

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

КАФЕДРА ИНФОРМАТИКИ И ПРИКЛАДНОЙ МАТЕМАТИКИ

Технологии создания и ведения баз данных в Microsoft Access

**Пособие
для студентов экономических специальностей**

Брест 2023

УДК 681.3
ББК 73
С58

Пособие содержит основные теоретические сведения по проектированию и разработке баз данных, практические рекомендации и пояснения по использованию системы управления базами данных Microsoft Access на примере выполнения задачи по созданию и ведению базы данных.

Цель пособия оказать теоретическую и практическую помощь студентам экономических специальностей всех форм обучения при выполнении лабораторных работ и подготовке к экзамену по дисциплине «Информационные технологии», написании курсовых и дипломных работ по специальности.

Составили: Аверина И.Н., доцент;
Гучко И.М., ст. преподаватель

Рецензенты: В. А. Головки, профессор, доктор технических наук., зав. кафедрой интеллектуальных информационных технологий учреждения образования «Брестский государственный технический университет»
Д. В. Грицук, кандидат физико-математических наук, доцент, зав. кафедрой прикладной математики и информатики учреждения образования «Брестский государственный университет имени А.С. Пушкина»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1. ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ	5
1.1. Понятие базы данных (БД)	5
1.2. Модели представления информации в БД	6
1.3. Реляционная БД	7
1.4. Понятие ключевого поля (полей) таблицы БД	7
1.5. Связи между таблицами в реляционных БД	7
1.5.1. Вид связи один–к–одному (1:1)	8
1.5.2. Вид связи один–ко–многим (1:M)	8
1.5.3. Вид связи многие–ко–многим (M:M)	8
1.6. Проектирование реляционной БД	9
1.7. Правила обеспечения целостности данных при ведении БД	10
1.8. Каскадные операции при ведении БД	10
1.9. Системы управления базами данных (СУБД)	11
1.10. Общая характеристика СУБД MS Access	11
1.11. Основные объекты БД MS Access	12
1.12. Файл БД MS Access	12
1.13. Интерфейс приложения MS Access	13
1.14. Способы создания таблиц в MS Access	15
1.15. Понятие запроса. Виды запросов в MS Access	15
1.16. Формы и отчеты как объекты БД MS Access	16
2. ПРАКТИЧЕСКАЯ ЧАСТЬ	17
Практическое задание 1. Создание файла БД Магазин СТРОИТЕЛЬ.	17
Практическое задание 2. Создание таблиц БД Магазин СТРОИТЕЛЬ	17
2.1. Создание таблицы с помощью Конструктора таблиц	18
2.2. Типы данных в MS Access	19
2.3. Описание свойств полей	20
2.4. Формирование маски ввода	22
2.5. Свойство поля «Значение по умолчанию»	23
2.6. Использование Мастера подстановки для создания фиксированного списка значений	23
2.7. Задание первичного ключа	25
2.8. Сохранение таблицы	25
Практическое задание 3. Установка связей между таблицами БД	25
2.9. Использование Мастера подстановки для создания связей между таблицами	26
2.10. Установка связей между таблицами БД в MS Access	29
2.11. Удаление (изменение) связей	30
Практическое задание 4. Заполнение таблиц БД записями.	30
2.12. Ввод данных в таблицу	31
2.13. Сортировка данных в таблице	31
Практическое задание 5. Проектирование запросов на выборку	31
2.14. Создание запросов выбора с помощью Конструктора запросов	31
2.14.1. Добавление (удаление) полей в Бланке запроса	32
2.14.2. Сортировка результатов выборки	33
2.14.3. Запуск запроса на выполнение	33
2.15. Построитель выражений	34
2.16. Формирование условий отбора в запросе	35
2.17. Формирование условий отбора для полей с типом данных Дата/Время	36
2.18. Сложные критерии выборки в запросах	36

Практическое задание 6. Параметрические запросы _____	38
2.19. Проектирование параметрических запросов _____	38
Практическое задание 7. Вычисляемые запросы для БД в MS Access _____	39
2.20. Вычисляемые поля в запросах для БД в MS Access _____	39
Практическое задание 8. Итоговые запросы для БД MS Access _____	43
2.21. Проектирование итоговых запросов для БД MS Access _____	43
Практическое задание 9. Перекрестные запросы для БД в MS Access _____	48
2.22. Проектирование перекрестных запросов для БД в MS Access _____	48
Практическое задание 10. Проектирование форм для БД в MS Access _____	53
2.23. Способы создания форм в MS Access _____	53
2.24. Создание формы с помощью инструмента Автоформа _____	54
2.25. Создание формы с помощью Мастера форм _____	54
2.26. Создание формы в режиме Конструктора _____	55
2.27. Структура формы _____	55
2.28. Свойства и события объектов формы _____	57
2.29. Размещение элемента управления "Поле" в форме _____	57
2.30. Разработка многотабличных (составных) форм в MS Access _____	60
2.31. Проектирование отчетов для БД в MS Access _____	61
2.32. Создание управляющей (кнопочной) формы для работы с БД _____	62
2.33. Использование макросов в MS Access _____	63
ПРИЛОЖЕНИЕ 1. Описание структур записей таблиц БД Магазин СТРОИТЕЛЬ _____	65
ПРИЛОЖЕНИЕ 2. Описание свойства поля "Маска ввода" _____	67
ПРИЛОЖЕНИЕ 3. Функции для обработки полей с типом данных Дата/Время _____	68
ПРИЛОЖЕНИЕ 4. Функции Iif() и Format() _____	69
ПРИЛОЖЕНИЕ 5. Примеры условий отбора _____	70
ЛИТЕРАТУРА _____	71

ВВЕДЕНИЕ

В любой предметной области объемы информации, требующей хранения и обработки, увеличиваются ежесекундно. В современном мире использование баз данных является неотъемлемой частью профессиональной деятельности человека.

Системы управления базами данных (СУБД) обеспечивают оптимальное хранение и эффективную обработку больших объемов информации, связанной между собой.

Пособие содержит основные теоретические сведения по проектированию и разработке реляционных баз данных, а также практические рекомендации и пояснения по использованию системы управления базами данных Microsoft Access. В пособии приводится пример создания и ведения базы данных в Microsoft Access и излагаются технологии создания ее основных объектов: таблиц, запросов, форм, отчетов и макросов.

Цель изучения пособия – получение знаний о современных технологиях организации, хранения и обработки информации, приобретение навыков по созданию баз данных и работе с ними при решении практических задач предметной области. Материал пособия соответствует базовой программе обучения по теме «Технологии баз данных» дисциплины «Информационные технологии».

Основной задачей пособия является развитие практических навыков при работе в реляционной СУБД: проектирование, создание и ведение базы данных с помощью запросов, разработка форм и отчетов, инструментов управления базой данных.

Изучение материала пособия основано на использовании базовых знаний информатики, умений и навыков работы с электронными таблицами.

Знания, умения и навыки, приобретенные в ходе изучения материала пособия, позволят студентам эффективно использовать современное прикладное программное обеспечение в процессе решения учебных, научных и практических задач, станут основой для освоения корпоративных информационных систем, применяемых организациями в современных условиях.

Владение базовыми технологиями работы в СУБД призвано развить информационно-коммуникативные компетенции будущих экономистов, помочь оперативно овладеть навыками профессиональной работы в специализированных автоматизированных информационных системах в своей будущей деятельности.

1. ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

1.1. Понятие базы данных (БД)

База данных (БД) – любая совокупность информации, связанная общей предметной областью, организованная специальным образом при помощи специальных программных средств (систем управления базами данных – СУБД) и хранящаяся на машинных носителях в виде одного или нескольких файлов. БД отображает состояние объектов, их свойства и взаимоотношения.

Предметная область – это часть реального мира, которая описывается и моделируется с помощью БД.

Объектом БД может быть предмет, физ. или юр. лицо (организация), событие, абстрактное понятие, т.е. любая сущность, информацию о которой требуется хранить в базе данных. Объект БД описывается набором значений некоторой совокупности атрибутов (характеристик) объекта. Атрибут – это информационное отображение свойства

объекта. Например, объект «Товар» характеризуется атрибутами: наименование, единица измерения, дата изготовления, цена, срок годности, поставщик и др.

Единицей информации в БД является запись.

Запись – это набор сведений об определенном объекте, событии или явлении. Запись состоит из полей. Поле – это отдельный атрибут (характеристика) объекта БД. Поле содержит однотипные данные.

Объем информации в БД измеряется количеством записей в ней.

1.2. Модели представления информации в БД

От способа организации информации в БД зависит возможность быстрого поиска, анализа и отображения данных. Набор принципов, определяющих организацию хранения информации в базе данных, называется моделью данных. Модель данных – это формально определенная логическая структура представления информации в БД.

По способу организации информации в БД различают следующие виды моделей:

I. Иерархическая модель. В такой модели любой объект может подчиняться только одному объекту вышестоящего уровня. В иерархической модели один объект главный, остальные – подчиненные (рис. 1).

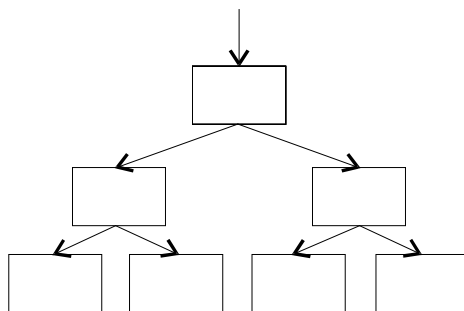


Рисунок 1 – Иерархическая древовидная структура модели БД

II. Сетевая модель. В сетевой модели организации информации любой объект может быть подчинен нескольким объектам, т.е. может быть и главным и подчиненным, и может участвовать в любом количестве взаимосвязей (рис. 2).

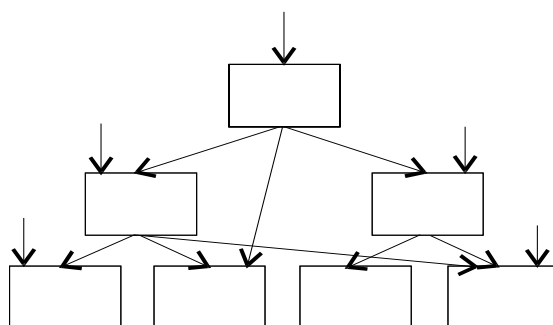


Рисунок 2 – Сетевая структура модели БД

III. Реляционная модель. В реляционной модели информация представляется в виде таблиц, связанных между собой совпадающими данными.

Таким образом, реляционная модель характеризуется основными свойствами:

1) в реляционной модели информация представляется в виде **таблиц**. Таблица хранит информацию об учитываемых в БД сущностях в виде записей.

Записи располагаются в строках таблицы БД. Каждая строка таблицы БД имеет одинаковую структуру и состоит из полей.

Поля записи располагаются в столбцах таблицы БД.

2) в реляционной модели данных таблицы между собой могут быть связаны посредством значений одного или нескольких совпадающих полей (ключей).

1.3. Реляционная БД

Реляционная БД (РБД) – это совокупность логически взаимосвязанных таблиц-отношений, состоящих из множества полей и записей, отражающих некоторую предметную область.

Отношение – информация об объектах одного типа, например, о товарах, заказах, поставщиках. В реляционной БД отношение хранится в виде таблицы.

Каждая таблица имеет имя, уникальное внутри БД. В свою очередь, каждая строка (запись) такой таблицы содержит информацию, относящуюся только к одному конкретному объекту, а каждый столбец (поле) таблицы имеет уникальное для своей таблицы имя. Каждое поле таблицы хранит однотипные данные.

Тип данных – определяет способ хранения информации в БД и возможности ее обработки. Набор допустимых типов данных зависит от СУБД.

Связь – способ, которым информация одной таблицы объединена с информацией другой таблицы. Связи (объединения) осуществляются с помощью совпадающих полей, которые называются **ключевыми**.

1.4. Понятие ключевого поля (полей) таблицы БД

Ключом называется набор атрибутов (полей), однозначно определяющий запись.

Ключевое поле (или поля) – это такое поле или набор полей, значения которого (или комбинация значений которых) для каждой записи таблицы уникально. Значения ключевого поля (полей) не могут быть пустыми (не допускает значение Null).

Ключи делятся на два вида: *простые* и *составные*.

Простой ключ состоит из одного поля.

Примеры полей, которые могут выступать в роли простого ключа: КодТовара, ТабельныйНомер, ИнвНомер, УНП, НомерПоПорядку.

Составной ключ состоит из нескольких полей. При этом в таблице для каждой записи могут повторяться значения отдельных полей, входящих в составной ключ. Но комбинация значений этих полей составного ключа повторений иметь не может. Составные ключи позволяют оптимизировать хранение информации в БД.

Пример набора полей для составного ключа: НомерТТН + ДатаТТН + Поставщик + КодТовара. Для этого примера в таблице могут быть записи с одинаковыми значениями и НомерТТН, и ДатаТТН, и Поставщик, и КодТовара. А вот запись, у которых совпадает комбинация их значений, может быть только одна, то есть от конкретного *Поставщика* в одной и той же накладной *НомерТТН* от одной и той же *ДатаТТН* может быть записан товар с *КодТовара* только один раз.

Таким образом, нужно подчеркнуть **основные назначения ключевых полей**:

1. Исключение повторяющихся записей в таблицах БД.
2. Организация **связей** между таблицами БД.

1.5. Связи между таблицами в реляционных БД

Связи устанавливаются между двумя таблицами БД по равенству значений одного поля в обеих таблицах. Связанные отношениями таблицы взаимодействуют по принципу «главная-подчиненная».

При связывании таблиц используются понятия **первичного** и **внешнего ключей**. Если в связи участвует ключевое поле, такая таблица, называется **главной**, а ее ключевое поле и есть **первичный ключ**.

Если во второй таблице в связи участвует не ключевое поле, то такая таблица считается **подчиненной**, а поле этой таблицы, по которому установлена связь с ней, называется **внешним ключом**. Внешний ключ определяет способ связи таблиц.

Одна и та же таблица может быть главной по отношению к одной таблице БД и подчиненной по отношению к другой.

Главная таблица в БД содержит, как правило, информацию об основных сущностях БД, без которой БД теряет смысл. Например, без *Поставщиков* не может быть *Товаров*, без *Товаров* и *Покупателей* – *УчетаПродаж* и т.п. Главная таблица, это, как правило, таблица-справочник.

В реляционных БД существуют три основных **вида связей** между таблицами:

- 1) один–к–одному (1:1)
- 2) один–ко–многим (1:M)
- 3) многие–ко–многим (M:M)

1.5.1. Вид связи один–к–одному (1:1)

Вид связи **один–к–одному** (1:1) означает, что каждой записи в главной таблице может соответствовать только одна запись в подчиненной таблице с таким же значением поля, по которому осуществляется связь.

В связи **один–к–одному** могут участвовать только ключевые поля обеих таблиц. В этом случае говорят, что одна таблица БД является продолжением другой. *Например*, таблица «Основные сведения о работниках» и таблица «Дополнительные сведения».

В схеме данных такой вид связи **один–к–одному** обозначается 1 1.

1.5.2. Вид связи один–ко–многим (1:M)

Вид связи **один–ко–многим** (1:M) означает, что каждой записи в главной таблице может соответствовать одна, несколько или ни одной записи в подчиненной таблице с таким же значением поля, по которому установлена связь.

В связи **один–ко–многим** может участвовать только ключевое поле главной таблицы и соответствующее ему не ключевое поле в подчиненной таблице (внешний ключ). *Например*, связь один–ко–многим установлена между ключевым полем *КодТовара* из таблицы *Товары* с полем *Товар* из таблицы *УчетПродаж*.

В схеме данных вид связи **один–ко–многим** обозначается 1 ∞. При этом обозначение 1 как бы говорит о том, что в таблице каждое значение этого поля встречается не более 1 раза, а символ ∞ означает, что такое значение в таблице может встретиться и много раз (или ни разу вообще).

Вид связи **один–ко–многим** наиболее часто используется в реляционных БД.

1.5.3. Вид связи многие–ко–многим (M:M)

Связь **многие–ко–многим** (M:M) означает, что в связи участвуют поля, которые не являются ключевыми полями в таблицах А и В (внешние ключи), и;

а) одна запись из **таблицы А** (выходная связь) может быть связана больше чем с одной записью **таблицы В** (принимающая);

б) одна запись из **таблицы В** может быть связана больше чем с одной записью **таблицы А**. Идентификация не является уникальной в обоих направлениях.

Схема связи **многие–ко–многим** (M:M) реализуется в реляционных СУБД только при помощи третьей соединительной таблицы, ключ связи которой состоит, как мини-

мум, из двух полей. То есть, если между таблицами имеются связи типа М:М, создается дополнительная таблица пересечений, с помощью которой связь М:М будет сведена к двум связям типа один–многим (1:М). При этом первичный ключ для соединительной таблицы – это, как правило, комбинация из внешних ключей.

1.6. Проектирование реляционной БД

Разработка БД и ее использование осуществляется специалистами, владеющими знаниями предметной области, и включает следующие этапы:

- проектирование БД;
- создание БД;
- внедрение БД, заполнение нормативно-справочной информации;
- ведение БД, ее администрирование и техническое сопровождение.

Проектирование БД самый важный этап. Хорошо организованная БД обеспечивает удобный доступ к хранящейся в ней информации. При правильном проектировании БД будет меньше затрачиваться времени и усилий на ввод данных в базу, внесение изменений и извлечение данных из базы.

Проектирование реляционной БД состоит из следующих основных этапов:

- постановка задачи;
- проектирование таблиц, определение полей и их свойств;
- определение связей между таблицами.

На этапе **постановки задачи** решают, какая информация будет храниться в планируемой БД, и как она будет использоваться. Как правило, постановка задачи включает перечень вопросов, ответы на которые должны быть получены из БД, список отчетов, которые должны формироваться по БД, многие другие задачи ведения БД. Исходя из этого, определяют, какие таблицы должны составлять БД и какие сведения (поля) должны быть включены в каждую таблицу.

Проектирование таблицы – это определение наборов полей, составляющих ее запись. При проектировании структуры записи таблицы решают, какие именно сведения об учитываемых в БД сущностях важны для решения поставленных перед БД задач.

При проектировании таблиц реляционной БД руководствуются следующими правилами:

- каждая запись (строка) в таблице должна быть уникальной, то есть не иметь повторений. Для реализации этого правила в каждую таблицу БД должно быть введено ключевое поле или поля.
- в структуру записи не следует включать поля, данные которых могут быть вычислены из других полей.
- в структуру записи не следует включать поля, значения которых будут одинаковыми для всех записей.
- каждое поле должно содержать информацию, относящуюся к теме таблицы.
- разные таблицы могут содержать поля с совпадающими данными, по этим полям в дальнейшем будет осуществляться связь информации между таблицами. При этом, название полей может не совпадать, но их тип, свойства и значения совпадать должны обязательно.

Проектирование связей между таблицами. Для определения связей выбирают пару таблиц, информация которых пересекается. Из них определяют главную и подчиненную. Главные таблицы содержат, как правило, справочную информацию, которая корректируется сравнительно редко. Подчиненные таблицы обрабатываются ежедневно

и используют информацию из различных справочных таблиц. При определении связей обращают внимание на ключевые поля таблиц.

Графическое отображение всех связей между таблицами БД называется **схемой данных**. На рисунке 3 приведен пример схемы данных для БД Магазин СТРОИТЕЛЬ, при этом в таблицах на рисунке выделены ключевые поля.

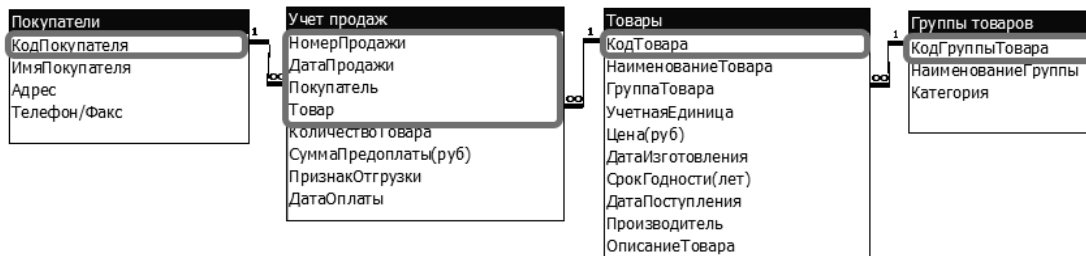


Рисунок 3 – Схема данных БД Магазин СТРОИТЕЛЬ

1.7. Правила обеспечения целостности данных при ведении БД

При ведении БД должна обеспечиваться целостность данных.

Целостность данных – это набор правил, которые поддерживают корректность связей между записями в связанных таблицах и обеспечивают защиту данных от случайных изменений или удалений. Правила обеспечения целостности данных заключаются в запрете следующих действий в БД:

- в подчиненную таблицу нельзя добавить запись, которая не связана с записью главной таблицы;
- в главной таблице нельзя изменять значение ключевого поля, если в подчиненной таблице существуют записи, которые с ней связаны;
- в главной таблице нельзя удалять записи, если в подчиненной таблице существуют связанные с ней записи.

При организации БД современные СУБД обеспечивают автоматизированный контроль целостности данных.

Режим обеспечения целостности данных устанавливается при создании конкретной связи между таблицами. Если опция установлена, то любая попытка выполнить одно из перечисленных выше действий, будет невозможна и появится предупреждение.

Установить режим обеспечения целостности данных можно, только если при задании связи между таблицами выполняются следующие условия:

- связанное поле главной таблицы является первичным ключом или альтернативным ключом (имеет уникальный индекс);
- связанные поля (первичный ключ главной таблицы и внешний ключ подчиненной таблицы) имеют один и тот же тип данных. Существует исключение: поле с типом данных **Счетчик** может быть связано с числовым полем, свойство которого *Размер поля* имеет значение Длинное целое;
- свойства *Размер поля* для обоих связываемых полей числового типа должны быть одинаковыми.

1.8. Каскадные операции при ведении БД

Жесткость условий целостности данных в связанных таблицах могут ослабить **каскадные операции** двух видов: **каскадное обновление** и **каскадное удаление**.

Если установлен режим «Каскадное обновление связанных полей», то любые изменения в значении ключевого поля главной таблицы, которая стоит на стороне «один»

в отношениях 1:M, ведут к автоматическому обновлению соответствующих значений во всех связанных записях в других таблицах.

При установке режима «Каскадное удаление связанных записей» при удалении записи из главной таблицы обеспечивается автоматическое удаление связанных записей в подчиненных таблицах.

Каскадные операции можно включать и выключать путем установки соответствующих опций при задании связей между таблицами в соответствующем функционале СУБД.

1.9. Системы управления базами данных (СУБД)

Система баз данных – это компьютеризованная система хранения записей, которая включает в себя непосредственно сохраняемые данные, аппаратное обеспечение, программное обеспечение (ПО) и пользователей.

СУБД (система управления базами данных) – это программная система (комплекс программ и языковых средств), которая обеспечивает все манипуляции с БД: создание на компьютере БД, ее ведение, обработку и использование данных, их защиту, организацию эффективного доступа пользователей к содержащимся в БД данным в рамках предоставленных им полномочий и др., т. е. СУБД реализуют сложный комплекс функций по централизованному управлению БД и обслуживают интересы ее пользователей.

СУБД можно рассматривать как программную оболочку, которая находится между базой данных и пользователем.

СУБД обеспечивает:

- набор средств для поддержки таблиц и отношений между ними;
- централизованный контроль защиты и целостности данных,
- интерфейс (диалог) для ввода и корректировки данных;
- интерфейс (диалог) для доступа к данным, их обработки, формирования отчетов на основе БД и другие операции и процедуры;
- средства программирования высокого уровня.

Каждая СУБД имеет свой язык и инструменты манипулирования данными, с помощью которых формируются обращения на поиск или выборку данных. Этот язык может также использоваться как язык программирования для СУБД. Причем он может быть как самостоятельным новым языком, так и частью или расширением общепринятого языка программирования.

Первые СУБД появились в середине 60-х годов XX в., они позволяли работать с иерархическими БД. Наиболее известной была иерархическая СУБД IMS фирмы IBM, известны также PC/Focus, Team-Up, Data Edge и др.

К известным сетевым СУБД относятся: DBMS, IDMS, TOTAL, VISTA, СЕТЬ, СЕТОР. КОМПАС и др.

Среди СУБД, использующих наиболее популярную реляционную модель представления данных, известны такие, как dBase, FoxBase, FoxPro, Paradox, Clarion, Clipper, Ingres, Oracle, MySQL, PostgreSQL, Microsoft Access и др.

1.10. Общая характеристика СУБД MS Access

В данном пособии рассматривается работа с СУБД Access, которая выпускается компанией Microsoft и работает в операционной системе Windows.

Microsoft Access (далее – Access) – это функционально полная реляционная СУБД, в которой предусмотрены все необходимые средства для организации, хранения и обработки данных, а также для управления ими при работе с большими объемами инфор-

мации. С помощью Access можно разрабатывать БД различной сложности: от индивидуальной, предназначенной для использования на отдельном компьютере, до систем масштаба предприятия, работающих в компьютерных сетях. Access, как одно из приложений пакета Microsoft Office, установленного на многих компьютерах, по праву считается настольной СУБД.

Первая версия Access вышла в 1992 году для Windows 3.0, последующие версии появляются с выходом новых версий пакета MS Office, например, 2003, 2007, 2010, 2013, 2016, 2019. Поскольку интерфейс Access, начиная с версии 2007, относительно схож, авторы пособия посчитали возможным обозначить версию Access, начиная с 2007 выпуска как 2007+. В практической части пособия рассматривается функционал версий Access 2003 и 2007+.

Access обладает большим набором инструментов для выполнения различных манипуляций с БД без программирования. Для начинающих пользователей Access большую помощь оказывают встроенные **Мастера**, которые предлагают последовательность шагов с подробными подсказками, для создания таблиц, форм, отчетов и выполнения других действий. Опытным пользователям полные возможности по созданию объектов БД предоставляют встроенные в Access **Конструкторы**. Все это позволяет быстро освоить работу в Access с БД, используя только визуальные средства.

1.11. Основные объекты БД MS Access

СУБД Access ориентирована на работу с объектами БД, к которым относятся: таблицы, запросы, формы, отчеты, страницы, макросы и модули.

Таблицы – основные объекты БД, предназначенные для хранения информации.

Запросы – объекты, которые применяются для ведения БД. Запросы позволяют извлечь нужные данные из одной или нескольких таблиц, представить их в удобном виде. С помощью запросов выполняют отбор данных, их сортировку и фильтрацию. С помощью запросов можно выполнять простейшие вычисления в таблицах, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными, импортированными из других источников, выполнять преобразование данных по заданному алгоритму и др.

Формы – объекты, предназначенные для удобного ввода и просмотра данных, а также для управления работой приложения.

Отчеты – объекты, предназначенные для формирования печатной формы.

Страницы доступа к данным – объекты, хранящиеся в отдельных HTML-файлах, для просмотра, редактирования и обработки данных в интерфейсе WEB-браузера.

Макросы – объекты БД, представляющие собой записанные без программирования наборы макрокоманд для автоматизированного выполнения часто требуемой последовательности действий.

Модули – объекты в виде программ, написанных на языке программирования Visual Basic for Application (VBA), для автоматизации нестандартных процедур обработки БД.

1.12. Файл БД MS Access

Особенностью БД, созданной в СУБД Access, является то, что все объекты БД, за исключением страниц доступа к данным, Microsoft Access может хранить в одном файле – файле БД с расширением **.mdb** в версии **Access 2003** или с расширением **.accdb** в версиях **Access 2007+**.

При этом отчеты можно экспортировать в файлы Microsoft Excel, текстовые файлы, файлы формата .pdf и .xps.

С целью защиты от модификации форм, отчетов и модулей база данных может быть сохранена в файле приложения с расширением **.mde** версии **Access 2003** или **.accde**

в версиях **Access 2007+**. При этом БД сжимается, оптимизируется использование памяти и повышается быстродействие БД.

1.13. Интерфейс приложения MS Access


Интерфейс приложения MS Access 2003 состоит из следующих компонентов:

строка заголовка – содержит кнопку системного меню, название приложения (Microsoft Access) и три кнопки, управляющие представлением окна на экране;

строка меню – содержит команды, каждая из которых имеет собственное ниспадающее меню. Команды меню **Файл, Правка, Вид, Вставка, Сервис, Окно, Справка** определяют операции, позволяющие пользователю выполнять необходимую работу. Содержание строки меню и набор операций изменяются в зависимости от средства, выбираемого пользователем для работы;

панель инструментов – состоит из графических пиктограмм, каждая из которых реализует часто используемую команду меню более быстрым и удобным способом;

окно БД – предназначено для выполнения всех операций по обработке объектов БД

В главном окне БД в Access 2003 представлены все основные режимы работы с объектами БД соответствующими кнопками на панели инструментов окна БД ( (рис. 4).

Открыть – позволяет открыть таблицу в режиме таблицы для редактирования и ввода новых записей, выполнить запрос, открыть форму, сформировать отчет, запустить макрос на выполнение.

Конструктор – обеспечивает переход в режим настройки структуры выбранного объекта.

Создать – открывает окно для выбора способа создания нового объекта текущего вида.

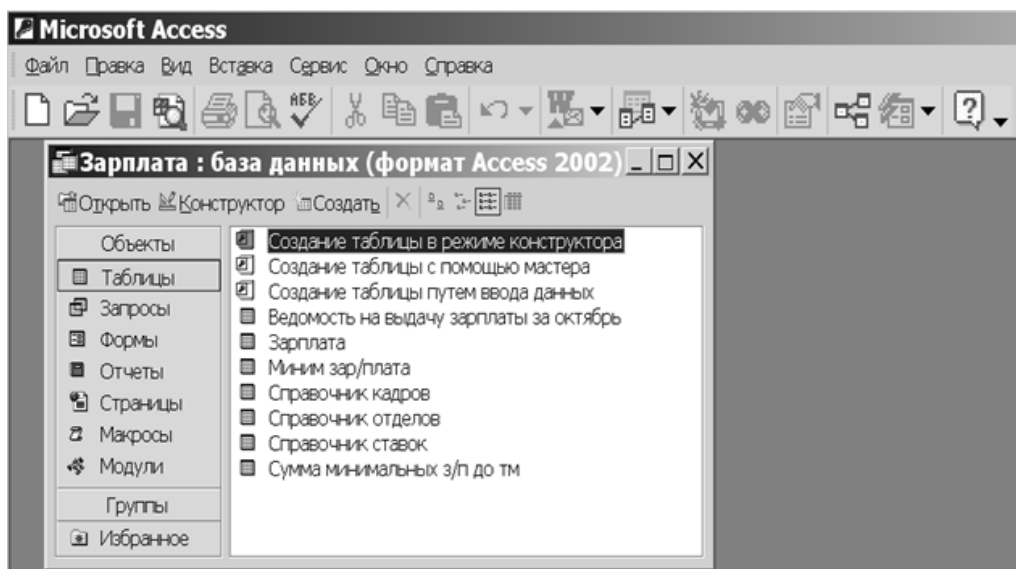




Рисунок 4 – Окно БД в интерфейсе MS Access 2003

Начиная с версии 2007 в пользовательском интерфейсе MS Access произошли существенные изменения. **Интерфейс приложения MS Access 2007+** включает следующие компоненты:

главное меню – открывается кнопкой Office  и включает команды *Открыть*, *Сохранить* и *Печать* файла;

панель быстрого доступа  включает кнопки для команд, которые обычно требуются чаще всего: *Сохранить*, *Отменить* и *Вернуть*. Панель можно *настраивать* и добавлять любые необходимые команды;

лента – это широкая полоса, расположенная в верхней части окна, на которой расположены группы команд, размещенные на разных вкладках.

Основные **вкладки ленты**: *Главная*, *Создание*, *Внешние данные* и *Работа с базами данных*, *Надстройки* (рис. 5).

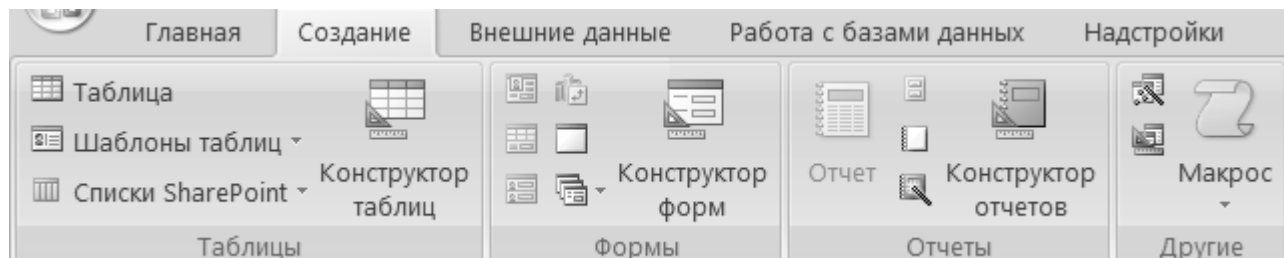



Рисунок 5 – Лента с группами команд и вкладками в интерфейсе MS Access 2007+

область переходов – это панель навигации для работы с объектами БД, которые сгруппированы по видам: *Таблицы*, *Запросы*, *Формы*, *Отчеты*, *Макросы*, *Модули*. Группы объектов в **области переходов** могут при необходимости разворачиваться или сворачиваться кнопкой  (рис. 6).

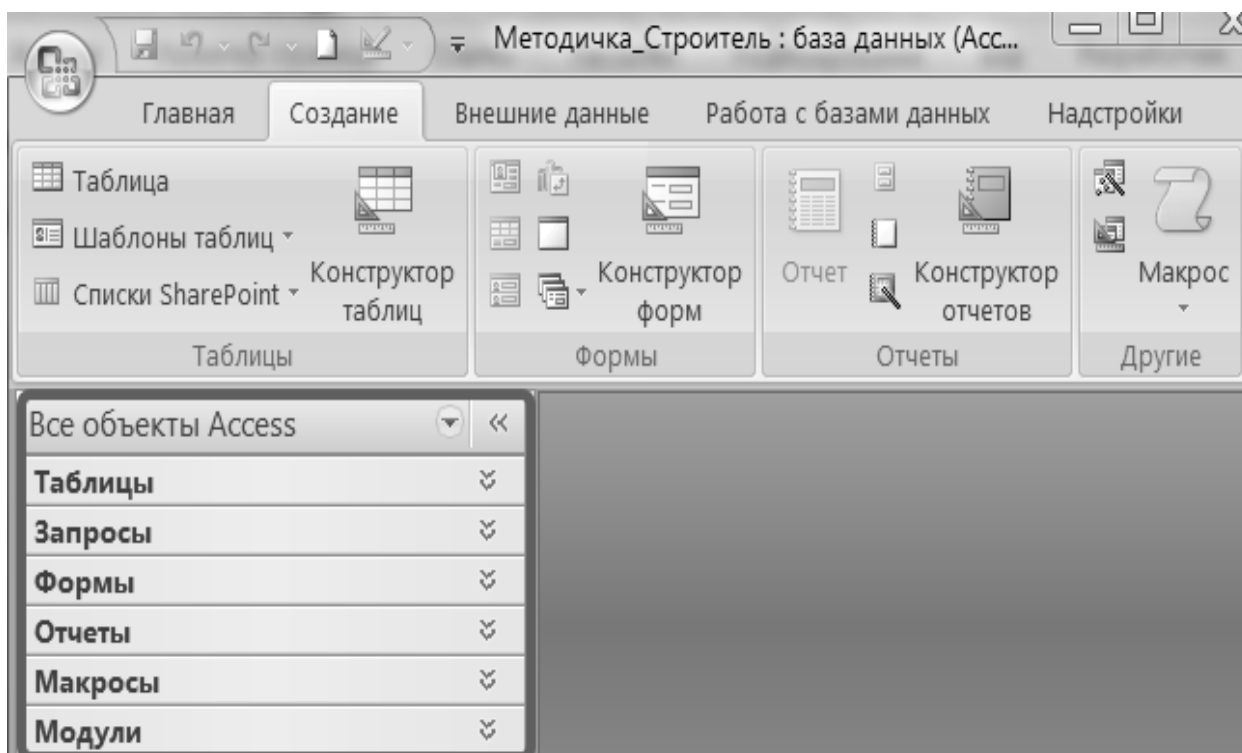


Рисунок 6 – Область переходов в интерфейсе MS Access 2007+


Для версий Access 2007+ эти команды для работы с объектами представлены в ленте на вкладке **Главная** в виде пиктограмм в группе **Режимы** (рис. 7) и на вкладке **Создание**.



Рисунок 7 – Режимы работы с объектами в ACCESS 2007+

Чтобы открыть объект БД (например, таблицу, форму или отчет) в Access 2007+, можно воспользоваться одним из способов:

- выполнить двойной щелчок левой кнопкой мыши на объекте в области переходов;
- выделить объект в области переходов и нажать клавишу **Enter**;
- в области переходов щелкнуть объект *правой кнопкой мыши* и выбрать в контекстном меню команду **Открыть**.

Вдоль нижней границы окна Access 2007+ может отображаться **строка состояния**, которая предназначена для вывода сообщений, например, о состоянии объекта. А имеющиеся справа в строке состояния элементы управления  позволяют быстро переключать различные **режимы** активного объекта.

1.14. Способы создания таблиц в MS Access

Таблицу в Access можно создать несколькими способами:

- построить новую «с нуля» непосредственно в **режиме таблицы**;
- спроектировать, воспользовавшись **Конструктором таблиц**;
- запустить **Мастер таблиц** – помощник, предлагающий создать таблицу в пошаговом режиме на базе типовых решений, имеющихся в Access;
- **импортировать** таблицу БД из файла какой-либо программы, например, текстового редактора или табличного процессора Microsoft Excel.

Записи одной таблицы могут содержать ссылки на данные другой таблицы, например в таблице со списком товаров могут храниться ссылки на справочник производителей товаров с их адресами и другими реквизитами. При этом записи, касающиеся разных товаров, могут указывать на одного и того же производителя. Такая взаимозависимость таблиц обеспечивается связью таблиц.

1.15. Понятие запроса. Виды запросов в MS Access

Запрос (query) есть механизм выбора и представления информации из базы данных. Запрос направляется пользователем или программой в СУБД для поиска отдельных записей в БД. Вопрос, сформированный по отношению к БД, и есть запрос.

Access позволяет создавать запросы на выборку и активные запросы. Они могут быть однотабличными (в качестве исходной информации используются данные одной таблицы) и многотабличными (в качестве исходной информации используются данные нескольких таблиц).

Запрос на выборку – это наиболее часто используемый тип запроса, который задает вопрос базе данных и отображает полученный ответ в виде динамического набора данных. Среди запросов на выборку можно выделить следующие разновидности:

- *простой запрос на выборку* – отбирает данные из взаимосвязанных таблиц или других запросов с учетом условий пользователя. Результатом является таблица, которая существует на экране, пока запрос не закрыт;
- *запрос с групповыми операциями (итоговый запрос)* – выполняет вычисления с использованием данных по группам записей;
- *запрос с параметром (параметрический запрос)* – это запрос, в котором критерий отбора или элемент формулы пользователь может задать с клавиатуры путем ввода нужного значения параметра при вызове запроса;
- *запрос с вычислениями* – позволяет выполнить вычисление для нового поля на основе данных полей текущей таблицы, запроса, других таблиц БД.

Перекрестный запрос обеспечивают создание результирующей таблицы в виде сводной таблицы по группам записей и при необходимости с выводом итогов. Таким образом с помощью данного перекрестного запроса можно сгруппировать большой объем информации и представить его в удобном для восприятия виде.

Запросы-действия вносят изменения в саму БД. Среди них выделяют следующие:

- *запросы на создание таблицы* – создают новую таблицу БД, в которой сохраняется результат запроса;
- *запросы на обновление* – изменяют информацию в полях таблицы БД;
- *запросы на добавление* – производят добавление записей из таблицы с результатами запроса в таблицу БД;
- *запросы на удаление* – удаляют группу записей из одной таблицы или нескольких взаимосвязанных таблиц базы, для которых задано каскадное удаление связанных записей.

SQL-запросы – формирование запросов с помощью последовательности SQL-инструкций специального структурированного языка запросов SQL (Structured Query Language), в которые можно включать выражения и вызов итоговых функций.

1.16. Формы и отчеты как объекты БД MS Access

Понятие **Форма** в БД означает структурированное экранное окно либо независимый элемент интерфейса с заранее установленными областями для ввода либо изменения информации; визуальный шаблон, который упорядочивает предоставляемые формой данные, что позволяет лучше их организовывать и просматривать.

Как объект БД форма используется:

- для ввода данных в таблицу, их корректировки;
- в качестве специального окна диалога для выбора;
- в виде кнопочной формы для открытия других объектов БД.

В форме может быть представлена информация из нескольких связанных таблиц. Представление данных в формах может быть различным: поля располагаются в виде столбца или таблицы.

Источником данных для формы могут являться записи таблицы либо запроса. Форма облегчает восприятие информации – это гибкий способ ее представления.

Инструментарий для форм ничем не отличается от табличного: данные в формах можно искать, заменять, сортировать, фильтровать, словом, делать с ними то же самое, что и в таблицах.

Отчеты в БД создаются для подготовки информации, содержащейся в таблицах и запросах, к выводу на печать. На этапе создания отчета над данными можно выполнить некоторые дополнительные действия: группировку и сортировку данных, а также произвести необходимые вычисления. Подготовка и автоматизированное формирование разнообразных отчетов – это основная цель ведения БД.

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

Практическое задание 1. Создание файла БД Магазин СТРОИТЕЛЬ.

Запустить СУБД Microsoft Access – из меню **Пуск** или с помощью ярлыка.

Для версии MS Access 2003:

- ✓ в окне **Microsoft Access** установить переключатель в положение **Новая база данных** и нажать кнопку **ОК**;
- ✓ в появившемся окне **Файл новой БД** указать, где создать файл (например, на рабочем диске локальной сети или на другом диске) и с каким именем (в нашем случае – **Магазин СТРОИТЕЛЬ**);
- ✓ нажать кнопку **Создать**.

После создания нового файла БД или открытия существующего в рабочей области окна приложения появляется **окно базы данных**, пример которого представлен на рисунке 4.

Для версий MS Access 2007+:

- ✓ Запустить СУБД Microsoft Access из меню **Пуск** или с помощью ярлыка. Отобразится меню создания БД Access.
- ✓ В окне раздела «Новая пустая база данных» (или в окне «Создать» раздела «Доступные шаблоны») выбрать команду **Новая база данных**.
- ✓ В области «Новая база данных» в поле *Имя файла* ввести имя файла БД (в нашем случае – **Магазин СТРОИТЕЛЬ**), указать путь к папке, где он должен быть сохранен.
- ✓ Нажать кнопку **Создать**.

При создании файла новой БД автоматически откроется окно создания новой таблицы в режиме таблицы (рис. 8).

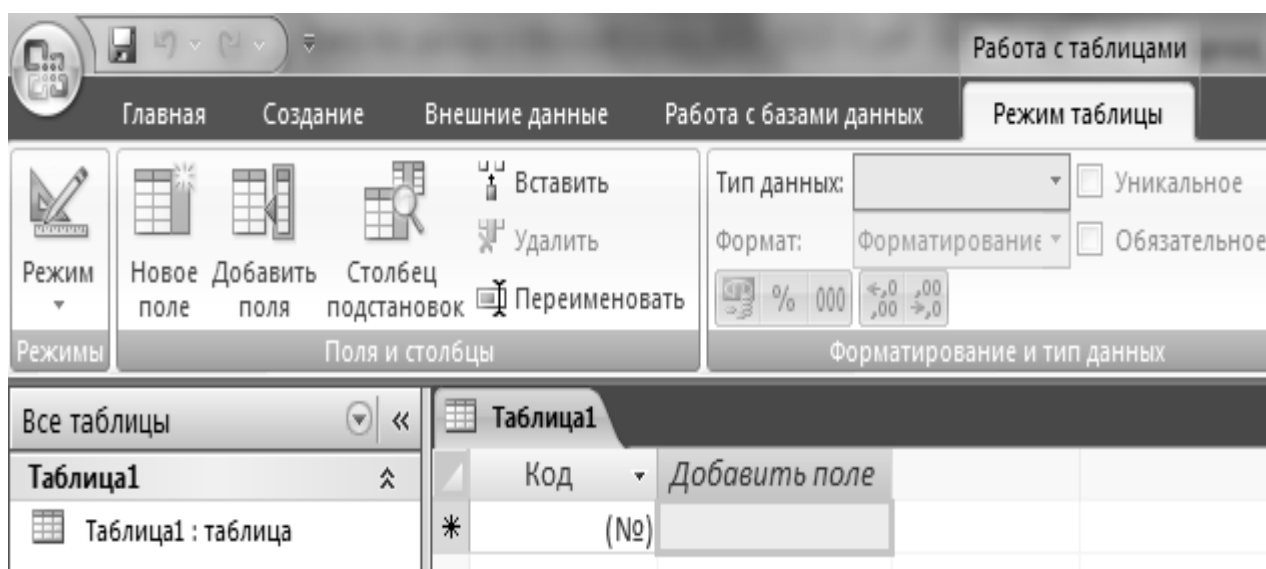


Рисунок 8 – Новая таблица в MS ACCESS 2007+

Практическое задание 2. Создание таблиц БД Магазин СТРОИТЕЛЬ

Чтобы создать таблицу в Access 2003, нужно перейти к списку таблиц в окне БД и нажать на панели инструментов кнопку **Создать**. Появится диалоговое окно *Новая таблица*, в котором необходимо выбрать нужный способ создания таблицы (рис. 9).

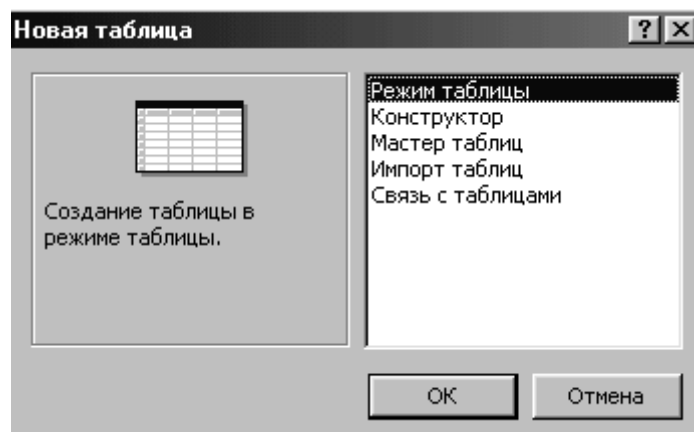
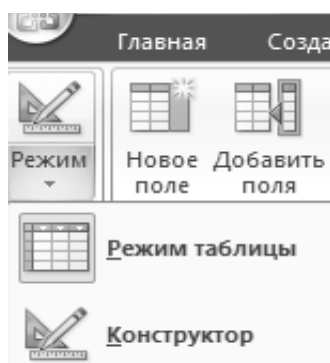


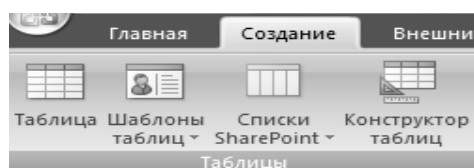
Рисунок 9 – Способы создания таблицы в MS Access 2003

Создание новой таблицы в версиях Access 2007+ осуществляется автоматически при создании файла БД. Изменить или выбрать способ проектирования таблицы можно с помощью следующих инструментов ленты:

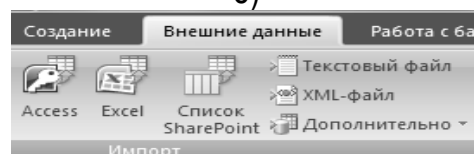
- ✓ на вкладке **Главная** в группе **Режимы** (рис. 10-а);
- ✓ на вкладке **Создание** в группе **Таблицы** (рис. 10-б);
- ✓ на вкладке **Внешние данные** в группе **Импорт** (рис. 10-в);



а)



б)



в)

Рисунок 10 – Способы создания таблицы в MS ACCESS 2007+

2.1. Создание таблицы с помощью Конструктора таблиц

Если для создания таблицы выбран режим **Конструктора**, то появляется окно с названием **Таблица1**, в котором определяется структура таблицы БД (рис. 10).

Создать таблицу БД – это значит задать структуру ее записи, то есть перечислить формирующие ее поля, определить тип данных их значений и указать необходимые свойства.

Окно **Конструктора таблицы** имеет две области:

Область полей таблицы. В столбце **Имя поля** вводятся названия полей. Имя поля должно быть смысловым и кратким, хотя и может содержать до 64 знаков. Имя поля не должно начинаться с пробела и при этом допустимы любые символы, за исключением: точки (.), восклицательного знака (!), квадратных скобок ([]) и апострофа ('). Повторение имен полей в таблице не допускается. Для каждого поля в графе **Тип данных** определяется тип вводимой информации путем выбора из списка (рис. 11). В графе **Описание** могут быть записаны комментарии.

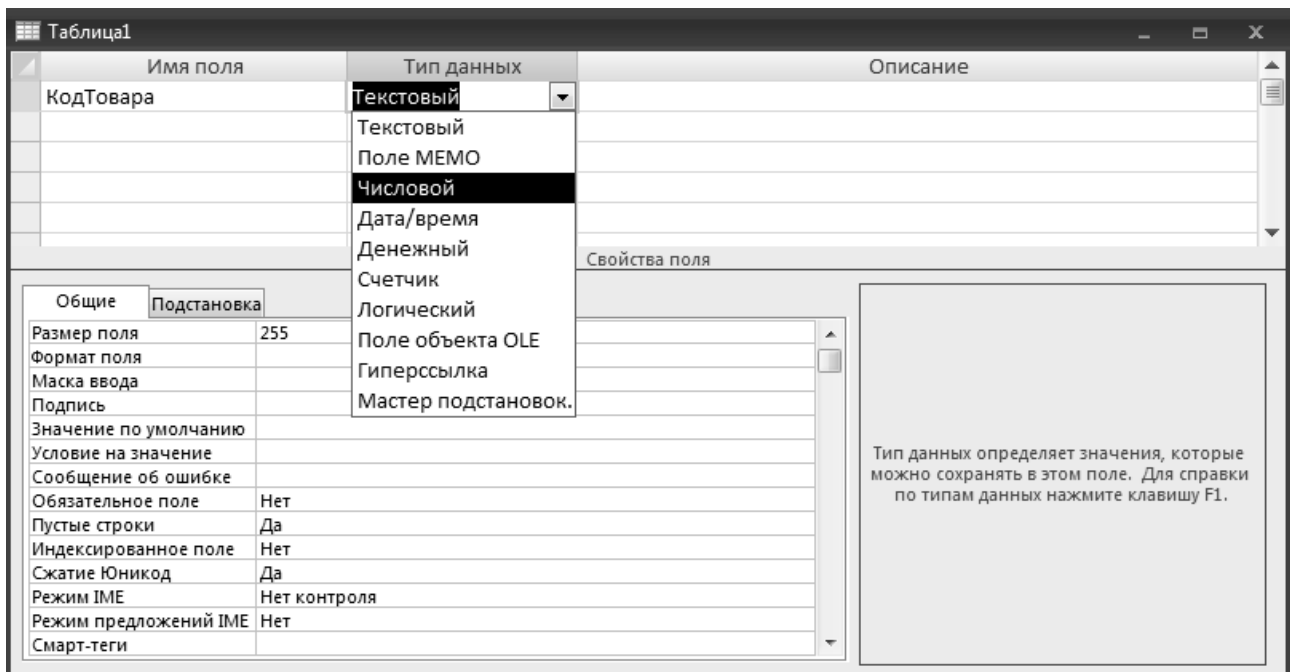


Рисунок 11 – Окно таблицы в режиме Конструктора с выбором типа данных

Область свойств поля. Здесь указываются отдельные характеристики каждого поля, необходимые для определения параметров сохранения данных в поле, их дальнейшего отображения и редактирования. Эта область содержит две вкладки: **Общие** и **Подстановка**. Вкладка **Подстановка** содержит список некоторых дополнительных параметров, необходимых, в частности, для настройки связей с полями других таблиц.

2.2. Типы данных в MS Access

СУБД MS Access допускает хранить информацию в БД следующих типов:

Тип данных	Содержимое
Текстовый	Используется для буквенно-символьной (текстовой) информации, не применяемой в вычислениях, длиной до 255 знаков.
Поле Мемо	Специальный тип для хранения больших текстов (от 255 до 65 535 знаков). Свойство поля МЕМО указывает, поддерживается ли форматирование текста.
Числовой	Применяется для хранения числовых значений (целых или дробных), которые используются в вычислениях (за исключением денежных сумм).
Дата / время	Используется для хранения календарных дат и времени.
Денежный	Тип данных для хранения денежных значений и предотвращения округления во время вычислений. Максимальная точность денежных сумм составляет 4 знака после запятой и 15 в целой части.
Счетчик	Применяется для автоматического заполнения поля последовательными номерами с шагом 1 при добавлении новой записи в таблицу. При удалении записей значения в полях с таким типом не обновляются. Может использоваться для ключевого поля таблицы.
Логический	Используется для логических полей-признаков, которые могут принимать только одно значение из пары значений, таких как: Да/Нет, Истина/Ложь или Вкл/Выкл.
Поле объекта OLE	Специальный тип данных, предназначенный для хранения объектов, созданных в других приложениях (диаграммы, картинки и др.), вставляемых внедрением или связыванием (OLE). Такие данные в таблице не хранятся, хранятся только указатели на них.

Тип данных	Содержимое
Гиперссылка	Позволяет хранить путь к файлу на жестком диске либо адрес в сети Интернет или Интранет, в том числе ссылки на объекты приложения Access, которые хранятся в БД.
Вложение (в Access 2007+)	Используется для хранения файлов (фотографий, изображений или файлов созданных с помощью приложений MS Office), аналогично вложениям электронной почты. Вложенные файлы можно просматривать и редактировать в соответствии с заданными для поля свойствами. Такой тип поля более рационально используют место для хранения, чем тип <i>Поле объекта OLE</i> , поскольку не создает растровые изображения исходного файла. Максимальная длина поля для сжатых вложений – 2 Гб, для несжатых – примерно 700 Кб
Вычисляемый (в Access 2010+)	Предназначен для создания вычисляемых полей: числовых, текстовых, денежных, дата/время, логических. Значение вычисляемого поля определяется выражением, записанным в поле и использующим другие поля текущей записи, некоторые встроенные функции и константы, связанные арифметическими, логическими или строковыми операторами
Мастер подстановок	Не является типом данных. Служит для запуска Мастера подстановок для создания поля с открывающимся списком для выбора значения поля из другой таблицы, запроса или фиксированного набора значений. Мастер подстановок может устанавливать связи между таблицами при их отсутствии.

2.3. Описание свойств полей

Каждое поле имеет определенный набор свойств **в зависимости от типа поля**. Некоторые типы полей имеют схожие наборы свойств полей.

Основные свойства полей:

Свойство поля	Применение свойства
Размер поля	Максимальная длина текстового поля (по умолчанию 50 знаков) или тип данных числового поля. Рекомендуется задавать минимально допустимое значение этого свойства, потому что обработка данных меньшего размера выполняется быстрее.
Формат поля	Задается формат представления данных при вводе на экран и печать.
Число десятичных знаков	Определяет число десятичных знаков при отображении числа – от 0 до 15 (по умолчанию 2 знака).
Новые значения	Указывает, какие значения (последовательные или случайные) присваиваются полю с типом данных «Счетчик» при добавлении новой записи.
Маска ввода	Символы редактирования, определяющие способы ввода данных. Содержит строковые константы и символы шаблона.
Подпись поля	Можно определить более описательное имя поля, которое будет отображаться в режиме таблицы и в качестве подписи в формах, отчетах и запросах. Если это значение не определено, в качестве подписи будет взято имя поля.
Значение по умолчанию	Значение, подставляемое в поле автоматически при добавлении новой записи.
Условие на значение	Задаёт ограничения на вводимые значения, тем самым позволяет осуществлять контроль над правильностью ввода данных.
Сообщение об ошибке	Текст, появляющийся при вводе значения, при котором нарушается условие, указанное в свойстве <i>Условие на значение</i>
Обязательное поле	Определяет, может ли данное поле содержать значения Null (т.е. оставаться пустым), или нужно обязательно вводить в это поле данные.

Свойство поля	Применение свойства
Пустые строки	Разрешает (установкой значения Да) ввод строки нулевой длины ("") в текстовом поле и поле МЕМО.
Индексированное поле	Посредством создания и использования индекса ускоряется доступ к этому полю для чтения.
Сжатие Юникод	Выполняется сжатие текстовых данных поля, если поле содержит менее 4096 символов. Если больше, то сжатия не будет.
Режим IME	Управление преобразованием символов в восточноазиатских версиях Windows.
Режим предложений IME	Управление преобразованием предложений в восточноазиатских версиях Windows.
Смарт-теги	Позволяет выбрать смарт-теги, применяемые к полю.
Только добавление	Если задано значение Да, ведется журнал значения поля.
Выравнивание текста	Выравнивание текста по умолчанию в элементе управления.
Формат текста	Для хранения данных в виде HTML-кода и использования RTF-форматирования следует выбрать значение <i>Формат RTF</i> . Чтобы хранить текст без форматирования, следует выбрать значение <i>Обычный текст</i> .
Отображать элемент выбора даты	Указывает, должен ли в приложении Access отображаться элемент выбора даты (элемент управления "Календарь"), когда пользователь изменяет значение поля.

Задача: для БД **Магазин СТРОИТЕЛЬ** в режиме **Конструктора таблиц** создать таблицы **ГруппыТоваров**, **Товары**, **Покупатели**, **УчетПродаж**. Структуры записей таблиц приведены на рисунке 12 (описание типов данных полей и таблиц – в **Приложении 1**).

ГруппыТоваров	Товары	Покупатели	УчетПродаж
<u>КодГруппыТовара</u>	<u>КодТовара</u>	<u>КодПокупателя</u>	<u>НомерПродажи</u>
НаименованиеГруппы	НаименованиеТовара	ИмяПокупателя	<u>ДатаПродажи</u>
Категория	ГруппаТовара	Адрес	<u>Покупатель</u>
	УчетнаяЕдиница	Телефон/Факс	<u>Товар</u>
	Цена(руб)		КоличествоТовара
	ДатаИзготовления		СуммаПредоплаты(руб)
	СрокГодности(лет)		ПризнакОтгрузки
	ДатаПоступления		ДатаОплаты
	Производитель		
	ОписаниеТовара		

Рисунок 12 – Состав таблиц БД Магазин СТРОИТЕЛЬ с выделенными ключевыми полями

Порядок выполнения:

- Для версии Access 2003: в окне БД выбрать объект **Таблицы**, далее п. м. **Создать**, в окне **Новая таблица** выбрать режим **Конструктор** (рис. 9);
- В графу **Имя поля** ввести имя формируемого поля. Нельзя использовать в имени точку, восклицательный знак и квадратные скобки. Нельзя начинать имя поля с пробела.
- В графе **Тип данных** выбрать из списка требуемый тип данных.
- В графу **Описание** ввести текст, поясняющий назначение и использование данного поля таблицы (Заполнять не обязательно!).
- В нижней части окна **Свойства поля** задать требуемые свойства каждого поля.

Замечание: для создания каждой следующей таблицы в версии Access 2007+ необходимо активизировать на ленте вкладку **Создание** и в группе инструментов **Таблицы** выбрать пиктограмму **Конструктор таблиц** (рис. 10-б).

2.4. Формирование маски ввода

Маска ввода определяет способы ввода данных в таблицу. Последовательность её создания:

- ✓ выбрать строку **Маска ввода** в области *Свойства поля* и нажать кнопку **Построить** (кнопка с тремя точками (...)), при этом подтвердить сохранение таблицы;
- ✓ в открывшемся окне **Мастер создания масок** (или **Создание Масок ввода**) выбрать из предлагаемого списка необходимую маску, если она есть в списке, например, для поля с типом данных *Дата/Время* можно выбрать маску **Краткий формат даты** (рис. 13);

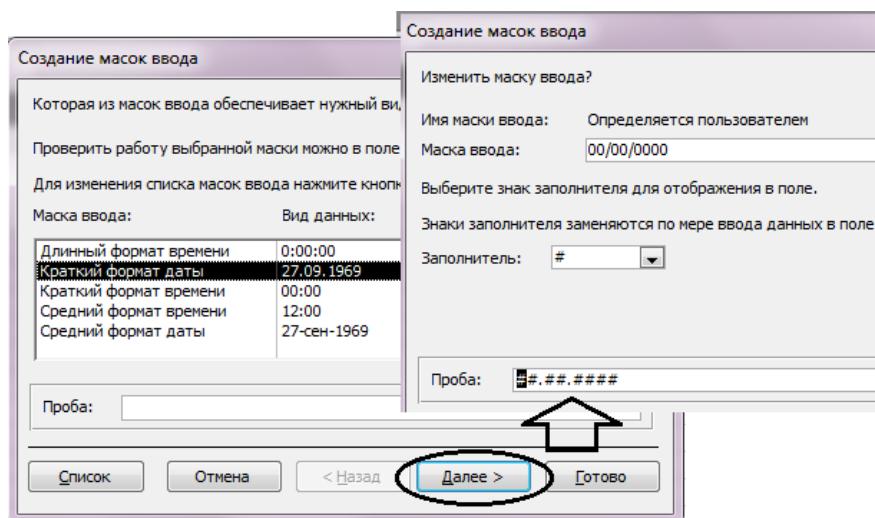


Рисунок 13 – Окна работы с Маской ввода для поля с типом данных *Дата/Время*

- ✓ если необходимой маски в списке нет, нажать кнопку **Список**. Откроется окно **Настройка масок ввода** (рис. 14);

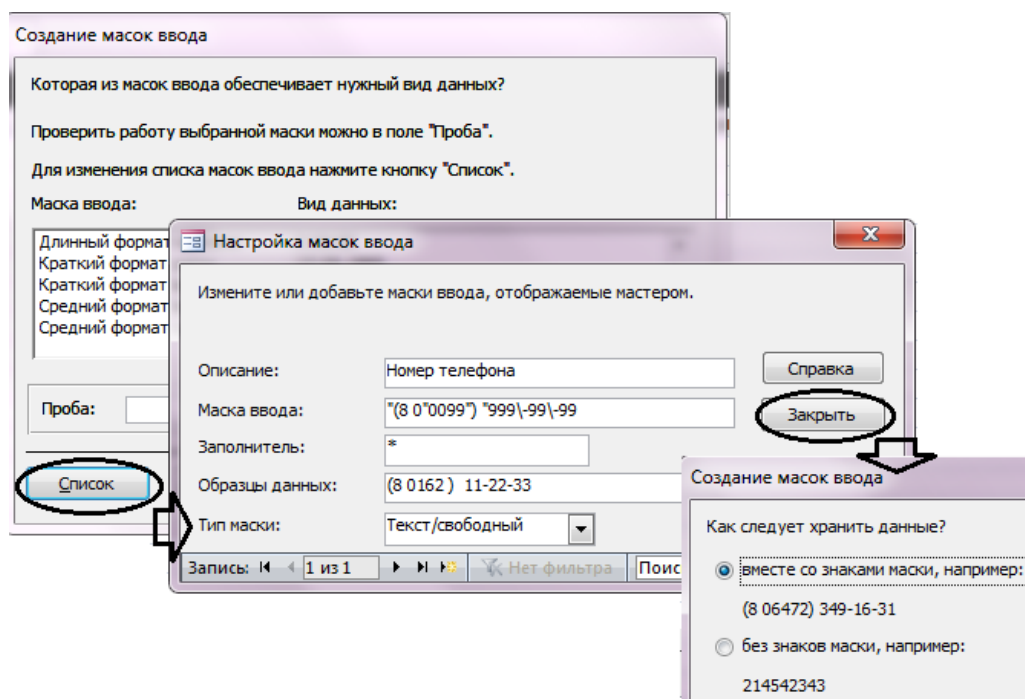


Рисунок 14 – Окна работы с Маской ввода для поля с тестовым типом данных

✓ в поле *Описание* ввести название создаваемой маски ввода; в поле *Маска ввода* ввести собственно саму маску ввода; в поле *Заполнитель* выбрать заполнитель символов маски (по умолчанию – знак нижнего подчеркивания); ввести образец данных в соответствующее поле (в качестве примера для пользователя) и выбрать *Тип маски*. Нажать кнопку **Заккрыть**;

✓ в окне **Мастер создания масок** появится только что созданная маска. Нажать кнопку **Далее**. В поле *Проба* ввести данные для проверки действия маски. После этого – кнопки **Далее** и **Готово**.

В случае необходимости маска может быть построена без помощи мастера (см. **Приложение 2**).

2.5. Свойство поля «Значение по умолчанию»

Например, в таблице **УчётПродаж** для автоматического заполнения поля *ДатаПродажи* значением текущей даты необходимо для свойства *Значение по умолчанию* использовать встроенную функцию **Date()**. Можно воспользоваться **Построителем выражений** (см. раздел 2.15. практической части).

2.6. Использование Мастера подстановки для создания фиксированного списка значений

Список значений для выбора в поле таблицы создается для удобства его заполнения и предотвращения случайных синтаксических ошибок при вводе одного и того же значения, которое затем может использоваться в критериях отбора. Порядок создания списка значений для поля *Категория* таблицы **ГруппыТоваров** следующий:

Открыть таблицу **ГруппыТоваров** в режиме *Конструктора*, установить курсор в строке *Категория* в столбец **Тип данных** и из раскрывающегося списка выбрать **Мастер подстановок** (рис. 15).

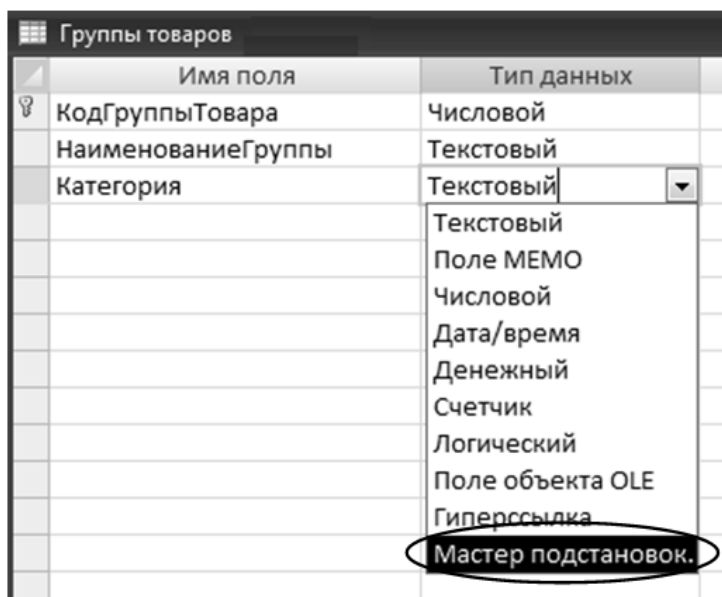


Рисунок 15 – Выбор инструмента Мастер подстановок

1) В первом диалоговом окне **Создание подстановки**, необходимо выбрать каким способом столбец подстановки будет получать значения для выбора. В данном случае необходимо выбрать вариант создания фиксированного набора значений (рис.16). Нажать кнопку **Далее**.

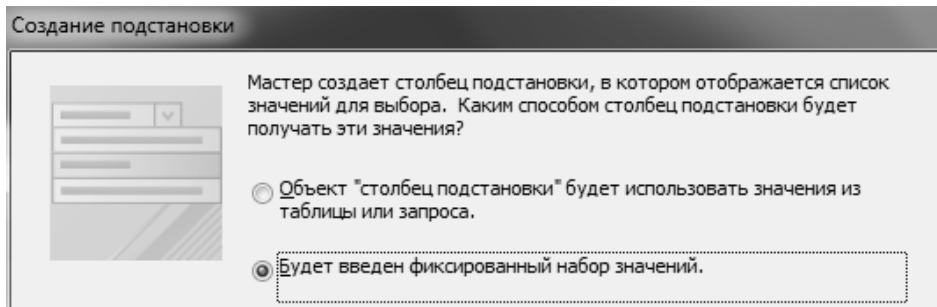


Рисунок 16 – Первый шаг Мастера подстановок

2) Во втором диалоговом окне **Создание подстановки** следует ввести в столбец данные для списка значений (рис.17). Нажать кнопку **Далее**.

3) В последнем диалоговом окне можно оставить всё без изменений, соглашаясь с заданной подписью столбца подстановки и нажать кнопку **Готово**.

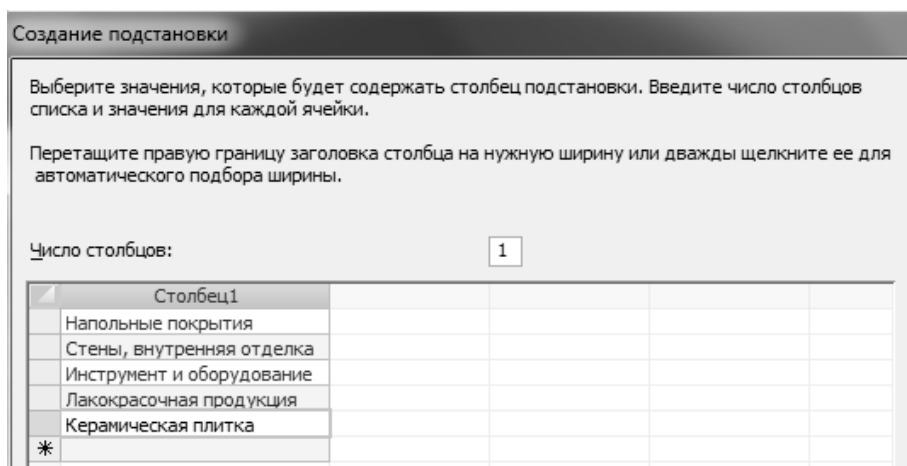


Рисунок 17 – Второй шаг Мастера подстановок

Столбец фиксированных значений можно создавать "вручную" в нижней части бланка Конструктора таблицы на вкладке **Подстановка**: в строке **Тип элемента управления** выбрать опцию "Поле со списком", в строке **Тип источника строк** – опцию "Список значений", а в строке **Источник строк** ввести элементы списка в кавычках через точку с запятой (рис. 18).

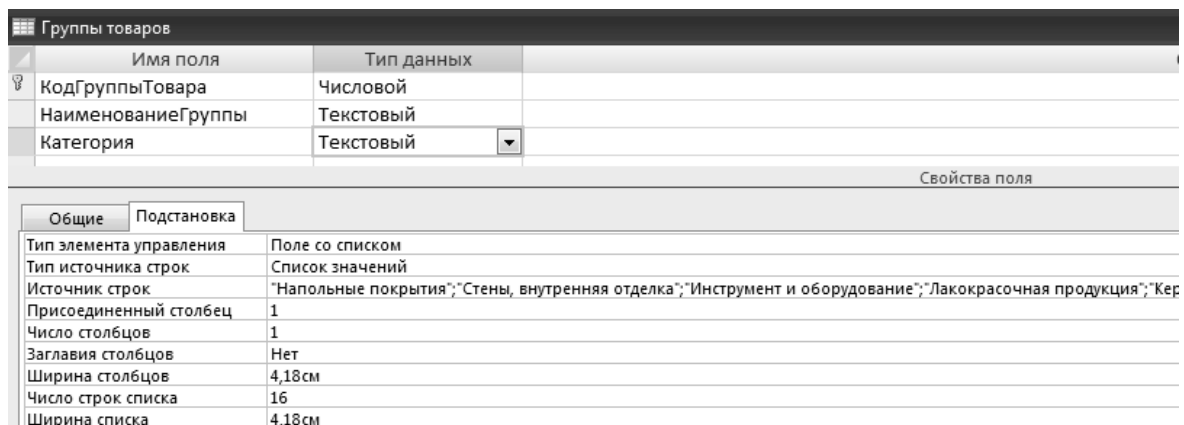


Рисунок 18 – "Ручное" создание фиксированного столбца значений

В результате создания такого списка, в режиме таблицы при переходе к данному полю для заполнения в правой части ячейки появится элемент управления "Список", с помощью которого можно выбрать значение, заносимое в данное поле (рис. 19).

Группы товаров			
КодГруппы	НаименованиеГруппы	Категория	
5	Ламинат 32 класс	Напольные покрытия	▼
6	Ламинат 33 класс	Напольные покрытия	
7	Паркетная доска	Стены, внутренняя отделка	
8	Линолеум	Инструмент и оборудование	
10	Керамическая плитка	Лакокрасочная продукция	
11	Керамогранит	Керамическая плитка и керамогранит	
14	Краска водно-дисперсионная	Лакокрасочная продукция	

Рисунок 19 – Выпадающий список значений в режиме таблицы

Аналогично создаются фиксированные списки значений для полей *УчетнаяЕдиница* и *Производитель* таблицы **Товары**.

2.7. Задание первичного ключа

Понятие ключевого поля (полей) рассмотрено в разделе 1.4. теоретической части.

Для задания ключевого поля в таблице необходимо:

1) в режиме **Конструктора таблицы** с помощью указателя мыши выделить поле, которое должно использоваться в качестве ключа (при определении составного ключа, необходимо одновременно выделить все поля, входящие в состав ключа, при нажатой клавише **CTRL**);

2) для версии Access 2003: нажать на панели инструментов кнопку **Ключевое поле**;


2) для версии Access 2007+: на вкладке ленты **Конструктор** в группе **Сервис** выбрать пиктограмму **Ключевое поле**.

Замечание: можно воспользоваться соответствующей командой из контекстного меню для выделенного поля.

2.8. Сохранение таблицы

После создания структуры таблицы в режиме Конструктора, её необходимо сохранить.

В MS Access 2003 необходимо нажать кнопку **Сохранить** на панели инструментов или соответствующую команду в п. м. **Файл** и ввести название таблицы, после чего на экране появляется вопрос «Создать ключевое поле сейчас?» (Да или Нет). Если выбирается ответ «Да», то Access создаст автоматически поле с именем «Код» и типом данных *Счетчик*, если «Нет», – то таблица будет создана без ключевого поля. В этом случае необходимо открыть созданную таблицу в режиме Конструктора и определить «вручную» ключевое поле.

В MS Access 2007+ запрос на сохранение таблицы происходит при переключении её из табличного вида в вид Конструктора. При пересохранении изменений в её структуре, можно выбрать команду **Сохранить** в главном меню, вызываемом кнопкой «Office» .

Практическое задание 3. Установка связей между таблицами БД

Задача: разработать схему данных для таблиц БД *Магазин СТРОИТЕЛЬ* путём установки связей между ними (рис. 20).

Следует обратить внимание на то, при вводе данных в поля *Покупатель* и *Товар* таблицы **УчетПродаж** значения берутся из соответствующих полей *КодПокупателя* таблицы **Покупатели** и *КодТовара* таблицы **Товары**. Аналогично, при заполнении поля *ГруппаТовара* таблицы **Товары** используются данные поля *КодГруппыТовара* из таблицы **ГруппыТоваров**. Для того, чтобы заполнение указанных полей было удобным, необходимо воспользоваться инструментом **Мастер подстановок**.

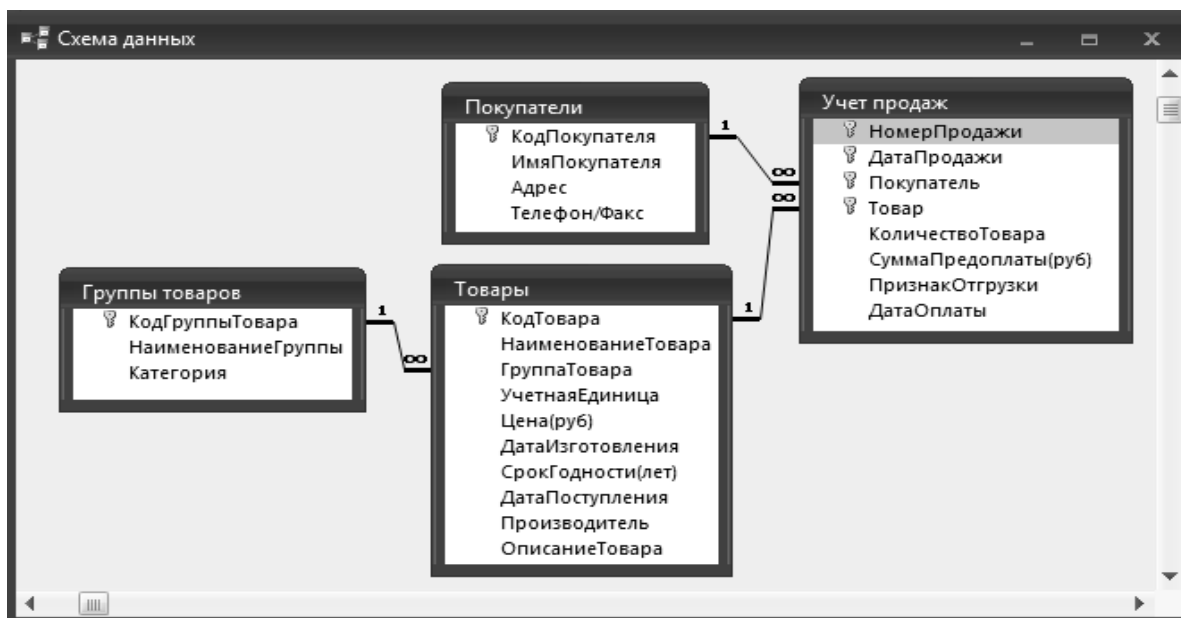


Рисунок 20 – Схема данных в БД Магазин СТРОИТЕЛЬ

2.9. Использование Мастера подстановки для создания связей между таблицами

Открыть таблицу **УчетПродаж** в режиме *Конструктора*, установить курсор в строку **Покупатель** в графу **Тип данных** и из раскрывающегося списка выбрать **Мастер подстановок**.

Последовательно действовать по шагам **Мастера**:

1) В первом диалоговом окне **Создание подстановки**, необходимо выбрать каким способом столбец подстановки будет получать значения для выбора. В данном случае можно оставить вариант по умолчанию, т. е. выбор из таблицы или запроса (рис. 21). Нажать кнопку **Далее**.

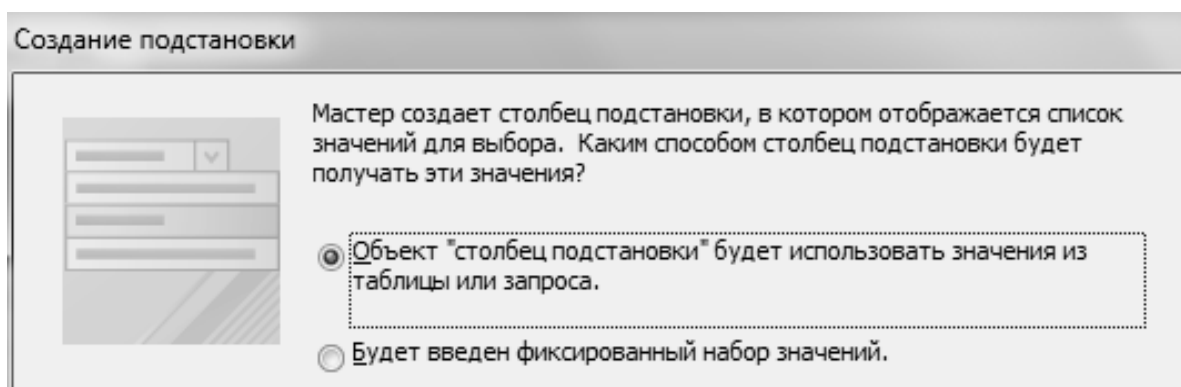


Рисунок 21 – Первый шаг Мастера подстановок

2) Во втором диалоговом окне **Создание подстановки** следует выбрать таблицу **Покупатели** (рис. 22). Нажать кнопку **Далее**.

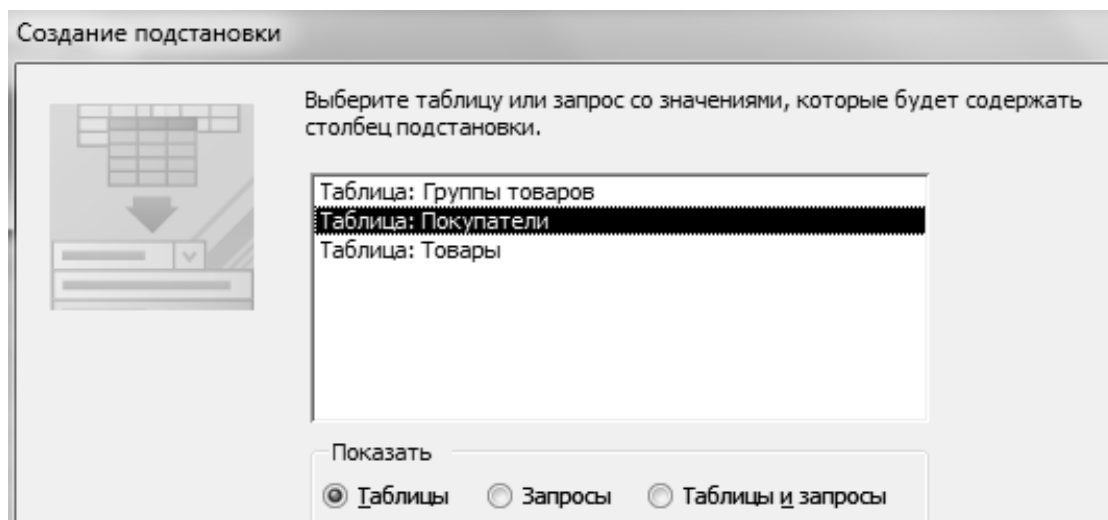


Рисунок 22 – Второй шаг Мастера подстановок

3) В третьем диалоговом окне необходимо отобразить отдельные поля таблицы с помощью кнопки **>** (либо все поля таблицы с помощью кнопки **>>**). В нашем случае выбираются поля **КодПокупателя**, **ИмяПокупателя** и **Адрес** (рис. 23). Нажать кнопку **Далее**.

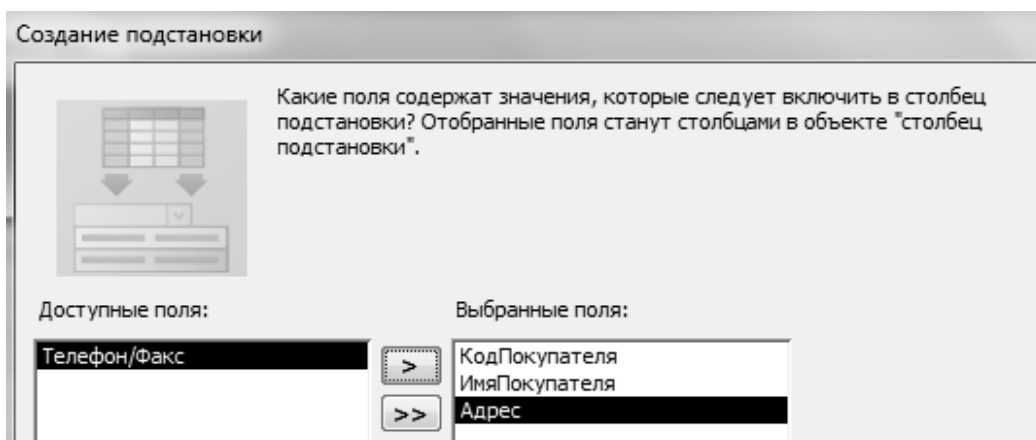


Рисунок 23 – Третий шаг Мастера подстановок

4) В следующем диалоговом окне можно указать способ сортировки элементов списка выбираемых значений (рис. 24). Нажать кнопку **Далее**.

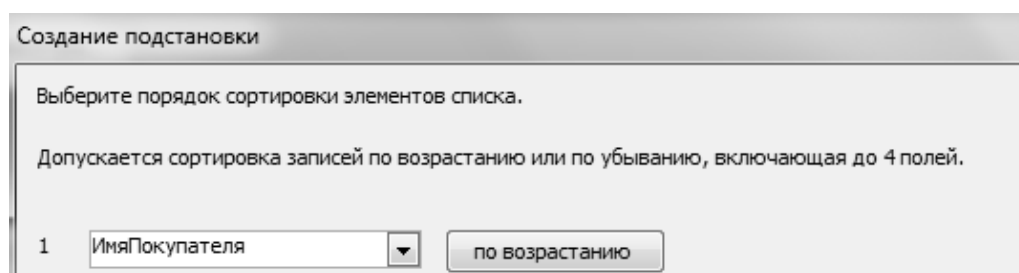


Рисунок 24 – Четвертый шаг Мастера подстановок

5) Затем можно подобрать ширину столбцов подстановки для того, чтобы данные поля просматривались при выборе из списка полностью (рис. 25). Нажать кнопку **Далее**.

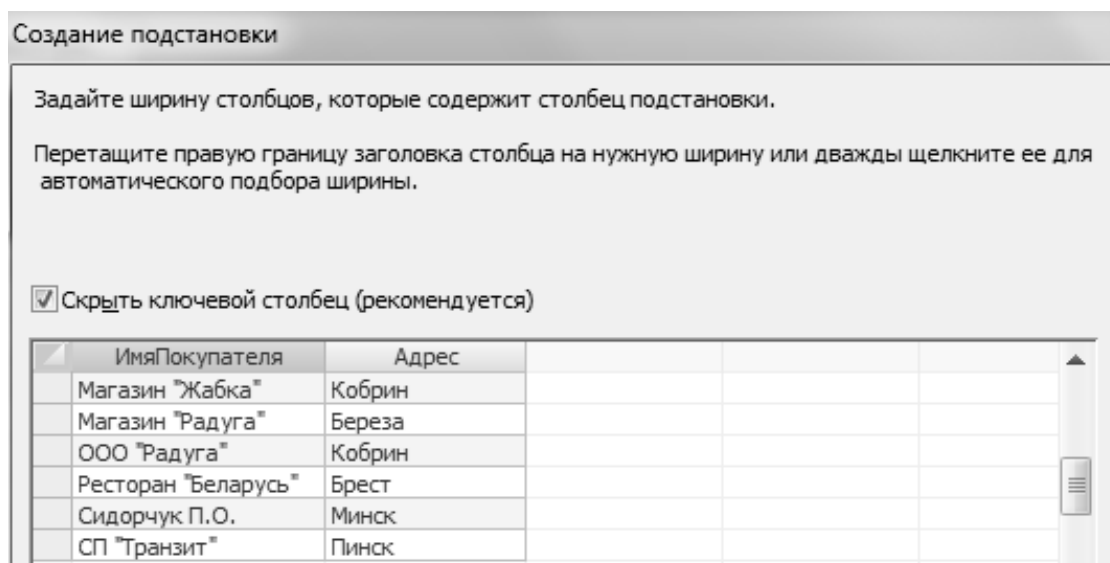


Рисунок 25 – Пятый шаг Мастера подстановок

6) В предпоследнем диалоговом окне можно оставить всё без изменений, соглашаясь с заданной подписью столбца подстановки (рис. 26), нажать кнопку **Готово**.

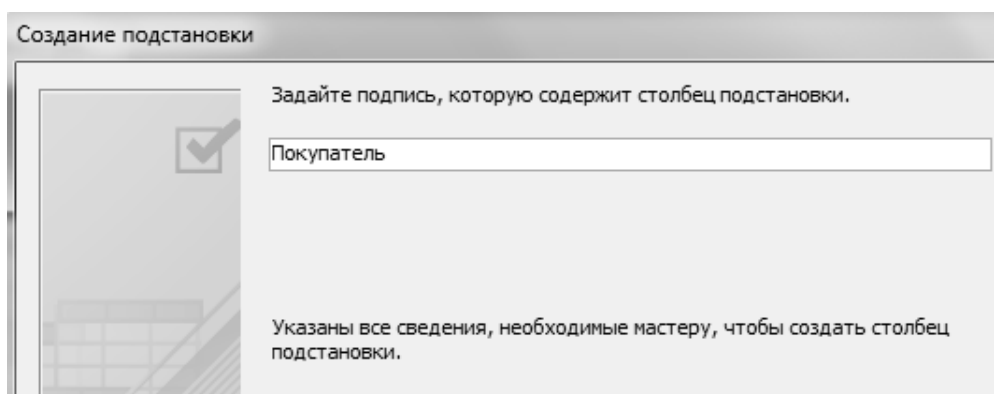


Рисунок 26 – Шестой шаг Мастера подстановок

7) В последнем диалоговом окне утвердительно ответить – **Да** (рис. 27).

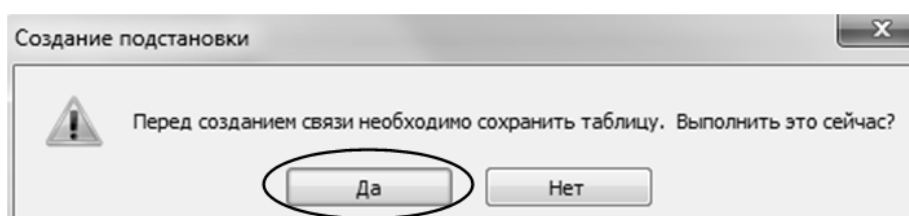


Рисунок 27 – Заключительный шаг Мастера подстановок

Для поля *Товар* нужно проделать ту же последовательность шагов, что и для поля *Покупатель*. Сохранить структуру таблицы **УчетПродаж**.

Аналогично используется Мастер подстановок для связи таблиц **ГруппаТоваров** и **Товары** по полю *ГруппаТовара*.

2.10. Установка связей между таблицами БД в MS Access

В MS Access межтабличные связи создаются и просматриваются с помощью окна

«**Схема данных**», которое вызывается с помощью пиктограммы **Схема данных** :

в версии Access 2003: на панели инструментов или соответствующей команды в пункте меню **Сервис**,

в версии Access 2007+: на вкладке ленты **Работа с базой данных** в группе **Отношения** (или **Показать** или **Скрыть**).

Откроется окно **Схема данных**, в котором будут отображены все созданные линии связи при использовании **Мастера подстановок** по ходу создания таблиц. Однако такие линии связи необходимо обеспечить целостностью следующим образом:

1) Находясь в окне **Схема данных**, активизировать левой кнопкой мыши имеющуюся линию связи.

2) Из контекстно-зависимого меню выбрать команду **Изменить связь...** (рис. 28).

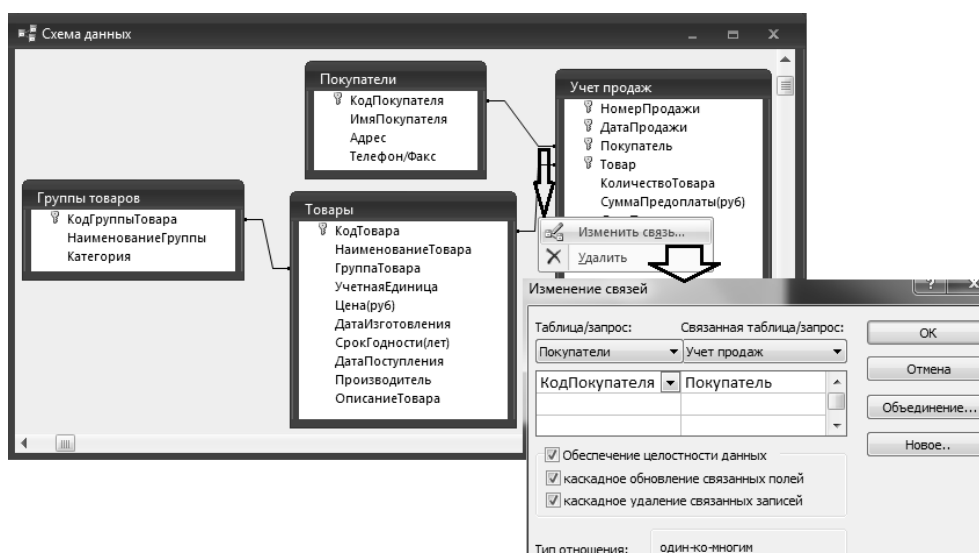


Рисунок 28 – Изменение линий связи в окне **Схема данных**

3) В диалоговом окне **Изменение связей** (рис. 28) установить флажок **Обеспечение целостности данных**. При необходимости, в этом же окне можно установить каскадные операции: **каскадное обновление связанных полей** и **каскадное удаление связанных записей**.

4) Нажать кнопку **Создать** (или **ОК**), после чего связи будут представлены в виде линий, соединяющих **первичный ключ** основной (или главной) таблицы с полем **внешнего ключа** связанной (или подчинённой) таблицы (рис. 29).

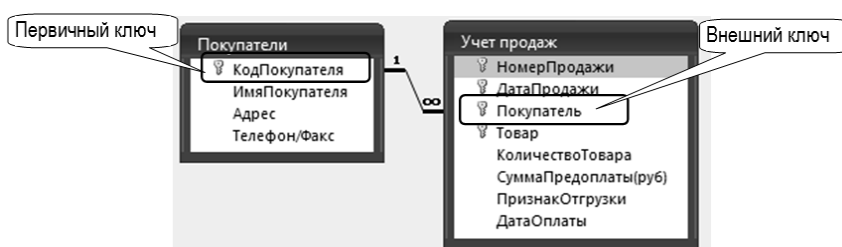


Рисунок 29 – Связь между таблицами, обеспеченная целостностью данных

5) Закрывать окно **Схема данных**, при этом на вопрос *Сохранить изменения макета Схема данных?* ответить – «Да».

Замечание 1: при открытии окна **Изменение связей** автоматически для выделенной связи устанавливается тип отношений между таблицами «**Один-ко-многим**», что означает следующее: каждый покупатель, **однозначно** определённый в таблице-справочнике **Покупатели** (т.е. один раз, что соответствует цифре **1** на линии связи) может **несколько** раз встречаться в таблице **Учет продаж** (т.е. **МНОГО** раз, что соответствует символу ∞ на линии связи). Это означает, что один и тот же покупатель может совершать покупки в магазине не один раз (рис. 30).

Покупатели		Учет продаж			
КодПокупателя	ИмяПокупателя	НомерПродаж	ДатаПродаж	Покупатель	Товар
1102	Демушкин П.И.	1	13.05.2016	Магазин "Верас"	Паркет ТЕМПА
1103	Левочкин С.В.	2	13.06.2016	Золотарев И.И.	Обои GRAND
1104	Крапивин А.Р.	3	14.06.2016	ИП Сидоров С.С.	Краска для пола KRAS
1105	Сидорчук П.О.	3	14.06.2016	ИП Сидоров С.С.	Валик ГРЕЙС с ручкой
1201	ИП Сидоров С.С.	4	13.07.2016	Сидорчук П.О.	Паркет CLASSIC
1202	ИП Петренко П.П.	5	15.07.2016	СП "Транзит"	Обои WILTAN
1203	ИП Иванов И.И.	5	15.07.2016	СП "Транзит"	Линолеум ТАКТ
1301	ООО "Радуга"	6	17.07.2016	Ресторан "Беларусь"	Краска интерьерная SWET
1302	СП "Транзит"	6	17.07.2016	Ресторан "Беларусь"	Валик ГРЕЙС с ручкой
1401	Ресторан "Беларусь"	7	09.08.2016	Левочкин С.В.	Диск зачистной по металлу
1402	Кафе "Прайм"	8	10.08.2016	СП "Транзит"	Краска для потолка СНЕЖОК
		9	11.08.2016	ООО "Радуга"	Ламинат EGAR

Рисунок 30 – Тип отношений «Один-ко-многим»

Замечание 2: если по какой либо причине в окне **Схемы данных** не отобразились все таблицы, то можно открыть окно **Добавление таблицы** по одноимённой команде из контекстно-зависимого меню (щелчок правой клавиши мыши) и выбрать нужную таблицу.

Если связи не были определены с помощью **Мастера подстановок**, то в окно **Схемы данных** необходимо добавить все таблицы и создать связи «вручную»: перетащить поле (как правило – это первичный ключ) из одной таблицы (в нашем случае, поле **КодПокупателя** из таблицы **Покупатели**) в поле с такими же значениями (внешний ключ) в другой таблице (в нашем случае, в поле **Покупатель** таблицы **УчетПродаж**).

Откроется диалоговое окно **Изменение связей**, в котором надо выполнить пункты 3-4 текущего раздела.

2.11. Удаление (изменение) связей

1. Открыть окно **Схема данных**.
2. Активизировать левой кнопкой мыши связь, которую необходимо удалить (изменить).
3. Правой кнопкой мыши вызвать контекстно-зависимое меню и выбрать команду **Удалить (Изменить)** соответственно.

Практическое задание 4. Заполнение таблиц БД записями.

Задача: заполнить данными сначала главные таблицы, а затем подчиненные таблицы БД **Магазин СТРОИТЕЛЬ**.

2.12. Ввод данных в таблицу

Чтобы перевести таблицу в режим ввода информации, нужно перейти в режим **Таблицы** (пункт 1.13 теоретического раздела). Поля заполняются последовательно. Переход от одного поля к другому удобно выполнять клавишей **Tab** (или комбинацией **Shift+Tab** – в обратном направлении). Если при проектировании таблицы для некоторых полей были предусмотрены значения по умолчанию, эти значения автоматически появятся в соответствующих полях. Записи в таблице можно перемещать, копировать и удалять теми же способами, что и в электронных таблицах, то есть сначала выделить строки, а потом выполнить необходимую операцию. Столбец можно выделить щелчком мыши по заголовку. Столбцы можно перемещать вправо и влево, пользуясь методом "drag and drop" (перетащить и бросить).

При необходимости можно вернуться в режим **Конструктора таблицы**. Это дает возможность что-либо подправить в структуре таблицы. Однако, в случае преобразования поля (числового или текстового типа) в меньшее по размеру, может произойти потеря данных.

2.13. Сортировка данных в таблице

Данные, находящиеся в таблице, можно отсортировать в порядке возрастания или убывания. Для этого нужно поместить курсор мыши в любую ячейку столбца, значения которого должны быть отсортированы и из контекстного меню выбрать команду **Сортировка...** Ускорить сортировку и поиск данных в таблице возможно с помощью установки свойства индексированного поля (полей).

Практическое задание 5. Проектирование запросов на выборку

2.14. Создание запросов выбора с помощью Конструктора запросов

- 1) В версии Access 2003: в окне БД выбрать вкладку **Запросы** и щелкнуть кнопку **Создать**, откроется окно диалога **Новый запрос**, в котором выбрать опцию **Конструктор** и нажать кнопку **ОК**;
- 1) В версии Access 2007+: на вкладке ленты **Создание** в группе **Запросы** (или **Другие**) выбрать пиктограмму **Конструктор запросов**;
- 2) в появившемся диалоговом окне **Добавление таблицы** из списка следует выбрать имя таблицы-источника или запроса-источника, на основе которых будет проводиться выбор данных, и щелкнуть кнопку **Добавить** (рис. 31);

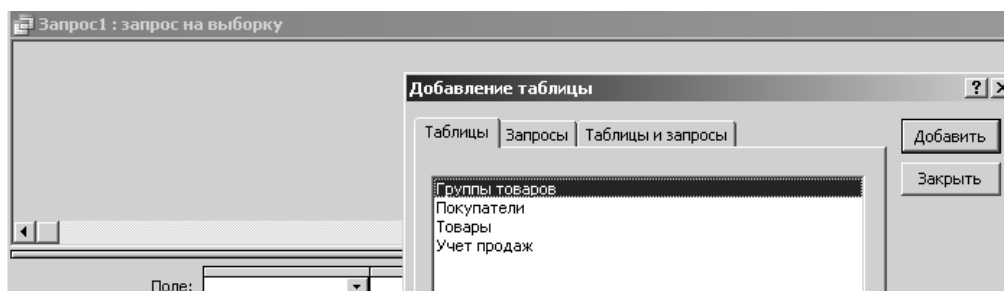


Рисунок 31 – Диалоговое окно **Добавление таблицы** при проектировании запроса

Если запрос формируется на основе нескольких таблиц, то действия, указанные в п. 2, повторяются для каждой таблицы-источника;

3) закрыть окно диалога **Добавление таблицы** кнопкой **Заккрыть** – на экране появится окно **Конструктора запроса** (рис. 32).

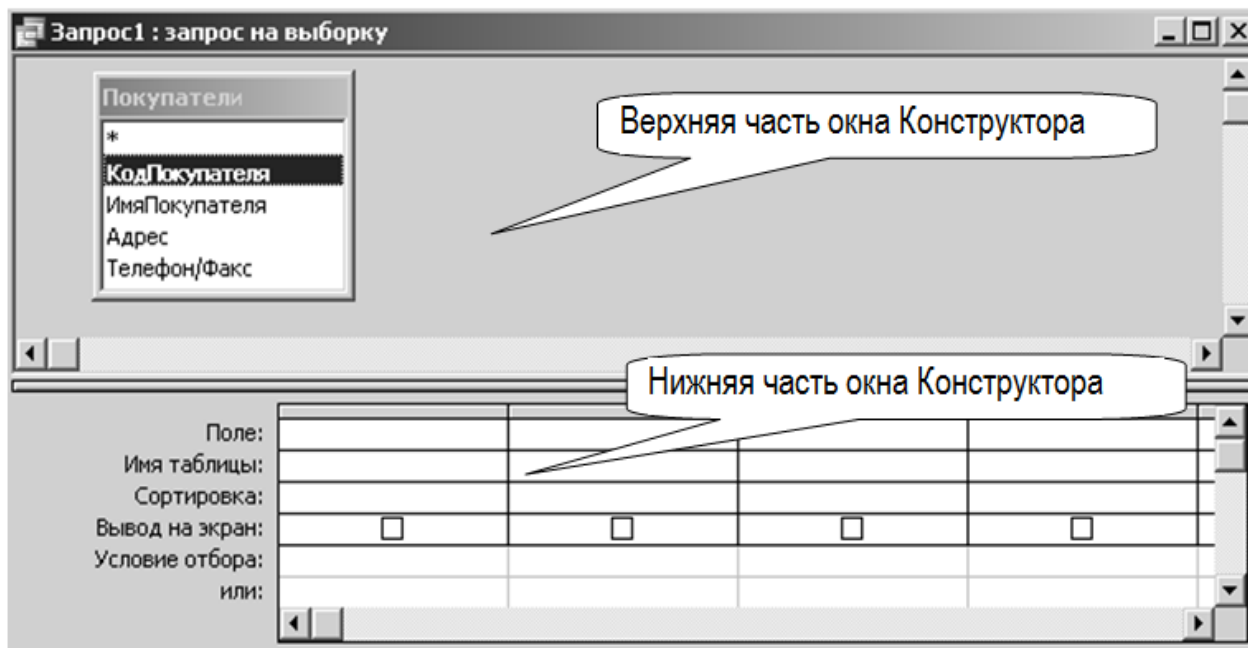


Рисунок 32 – Окно Конструктора запроса

Окно **Конструктора запроса** состоит из двух частей.

В **верхней части** окна **Конструктора запроса** размещается **Схема данных запроса**, которая содержит список таблиц-источников и отражает связь между ними.

В **нижней части** окна находится **Бланк запроса (бланк QBE)**, в котором каждая строка выполняет конкретную функцию:

- в строке **Поле** указываются имена полей, участвующих в запросе;
- в строке **Имя таблицы** отображаются имена таблиц, из которых взяты поля;
- в строке **Сортировка** задается порядок сортировки записей в запросе;
- в строке **Вывод на экран** осуществляется пометка полей, выводимых в результате выполнения запроса (посредством установки флажков);

➤ в строке **Условие отбора** задаются условия отбора записей по значению поля. Условие отбора – это выражение, которое состоит из операторов сравнения и операндов, используемых для сравнения. В качестве операндов могут быть выбраны некоторые *заданные значения* и *идентификаторы* (имена полей, таблиц, запросов, форм, отчетов и т.д.). Идентификаторы необходимо заключать в квадратные скобки. Если надо указать ссылку на поле в конкретной таблице, форме, отчете, то прописывается полное имя поля в виде: **[Имя таблицы]![Имя поля]**.

➤ строка **"или"** позволяет объединять условия отбора логической операцией ИЛИ, причем она может состоять из нескольких строк бланка запроса.

С помощью инструментария окна **Конструктора** запрос формируется в два этапа:

- выбираются поля для вывода значений в результат запроса;
- указываются критерии для выборки, группирования и сортировки данных.

2.14.1. Добавление (удаление) полей в Бланке запроса

Выбор полей с таблицы-источника (из верхней части окна **Запрос на выборку**) в **Бланк запроса** (нижнюю часть окна) можно выполнить следующими способами:

- ✓ путем «перенести-и-бросить»;
- ✓ путем выбора поля таблицы в верхней части окна двойным щелчком мыши;
- ✓ с помощью выбора нужного поля из списка, находящегося непосредственно в ячейке строки **Поле Бланка запроса**.


В строке **Поле** появится имя поля, а в строке **Имя таблицы** автоматически появится имя таблицы-источника; в строке **Вывод на экран** будет установлен флажок.

Замечание: для включения всех полей в бланк запроса нужно выбрать из таблицы, находящейся в верхней части **Конструктора запроса**, пункт * , при этом отключить режим вывода на экран для дополнительных полей, по которым задается критерий отбора.

Удалять поля в нижней части Бланка запроса можно следующими способами:

1 способ: выделить столбец с помощью стрелки ↓ (которая появляется при подведении курсора мыши к верхней части столбца) и нажать кнопку **Delete**;

2 способ: установить курсор в столбец для удаления и выбрать:

в версии Access 2003: из пункта **Правка** команду **Удалить столбец**; в версии Access 2007+: на вкладке ленты **Конструктор** в группе инструментов **Настройка запроса** пиктограмму  **Удалить столбцы**



2.14.2. Сортировка результатов выборки

Для сортировки результатов выборки по одному полю необходимо перейти на строку **Сортировка** нужного поля и в раскрывающемся списке выбрать способ сортировки *по возрастанию* или *по убыванию*.

Для отказа от сортировки по выбранному ранее полю следует выбрать значение **Отсутствует**.

2.14.3. Запуск запроса на выполнение

Для запуска запроса на выполнение из окна **Конструктора** надо:

в версии Access 2003: на панели инструментов нажать кнопку «**Запуск**»  или выполнить из пункта меню **Запрос** команду **Запуск**;
в версии Access 2007+: на вкладке ленты **Конструктор** в группе инструментов **Результаты** нажать пиктограмму  **Выполнить**.

Результаты выборки данных по запросу выводятся на экран в режиме таблицы.

Запрос1: вывести в алфавитном порядке список тех клиентов, которым товар не отгружен.

Бланк **Конструктора Запроса1** представлен на рисунке 33.

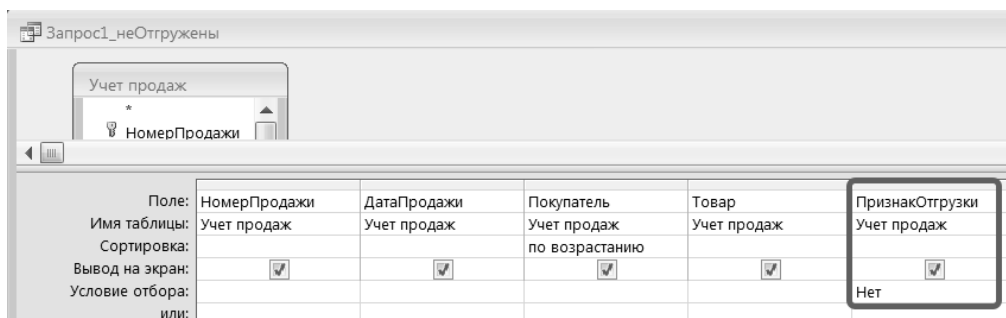


Рисунок 33 – Запрос1 в режиме Конструктора

2.15. Построитель выражений

Построитель выражений предназначен для составления математических, логических и других выражений при изменении свойств полей таблиц, при задании условий отбора и создании вычисляемых полей в запросах. Этот инструмент обеспечивает простой доступ к именам полей и элементов управления в базе данных, а также ко многим встроенным функциям, которые можно включать в выражения.

Построитель выражений в разных случаях (или в строке ввода свойств для поля таблицы в режиме Конструктора, или для нового Поля в бланке запроса, или в строке Условие отбора для задания критерия в бланке запроса) вызывается одним из следующих способов:

1 способ: выбрать команду **Построить** из контекстного меню (вызываемого правой кнопкой мыши);

2 способ:

в версии Access 2003: нажать на панели инструментов



кнопку **Построить**

в версии Access 2007+: выбрать на вкладке ленты панели инструментов **Сервис**, для использования в бланке запроса – в группе **Настройка запроса** пиктограмму



Построитель

Окно построителя выражений состоит из трех разделов (рис. 34):

В **верхней части** окна построителя расположено поле построения выражения.

В **средней части** окна построителя находятся кнопки с часто используемыми операторами. При нажатии на одну из этих кнопок построитель вставит соответствующий оператор в текущую позицию поля выражения.

Замечание. Чтобы вывести полный список операторов, можно выбрать папку **Операторы** в нижней левой области и нужный тип в средней – в правой области будут выведены все операторы выбранного типа.

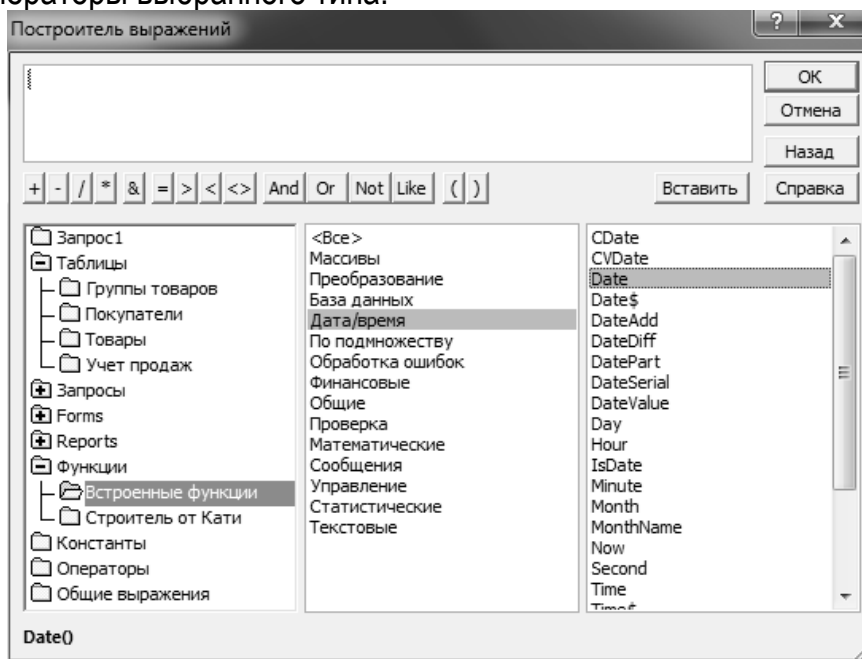


Рисунок 34 – Окно Построителя выражений

В **нижней части** окна построителя находятся три области.

В *левой области* содержится список источников данных (таблицы, запросы, формы, объекты базы данных) и средства для записи выражений (встроенные и определенные пользователем функции, константы, операторы и общие выражения).

В *средней области* задаются определенные элементы или типы элементов для папки, заданной в левой области. Например, если выбрать в левой области **Встроенные функции**, то в средней области появится список всех типов функций Access.

В *правой области* выводится список значений (если они существуют) для элементов, заданных в левой и средней областях. Например, если выбрать в левой области **Встроенные функции** и тип **Дата/время** в средней области, то в правой области будет выведен список всех встроенных функций выбранного типа (рис. 34).

Поиск нужных элементов начинается с левой области. Списки объектов в левой, средней и правой областях имеют иерархическое подчинение.

Последовательность вставки элемента выражения в верхнее поле ввода:

- 1) в левой области выбрать необходимую папку;
- 2) открыть папку для выбора нужного объекта (элемента) – двойным щелчком мыши;
- 3) последовательно в средней и правой областях выбрать нужный элемент и перенести его в поле ввода выражения (в верхнюю часть окна) кнопкой **Вставить** или дважды щелкнуть элемент мышью;
- 4) в средней части окна Построителя выбрать нужный оператор и щелкнуть мышью для переноса его в создаваемое выражение;
- 5) после формирования нужного выражения нажать кнопку **OK**.

Пример выбора **поля таблицы** рассмотрен на рисунке 39 при создании вычисляемого запроса, а пример выбора **встроенной функции** в окне **Построителя выражений** приведен на рисунке 34.

Замечание 1: при переносе операторов, функций или других элементов выражений из нижней части в верхнее поле построения могут появляться слова «Выражение», которые следует удалять.

Замечание 2: при выделении встроенной функции в правой области нижней части окна **Построителя**, её синтаксис будет отображен внизу окна и будет возможен доступ к подсказке по данной функции с помощью кнопки **Справка**.

2.16. Формирование условий отбора в запросе

Список операторов, используемых при задании выражений в условиях отбора:

➤ операторы сравнения:

= (равно)	> (больше)	>= (больше или равно, не меньше)
< > (не равно)	< (меньше)	<= (меньше или равно, не больше)

BETWEEN – позволяет задать диапазон значений.

Синтаксис: **Between «Выражение» And «Выражение»** (например, логическому выражению **>=10 AND <=20** эквивалентна запись **BETWEEN 10 And 20**).

IN – позволяет задавать используемый для сравнения список значений (операндом является список, заключенный в круглые скобки). Например: **IN("Брест", "Минск", "Гродно")** означает то же самое, что и логическое выражение **"Брест" OR "Минск" OR "Гродно"**.

➤ логические операторы:

AND – логическое **И** или логическое умножение (конъюнкция). Объединяет два и более условий и возвращает истину только при одновременном выполнении всех условий (например, от 10 и до 20: $\geq 10 \text{ AND } \leq 20$).

OR – логическое **ИЛИ**, эквивалент логического сложения (дизъюнкция). Связывает два или больше условий, но возвращает истину при выполнении хотя бы одного условия (например, до 50 или свыше 100: $< 50 \text{ OR } > 100$).

NOT – отрицание (например: **Is Not Null** – поле, содержащее какое-либо значение).

➤ **оператор LIKE** – проверяет соответствие *текстового* или *Мето поля* по заданному шаблону символов:

Символы шаблона	Соответствие в выражении
?	Любой один текстовый символ
*	Соответствует любой цифре или любому символу
#	Любая одна цифра
[список знаков]	Любой один знак в «списке знаков»
[!список знаков]	Любой один знак, который не входит в «список знаков»

Примеры создания условий отбора см. в Приложении 5.

2.17. Формирование условий отбора для полей с типом данных Дата/Время

Для того, чтобы сообщить Access, что выражении задается дата (или время), необходимо значение даты (или времени) заключить в знак #. Например: #15.10.16#.

Основные функции для обработки значений полей с типом данных Дата/Время см. в Приложении 3. Примеры создания условий отбора см. в Приложении 5.

2.18. Сложные критерии выборки в запросах

Часто приходится выбирать записи по условию, которое задается для нескольких полей таблицы или по нескольким условиям для одного поля. В этом случае применяются два вида запросов:

- ✓ **«И-запросы»** (выбор записей только при условии выполнения всех условий);
- ✓ **«ИЛИ-запросы»** (выбор записей при выполнении хотя бы одного из условий).

При задании «ИЛИ-запроса» каждое условие выборки должно размещаться на отдельной строке Бланка запроса. При задании «И-запроса» каждое условие выборки должно размещаться на одной строке, но в разных полях Бланка запроса. Эти операции для условий отбора по одному полю могут быть заданы явно с помощью операторов **OR** и **AND** соответственно.

Запрос2: выбрать записи по продажам обоев GRAND в количестве рулонов более 5 штук.

Запрос2 строится по таблицам «Товары» и «УчетПродаж», которые добавляются в верхнюю часть конструктора запроса.

Данный запрос относится к виду **«И-запросы»** с критериями по двум разным полям таблицы. В результате запроса сначала выбираются записи с наименованием товара **обои GRAND**, затем среди которых выбираются те, количество продаж которых **более 5-ти**. Бланк Конструктора Запроса2 представлена на рисунке 35.

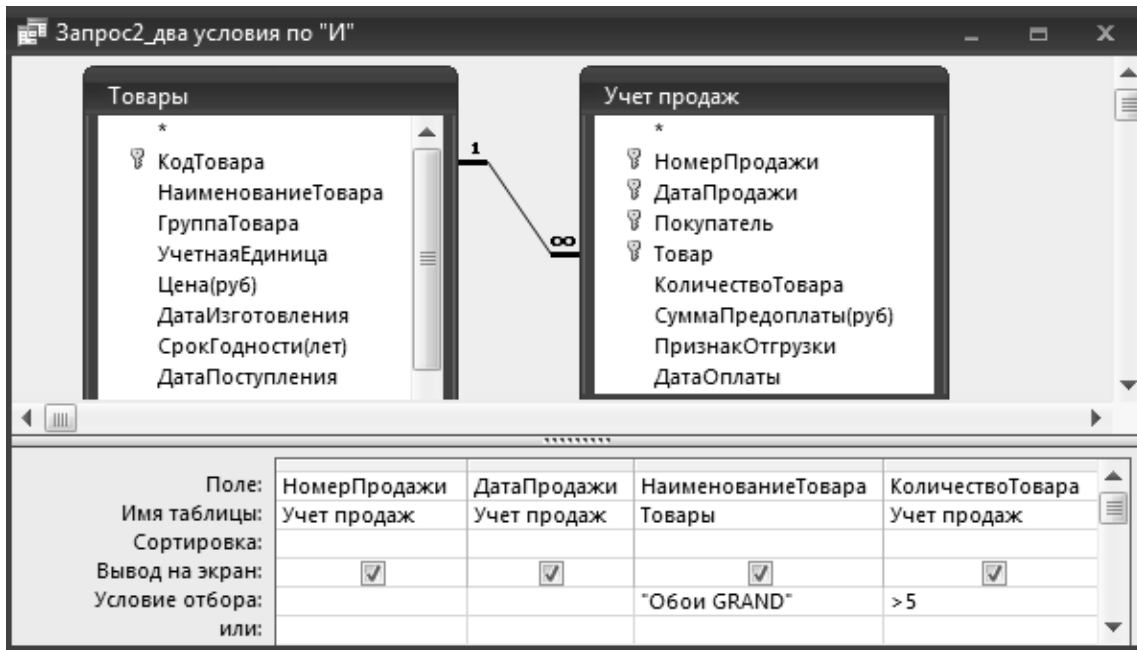


Рисунок 35 – Запрос2 в режиме Конструктора

Запрос3: выбрать всю информацию о товарах из групп ламинатов 32-го и 33-го классов.

Данный запрос относится к виду «ИЛИ-запросы» с несколькими критериями отбора по одному полю таблицы. В результате запроса сначала выбираются товары, для которых значение поля **НаименованиеГруппы** совпадает с одним, либо с другим критерием.

В сформированном запросе использована возможность вывода всех полей таблицы **Товары** с помощью символа «*».

Бланк **Конструктора Запроса3** представлен на рисунке 36.

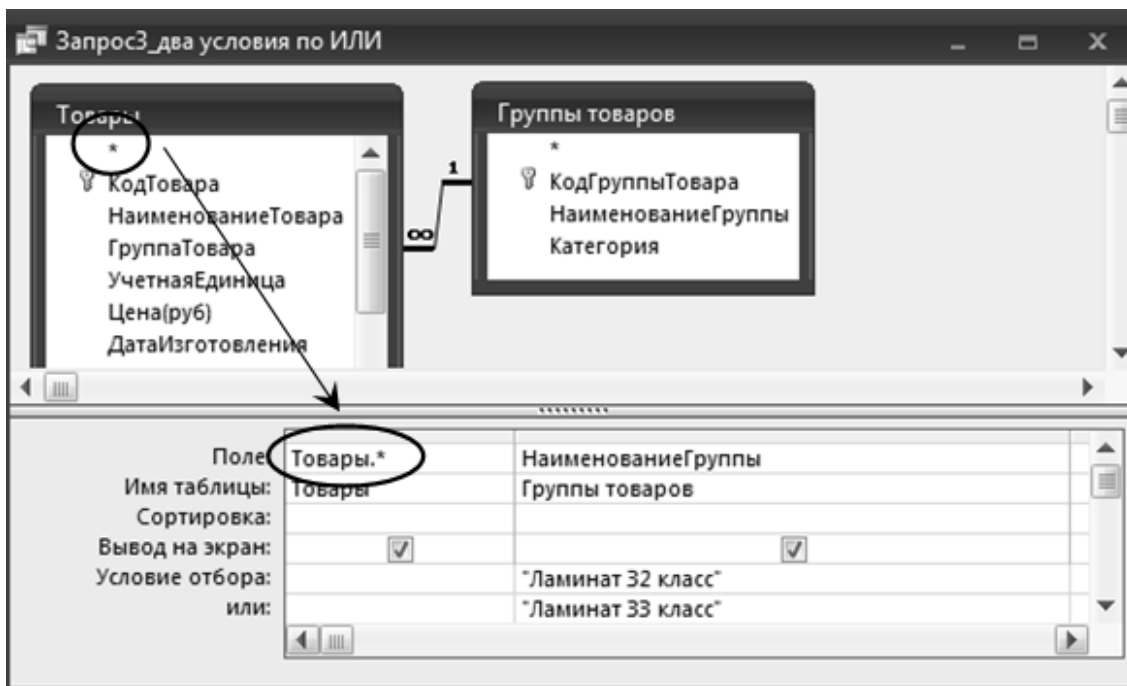


Рисунок 36 – Запрос3 в режиме Конструктора

Замечание: Возможен другой вариант задания двух критериев в одной строке *Условие отбора* для поля **НаименованиеГруппы**:

"Ламинат 32 класс" Or "Ламинат 33 класс".

Практическое задание 6. Параметрические запросы

2.19. Проектирование параметрических запросов

При создании запроса в **условии отбора** или в **выражении для вычисляемых полей** можно использовать **параметры**.

Параметр – это шаблон, заключенный в квадратные скобки, который при выполнении запроса будет заменен значением, введенным с клавиатуры.

Запрос с параметрами выводит одно или несколько predetermined диалоговых окон, в которых пользователь должен ввести значения параметров.

Параметрический запрос каждый раз при выполнении требует ввода с клавиатуры значений параметров, требуемых по заданному условию запроса. Тем самым, **параметрический запрос** имеет универсальный характер условия или формулы, в зависимости от того, где используется параметр.

Преимущества параметрических запросов:

- нет необходимости при изменении критерия условия или значения константы в формуле модифицировать запрос в режиме **Конструктора запросов**;
- удобно использовать в формах и отчетах, т. к. каждый раз при их открытии Access запрашивает у пользователя требуемый параметр. Получаются тем самым настраиваемые формы и универсальные отчеты.

Для использования **параметра** в строке *Условие отбора* нужно в выражение для условия вместо конкретной константы записать произвольное имя параметра, заключенное в квадратные скобки. **Имя параметра** может быть даже фразой из нескольких слов, главное, чтобы имя параметра не совпало ни с одним полем таблиц и запросов текущей БД. При выполнении запроса MS Access то, что заключено в [] пытается найти среди всех полей БД, и, если не находит, то рассматривает то, что заключено в [], как параметр, и просит пользователя в экранном диалоге задать этому параметру необходимое значение путем ввода с клавиатуры.

Например, для поля с типом Дата/Время или числового типа для условия отбора за период дат или интервал чисел в качестве параметров для задания границ диапазона можно записать в квадратных скобках следующие виды приглашения в выражении:

>=[Введите начальную дату:] And <=[Введите конечную дату:]

Between [НачальнаяДата] **And** [КонечнаяДата]

>=[ДатаС] **And** <=[ДатаПо]

Between [С] **And** [По]

>=[нач] **And** [кон]

Between [цена1] **And** [цена2]

Классический пример использования **параметра в выражении для вычисляемого поля** – это формула для перевода значения поля таблицы *Цена(руб)* в валютное значение по курсу, задаваемому каждый раз с клавиатуры. Выражение может иметь вид, например: **[Цена(руб)]/[КурсВалюты]**, где **[КурсВалюты]** – это параметр.

Запрос4: *Выбрать продажи за период, задаваемый с клавиатуры.*

Запрос4 относится к виду «**И-запросы**» с двумя критериями отбора по одному и тому же полю таблицы. Бланк **Конструктора Запроса4** представлен на рисунке 37.

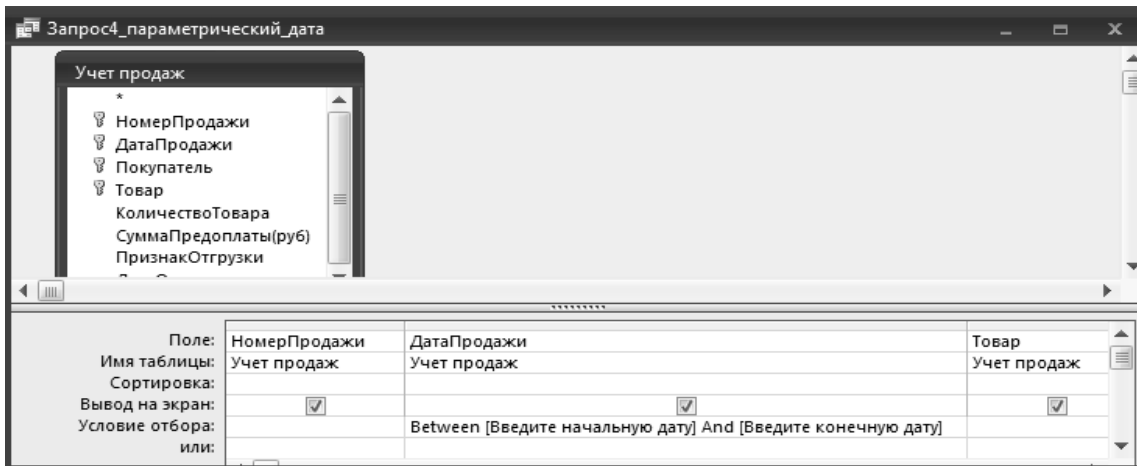


Рисунок 37 – Запрос4 в режиме Конструктора

При выполнении данного запроса сначала будет выведено первое предопределённое диалоговое окно «**Введите значение параметра**» с запросом ввести начальную дату границы временного интервала, затем будет выведено другое окно с запросом ввести конечную дату интервала (рис. 38).

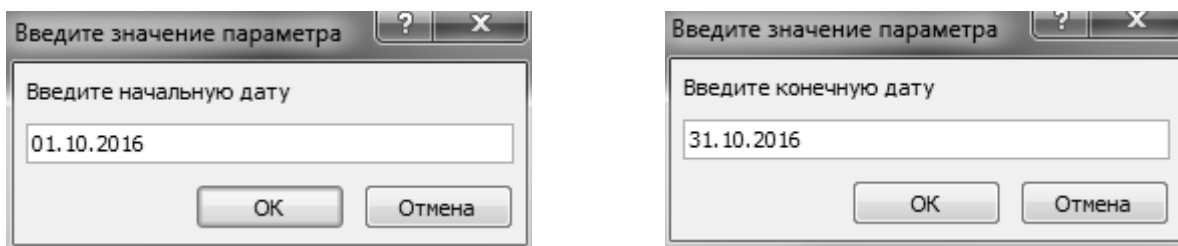


Рисунок 38 – Ввод границ временного интервала при выполнении запроса4

Практическое задание 7. Вычисляемые запросы для БД в MS Access

2.20. Вычисляемые поля в запросах для БД в MS Access

Существует ряд вычислений, которые можно выполнить в запросе. Например, найти сумму или перемножить значения двух полей, вычислить дату, отстоящую от текущей на несколько месяцев, заполнить поле значениями в зависимости от выполнения условия.

Результаты вычислений, выводящиеся в поле, не запоминаются в базовой таблице. Вычисления снова производятся всякий раз, когда выполняется запрос, поэтому результаты всегда представляют текущее содержимое базы данных. Обновить вычисленные результаты вручную невозможно.

Для определения вычисляемого поля можно использовать встроенные функции Access или выражения, создаваемые пользователем. Выражения, определяемые пользователем, дают возможность выполнять действия с числами, датами и текстовыми значениями в каждой записи с использованием данных из одного или нескольких полей.

Результаты вычислений не должны обязательно отображаться в поле. Их можно использовать в условиях отбора для определения записей, которые выбираются в запросе, или для определения записей, над которыми производятся какие-либо действия.

Новое вычисляемое поле создается прямо в Бланке запроса. Выражение записывается в пустую ячейку **Поле** с помощью **Построителя выражений** и начинается с **названия** для нового поля, **двоеточия**, после которого следует само **выражение**.

Запрос5: Вычислить стоимость товаров в рублях. Определить окончательный размер доплаты за покупку с учетом предоплаты.

В данном запросе создаются последовательно два новых вычисляемых поля.

Сначала создается поле для вычисления стоимости товаров в рублях – *Стоимость покупки (руб)*:

СтоимостьПокупки (руб): [Товары].[Цена(руб)] * [УчетПродаж].[КоличествоТовара]

Затем **Запрос5** сохраняется для того, чтобы далее иметь возможность в следующем вычисляемом выражении для нового поля использовать уже созданные поля.

Для формирования выражения нового вычисляемого поля *Сумма доплаты* в **Построителе выражений** в левой области списка источников данных открывают папку текущего **Запроса5**, выбирают поле *СтоимостьПокупки(руб)* и нажимают кнопку **Вставить** или выполняют мышью двойной щелчок (рис. 39):

СуммаДоплаты: [СтоимостьПокупки (руб)] – [УчетПродаж].[СуммаПредоплаты]

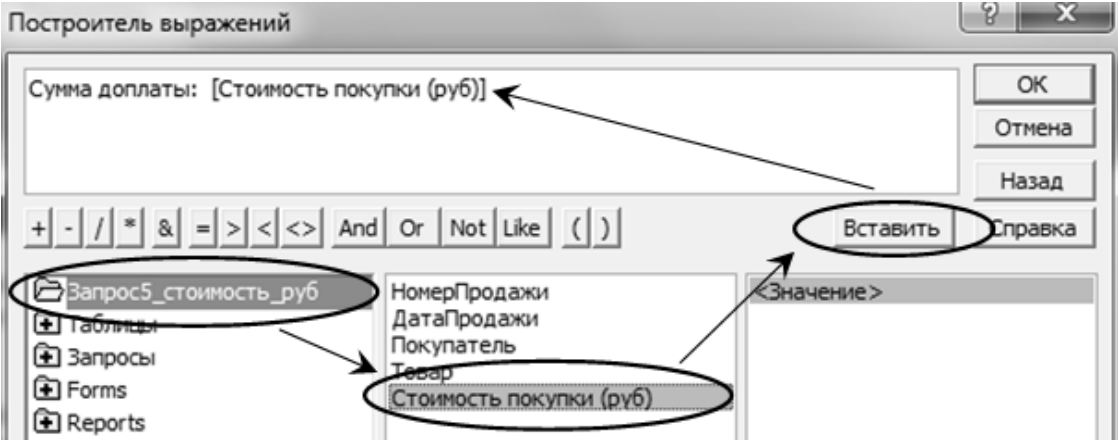


Рисунок 39 – Выбор поля текущего Запроса5 в окне Построителя выражений

Бланк **Конструктора Запроса5** представлен на рисунке 40.

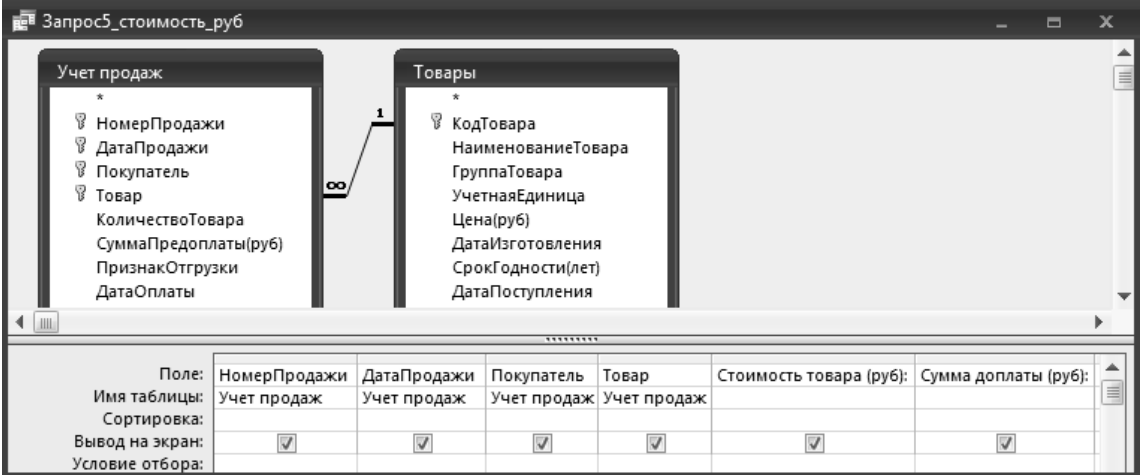


Рисунок 40 – Запрос5 в режиме Конструктора

Запрос6: Подсчитать стоимость товаров в валюте, курс которой вводится с клавиатуры как параметр.

Для ввода курса валюты каждый раз при выполнении запроса необходимо в выражение для вычисляемого поля включить параметр с именем *[ВведитеКурсВалюты]* (см. раздел 2.19 в практическом задании 6):

СтоимостьТовараВал:

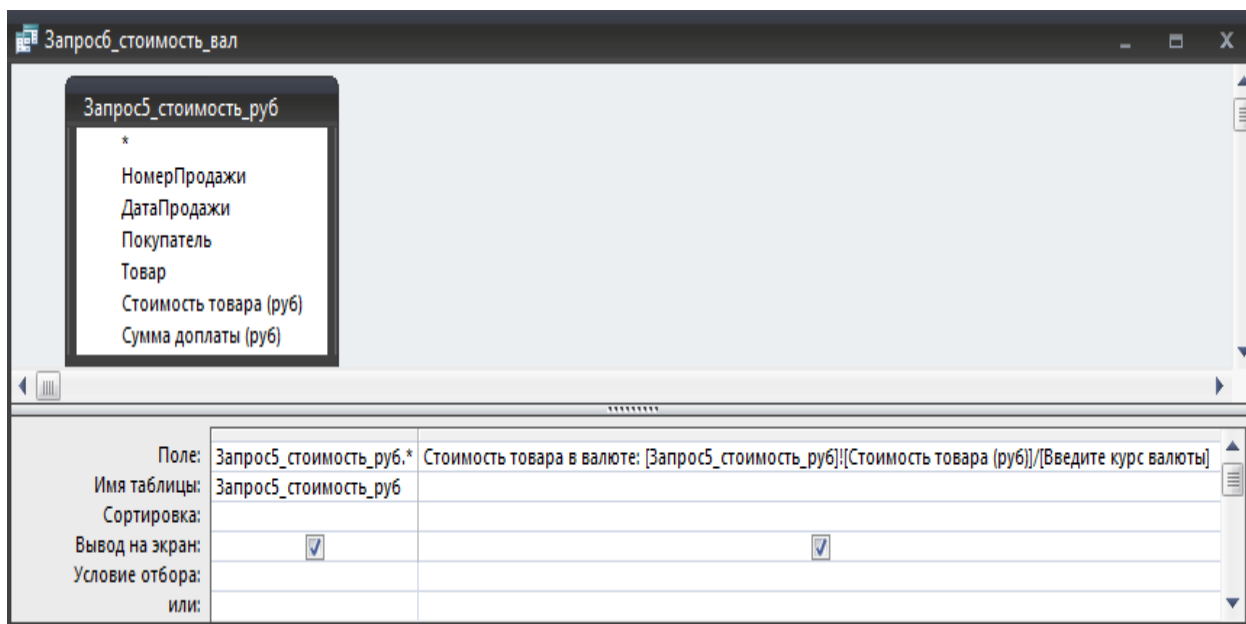
[Товары]![Цена(руб)] * [УчетПродаж]![КоличествоТовара] / [ВведитеКурсВалюты]

Для создания данного запроса можно использовать готовый Запрос5, в котором рассчитана стоимость в рублях. Тогда выражение будет иметь вид:

СтоимостьТовараВал:

[Запрос5_стоимость_руб]![СтоимостьТовара(руб)] / [ВведитеКурсВалюты]

Бланк Конструктора Запроса6 представлен на рисунке 41.



Рисунке 41 – Запрос6 в режиме Конструктора

Запрос7: *Определить дату истечения срока годности товара, исходя из даты его изготовления и срока хранения. Подсчитать, сколько месяцев осталось до истечения срока годности товара.*

Для товаров с истекшим сроком годности вывести сообщение "списать".

Выражения для новых вычисляемых полей имеют вид:

ДатаИстеченСрокаГодн:

DateAdd("yyyy"; [Товары]![СрокГодности(лет)]; [Товары]![ДатаИзготовления])

ДоИстеченГоднОсталосьМесяцев: ([ДатаИстеченСрокаГодн] – Date()) / 30

Списание: Iif([ДоИстеченГоднОсталосьМесяцев]<0; "списать"; "")

Бланк Конструктора Запроса7 представлен на рисунке 42.

Замечание 1: чтобы в вычислениях использовать уже созданные в текущем запросе вычисляемые поля, необходимо запрос сохранять при каждом новом вычислении и тогда созданные поля будут доступны в **Построителе выражений** для формирования следующих полей;

Замечание 2: синтаксис использованных функций **DateAdd()**, **Date()**, **Iif()** приведен в **Приложении 3** и **Приложении 4**.

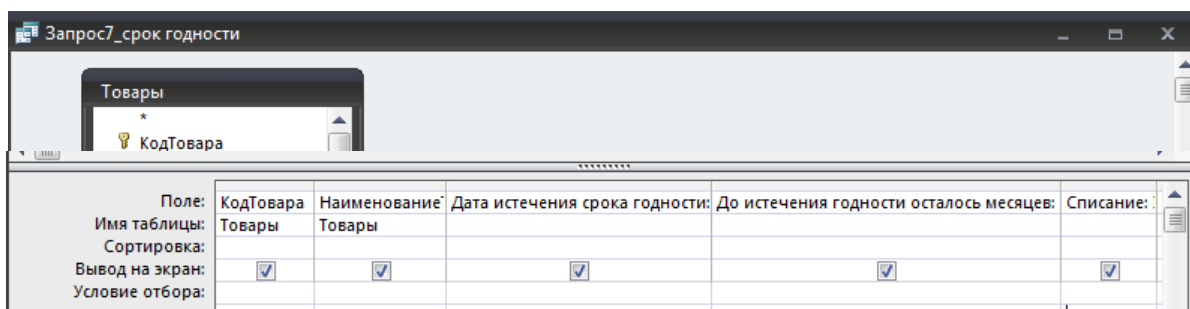


Рисунок 42 – Запрос7 в режиме Конструктора

Запрос8: Вывести список товаров, до истечения срока годности которых осталось менее месяца, и вычислить для них цену со скидкой 50% от цены.

Для создания **Запроса8** можно использовать предыдущий **Запрос7**, в котором было рассчитано количество месяцев до окончания срока хранения товара.

Выражение для вычисляемого поля *ЦенаСоСкидкой* имеет вид:

ЦенаСоСкидкой: [Товары].[Цена(руб)] * 0,5

Бланк **Конструктора Запроса8** представлен на рисунке 43.

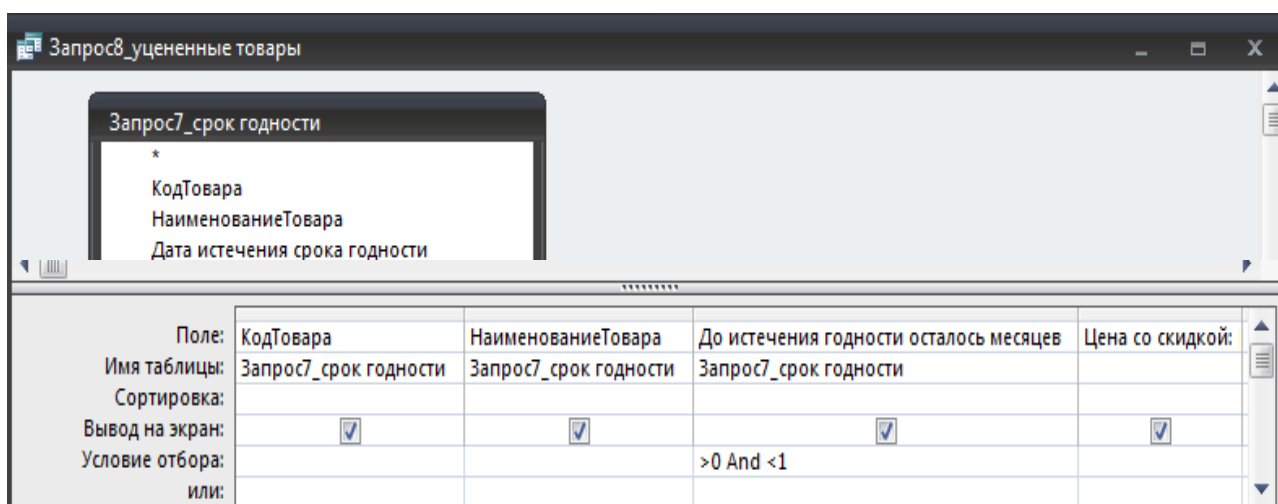


Рисунок 43 – Запрос8 в режиме Конструктора

Запрос9: Для покупателей, не оплативших товар в день покупки, определить, через сколько дней от них поступила оплата.

Выражение для вычисляемого поля *КолДнейДоПолнойОплаты* имеет вид:

КолДнейДоПолнойОплаты:

[УчетПродаж].[ДатаОплаты] – [УчетПродаж].[ДатаПродажи]

Замечание 1: разница между двумя значениями полей с типом данных Дата/Время выражается целым числом.

Замечание 2: значение данного поля можно посчитать по другой формуле с использованием встроенной функции **DateDiff()**, синтаксис которой приведен в **Приложении 3:**

DateDiff("d";[УчетПродаж].[ДатаПродажи];[УчетПродаж].[ДатаОплаты])

Бланк **Конструктора Запроса9** представлен на рисунке 44.

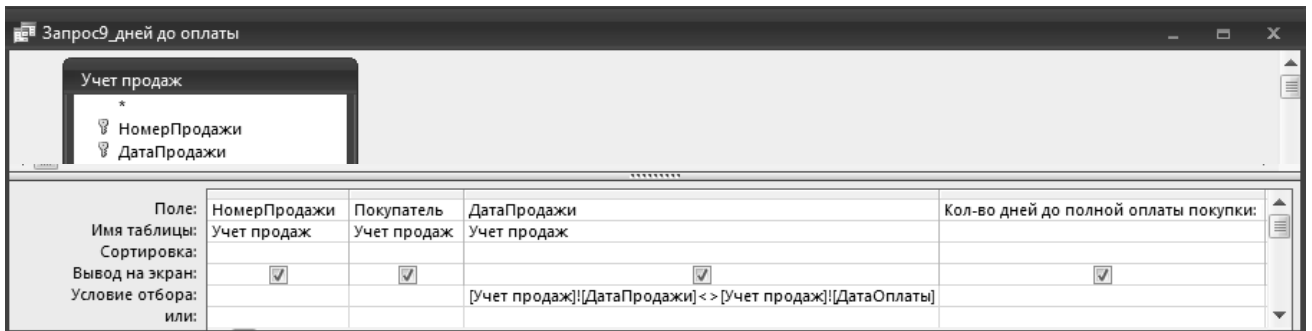


Рисунок 44 – Запрос9 в режиме Конструктора

Запроса10: Для покупателей, которым был продан товар в субботу, предоставить скидку 3% от стоимости товара, в воскресенье – 5%.

Сначала создается поле *ДеньНедПродаж*, в котором вычисляется название дня недели продаж:

ДеньНедПродаж: `WeekdayName(Weekday([УчетПродаж]![ДатаПродажи]);0;1)`

После сохранения запроса создается поле *Скидка* для вычисления значения с использованием функции `lif()`, синтаксис которой описан в Приложении 4:

Скидка: `lif([ДеньНедПродаж]="Воскресенье";
0,05 * [Запрос5_стоимость_руб]![СтоимостьТовара(руб)];
lif([ДеньНедПродаж]="суббота";
0,03 * [Запрос5_стоимость_руб]![СтоимостьТовара(руб)]; 0))`

Бланк Конструктора Запроса10 представлен на рисунке 45.

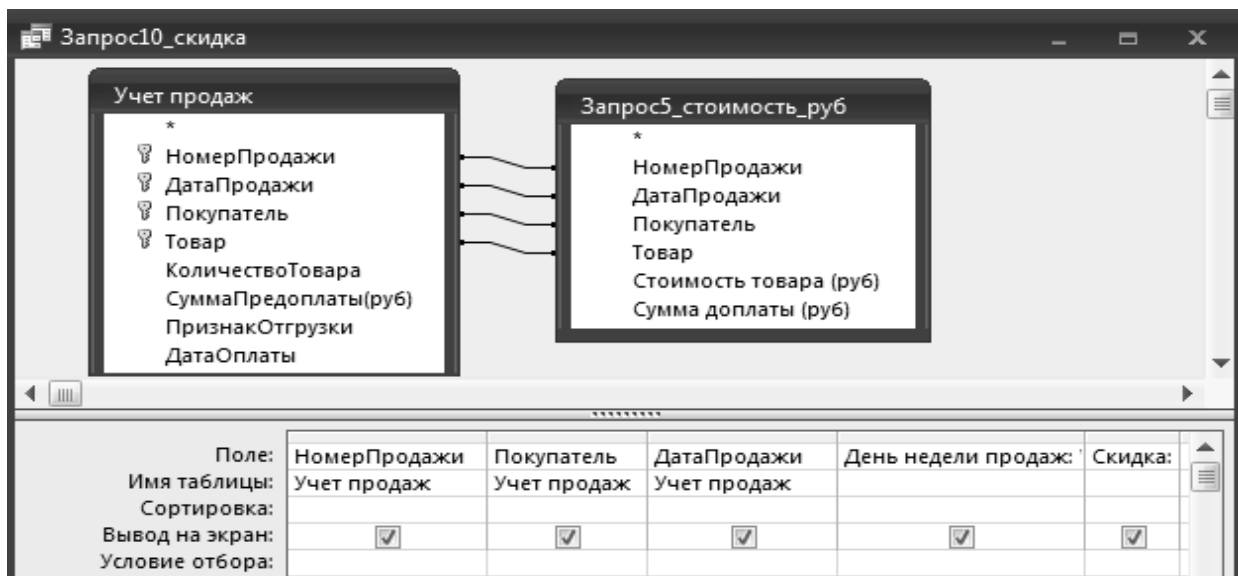



Рисунок 45 – Запрос10 в режиме Конструктора


Практическое задание 8. Итоговые запросы для БД MS Access

2.21. Проектирование итоговых запросов для БД MS Access

Запросы, которые выполняют вычисления для групп записей, называются итоговыми. Можно вычислить сумму, среднее, минимальное и максимальное значения полей записей, входящих в группу.

Для создания итогового запроса необходимо произвести следующие изменения в окне **Конструктора** запроса:

в версии Access 2003: вызвать команду **Групповые операции Вид** или нажать на панели инструментов на кнопку 

в версии Access 2007+: на вкладке ленты **Конструктор** в группе инструментов **Показать или скрыть** выбрать пиктограмму  **Итоги**

После этого в Бланке запроса появится новая строка **Групповые операции**.

В строке **Групповая операция** для соответствующего критерию запроса поля указывается из раскрывающегося списка установок групповых операций тип выполняемой групповой установки или выбирается вид групповой функции для вычисления итогового значения:

Групповая установка	Результат
Группировка	Определяет группы записей, для которых рассчитываются итоговые значения. Группирует записи по каждому имеющемуся полю, а в результате выполнения запроса выводит набор записей, включающий по одной строке для каждого уникального значения полей запроса.
Выражение	Определяет в запросе вычисляемое поле. Создание вычисляемого поля позволяет использовать в выражении несколько функций.
Условие	Задаёт условия отбора для поля, не используемого для определения группы. После выбора параметра «Условие» поле делается скрытым (снимается флажок в строке «Вывод на экран»).

Групповая функция	Результат
Sum	Сумма всех значений поля внутри группы записей
Avg	Среднее значение поля среди всех значений внутри группы
Min	Минимальное значение поля внутри группы записей
Max	Максимальное значение поля внутри группы записей
Count	Количество непустых значений поля записей внутри группы
StDev	Среднеквадратичное отклонение значений поля
Var	Дисперсия значений поля внутри группы записей
First	Возвращают значение поля из первой записи результирующего набора запроса
Last	Возвращают значение поля из последней записи результирующего набора запроса

При проектировании итоговых запросов следует понимать, что режим **Группировка** задается для тех полей, которые в формулировке задания звучат со словом **каждый** или **одинаковый**. Например, в задании к запросу фраза для «каждого товара» указывает, что в запросе для поля **Товар** должен быть задан режим **Группировка** в строке **Групповые операции**.

Запрос11: *Определить, сколько раз продавался каждый товар, на какую сумму и в каком объёме.*

Можно воспользоваться ранее созданным **Запросом5**, в котором была рассчитана стоимость товара, или опять выполнить вычисление стоимости, включив в выражение формулу **[Товары]![Цена(руб)] * [УчетПродаж]![КоличествоТовара]**

Бланк Конструктора Запроса11 представлен на рисунке 46.

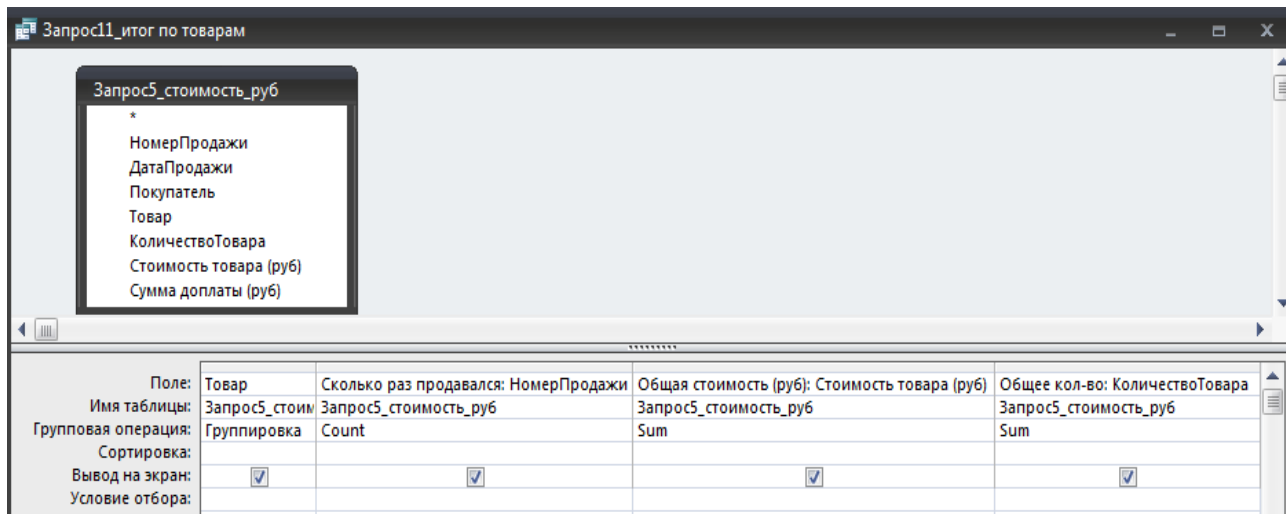


Рисунок 46 – Запрос11 в режиме Конструктора

Запрос12: Определить общее количество проданных товаров каждого производителя по каждой группе товаров.

Бланк Конструктора Запроса12 представлен на рисунке 47.

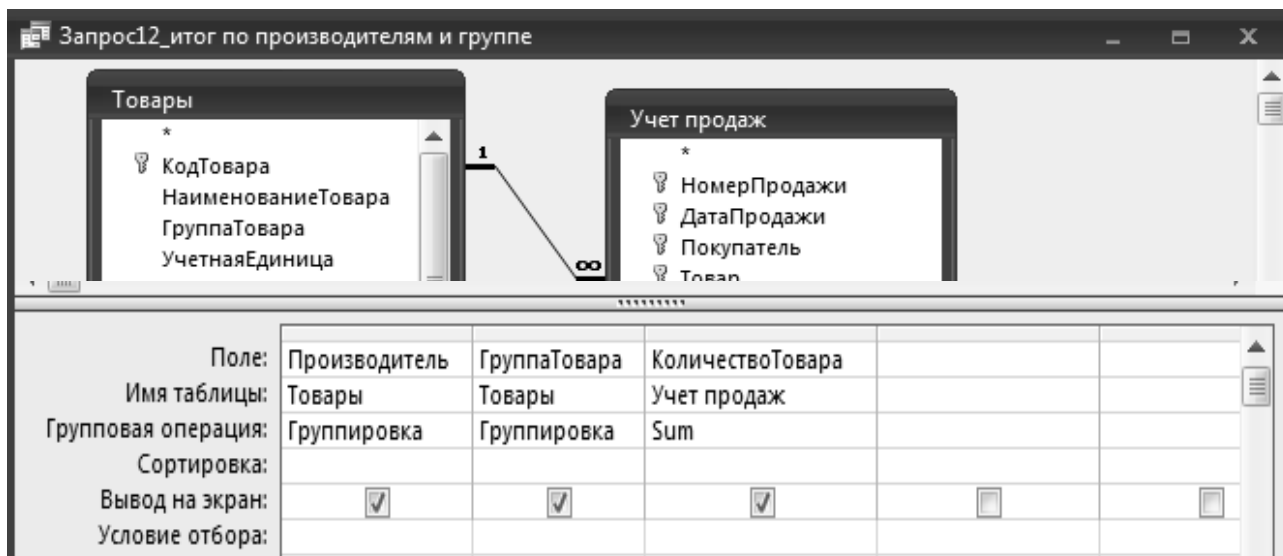


Рисунок 47 – Запрос12 в режиме Конструктора

Запрос13: Определить, на какую сумму продано товаров по каждой группе товаров из категории "Напольные покрытия".

В Запросе13 используется групповая установка **Выражение** для обозначения вычисления значения стоимости товара непосредственно в текущем запросе и обработка его групповой функцией **Sum**:

Общая сумма: $Sum([Товары]![Цена(руб)] * [УчетПродаж]![КоличествоТовара])$
 Бланк Конструктора Запроса13 представлен на рисунке 48.

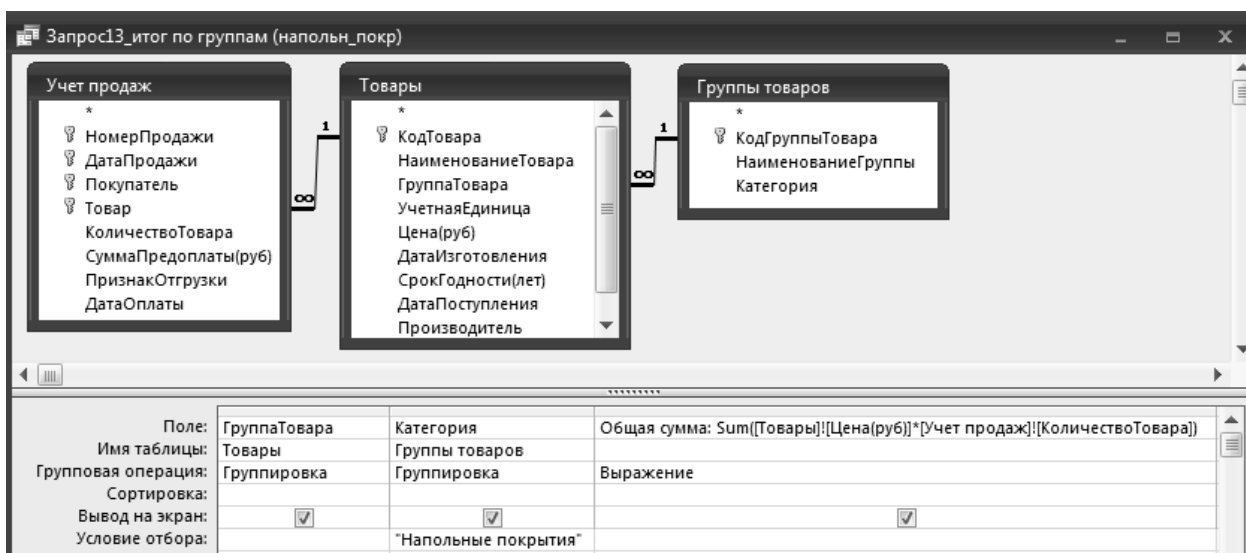


Рисунок 48 – Запрос13 в режиме Конструктора

Запрос14: Определить итоговую сумму продаж за каждый день недели.

В **Запросе14** используется созданный **Запрос5**, в котором была рассчитана стоимость товара, и добавляется новое поле, в котором определяется название дня недели продаж:

День недели: `Format([УчетПродаж].[ДатаПродаж];"dddd")`

Бланк Конструктора **Запроса14** представлен на рисунке 49.

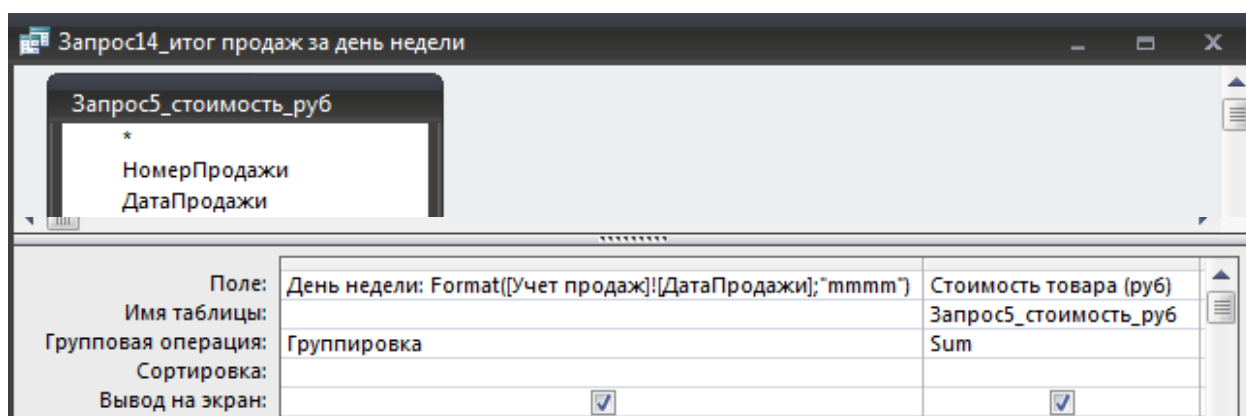


Рисунок 49 – Запрос14 в режиме Конструктора

Запрос15: Определить количество наименований товаров, у которых интересный год изготовления задается с клавиатуры.

В **Запросе15** используется групповая установка **Условие** для обозначения критерия отбора по данному полю, которое при этом не отображается в результате запроса. Критерий отбора выступает в виде параметра. Выражение, построенное в **Построителе** для извлечения значения года поля *Дата изготовления*, имеет вид:

ГодИзготовления: `Year([Товары].[ДатаИзготовления])`

Замечание 1: для извлечения значения года из поля с типом Дата/Время можно использовать другие встроенные функции, синтаксис которых приведен в **Приложении 3** и **Приложении 4:**

Год изготовления: DatePart("yyyy";[Товары]![ДатаИзготовления])

Год изготовления: Format([Товары]![ДатаИзготовления];"yyyy")

Замечание 2: в результате данного запроса будет выведено одно число.

Бланк Конструктора Запроса15 представлен на рисунке 50.

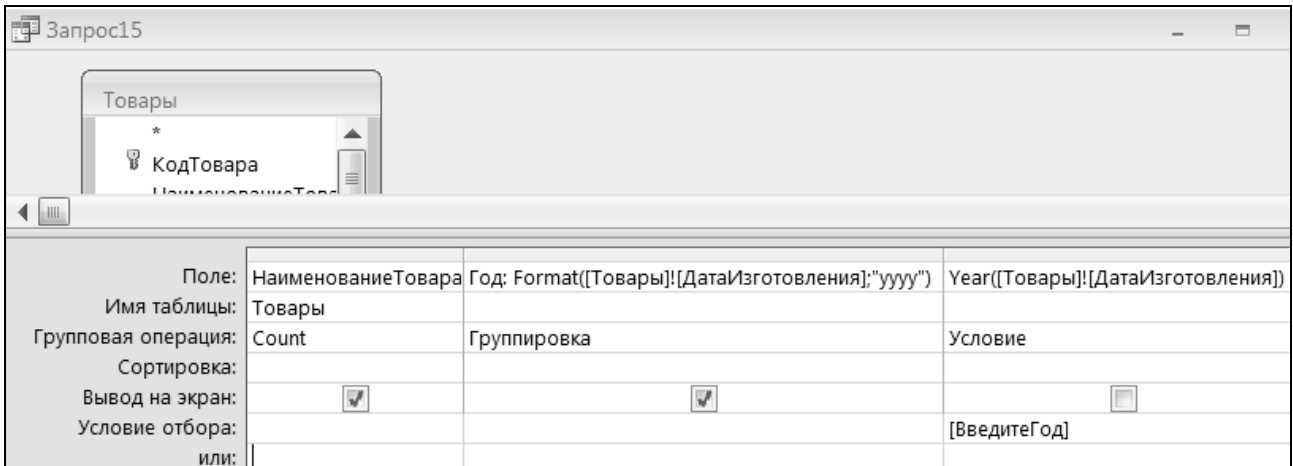


Рисунок 50 – Запрос15 в режиме Конструктора

Запрос16: Определить, на какую сумму было продано товаров за каждый месяц определенного квартала, номер которого вводится с клавиатуры.

Выражения, построенные для преобразования поля *ДатаПродажи*, имеет вид:

Месяц: MonthName(Month([УчетПродаж]![ДатаПродажи]))

Квартал: DatePart("q";[УчетПродаж]![ДатаПродажи])

Замечание: для извлечения значения месяца и квартала из поля с типом Дата/Время можно также использовать встроенную функцию **Format()**, синтаксис которой приведен в **Приложении 4**:

Месяц: Format([УчетПродаж]![ДатаПродажи];"mmmm")

Квартал: Format([УчетПродаж]![ДатаПродажи];"q")

Бланк Конструктора Запроса16 представлен на рисунке 51.

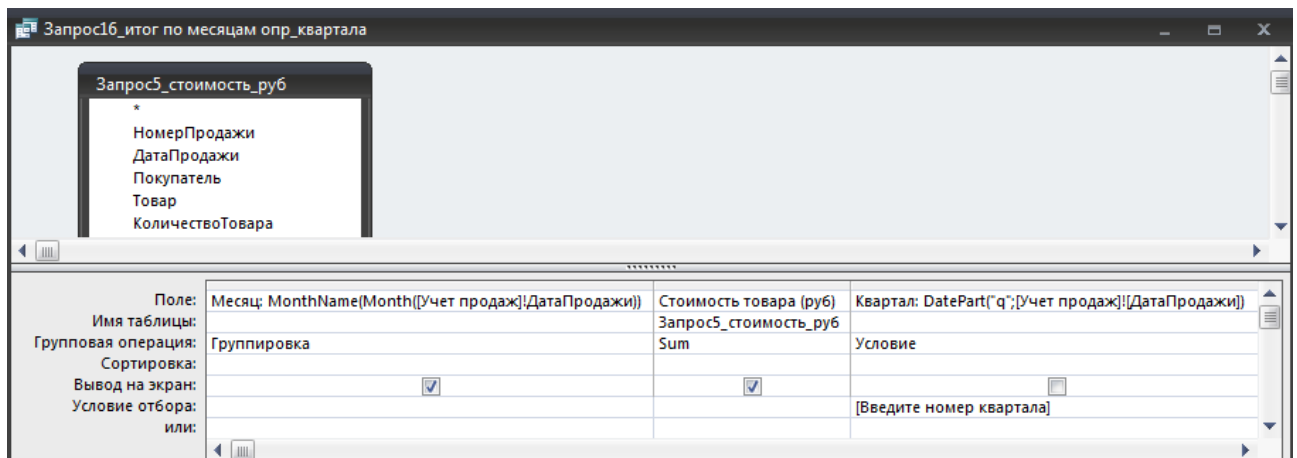


Рисунок 51 – Запрос16 в режиме Конструктора

Запрос17: Подсчитать по каждому покупателю количество неоплаченных покупок.

Покупка считается неоплаченной, если в БД не заполнена дата оплаты. В запросе для задания условия отбора по полю *ДатаОплаты* используется встроенная функция **IsNull** (из категории *Проверка*), которая при обработке пустого поля (не содержащего никаких данных) возвращает значение **ИСТИНА**, в противном случае – значение **ЛОЖЬ**.

Бланк **Конструктора Запроса17** представлен на рисунке 52.

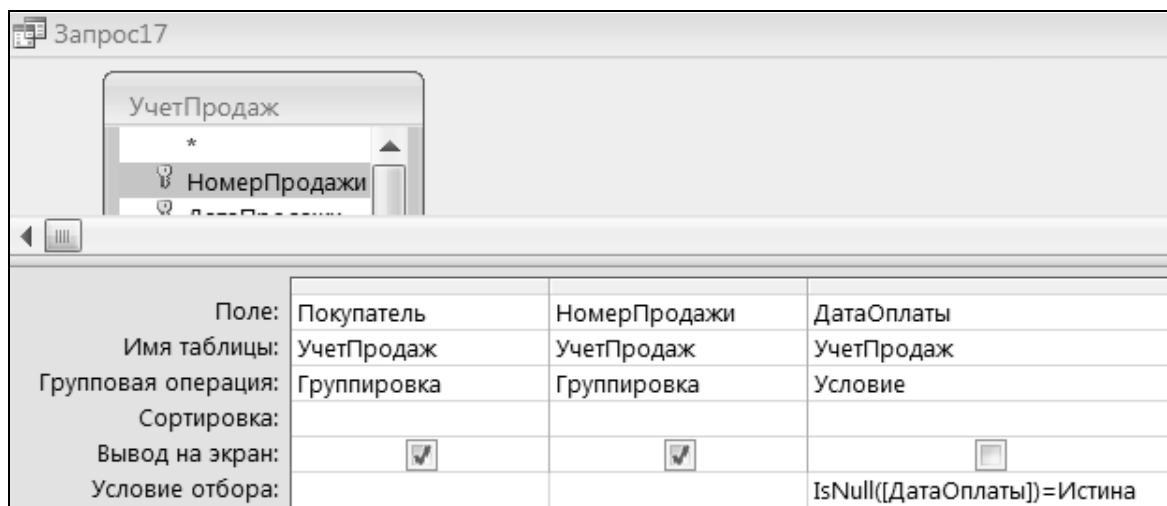


Рисунок 52 – Запрос17 в режиме Конструктора

Практическое задание 9. Перекрестные запросы для БД в MS Access

2.22. Проектирование перекрестных запросов для БД в MS Access

Перекрестный запрос – это запрос, в котором происходит статистическая обработка данных, результаты которой выводятся в виде таблицы, очень похожей на сводную таблицу Excel. Такой запрос подсчитывает сумму, среднее, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк. Перекрестный запрос представляет информацию в более компактном виде, чем обычные запросы. Такие запросы можно создавать как в режиме **Конструктора**, так и с помощью **Мастера**.

Последовательность действий при создании перекрестного запроса **в режиме Конструктора запроса**:

- ✓ сначала создать обычный запрос-выборку (см. пп.1 – 3 раздела 2.14);
- ✓ далее необходимо преобразовать запрос-выборку в **Перекрестный**, для этого:


в версии Access 2003: находясь в окне Конструктора из пункта меню Запрос выбрать команду Перекрестный ;	в версии Access 2007+: на вкладке ленты Конструктор в группе инструментов Тип запроса выбрать пиктограмму Перекрестный
---	---
- ✓ для полей, значения которых будут заголовками строк, выбрать в строке **Перекрестная таблица** опцию **Заголовки строк** и оставить в строке **Групповая операция** значение **Группировка**;
- ✓ для полей, значения которых будут представлены в запросе в качестве заголовков столбцов, выбрать в строке **Перекрестная таблица** опцию **Заголовки столбцов** и оставить в строке **Групповая операция** значение **Группировка**;

✓ для расчетных (итоговых) полей, значения которых будут находиться на пересечении строк и столбцов, выбрать в строке **Перекрестная таблица** опцию **Значение**, а в строке **Групповая операция** из списка выбрать нужную статистическую функцию.

Запрос18: Для каждой группы товаров по каждому производителю подсчитать среднюю цену. В итоговом столбце вывести среднюю цену товаров по каждой группе по всем производителям. Результат запроса представить в виде перекрестной таблицы.

Последовательность действий при создании перекрестного запроса с помощью **Мастера запросов**:

в версии Access 2003: находясь в окне БД выбрать вкладку **Запросы**; нажать кнопку **Создать**;

в версии Access 2007+: на вкладке ленты выбрать кнопку **Создать** в группе инструментов **Другие** (или в группе **Запросы**) выбрать пиктограмму  **Мастер запросов**

✓ в появившемся диалоговом окне **Новый запрос** выбрать опцию **Перекрестный запрос** (рис. 53) и нажать кнопку **ОК**;

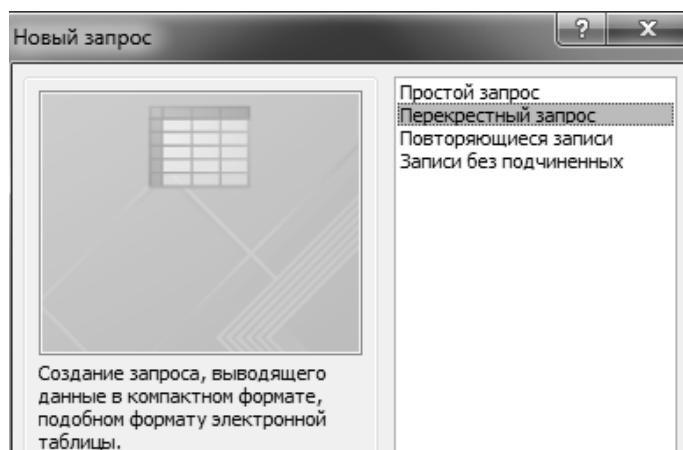


Рисунок 53 – Окно выбора "Новый запрос"

✓ в появившемся окне **Создание перекрестных таблиц** выбрать из списка таблицу-источник **Товары** (рис. 54). Нажать кнопку **Далее**.

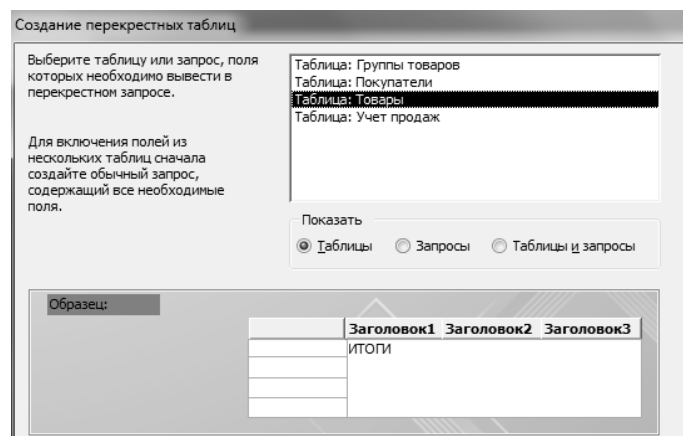


Рисунок 54 – Первый шаг Мастера создания перекрестного запроса

✓ на втором шаге **Мастера** выбрать с помощью кнопки **>** поле *ГруппаТовара*, значения которого будут использованы в качестве заголовков строк (рис. 55). Нажать кнопку **Далее**.

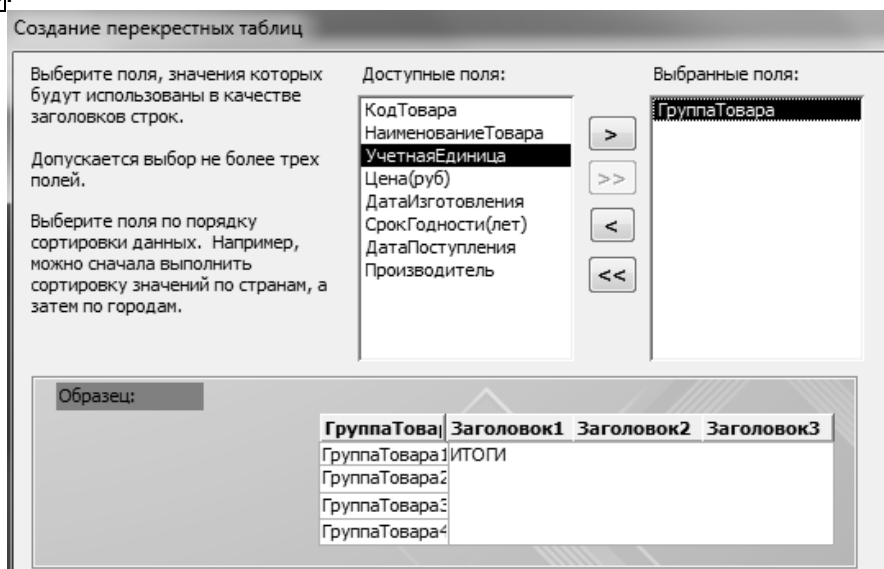


Рисунок 55 – Второй шаг Мастера создания перекрестного запроса

✓ на третьем шаге **Мастера** выбрать поле *Производитель*, значения которого будут использованы в качестве заголовков столбцов (рис. 56). Нажать кнопку **Далее**.

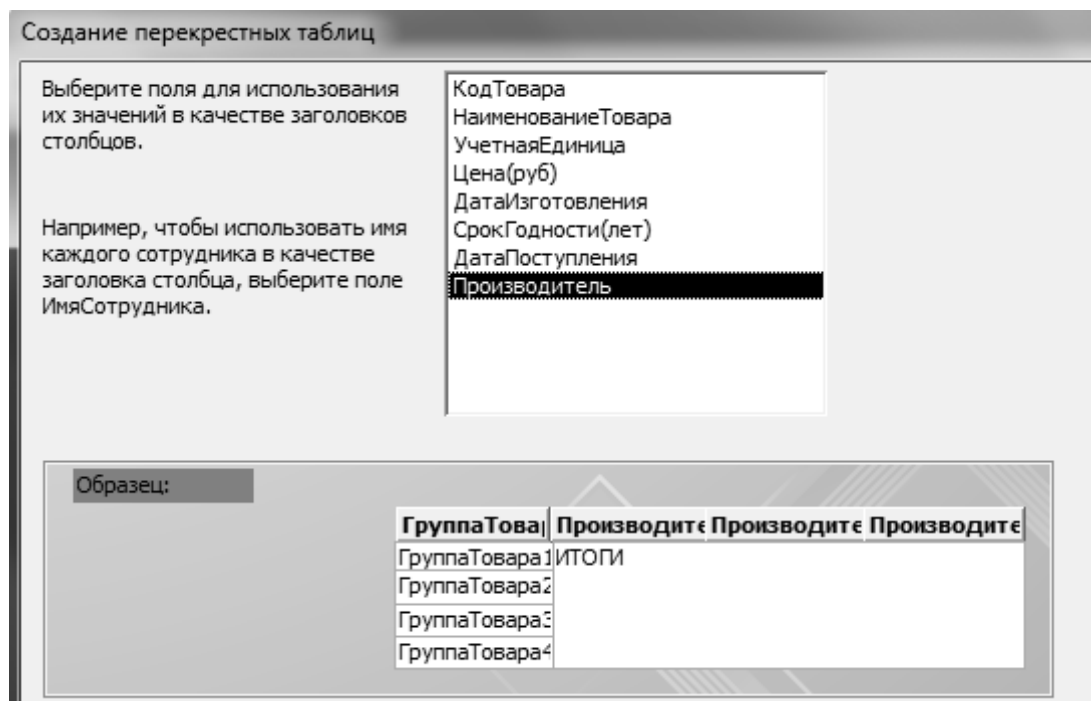


Рисунок 56 – Третий шаг Мастера создания перекрестного запроса

✓ на следующем шаге **Мастера** выбрать сначала поле, по которому будут производиться вычисления на пересечении строк и столбцов, а затем необходимую операцию. На вопрос "Вычислить итоговое значение для каждой строки?" ответить "Да" в виде установленного флажка (рис. 57). Нажать кнопку **Далее**.

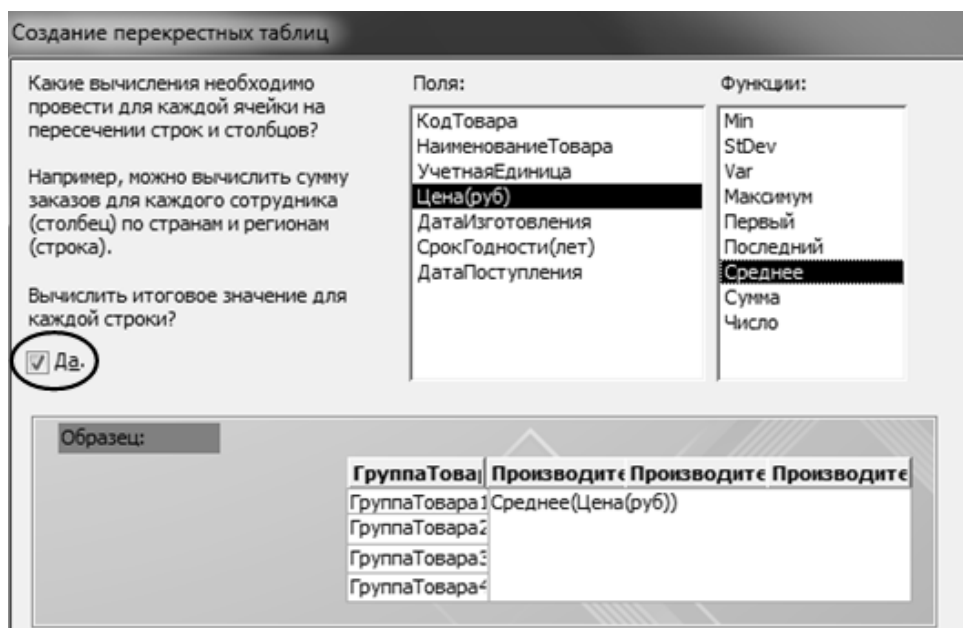


Рисунок 57 – Четвертый шаг Мастера создания перекрестного запроса

✓ на последнем шаге **Мастера** ввести имя запроса и выбрать один из дальнейших вариантов действий: либо перейти в режим **Таблицы** для просмотра результата, либо перейти в режим **Конструктора** для возможного изменения структуры запроса (рис. 58). Нажать кнопку **Готово**.

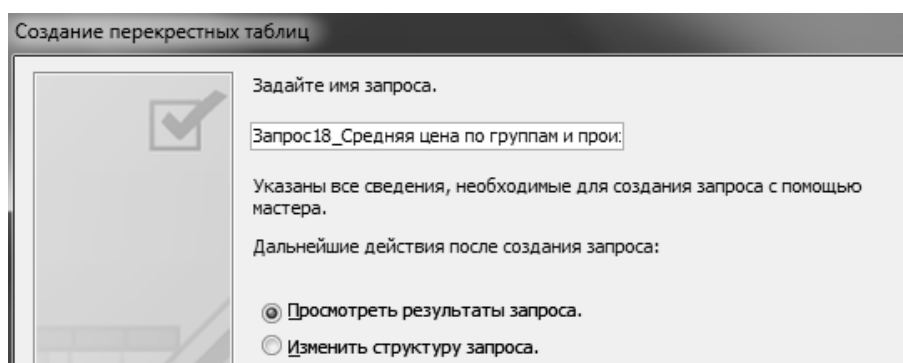


Рисунок 58 – Заключительный шаг Мастера создания перекрестного запроса

Результат запроса в виде таблицы в представлен на рисунке 59:

ГруппаТовара	Итоговое значение Цена(руб)	Беларусь	Германия	Италия	Китай	Литва	Польша	Россия
Ламинат 32 класс	13,00р.							
Ламинат 33 класс	78,00р.		78,00р.					
Паркетная доска	114,00р.		150,00р.					
Линолеум	10,20р.		10,20р.					
Краска водно-дисперсионная	21,30р.	21,30р.						
Краска-эмаль	98,50р.					188,00р.	9,00р.	
Лак для паркета	70,00р.							70,00р.
Обои бумажные	9,00р.				9,00р.			
Обои флизелиновые	53,50р.			53,50р.				
Инструмент слесарный	3,50р.				3,50р.			
Инструмент малярный	4,55р.							

Рисунок 59 – Результат Запроса18 в режиме Таблицы

Бланк Конструктора Запроса18 представлен на рисунке 60.

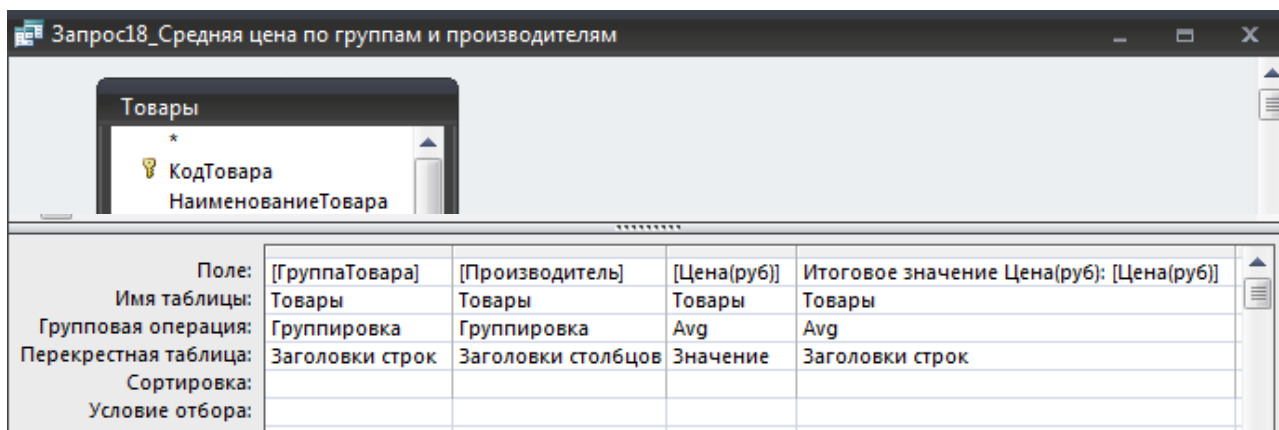


Рисунок 60 – Запрос18 в режиме Конструктора

Запрос19: Подсчитать сумму продаж по каждому товару за каждый месяц. В итоговом столбце вывести максимальное проданное количество каждого товара за все время. Результат запроса представить в виде перекрестной таблицы.

Выражения, построенные в Построителе для преобразования поля *ДатаПродажи* и вычисления *стоимости продаж* имеют вид:

Месяц: MonthName(Month([УчетПродаж]!ДатаПродажи))

Общая стоимость: [УчетПродаж]![КоличествоТовара] * [Товары]![Цена(руб)]

Бланк Конструктора Запроса19 представлен на рисунке 61.

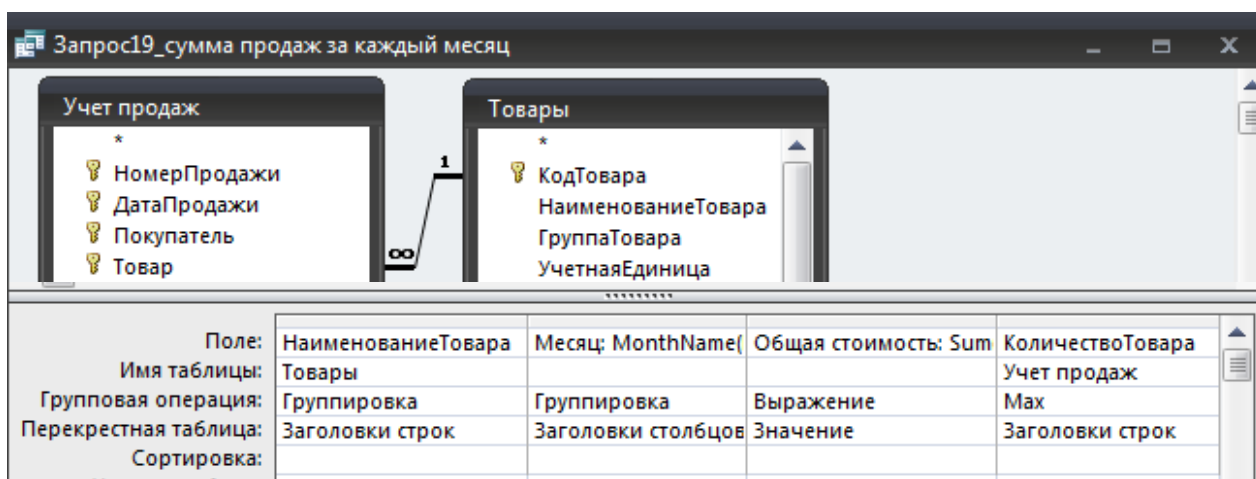


Рисунок 61 – Запрос19 в режиме Конструктора

Запрос20: Подсчитать, сколько раз продавался каждый товар за каждый день недели определенного месяца, название которого вводится с клавиатуры как параметр. Результат запроса представить в виде перекрестной таблицы.

Выражения в **Запросе20**, построенные в Построителе для преобразования поля *ДатаПродажи*, имеют следующий вид:

День недели: WeekdayName(Weekday([УчетПродаж]![ДатаПродажи];2))

Месяц: MonthName(Month([УчетПродаж]![ДатаПродажи]))

Замечание: для определения типа данных параметра необходимо его добавить в окно **Параметры запроса** (рис. 62), которое открывается следующим образом:
в версии Access 2003: выбрать команду **Параметры** из пункта меню **Запрос**.
в версии Access 2007+: на вкладке ленты **Создание** в группе инструментов **Показать или скрыть** (или в группе **Запросы**) выбрать пиктограмму **Параметры**.

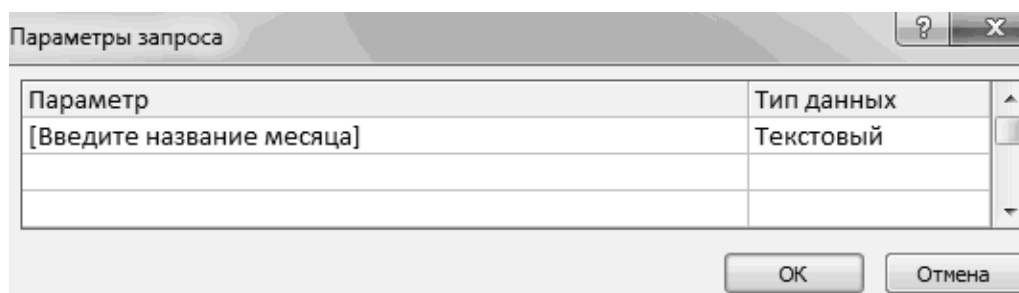


Рисунок 62 –. Окно **Параметры запроса**

Бланк **Конструктора Запроса20** представлен на рисунке 63.

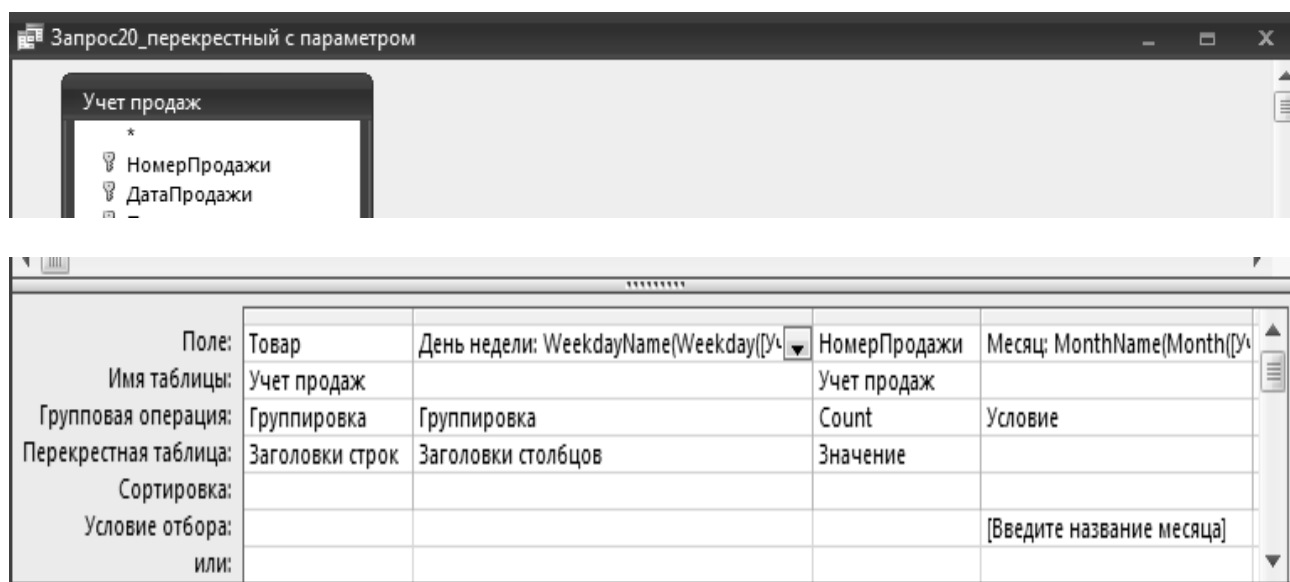


Рисунок 63 –. **Запрос20** в режиме **Конструктора**

Практическое задание 10. Проектирование форм для БД в MS Access

2.23. Способы создания форм в MS Access

Формы – это диалоговый графический интерфейс для работы пользователя с БД. Любая форма, с помощью которой надо просматривать, вводить или редактировать записи таблиц БД, должна быть предварительно сконструирована.

Возможны следующие способы создания форм:

- **Автоформа** (полностью автоматизированное средство);
- с помощью **Мастера** (в режиме диалога);
- в режиме **Конструктора**;
- **Диаграмма** – создание формы с диаграммой;
- **Сводная таблица** – создание формы со сводной таблицей MS Excel.

2.24. Создание формы с помощью инструмента Автоформа

Порядок создания Автоформы следующий:

в версии Access 2003:

1 способ: находясь в окне БД на вкладке **Таблицы** выделить таблицу, для которой создаётся форма, и выбрать команду **Автоформа** из п. меню **Вставка** (или кнопку «Новый объект» на панели инструментов)

2 способ: в окне БД на вкладке **Формы** нажать кнопку **Создать**; в окне **Новая форма** задать режим – **Автоформа: в столбец**, **Автоформа: ленточная**, **Автоформа: табличная** и пр.; выбрать из списка в качестве источника данных для формы таблицу или запрос.

в версии Access 2007+:

находясь в левой части окна приложения в области переходов в списке объектов **Таблицы** выделить таблицу, для которой создаётся форма, и на вкладке ленты **Создание** в группе инструментов **Формы** выбрать пиктограмму



Формы

Задача: создать форму по таблице **Учет продаж** для ведения учета.

2.25. Создание формы с помощью Мастера форм

Для создания формы с помощью Мастера выполняется следующая последовательность шагов:

1) в версии Access 2003: находясь в окне БД на вкладке **Формы** нажать кнопку **Создать**; в окне **Новая форма** задать режим **Мастер форм**;

1) в версии Access 2007+: на вкладке ленты **Создание** в группе инструментов **Формы** (может находиться в дополнительном меню **Другие формы**) выбрать пиктограмму



Мастер форм;

2) следовать шагам Мастера в окне **Создание форм**:

на 1-м шаге: сначала выбрать из раскрывающегося списка в качестве источника данных для формы таблицу **Учёт продаж**, затем выбрать все поля для формы посредством кнопки **>>** (рис. 64), нажать кнопку **Далее**;

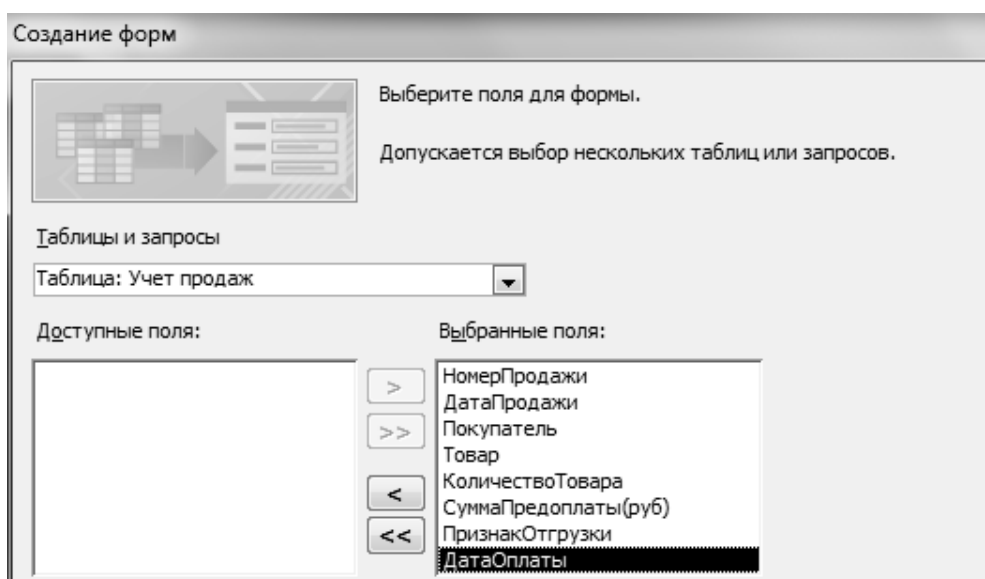


Рисунок 64 – Первый шаг Мастера форм

на 2-м шаге: выбрать внешний вид формы, например, "в один столбец", нажать кнопку **Далее**;

на 3-м шаге: выбрать стиль оформления формы, например, "Стандартная", нажать кнопку **Далее**;

на 4-м шаге: задать имя форме (в нашем случае – Учет продаж) и выбрать один из вариантов дальнейших действий: либо открыть форму для просмотра данных, либо переключиться в режим Конструктора для дальнейшего её редактирования, нажать кнопку **Готово**.

В результате получится вид формы для первой записи, представленный на рисунке 65.


Поле	Значение
НомерПродажи	1
ДатаПродажи	13.05.2016
Покупатель	Магазин "Верас"
Товар	Паркет ТЕМРА
КоличествоТовара	45
СуммаПредоплаты(руб)	3 000,00
ПризнакОтгрузки	<input checked="" type="checkbox"/>
ДатаОплаты	15.05.2016

Рисунок 65 – Вид формы в режиме просмотра данных

Просматривать записи в форме можно используя кнопки внизу окна формы.

2.26. Создание формы в режиме Конструктора

Для создания формы в режиме Конструктора необходимо:

в версии Access 2003: находясь в окне в версии Access 2007+: на вкладке ленты БД на вкладке **Формы** нажать кнопку **Создание** в группе инструментов **Формы** **Создать**; в окне **Новая форма** задать выбрать пиктограмму режим **Конструктор**.  **Конструктор форм**).

Процесс создания формы состоит:

- ✓ в размещении объектов в форме (текстовой информации; полей ввода; кнопок управления; рисунков или других графических изображений);
- ✓ определении свойств для размещенных объектов, а также связанных с ними событий и выполняемых действий;
- ✓ определении свойств самой формы как объекта (размера, возможности использования формы либо для просмотра данных, либо для их ввода и редактирования и пр.).

2.27. Структура формы

В режиме Конструктора может производиться настройка **областей** и **объектов** формы.

Области окна формы (рис. 66):

- ✓ область заголовка;
- ✓ область верхнего колонтитула;
- ✓ область данных;
- ✓ область нижнего колонтитула;
- ✓ область примечаний.

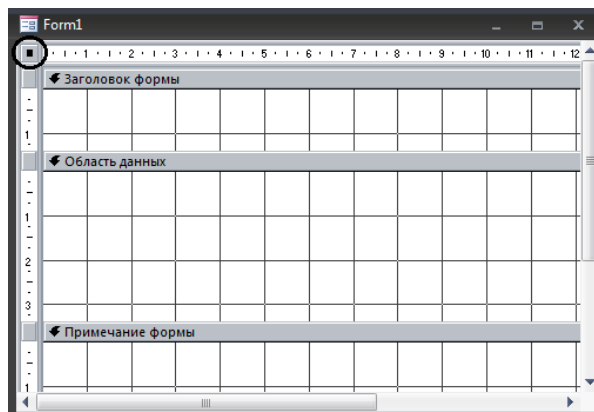

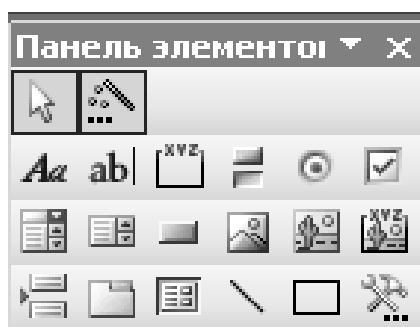


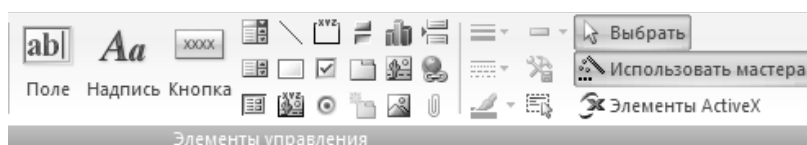
Рисунок 66 – Области окна формы в Конструкторе

Замечание: для отображения областей нужно выбрать соответствующие команды из контекстного меню, вызываемого щелчком правой кнопки мыши по области формы.

Элементы управления (ЭУ) в форме размещаются с помощью **Панели элементов управления**, которая в версии Access 2003 вызывается из пункта меню **Вид** или пиктограммы  на панели инструментов, а в версии Access 2007+ располагается на вкладке ленты **Конструктор** (рис. 67).



версия Access 2003



версия Access 2007+

Рисунок 67 – Панель элементов управления

Для размещения на форме нового ЭУ его нужно выбрать щелчком мыши на панели элементов, перейти на бланк конструктора, при этом курсор должен принять форму крестика +, и нарисовать прямоугольник, удерживая левую кнопку мыши. В этом прямоугольнике будет размещен созданный ЭУ.

В Access существуют следующие типы элементов управления (ЭУ):

- ✓ **Выбор объектов.** Позволяет выделить определённый элемент управления или группу ЭУ, если держать нажатой клавишу **Shift**.
- ✓ **Мастера.** Если эта кнопка нажата, то при размещении на форме ЭУ запускается мастер, помогающий задать параметры элемента.
- ✓ **Надпись.** Описательный текст.
- ✓ **Текстовое поле** для ввода и редактирования текста.
- ✓ **Группа переключателей.**
- ✓ **Выключатель.** Может быть в двух состояниях: включено и выключено.


- ✓ **Переключатель.** Несколько переключателей обычно объединяются в группу и позволяют выбрать одно из взаимоисключающих значений, например муж или жен.
- ✓ **Флажок.** Используется для включения и отключения параметра.
- ✓ **Поле со списком.** В поле можно ввести новое значение или выбрать существующее из списка.
- ✓ **Список,** содержит значения, из которых можно сделать выбор.
- ✓ **Кнопка.** Используется для выполнения определенного действия или ряда действий.
- ✓ **Рисунок.** Не меняется при переходе от одной записи к другой.
- ✓ **Свободная рамка объекта.** Может отображать объект OLE.
- ✓ **Присоединённая рамка объекта.** Отображает объекты OLE, хранящиеся в записях таблиц, например, рисунки, фотографии.
- ✓ **Разрыв страницы.**
- ✓ **Набор вкладок.** Позволяет разделить форму на несколько вкладок.
- ✓ **Подчинённая форма/отчёт.** В форму добавляется информация из дополнительной таблицы.
- ✓ **Линия.**
- ✓ **Прямоугольник.**
- ✓ **Другие элементы, в том числе ActiveX.** Позволяют добавить формам и отчетам ещё некоторые функциональные возможности.

2.28. Свойства и события объектов формы


Как и форма в целом, так и каждый из ее элементов обладает свойствами, которые позволяют определить его внешний вид, размер, местоположение в форме, режим ввода/вывода, а также привязать к элементу выражение, макрос или программу (т.е. встроенные события, которые выполняются при наступлении связанных с ЭУ действий).

Для того, чтобы получить доступ к свойствам и событиям объекта формы необходимо выделить нужный объект, а затем:

в версии Access 2003: выбрать команду **Свойства** из пункта меню **Вид** или нажать соответствующую кнопку на панели инструментов.


в версии Access 2007+: на вкладке ленты **Конструктор** в группе инструментов **Сервис** выбрать пиктограмму  **Страница свойств.**

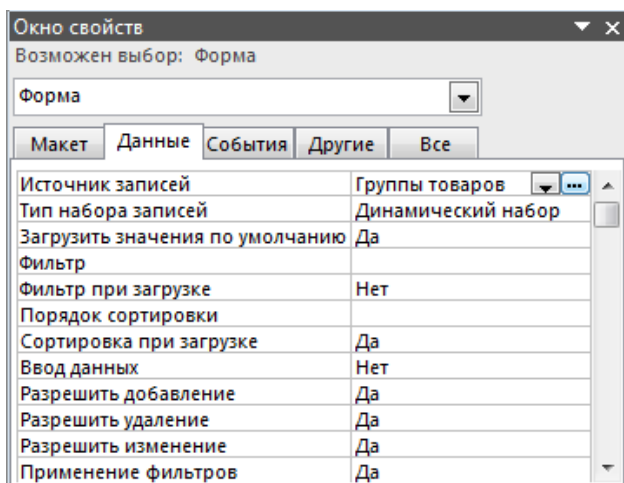
Замечание 1: для отображения окна свойств выбирается команда **Свойства** из контекстного меню, вызываемого щелчком правой кнопки мыши по выделенному объекту.

Замечание 2: для отображения окна свойств самой формы, необходимо выбрать команду **Свойства** из контекстного меню, вызываемого щелчком правой кнопки мыши по кнопке , расположенной в левом верхнем углу формы на пересечении горизонтальной и вертикальной линеек.

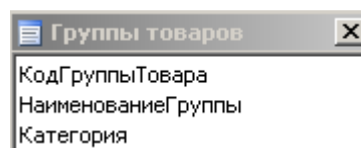
В окне **Свойств** все свойства разбиты по группам на вкладках **Макет, Данные, События, Другие.**

2.29. Размещение элемента управления "Поле" в форме

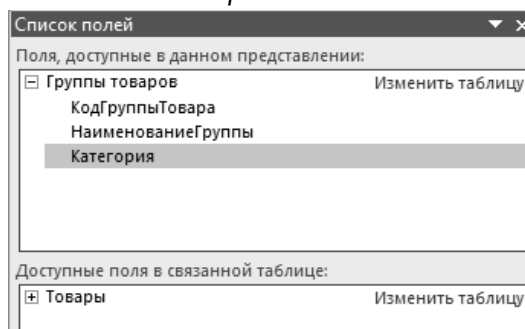
При создании формы в режиме Конструктора, необходимо связать её с источником данных. Для этого, в свойстве формы (см. **замечание 2** предыдущей темы) на вкладке **Данные** в строке **Источник записей** с помощью элемента управления "список" () выбрать источник данных, т.е. таблицу или запрос (рис. 68-а).



а)



в версии ACCESS 2003





в версии ACCESS 2007+

б)

Рисунок 68 – Окно свойств формы и список полей источника данных

Далее в окне **Список полей** (рис. 68-б) отобразятся поля текущего источника данных, которые необходимо переместить методом "перетащить и бросить" в область данных формы. Такие поля называются **присоединенными**, поскольку они связаны с данными таблиц или запросов.

Замечание: для открытия окна **Список полей**:

в версии Access 2003 вызывается в Версии ACCESS 2007+ на вкладке ленты соответствующая команда из пункта **Конструктор** в группе инструментов **Сервис** меню **Вид** или на панели инструментов выбирается пиктограмма  **пиктограмма**  **Добавить поля**

Кроме присоединенных полей, существуют **свободные** поля, которые, например, могут использоваться для отображения результатов вычислений или для приёма данных, вводимых пользователем.


Задача: в созданной форме **Учет продаж** добавить вычисляемые поля: в области заголовка – поле текущей даты, а в области данных – поле для отображения дня недели продаж.

Открыть форму **Учет продаж** в режиме **Конструктора**. Способ переключения между режимами работы приведены в п. 1.13. практической части (для Access 2007+ – рисунок 7-в).

Для дальнейшего редактирования формы, т.е. добавления дополнительных ЭУ, возможно необходимо будет выполнить следующие манипуляции:


✓ изменить размеры окна **Конструктора** с помощью стрелок \leftrightarrow , \updownarrow на границах окна;

✓ изменить размеры области формы (область заголовка, данных, примечания) с помощью стрелок \updownarrow , \leftrightarrow на границах области;

✓ переместить при необходимости уже имеющиеся выделенные поля формы, если два отдельных ЭУ (надпись и поле ввода) образуют связанную группу: подвести указатель к границе до появления «ладони»  (или символа \updownarrow), зажать кнопку мыши, перетащить в другое место и отпустить;

- ✓ переместить при необходимости отдельно надпись или поле ввода с помощью маркера перемещения – квадратика ■ – в верхнем левом углу элемента;
- ✓ изменить размеры выделенного ЭУ, используя маркеры размеров, расположенные на углах и серединах границ элемента;
- ✓ изменить внешний вид элемента управления (цвет шрифта, границы, фона).

Порядок создания свободных (вычисляемых) полей в форме:

1. Выбрать на **Панели элементов управления** элемент **Поле** () , нажать мышью место в нужной области формы, в которой предполагается разместить данный элемент. В форме появится связанный объект, состоящий из выделенного поля ввода и его надписи (рис. 69).

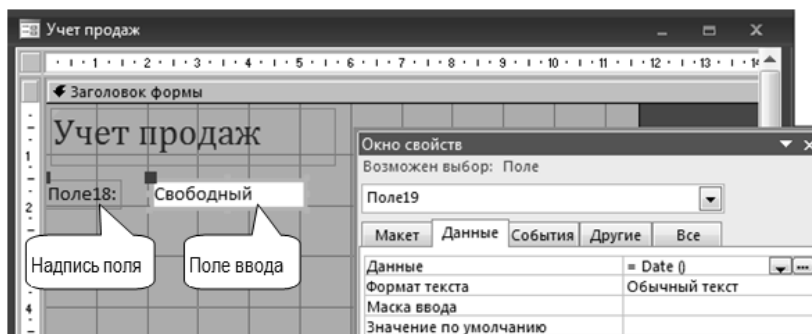



Рисунок 69 – Создание вычисляемого поля в форме

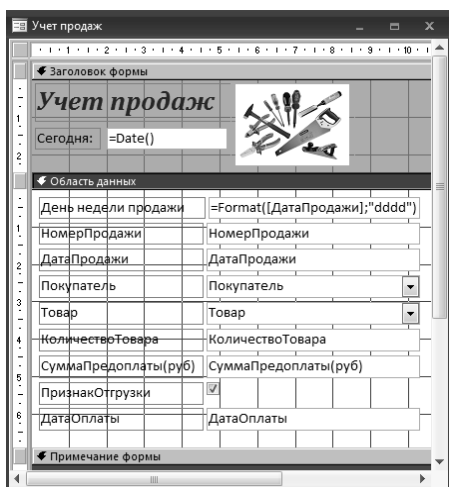
2. Далее для выделенного нового поля ввода необходимо открыть окно **Свойства** (см. п. 2.28. практической части), в котором на вкладке **Данные** выбрать свойство **Данные** (рис. 70). Для добавления выражения с использованием встроенных функций целесообразно использовать **Построитель выражений**, который можно открыть кнопкой  в правой части строки свойства **Данные** (использование **Построителя выражений** см. в п. 2.15 практической части).

3. Затем выделить надпись к полю ввода, открыть окно **Свойства** для этого элемента, чтобы ввести в свойстве **Подпись** подходящий текст с названием к полю ввода.

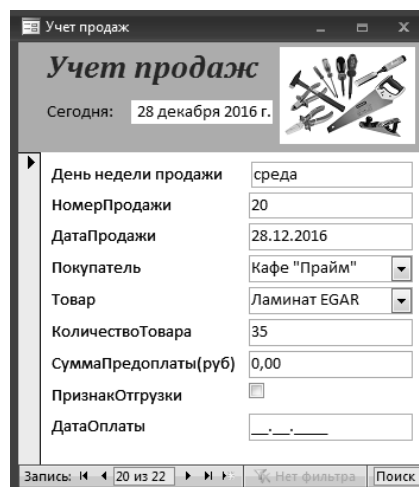
4. Закрыть окно свойств и откорректировать ширину поля и надписи по содержимому.

5. Перейти в режим **Формы** и просмотреть записи, используя кнопки внизу окна формы.

Отредактированная в режиме **Конструктора** форма **Учет продаж** и её вид в режиме **Формы** для одной из записей, представлен на рисунке 70.



Режим Конструктора



Режим формы

Рисунок 70 – Отредактированная форма Учет продаж


Задача: Создать многотабличную форму для просмотра и ввода записей в таблице **Товары** и **УчетПродаж**.

2.30. Разработка многотабличных (составных) форм в MS Access

Многотабличная форма создается для работы на основе нескольких взаимосвязанных (как правило, связью "Один-ко-многим") таблиц и может состоять из одной формы или из основной и одной или нескольких подчиненных вложенных форм. Такой вид формы может быть создан в режиме **Конструктора форм** или с помощью **Мастера форм**. Однако в Access наиболее технологичным является способ первоначального создания форм с помощью **Мастера** и доработка их в режиме **Конструктора**.

Порядок создания многотабличной формы с помощью Мастера:

1) в версии Access 2003: находясь в окне БД на вкладке **Формы** нажать кнопку **Создать**; в окне **Новая форма** задать режим **Мастер форм**;

1) в версии Access 2007+: на вкладке ленты **Создание** в группе инструментов **Формы** (может находиться в дополнительном меню **Другие формы**) выбрать пиктограмму  **Мастер форм**;

2) следовать шагам Мастера в окне **Создание форм**:

на 1-м шаге: сначала выбрать из раскрывающегося списка "Таблицы и запросы" в качестве источника данных для формы главную таблицу-справочник **Товары** и выбрать все поля этой таблицы в область "Выбранные поля" посредством кнопки **>>**, затем на этом же шаге выбрать таблицу **Учёт продаж** и выбрать все поля этой таблицы в область "Выбранные поля" посредством кнопки **>>**, нажать кнопку **Далее**;

на 2-м шаге: выбрать вид представления данных – **Подчинённые формы** (рис. 71), нажать кнопку **Далее**;

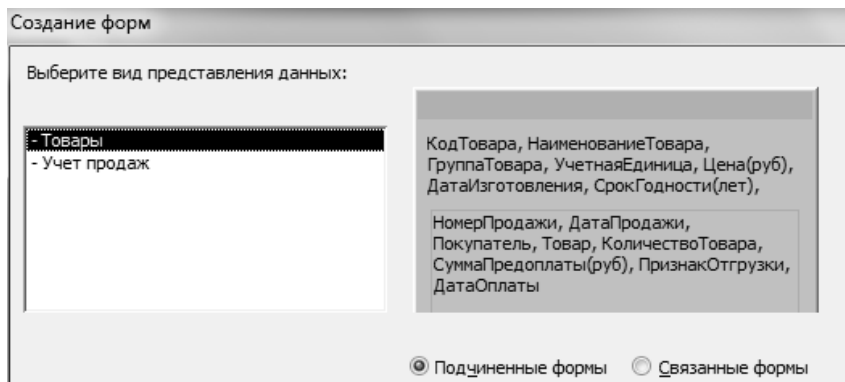


Рисунок 71 – Второй шаг Мастера создания многотабличной формы

на 3-м шаге: выбрать внешний вид *подчиненной* формы, например, "Табличный", нажать кнопку **Далее**;

на 4-м шаге: выбрать стиль оформления формы, например, "Стандартная", нажать кнопку **Далее**;


на 5-м шаге: задать имя двум формам: основной и подчиненной. На этом же шаге выбрать один из вариантов дальнейших действий: либо открыть форму для просмотра данных, либо переключиться в режим Конструктора для дальнейшего её редактирования, нажать кнопку **Готово**.

В результате получим вид формы, в которой можно просматривать информацию о каждом товаре в отдельности, а также информацию о его продажах (рис. 72).

Учет продаж	НомерПродаж	ДатаПродажи	Покупатель	Товар	Количество
	2	13.06.2016	Золотарев И.И.	Обои GRAND	3
	15	04.10.2016	Сидорчук П.О.	Обои GRAND	6
	19	18.11.2016	ИП Сидоров С.С.	Обои GRAND	6

Рисунок 72 – Вид многотабличной формы в режиме просмотра данных

Просматривать записи в форме можно используя кнопки внизу окна формы.

Замечание: в версии Access 2007+ составную форму для главных таблиц (участвующих в связях в отношении "ОДИН") можно создавать автоматически, выделив её в списке объектов БД и выбрав на вкладке ленты **Создание** в группе инструментов **Формы** пиктограмму  **Формы**.

Задача: Создать отчет о продажах в разрезе покупателей.

2.31. Проектирование отчетов для БД в MS Access

Способы создания отчетов:

- Автоотчет (автоматизированное средство);
- с помощью Мастера (в режиме диалога);
- в режиме Конструктора;
- Диаграмма – создание отчета с диаграммой;
- Почтовые наклейки – создание отчетов для почтовых наклеек.

Порядок создания отчета с помощью **Мастера**:

1) в версии Access 2003: находясь в окне БД на вкладке **Отчеты** нажать кнопку **Создать**; далее в окне **Новый отчет** задать режим **Мастер отчетов** и выбрать из раскрывающегося списка в качестве источника данных для отчета **таблицу** или **запрос**;

1) в версии Access 2007+: на вкладке ленты **Создание** в группе инструментов **Отчеты** выбрать пиктограмму

 **Мастер отчетов**;

2) следовать шагам **Мастера отчетов** в окне **Создание отчета**:

на 1-м шаге – выбрать поля для отчета (кроме полей выбранной таблицы, например, можно добавлять числовые поля из связанных таблиц для подведения итогов);

на 2-м шаге – указать поле, по которому должна осуществляться группировка записей в отчете (в нашем случае – поле *Покупатель*);

на 3-м шаге – выбор порядка сортировки в группах и задания итогов по группам, используя кнопку **Итоги**;

на 4-м шаге – выбрать макет отчета;

на 5-м шаге – выбрать стиль для отчета;

на 6-м шаге – задать имя отчету (можно именем таблицы-источника).

Пример отчета о продажах по покупателям приведен на рисунке 73.

Отчет по покупателям


ИмяПокупателя	НомерПродажи	ДатаПродажи	Товар	КоличествоТовара	Стоимость товара (руб)
Магазин "Верас'	1	13.05.2016	Паркет ТЕМПА	45	6 750,00р.
Итоги для 'КодПокупателя' = 1010 (1 запись)					
Sum				45	6 750,00р.
Золотарев И.И.	2	13.06.2016	Обои GRAND	3	160,50р.
	17	13.11.2016	Обои WILTAN	4	36,00р.
Итоги для 'КодПокупателя' = 1101 (2 записей)					
Sum				7	196,50р.

Рисунок 73 – Отчет о продажах в разрезе покупателей

Задача: Создать кнопочную форму для управления работой с БД.

2.32. Создание управляющей (кнопочной) формы для работы с БД

1. Создать пустую форму (т.е. без привязки к источнику данных) в режиме **Конструктора** (см. п. 2.26. практической части).

2. Добавить в область данных текст заголовка с помощью ЭУ **Надпись** , изменив при необходимости её оформление (цвет шрифта, фона, границы, размеры и т.д.)

3. На **Панели элементов управления** активизировать кнопку **Мастера** .

4. Далее добавлять на форму кнопки управления в следующем порядке:

4.1. выбрать ЭУ **Кнопка** () на Панели ЭУ и начертить с помощью мыши прямоугольник в области данных – загрузится **Мастер создания кнопок**.

4.2. в окне **Создание кнопок** выполнить следующие действия:

на 1-м шаге выбрать необходимую опцию из списка **Категории**, например, **Работа с формой**, а из списка **Действия** – конкретное действие, например, **Открыть форму** (рис. 74), нажать кнопку **Далее**;

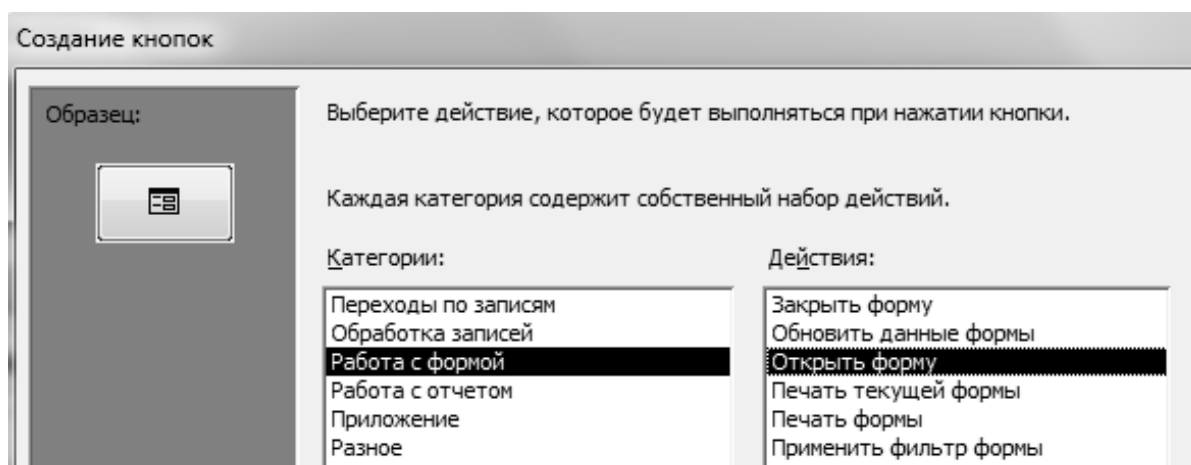


Рисунок 74 – Первый шаг Мастера создания кнопок

на 2-м шаге выбрать открываемый объект, например, **составную форму** (рис. 75), нажать кнопку **Далее**;

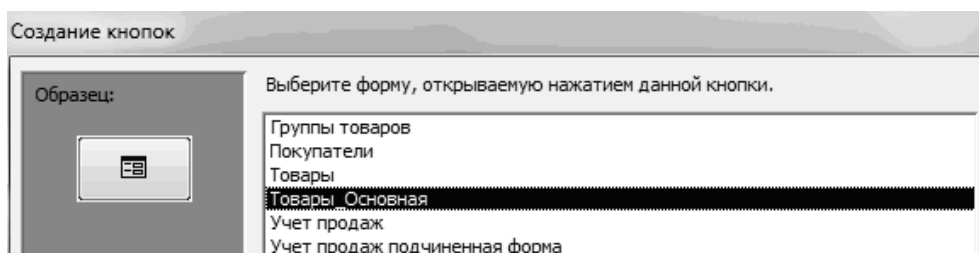


Рисунок 75 – Второй шаг Мастера создания кнопок

на 3-м шаге выбрать один из двух вариантов отображения данных в форме – открыть форму для **отобранных** записей или с выводом **всех** записей, нажать кнопку **Далее**;

на 4-м шаге выбрать, что будет размещено на кнопке: текст или рисунок, нажать кнопку **Далее**;

на 5-м шаге выбрать имя для кнопки, нажать кнопку **Готово**.

5. Можно упорядочить и красиво разместить кнопки на форме (предварительно выдлив их всех или их часть при нажатой клавише **SHIFT**) с помощью команд **выравнивания** (по верхнему краю, по нижнему краю и пр.), и команд определения **размера** (по самому высокому, по самому широкому и пр.), которые находятся в контекстном меню для выделенных объектов.

6. Сохранить форму (например, под именем **Управляющая форма**).

Пример управляющей формы для БД Магазин СТРОИТЕЛЬ приведен на рисунке 76.

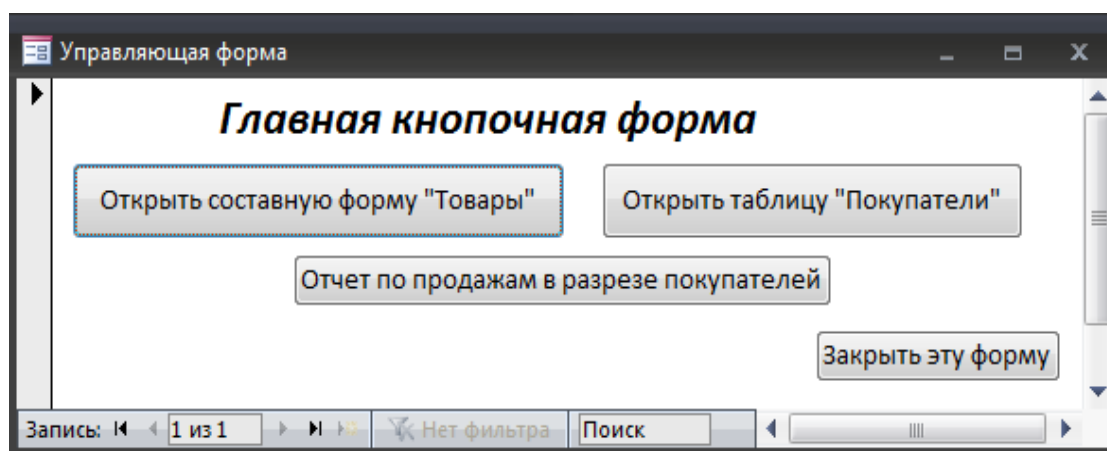



Рисунок 76 – Управляющая кнопочная форма БД Магазин СТРОИТЕЛЬ

2.33. Использование макросов в MS Access

Макрос – это средство для автоматизации задач и добавления функциональных возможностей формы, отчеты и элементы управления без программирования. В MS Access макросы записываются в виде набора макрокоманд для выполнения.

Макрос создается с помощью **Конструктора макросов**, который открывается следующим образом:

в версии Access 2003: в окне БД на вкладке **Макросы** кнопкой **Создать**

в версии Access 2007+: на вкладке ленты **Создание** в группе инструментов **Другие** (или в группе **Макросы и код**) пиктограммой  **Макрос**.

Пример создания макроса для открытия таблицы **Покупатели** приведен на рисунке 77.

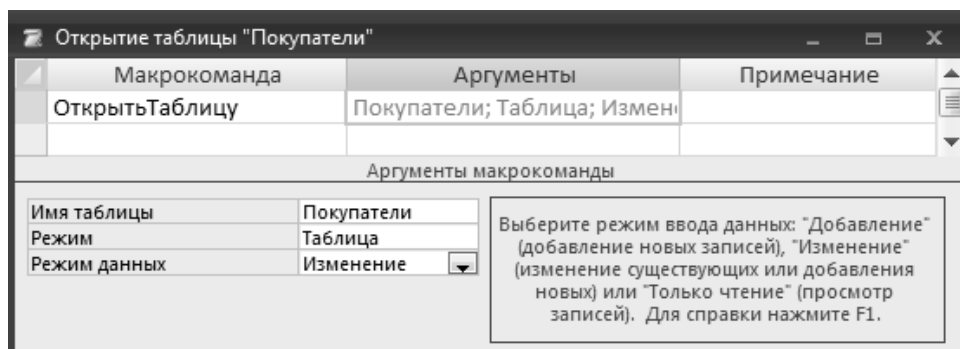


Рисунок 77 – Окно Конструктора макроса

Замечание: в **Мастере создания кнопок** может не быть обработок некоторых действий (например, открытие таблицы), поэтому необходимо создавать для этого макрос, который будет находиться в категории "Разное" в **Мастере** для выбора действия, выполняемого при нажатии кнопки.

ПРИЛОЖЕНИЕ 1. Описание структур записей таблиц БД *Магазин СТРОИТЕЛЬ*
Таблица1 «ГруппыТоваров»

Имя поля	Тип данных	Размер	Другие свойства
КодГруппыТовара	Числовой	Длинное целое	Ключ
НаименованиеГруппы	Текстовый	50	Обязательное поле; Индексированное – Да (Совпадения не допускаются)
Категория	Текстовый	50	Обязательное поле; Индексированное – Да (Допускаются совпадения); <i>Подстановка:</i> Тип элемента управления – Поле со списком

Таблица2 «Товары»

Имя поля	Тип данных	Размер	Другие свойства
КодТовара	Счетчик	Длинное целое	Ключ
НаименованиеТовара	Текстовый	255	Обязательное поле; Индексированное – Да (Допускаются совпадения)
ГруппаТовара	Числовой	Длинное целое	Обязательное поле; Индексированное – Да (Допускаются совпадения)
УчетнаяЕдиница	Текстовый	25	Обязательное поле; <i>Подстановка:</i> Тип элемента управления – Поле со списком
Цена(руб)	Денежный		Формат поля – С разделителями разрядов; Число десятичных знаков – 2; Обязательное поле;
ДатаИзготовления	Дата/Время		Обязательное поле; Маска ввода – краткий формат даты (00.00.0000;0;_)
СрокГодности(лет)	Числовой	Одинарное с плавающей точкой	Обязательное поле; Число десятичных знаков – 1
ДатаПоступления	Дата/Время		Обязательное поле; Маска ввода – краткий формат даты (00.00.0000;0;_)
Производитель	Текстовый	50	Обязательное поле; <i>Подстановка:</i> Тип элемента управления – Поле со списком
ОписаниеТовара	Поле MEMO		

Таблица3 «Покупатели»

Имя поля	Тип данных	Размер	Другие свойства
КодПокупателя	Счетчик	Длинное целое	Ключ
ИмяПокупателя	Текстовый	30	Обязательное поле; Индексированное – Да (Совпадения не допускаются)
Адрес	Текстовый	255	Обязательное поле
Телефон/Факс	Текстовый	15	Маска ввода "(8 0"0099") "999\~99\~99;0;*

Таблица4 «УчетПродаж»

Имя поля	Тип данных	Размер	Другие свойства	
<u>НомерПродажи</u>	Составной ключ	Числовой	Длинное целое	Обязательное поле; Индексированное – Да (Допускаются совпадения)
<u>ДатаПродажи</u>		Дата/время		Обязательное поле; Маска ввода – краткий формат даты (00.00.0000;0;_); Значение по умолчанию – Date() Индексированное – Да (Допускаются совпадения)
<u>Покупатель</u>		Числовой	Длинное целое	Обязательное поле; Индексированное – Да (Допускаются совпадения); Подстановка: значения из таблицы «Клиенты»
<u>Товар</u>		Числовой	Длинное целое	Обязательное поле; Индексированное – Да (Допускаются совпадения); Подстановка: значения из таблицы «Товары»
КоличествоТовара	Числовой	Целое	Обязательное поле;	
СуммаПредоплаты(руб)	Денежный	Денежный	Обязательное поле; Формат поля – С разделителями разрядов; Число десятичных знаков – 2; Значение по умолчанию – 0	
ПризнакОтгрузки	Логический		Обязательное поле; Формат – Да/Нет; Подстановка: Тип элемента управления – Флажок	
ДатаОплаты	Дата/время		Маска ввода – краткий формат даты (00.00.0000;0;_)	

ПРИЛОЖЕНИЕ 2. Описание свойства поля "Маска ввода"

Описание маски ввода может состоять из трех разделов, разделенных знаком точка с запятой, например, (999) 000-0000!; 0 ;" _".

Раздел	Значение
Первый	Сама маска ввода.
Второй	Указывает, следует ли сохранять текстовые символы: ✓ 0 – означает, что текстовые символы сохраняются вместе с введенными значениями; ✓ 1 или пустое значение – означает, что сохраняются только введенные символы.
Третий	Символ, выводящийся в маске ввода на месте пустых символов. Допускается использование любого символа; для отображения пробела, следует ввести " " (прямые кавычки, пробел, прямые кавычки). Если данный раздел описания оставить пустым, то для представления пустых символов используется символ подчеркивания (_).

В приведенной ниже таблице указано, как MS Access интерпретирует символы, содержащиеся в первой части описания свойства *Маска ввода* (InputMask).

Символ	Описание
0	Цифра (от 0 до 9, ввод обязателен).
9	Цифра или пробел (ввод не обязателен).
#	Цифра или пробел (ввод не обязателен; пустые символы преобразуются в пробелы, допускаются символы плюс и минус).
L	Буква (от A до Z или от A до Я, ввод обязателен).
?	Буква (от A до Z или от A до Я, ввод не обязателен).
A	Буква или цифра (ввод обязателен).
a	Буква или цифра (ввод необязателен).
&	Любой символ или пробел (ввод обязателен).
C	Любой символ или пробел (ввод необязателен).
<	Указывает перевод всех следующих символов на нижний регистр.
>	Указывает перевод всех следующих символов на верхний регистр.
!	Указывает заполнение маски ввода справа налево, а не слева направо. Заполнение маски символами всегда происходит слева направо. Восклицательный знак в маске ввода можно помещать в любую позицию.
\	Указывает ввод любого следующего символа в качестве текстовой константы. Используется для отображения всех перечисленных символов как текстовых констант (например, \A выводится как символ «А»).
Пароль	Значение Пароль, заданное для свойства Маска ввода, создает поле для ввода пароля. Любой символ, введенный в поле, сохраняется как символ, но отображается как звездочка (*).

ПРИЛОЖЕНИЕ 3. Функции для обработки полей с типом данных Дата/Время

В MS Access используются следующие встроенные функции для обработки данных с типом Дата/Время:

Date() – возвращает текущую системную дату;

Day(дата) – возвращает значение дня месяца от 1 до 31;

Month(дата) – возвращает значение месяца от 1 до 12;

MonthName(Month(дата)) – возвращает название месяца с Января по Декабрь;

Year(дата) – возвращает значение года от 100 до 9999;

Weekday(дата) – по умолчанию возвращает целое **число** от 1 (Воскресенье) до 7 (Суббота), соответствующее дню недели;

WeekdayName(Weekday(дата)) – возвращает **название** дня недели с Понедельника по Воскресенье;

Hour(дата) – возвращает целое число от 0 до 23, представляющее значение часа в дате;

DatePart("интервал"; дата) – возвращает число дней, недель, месяцев и т.д. в соответствии со значением аргумента **интервал** для указанной даты:

"q" – определение квартала (значение от 1 до 4);

"ww" – определение номера недели в году (значение от 1 до 53).

DateAdd("интервал"; число; дата) – возвращает новую дату, равную указанной дате, увеличенной на указанное **число** в соответствии с установленной маской, указанной в аргументе "интервал". Возвращает данные типа **Дата/время**.

Значения маски:

Значение	Описание	Значение	Описание
уууу	Год	w	День недели
q	Квартал	ww	Неделя
m	Месяц	h	Часы
y	Дней в году	n	Минуты
d	День	s	Секунды

DateDiff("интервал"; дата1; дата2) – определяет интервал между двумя датами в указанных единицах измерения (днях, месяцах, кварталах, годах). Здесь значение аргумента "интервал" имеет то же значение, что и в функции DateAdd.

Замечание: "дата1" – начальная дата, "дата2" – конечная дата, причем "дата1" должна быть ранее "дата2", то есть "дата1" ≤ "дата2".

ПРИЛОЖЕНИЕ 4. Функции `lif()` и `Format()`

Функция **If(условие; еслиИстина; еслиЛожь)** возвращает один из двух аргументов в зависимости от результата вычисления выражения. В **Мастере функций** находится в категории "Управление".

Аргумент	Назначение
условие	Выражение, значение которого нужно вычислить
еслиИстина	Значение или выражение, возвращаемые, если значением выражения является "Истина" (1)
еслиЛожь	Значение или выражение, возвращаемые, если значением выражения является "Ложь" (0)

Функция **Format("выражение"; инструкция форматирования)** – возвращает строку, содержащую выражение, отформатированное согласно инструкциям форматирования. В **Мастере функций** находится в категории "Текстовые".

Для выражений с типом Дата/Время можно применять следующие символы в инструкции форматирования:

Символ	Описание
c	Полный формат даты
dd	День месяца (от 1 до 30)
ddd	Первые две буквы названия дня недели (от Пн до Вс)
dddd	Полное название дня недели (от Понедельник до Воскресенье)
w	День недели (от 1 до 7)
ww	Неделя года (от 1 до 53)
mm	Месяц года (от 1 до 12)
mmm	Первые три буквы названия месяца (от Янв до Дек)
mmmm	Полное название месяца (от Январь до Декабрь)
q	Квартал года (от 1 до 4)
y	День года (от 1 до 366)
yy	Последние две цифры года (от 01 до 99)
yyyy	Возвращает 4-ёх-значное значение года (от 100 до 9999)

ПРИЛОЖЕНИЕ 5. Примеры условий отбора

Требуемый результат	Поле / Таблица	Выражение в Условии отбора
Продажи в Москву	Адрес Покупатели	"Москва*"
Продажи в Москву или Васюки	Адрес Покупатели	"Москва*" Or "Васюки*"
Продажи 2 февраля 2023 г.	ДатаПродажи УчетПродаж	=#2.02.23#
Продажи с 05.01.2023 и до 10.01.2023 включительно	ДатаПродажи УчетПродаж	Between #5-январь-23# And #10-январь-23#
Продажи за 2022 г.	ДатаПродажи	Year([ДатаПродажи])=2022
Продажи в четвертом квартале всех лет	ДатаПродажи УчетПродаж	DatePart("q";[ДатаПродажи])=4
Продажи за период, задаваемый с клавиатуры	ДатаПродажи УчетПродаж	>=[ДатаС] And <=[ДатаПо]
Товары в весовых единицах измерения	УчетнаяЕдиница Товар	In ("кг"; "г"; "мг")
Не жидкие товары, продаваемые на разлив	УчетнаяЕдиница Товар	Not "л"
Поставки с давностью более 30 дней	ДатаПоступления Товар	< Date() – 30
Поступления за текущий месяц текущего года	ДатаПоступления Товар	Year([ДатаПоступления]) = Year(Now()) And Month([ДатаПоступления]) = Month(Now())
Производители, названия которых начинаются на букву "К"	Производитель Товар	Like "К*"
Товары, в описании которых указано "для внешних работ"	ОписаниеТовара Товар	Like "**для внешних работ**"
Продажи покупателям, с названиями на буквы от А до Г	Покупатель Товар	Like "[А-Г]*"
Группы товаров с последними цифрами кода группы 99	ГруппаТовара Товар	Right([ГруппаТовара]; 2)="99"
Оплаченные, но не отгруженные покупки	ПризнакОтгрузки УчетПродаж	Нет
	ДатаОплаты УчетПродаж	IsNull([УчетПродаж]![ДатаОплаты])
Товары с ценами до 5 \$ по курсу, задаваемому с клавиатуры	Цена Товар	<5*[Курс\$]

ЛИТЕРАТУРА

1. Электронный учебно-методический комплекс по дисциплине «Базы данных» для специальностей: 1-28 01 02 «Электронный маркетинг», 1-28 01 01 «Экономика электронного бизнеса» [Электронный ресурс] / Брестский государственный технический университет, Кафедра ИиПМ; сост.: И. М. Гучко. – Брест : БрГТУ, 2022. Режим доступа: <https://rep.bstu.by/handle/data/32660>. – Дата доступа: 01.03.2023.
2. Бекаревич, Ю. Б. Самоучитель Access 2010 / Ю. Б. Бекаревич, Н. В. Пушкина. – СПб. : БХВ-Петербург, 2011. – 432 с.
3. Быков, В. Л. Система автоматизированного проектирования баз данных Microsoft Access 2010: учебно-методическое пособие / В. Л. Быков, И. М. Гучко, А. М. Кулешова. – Брест : БрГТУ, 2014. – 75 с.
4. Гучко, И. М. Создание баз данных в среде СУБД Microsoft Access: пособие для самостоятельной работы студентов экономических специальностей / И. М. Гучко, Е. Н. Рубанова, И. Н. Аверина. – Брест : Изд-во БрГТУ, 2018. – 72 с.
5. Кузин, А. В., Демин, В. М. Разработка баз данных в системе Microsoft Access: учебник. – М. : ИНФРА-М, 2014. – 223 с.
6. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг. – М. : Вильямс, 2017. – 1436 с.
7. Кириллов, В. В. Введение в реляционные базы данных / В. В. Кириллов, Г. Ю. Громов. – СПб : БХВ-Петербург, 2012. – 464 с.
8. Левчук, Е. А. Технологии организации, хранения и обработки данных / Е. А. Левчук. – 3-е изд. – Мн. : Выш. шк., 2007. – 239 с.
9. Осипов, Д.Л. Технологии проектирования баз данных. – М.: ДМКПресс, 2019. – 498 с.
10. Оскерко, В. С. Технологии баз данных: учеб. пособие/ В. С. Оскерко, З. В. Пунчик. Мн. : БГЭУ, 2015. – 215 с.
11. Разорёнова, Т. Р. Управление базами данных : учебно-методическое пособие для студентов спец. 1-96 01 01, 1-26 02 02, 1-25 01 08, 1-25 01 07, 1-25 01 07 31 / Т. Р. Разорёнова, О. В. Альшевская. – Мн. : БНТУ, 2010. – 99 с. – Рекомендовано УМО РБ. – ISBN 978-985-525-378-6.
12. Редько, В. Н. Базы данных и информационные системы / В. Н. Редько, И. А. Бассараб. – М. : Знание, 2011. – 602 с.
13. Рясова, С. Е. Компьютерные информационные технологии : учебно-методический комплекс для студентов экономических специальностей : в 3 частях [Электронный ресурс] / С. Е. Рясова ; Полоцкий государственный университет. – Новополоцк : ПГУ, 2017. – Ч. 3. Кн. 2 : Технологии баз данных и знаний. – 2017. – 141 с. Режим доступа: <https://elib.psu.by/handle/123456789/23787>. – Дата доступа: 01.03.2023.
14. Стружкин, Н. П. Базы данных: проектирование. Учебник для академического бакалавриата / Н. П. Стружкин, В. В. Годин. – М. : Издательство Юрайт, 2019. – 477 с.
15. Сурядный, А. С. Microsoft Access 2010. Лучший самоучитель. – М. : Астрель, ВКТ, 2012. – 488 с.

Учебное издание

Составители:

Аверина Ирина Николаевна

Гучко Ирина Михайловна

Технологии создания и ведения баз данных в Microsoft Access

Пособие

для студентов экономических специальностей

***Текст печатается в авторской редакции,
орфографии и пунктуации***

Ответственный за выпуск: Аверина И.Н.

Редактор: Митлошук М. А.

Компьютерная верстка: Вашкевич Ю. А.

Подписано в печать 22.03.2023 г. Формат 60x84 1/16. Бумага «Performer».
Гарнитура « Arial Narrow». Усл. печ. л. 4,19. Уч. изд. л. 4,5. Заказ № 188. Тираж 19 экз.
Отпечатано на ризографе учреждения образования «Брестский государственный
технический университет». 224017, г. Брест, ул. Московская, 267.
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/235 от 24.03.2014 г.