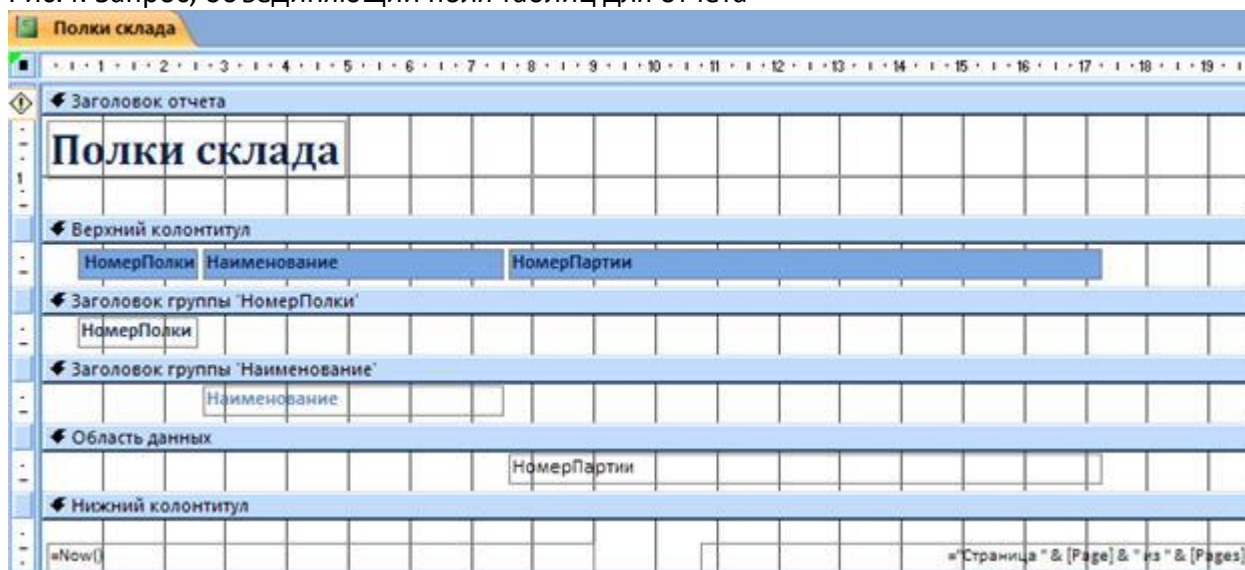


Рис.5. Макет отчета СКЛАД

5. Просмотрите отчет в режиме ВИД. Подкорректируйте расположение и форматы полей. Сохраните и закройте отчет СКЛАД.

6. Вернитесь в запрос СКЛАД. Внесите изменения: добавьте операцию группировки и статистическую функцию COUNT для поля НОМЕР_ПАРТИИ). Сохраните и закройте запрос.

7. Создайте другой отчет для измененного запроса. Обратите внимание на кнопку ИТОГИ в мастере отчетов. Отформатируйте бланк отчета, например, как показано на рис.6

Полки склада		
НомерПолки	Наименование	Count-НомерПартии
1	блокнот в мягком переплете	1
	блокнот в твердом переплете	2
Всего	3 партий	
2	блокнот в мягком переплете	1
	блокнот в твердом переплете	1
Всего	2 партий	
ИТОГО		5
29 марта 2009 г.		Страница 1 из 1

Рис.6. Изменения в отчете СКЛАД

Задание 4

1. Разработайте отчет, включающий сведения о товарах каждого поставщика. Используйте вычисляемые поля.
2. Создайте отчет на основании запроса с параметрами. Убедитесь в том, что при открытии бланка отчета появляются диалоговые окна для ввода значений параметров.
3. Разработайте аналитический отчет о покупательском спросе на отдельные категории товаров.