

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

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

**КАФЕДРА ИНФОРМАТИКИ И ПРИКЛАДНОЙ МАТЕМАТИКИ**

## **Методические указания**

**по курсу «Компьютерные информационные технологии»  
2-й раздел: «Технологии баз данных и знаний»  
для студентов экономических специальностей**

Методические указания представляют собой руководство по выполнению лабораторных работ по курсу «Компьютерные информационные технологии» (2-й раздел – «Технологии баз данных и знаний») для студентов первого курса экономических специальностей дневной и заочной форм обучения.

В указаниях приводится теоретический материал, рекомендации и пояснения для выполнения лабораторных работ в среде СУБД Microsoft Access, список литературы для самостоятельного изучения и вопросы к экзамену.

Указания составлены в помощь студентам при выполнении лабораторных работ и подготовке к экзамену по названной дисциплине.

Составители: Гучко И.М., ст. преподаватель;  
Рубанова Е.Н., ст. преподаватель

# СОДЕРЖАНИЕ

<b>1. ВВЕДЕНИЕ</b>	<b>5</b>
<b>2. ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ</b>	<b>5</b>
2.1. Определение базы данных (БД)	5
2.2. Модели данных	5
2.3. Реляционная модель данных	6
2.3.1. Базовые понятия реляционных баз данных (РБД)	7
2.3.2. Правила нормализации реляционной БД	7
2.4. Системы управления базами данных (СУБД)	7
2.4.1. Реляционная система управления базами данных (РСУБД)	8
2.5. Проектирование базы данных	8
2.6. Общая характеристика СУБД ACCESS	9
2.6.1. Основные объекты СУБД Access	9
2.7. Таблица как объект БД	9
2.8. Определение ключевого поля таблицы	9
2.9. Межтабличные связи в РБД	10
2.10. Целостность данных	11
2.11. Каскадные операции	11
2.12. Типы отношений между таблицами	12
2.13. Понятие запроса к базе данных	12
2.14. Формы и отчеты как объекты БД	13
<b>3. ПРАКТИЧЕСКАЯ ЧАСТЬ</b>	<b>13</b>
3.1. Порядок создания базы данных в MS Access	13
3.2. Режимы работы с объектами базы данных	15
3.3. Способы создания таблиц в MS Access	16
3.4. Создание таблицы с помощью Конструктора	16
3.5. Типы данных в MS Access	17
3.6. Описание свойств полей	18
3.6.1. Формирование маски ввода	20
3.6.2. Свойство «Значение по умолчанию»	21
3.7. Использование Мастера подстановки для создания фиксированного списка значений	21
3.8. Задание первичного ключа	24
3.9. Сохранение таблицы	24
3.10. Использование Мастера подстановки для создания связей между таблицами	25
3.11. Установка связей между таблицами	28
3.12. Удаление (изменение) связей	29
3.13. Ввод данных в таблицу	30
3.14. Сортировка данных в таблице	30

3.15. Создание запросов выбора с помощью Конструктора	30
3.15.1. Добавление (удаление) полей в Бланке запроса	32
3.15.2. Сортировка результатов выборки	32
3.15.3. Запуск запроса	32
3.16. Построитель выражений	33
3.17. Формирование условий отбора	35
3.18. Формирование условий отбора для полей с типом данных Дата/Время	36
3.19. Сложные критерии выборки	36
3.20. Проектирование параметрических запросов	37
3.21. Вычисляемые поля в запросах	38
3.22. Проектирование итоговых запросов	43
3.23. Проектирование перекрестных запросов	48
3.24. Способы создания форм	53
3.25. Создание формы с помощью инструмента Автоформа	53
3.26. Создание формы с помощью Мастера форм	54
3.27. Создание формы в режиме Конструктора	55
3.28. Структура формы	55
3.29. Свойства и события объектов формы	57
3.30. Размещение элемента управления "Поле" в форме	57
3.31. Разработка многотабличных (составных) форм	60
3.32. Проектирование отчетов	62
3.33. Создание управляющей (кнопочной) формы	63
3.34. Использование макросов	64
<b>ПРИЛОЖЕНИЕ 1. Описание структур таблиц БД Магазин "Строитель"</b>	<b>65</b>
<b>ПРИЛОЖЕНИЕ 2. Описание свойства поля "Маска ввода"</b>	<b>67</b>
<b>ПРИЛОЖЕНИЕ 3. Функции для обработки полей с типом данных Дата/Время</b>	<b>68</b>
<b>ПРИЛОЖЕНИЕ 4. Функции If( ) и Format( )</b>	<b>69</b>
<b>ПРИЛОЖЕНИЕ 5. Примеры условий отбора</b>	<b>70</b>
<b>СПИСОК ВОПРОСОВ К ЭКЗАМЕНУ</b>	<b>71</b>
<b>ЛИТЕРАТУРА</b>	<b>73</b>

## 1. ВВЕДЕНИЕ

Курс «Технологии баз данных и знаний» является вторым разделом дисциплины «Компьютерные информационные технологии».

Целью изучения данного раздела является получение знаний о современных технологиях организации, хранения и обработки данных, приобретение навыков по созданию баз данных и работе с ними при решении практических задач в своей предметной области, а также ознакомление с достижениями в области технологий баз данных.

Предмет изучения – технология проектирования реляционных баз данных и ее реализация на основе базовых возможностей СУБД Microsoft Access.

Освоение курса «Технологии баз данных и знаний» предполагает получение знаний:

- ✓ о представлениях экономической информации в автоматизированных информационных системах;
- ✓ концепциях моделирования данных, принципах организации баз данных и их проектирования;
- ✓ назначении, архитектуре, функциональных возможностях современных систем управления базами данных (СУБД).

Кроме того, в данном курсе приобретаются практические навыки в среде СУБД: проектирования и создания баз данных; формирования запросов к базе данных; проектирования приложений базы данных – форм и отчетов; управления базой данных.

Изучение материала этого раздела основано на использовании знаний, умений и навыков, полученных студентами после усвоения материала первого раздела «Техническое и программное обеспечение информационных технологий». Знания, умения и навыки, приобретенные в ходе изучения материала указанного раздела, позволят студентам эффективно использовать современное прикладное программное обеспечение в процессе решения учебных, научных и практических задач, станут основой для усвоения материала третьего раздела учебной программы «Корпоративные информационные системы».

## 2. ТЕОРЕТИЧЕСКИЙ МАТЕРИАЛ

### 2.1. Определение базы данных (БД)

База данных (database) – это поименованная совокупность взаимосвязанных данных, организованных на машинном носителе средствами СУБД, отображающая отношения и свойства объектов в некоторой предметной области.

Объектом может быть предмет, вещество, событие, лицо, явление, абстрактное понятие, т. е. все то, что характеризуется набором значений некоторой совокупности атрибутов – информационного отображения свойств объекта. Например, объект «книга» характеризуется атрибутами: наименование, авторы, количество страниц, тираж, цена и т. п.

### 2.2. Модели данных

Ядром любой БД является модель данных, т. е. совокупность структур данных и операций их обработки. По способу установления связей между данными различают следующие виды моделей:

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

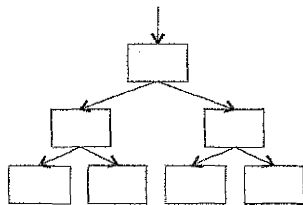


Рисунок 1 – Иерархическая древовидная структура модели БД

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

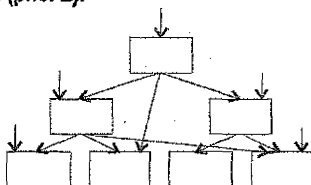


Рисунок 2 – Сетевая структура модели БД

**III. Реляционная модель.** В реляционной модели информация представляется в виде прямоугольных таблиц. Каждая таблица состоит из строк и столбцов и 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.

Реляционные базы данных (РБД), в отличие от иерархических и сетевых моделей, позволяют организовывать связи между таблицами в любой момент. Для этого в РБД реализован механизм внешних ключей. В каждой таблице БД имеется хотя бы одно поле, служащее ссылкой для другой таблицы. В терминологии РБД такие поля называются полями внешних ключей. С помощью внешних ключей можно связывать любые таблицы БД на любом этапе работы с БД.

### 2.3. Реляционная модель данных

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

Реляционная модель данных была предложена Е. Коддом, известным американским специалистом в области баз данных. Основные концепции этой модели были впервые опубликованы в 1970 г. Будучи математиком по образованию, Кодд предложил использовать для обработки данных аппарат теории множеств (объединение, пересечение, разность, декартово произведение). Он показал, что любое представление данных сводится к совокупности двумерных таблиц особого вида, известного в математике как отношение (по-английски – relation, отсюда и название – реляционные базы данных).

Одна из главных идей Кодда заключалась в том, что связь между данными должна устанавливаться в соответствии с их внутренними логическими взаимоотношениями. Второй важный принцип, предложенный Коддом, заключается в том, что в реляционных системах одной командой могут обрабатываться целые файлы данных, в то время как ранее одной командой обрабатывалась только одна запись.

### **2.3.1. Базовые понятия реляционных баз данных (РБД)**

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

**Атрибут** – определенная часть информации о некотором объекте – например, адрес клиента или зарплата сотрудника. Атрибут обычно хранится в виде столбца (поля) таблицы.

**Тип данных** – понятие, которое в реляционной модели полностью эквивалентно соответствующему понятию в алгоритмических языках. Набор поддерживаемых типов данных определяется СУБД и может сильно различаться в разных системах.

**Связь** – способ, которым связана информация в одной таблице с информацией в другой таблице. Связи осуществляются с помощью совпадающих полей, которые называются ключом.

**Объединение** – процесс объединения таблиц или запросов на основе совпадающих значений определенных атрибутов.

### **2.3.2. Правила нормализации реляционной БД**

**Нормализация** представляет собой процесс реорганизации данных путем ликвидации повторяющихся групп и иных противоречий с целью приведения таблиц к виду, позволяющему осуществлять непротиворечивое и корректное редактирование данных. Окончательная цель нормализации сводится к получению такого проекта БД, в котором каждый факт появляется только в одном месте, т. е. исключена избыточность информации.

1. Каждое поле любой таблицы должно быть уникальным.

2. Каждая таблица должна иметь уникальный идентификатор (первичный ключ), который может состоять из одного или нескольких полей таблицы.

3. Для каждого значения первичного ключа должно быть одно и только одно значение любого из столбцов данных, и это значение должно относиться к объекту таблицы (т. е. в таблице не должно быть данных, которые не относятся к объекту, определяемому первичным ключом, а также информация в таблице должна полностью описывать объект).

4. Должна иметься возможность изменять значения любого поля (не входящего в первичный ключ), и это не должно повлечь за собой изменения другого поля (т. е. не должно быть вычисляемых полей).

### **2.4. Системы управления базами данных (СУБД)**

Поддержание баз данных в компьютерной среде осуществляют программные средства – **системы управления базами данных** (database management system), которые представляют собой совокупность программных и языковых средств общего или специализированного назначения, необходимых для создания баз данных на машинных носи-

телях, поддержания их в актуальном состоянии и организации доступа к ним различных пользователей в условиях принятой технологии обработки данных.

СУБД – это управляющие программы, которые обеспечивают все манипуляции с базами данных: создание базы, ее ведение, ее использование многими пользователями и др., т. е. реализуют сложный комплекс функций по централизованному управлению базой данных и обслуживают интересы пользователей.

СУБД можно рассматривать как программную оболочку, которая находится между базой данных и пользователем. Она обеспечивает централизованный контроль защиты и целостности данных, доступ к данным, их обработку, формирование отчетов на основе базы данных и другие операции и процедуры.

#### **2.4.1. Реляционная система управления базами данных (РСУБД)**

Набор средств для управления РБД называется реляционной системой управления базами данных, которая может содержать утилиты, приложения, службы, библиотеки, средства создания приложений и другие компоненты. Будучи связанной посредством общих ключевых полей, информация в РБД может объединяться из множества таблиц в единый результирующий набор.

#### **2.5. Проектирование базы данных**

Проектирование базы данных (БД) состоит из двух основных частей: логического и физического моделирования. Во время логического моделирования разрабатывается модель базы данных, не зависящая от конкретной СУБД. Во время физического моделирования создается модель под конкретную систему управления базами данных.

Процесс проектирования базы данных состоит из следующих этапов:

1. Сбор информации;
2. Идентификация объектов;
3. Моделирование объектов;
4. Идентификация типов информации для каждого объекта;
5. Идентификация отношений;
6. Нормализация;
7. Преобразование к физической модели;
8. Создание базы данных.

Пункты 1-6 представляют собой логическое моделирование, а 6-7 – физическое моделирование базы данных.

На первом этапе проектирования базы необходимо определить назначение базы данных, режимы ее использования и основные алгоритмы, реализующие реальные бизнес-процессы и изучить предметную область ее использования с целью создания модели.

В процессе идентификации базы определяется, какие данные будут обрабатываться, какие объекты будут при этом создаваться, а также какие связи будут реализованы. Далее необходимо определить все свойства (атрибуты) объектов и данных, т. е. принять решение по следующим вопросам:

- какие значения будут храниться в полях;
- сколько места необходимо для хранения данных в поле;
- формат представления данных;



- какие операции возможно будут производиться со значениями в полях ;
- сортировка, группировка и т. д.

## 2.6. Общая характеристика СУБД ACCESS

Microsoft Access – это функционально полная реляционная СУБД, в которой предусмотрены все необходимые средства для определения и обработки данных, а также для управления ими при работе с большими объемами информации. Различные ее версии входят в состав программного пакета MS Office и работают в среде Windows.

### 2.6.1. Основные объекты СУБД Access

База данных в Access может содержать различные элементы, называемые объектами. Объектами в Access являются:

**Таблица** – объект, используемый для хранения пользовательских данных.

**Запрос** – объект, который позволяет пользователю получить нужные данные из одной или нескольких таблиц.

**Форма** – объект, предназначенный для ввода данных, отображения их на экране или управления работой приложения.

**Отчет** – объект, предназначенный для формирования выходного документа, который может быть распечатан.

**Страницы доступа к данным** позволяют редактировать, просматривать и обрабатывать данные, используя интернет-браузер.

**Макрос** – набор команд, который позволяет автоматизировать часто выполняемые операции.

**Модуль** – программы, написанные на языке программирования Visual Basic, которые могут разрабатываться пользователем для реализации нестандартных процедур при создании приложения.

### 2.7. Таблица как объект БД

Таблица – фундаментальная структура СУБД. В MS Access – это объект, предназначенный для хранения данных в виде *записей* (строк) и *полей* (столбцов). В записи собраны сведения об одном экземпляре объекта предметной области, а каждое поле содержит одну характеристику (один атрибут) этого объекта.

Таблицу в Access можно создать несколькими способами:

- построить новую таблицу «с нуля» непосредственно в **режиме таблицы**;
- спроектировать таблицу, воспользовавшись **Конструктором**;
- запустить **Мастер таблиц** – специальную программу, предлагающую 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 Excel.

### 2.8. Определение ключевого поля таблицы

В теории реляционных баз данных таблица представляет собой изначально неупорядоченный набор записей. Единственным способом идентифицировать определенную запись в этой таблице – это указать набор атрибутов, который был бы уникальным для этой записи.

**Ключом** называется набор атрибутов, однозначно определяющий запись.

Существуют следующие виды ключей:

- **Первичный ключ** – это одно или несколько полей (столбцов), значения которых однозначно определяют каждую запись в таблице. Первичный ключ не допускает значение Null и всегда должен иметь уникальный индекс. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах. Первичный ключ может быть естественным или искусственным. Ключ, состоящий из информационных полей таблицы (полей, содержащих полезную информацию об описываемых объектах), называется естественным ключом. Теоретически, естественный ключ всегда можно сформировать, в этом случае он носит название «интеллектуальный ключ». Искусственный ключ – это дополнительное служебное поле, назначение которого – служить первичным ключом. Значения этого поля генерируются искусственно в виде числовой последовательности. Как правило, в качестве ключевых полей указываются поля, которые имеют неповторяющиеся данные или создаются поля с типом данных *Счетчик*. В любом случае, поле ключа не должно содержать повторяющихся данных.

- **Внешний ключ** – представляет собой одно или несколько полей (столбцов), содержащих ссылку на поле или поля первичного ключа в другой таблице. Внешний ключ определяет способ связи таблиц.

Ключи также делятся на два класса: простые и составные.

**Простой ключ** состоит из одного атрибута, **составной ключ** состоит из нескольких атрибутов. Применение составных ключей усложняет объединение таблиц.

**Индекс** – средство, ускоряющее поиск и сортировку данных в таблице. Существенное повышение скорости выполнения запросов приносит индексирование полей, расположенных по обе стороны отношения, или создание связи между этими полями, а также индексирование всех полей, используемых для задания условий отбора в запросе. Индекс может быть простым (состоять из одного атрибута) или составным (состоять из нескольких атрибутов).

## 2.9. Межтабличные связи в РБД

Связи между таблицами устанавливаются по равенству значений первичного и внешнего ключей. Таблица, в которой используется для связи первичный ключ, называется **главной**, а таблица с внешним ключом – **подчиненной**.

В приложении Access межтабличные связи используются для того, чтобы продемонстрировать как связать таблицы для использования их в объекте базы данных. Существует несколько причин для создания межтаблических связей перед созданием других объектов базы данных (форм, запросов, отчетов):

- межтабличные связи представляют сведения для структуры запросов;
- межтабличные связи представляют сведения для структуры форм и отчетов;
- межтабличные связи являются основой, с помощью которой можно обеспечить целостность данных.

После создания таблиц, полей и связей необходимо еще раз проверить структуру созданной базы данных и выявить все неточности и ошибки. Это нужно делать до заполнения таблиц данными!

Связь между таблицами устанавливается путем определения в одной таблице (подчиненной) поля, соответствующего ключу другой таблицы (главной). Установленная связь свяжет записи, содержащиеся в заданном поле одинаковые значения.

## 2.10. 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.

Установить целостность данных можно, если выполнены 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;
- ✓ обе таблицы принадлежат одной базе данных MS Access. Для установки целостности данных база данных, в которой находятся таблицы, должна быть открыта. При установке целостности данных необходимо помнить следующие правила:
  - в подчиненной таблице нельзя вводить записи, которые не связаны 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.

Устанавливается опция обеспечения целостности данных при создании конкретной связи. Если опция установлена, то любая попытка выполнить одно из перечисленных выше правил приведет к выводу на экран предупреждения, а само действие будет отменено.

## 2.11. Каскадные операции

Целостность данных в связанных таблицах обеспечивают каскадные операции двух видов:

- операции каскадного обновления;
- операции каскадного удаления.

Эти операции можно включать и выключать путем установки соответствующих флажков: «Каскадное обновление связанных полей» и «Каскадное удаление связанных полей».

Если установлен флажок «Каскадное обновление связанных полей», то любые изменения в значении ключевого поля в главной таблице, которая стоит на стороне «один» в отношениях 1:М, ведут к автоматическому обновлению соответствующих значений во всех связанных записях.

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

## 2.12. Типы отношений между таблицами

Существует три типа отношений между таблицами:

**Один-к-одному (1:1).** Значению ключа в каждой записи в главной таблице могут соответствовать значения в связанном поле только в одной записи подчиненной таблицы. В этом случае связь между таблицами может быть установлена только через ключевые поля обеих таблиц.

**Один-ко-многим (1:M).** Значению ключа в каждой записи в главной таблице могут соответствовать значения в связанном поле (полях) в нескольких записях подчиненной таблицы. Этот тип отношения довольно часто используется в реляционных БД.

**Много-ко-многим (M:M).** Возникает между двумя таблицами, когда одна запись из первой таблицы А (выходная связь) может быть связана больше чем с одной записью другой таблицы В (принимающая), в свою очередь, одна запись из другой таблицы может быть связана больше чем с одной записью первой таблицы. Эта схема реализуется только при помощи третьей соединительной таблицы, ключ связи которой состоит, как минимум, из двух полей. Эти поля являются полями внешнего ключа в таблицах А и В. Первичный ключ для соединительной таблицы – это обычно комбинация из внешних ключей.

Если между таблицами имеются связи типа M:M, создается дополнительная таблица пересечений, с помощью которой связь M:M будет сведена к двум связям типа 1:M. Access не позволяет определить прямую связь M:M между двумя таблицами.

## 2.13. Понятие запроса к базе данных

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

Access позволяет создавать запросы на выборку и активные запросы. Они могут быть однотоабличными (в качестве исходной информации используются данные одной таблицы) и многотоабличными (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:

- *простой запрос на выборку* – выбирает данные из взаимосвязанных таблиц или других запросов с учетом условий пользователя. Результатом его является таблица, которая существует до закрытия запроса;
- *запросы с групповыми операциями (итоговый запрос)* – выполняют вычисления с использованием данных из групп записей;
- *запросы с параметром (параметрический запрос)* – это запросы, в которых критерий отбора может задать сам пользователь, введя нужный параметр при вызове запроса;
- *запросы с вычислениями* – позволяют осуществлять вычисления над полями БД;
- *перекрестные запросы* – обеспечивают создание результирующей таблицы на основе расчетов, полученных при анализе группы таблиц. С помощью данного запроса можно сгруппировать большой объем информации и представить его в удобном для восприятия виде.

**Активные запросы**, или 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 агрегатных функций.

## 2.14. Формы и отчеты как объекты БД

Понятие «**Форма**» в БД означает структурированное окно, экранное поле 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. Форма дает возможность вводить и просматривать информацию в БД.

Как объект базы данных 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.

Как правило, **отчеты** 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.

## 3. ПРАКТИЧЕСКАЯ ЧАСТЬ

### Практическое задание 1. Создание файла базы данных.

**Условие:** создать файл базы данных **Магазин «Строитель»**.

#### Порядок выполнения:

#### 3.1. Порядок создания базы данных в MS Access

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** (рис. 3). В свою очередь, рабочее поле окна БД предназначено для отображения списка объектов выбранного типа.

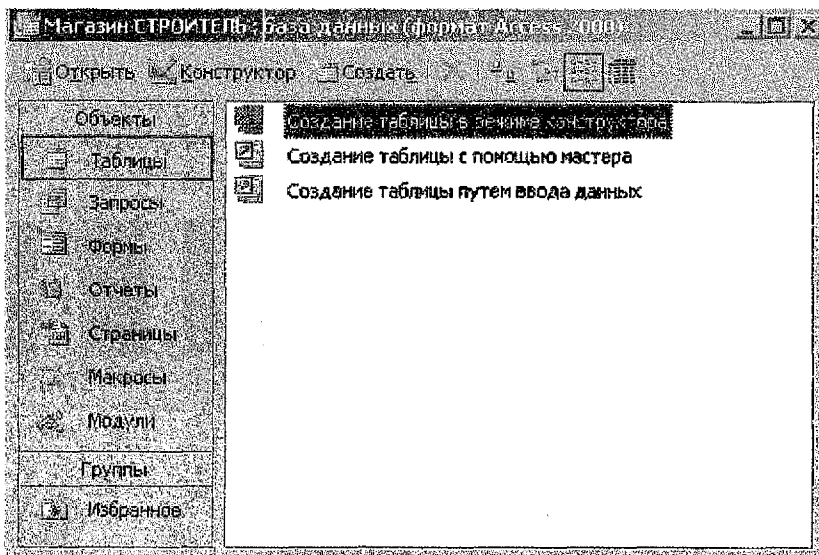


Рисунок 3 – Окно БД в MS ACCESS 2003

### Для версий MS ACCESS 2007/2010:

- ✓ Запустить СУБД Microsoft Access из меню **Пуск** или с помощью ярлыка. Отобразится меню создания базы данных 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, в которую он должен быть сохранен.
- ✓ Нажать кнопку **Создать**.

При создании новой БД автоматически создается новая таблица в режиме таблицы (рис. 4).

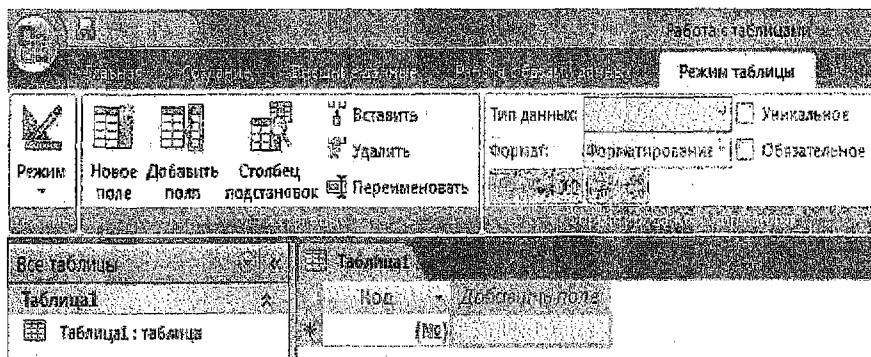


Рисунок 4 – Новая таблица в MS ACCESS 2007

### 3.2. Режимы работы с объектами базы данных

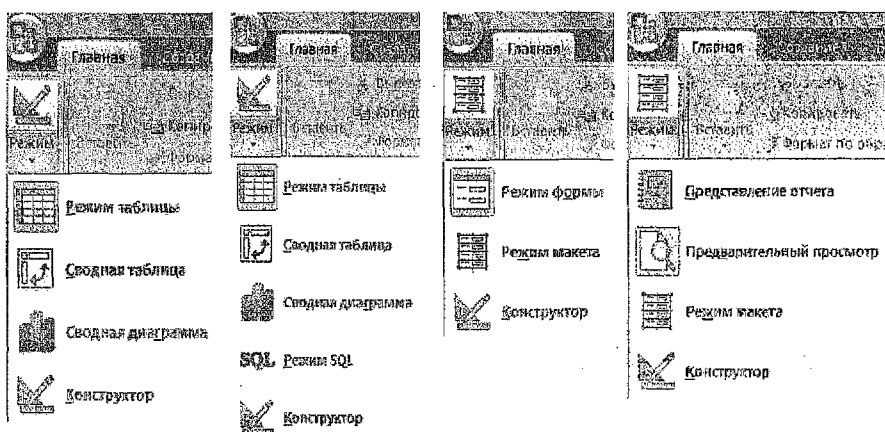
Основные режимы работы с уже созданными объектами БД:

**Открыть** – позволяет перейти в режим редактирования таблицы, выполнения запроса, загрузки формы, построения отчета, запуска макроса.

**Конструктор** – обеспечивает переход к режиму настройки выбранного объекта.

Для ACCESS 2003 данные команды представлены соответствующими кнопками на панели инструментов окна БД (Изменить, Конструктор).

Для версий ACCESS 2007/2010 эти и другие команды представлены в виде пиктограмм на вкладке Главная в группе Режимы (рис. 5).



а) для таблиц

б) для запросов

в) для форм

г) для отчетов

Рисунок 5 – Режимы работы с объектами в ACCESS 2007

## Практическое задание 2. Создание таблиц.

### 3.3. Способы создания таблиц в MS Access

Чтобы создать таблицу в ACCESS 2003, нужно перейти к списку таблиц в окне БД и нажать на панели инструментов кнопку . Появится диалоговое окно *Новая таблица*, в котором необходимо выбрать нужный способ создания таблицы (рис. 6).

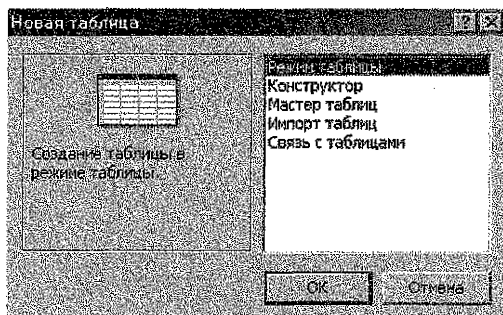


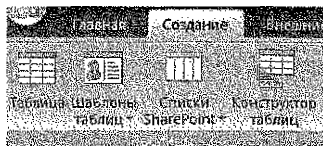
Рисунок 6 – Способы создания таблицы в MS ACCESS 2003

Создание новой таблицы в версиях ACCESS 2007/2010 осуществляется автоматически с созданием файла БД. Изменить способ её проектирования или выбрать другой можно с помощью следующих инструментов (рис. 7):

- ✓ на вкладке **Главная** в группе **Режимы**;
- ✓ на вкладке **Создание** в группе **Таблицы**;
- ✓ на вкладке **Внешние данные** в группе **Импорт**.



а)



б)



в)

Рисунок 7 – Способы создания таблицы в MS ACCESS 2007-2010

### 3.4. Создание таблицы с помощью Конструктора

Если для создания таблицы выбран режим **Конструктора**, то появляется окно с названием *Таблица1*, в котором определяется структура таблицы БД (рис. 8).



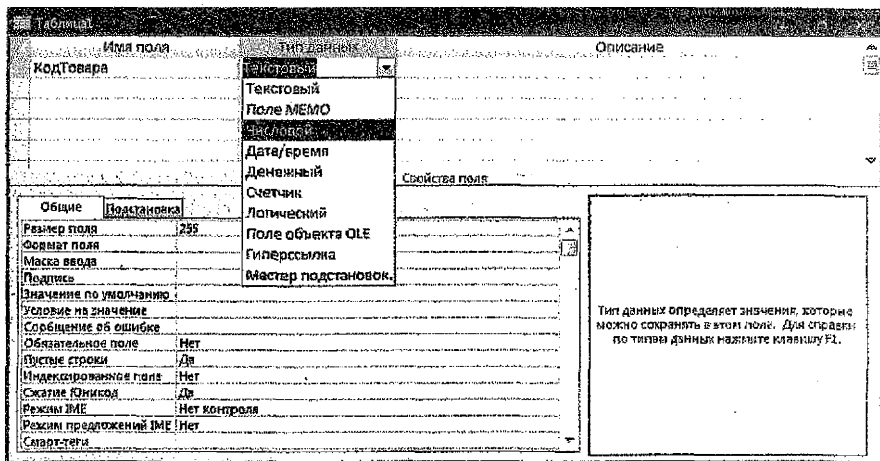


Рисунок 8 – Окно таблицы в режиме Конструктора

Условно это окно можно поделить на две области:

- **Область полей таблицы.** В этой части окна в столбце *Имя поля* вводится **название** полей (имя может содержать не более 64 знаков, при этом допустимы любые символы, кроме точки, восклицательного знака и угловых скобок, при этом повторение имен полей не допускается), определяется **тип данных** каждого поля в соответствующей графе, а также делаются краткие **комментарии** по поводу того, какие данные находятся в этом поле (графа *Описание*).

- **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.

### 3.5. Типы данных в MS Access

Для каждого поля таблицы необходимо указать тип данных, содержащихся в нем. Тип данных выбирается из списка, который можно вызвать щелчком мыши в одноименном столбце в режиме Конструктора (рис. 8).

Access оперирует следующими типами данных:

Тип данных	Содержимое
Текстовый	Используется для буквенно-цифровых символов, включая текст, а также текст и числа, не применяемые в вычислениях. Может содержать до 255 символов.
Числовой	Применяется для хранения числовых значений (целых или дробных), которые используются в вычислениях (за исключением денежных сумм).

<b>Тип данных</b>	<b>Содержимое</b>
Денежный	Данные о денежных суммах, хранящихся с 4 знаками после запятой и 15 в целой части. Применяется для предотвращения округления во время вычислений.
Счетчик	Используется для формирования уникального первичного ключа таблицы. При добавлении в таблицу новой записи значение этого поля увеличивается на 1 (единицу). Значения в таких полях не обновляются.
Логический	Используется для логических значений: Да/Нет, Истина/Ложь, Вкл/Выкл, т. е. одно из двух возможных значений.
Дата / время	Используется для хранения значений дат и времени.
Поле Мемо	Абзацы, длинные тексты и т. д., размер которых превышает 255 символов, или для текста, в котором используется RTF – форматирование. Свойство поля MEMO указывает, поддерживает ли поле форматированный текст.
Поле объекта OLE	Объекты, созданные в других программах, поддерживающих протокол OLE. Картинки, диаграммы и т. д.
Гиперссылка	Текст, комбинация текста и цифр, являющихся адресом гиперссылки (в том числе ссылки на объекты приложения Access, которые хранятся в базе данных).
Вложение (для Access 2010)	Используется для хранения двоичных файлов (файлов, которые нельзя прочесть с помощью текстового редактора), таких как цифровые изображения (фотографии и др. изображения) или файлов, созданных с помощью других приложений Microsoft Office. Используя поле с типом данных «Вложение», можно вложить в одну запись более одного файла.
Мастер подстановок	Используется для запуска мастера подстановок, позволяющего создавать поле, в котором в виде раскрывающегося списка отображаются значения из другой таблицы, запроса или фиксированного списка значений (фактически мастер подстановок не является типом данных).

### 3.6. 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:

<b>Свойство поля</b>	<b>Применение свойства</b>
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.

<b>Свойство поля</b>	<b>Применение свойства</b>
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, автоматически назначаемое полю 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.
Пустые строки	Разрешает (установкой значения Да) ввод строки нулевой длины ("" ) в текстовом поле и поле MEMO.
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 восточно-азиатских версиях Windows.
Режим предложений IME	Управление преобразованием предложений в восточно-азиатских версиях Windows.
Смарт-теги	Позволяет выбрать смарт-теги, применяемые к полю.
Только добавление	Если задано значение Да, ведется журнал значения поля.
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> .
Выравнивание текста	Выравнивание текста по умолчанию в элементе управления.
Отображать элемент выбора даты	Указывает, должен ли в приложении Access отображаться элемент выбора даты (элемент управления "Календарь"), когда пользователь изменяет значение поля.


**Условие:** в режиме Конструктора создать таблицы «Группы товаров», «Товары», «Покупатели» и «Учет продаж» (описание структур таблиц см. в Приложении 1).

### Порядок выполнения:

1. Для версии ACCESS 2003: в окне БД выбрать объект Таблицы, далее пункт меню Создать, далее в окне Новая таблица выбрать режим Конструктор (рис. 6);
  2. В графу Имя поля ввести имя формируемого поля. Нельзя использовать в имени точку, восклицательный знак и квадратные скобки. Нельзя начинать имя поля с пробела.
  3. В графе Тип данных выбрать из списка требуемый тип данных.
  4. В графу Описание ввести текст, поясняющий назначение и использование данного поля таблицы (Заполнять не обязательно).
  5. В нижней части окна Свойства поля задать требуемые свойства каждого поля.
- Замечание:** для создания каждой следующей таблицы в версии ACCESS 2007/2010 необходимо активизировать на ленте вкладку Создание и в группе инструментов Таблицы выбрать пиктограмму Конструктор таблиц (рис. 7-б).

#### 3.6.1. 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. 9);

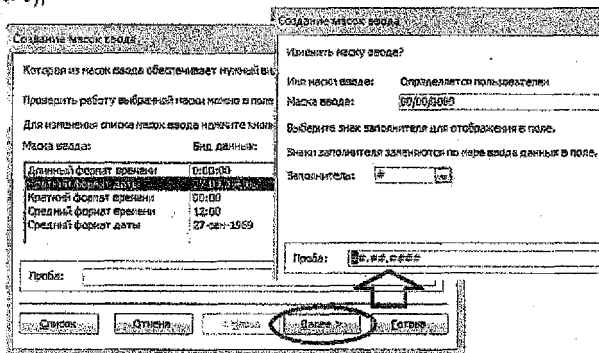


Рисунок 9 – Окна работы с Маской ввода для поля с типом данных Дата/Время

- ✓ если необходимой маски в списке нет, нажать кнопку **Список**. Откроется окно **Настройка масок ввода** (рис. 10);

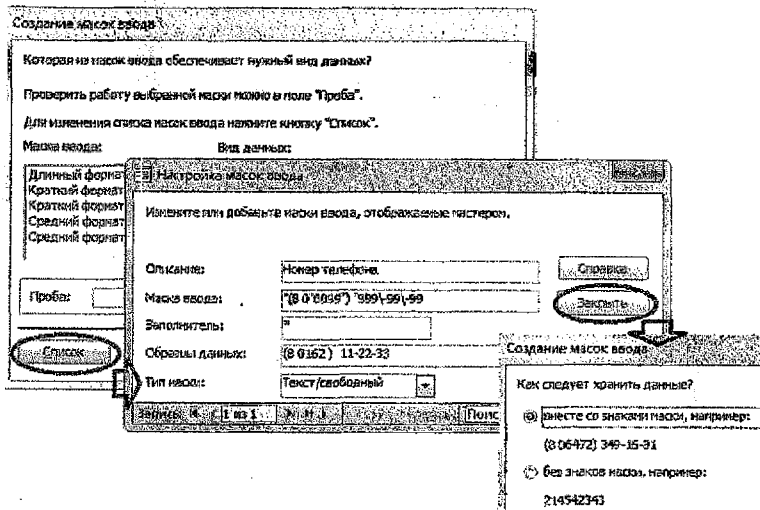


Рисунок 10 – Окна работы с Маской ввода для поля с тестовым типом данных

- ✓ 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 **Закрыть**;
- ✓ 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**).

### 3.6.2. Свойство «Значение по умолчанию»

Например, в таблице **Учёт продаж** для автоматического заполнения поля **ДатаПродажи** значением текущей даты необходимо для свойства **Значение по умолчанию** использовать встроенную функцию **Date()**. Можно воспользоваться **Построителем выражений** (см. раздел 3.16. практической части).

### 3.7. 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 **Группы товаров** следующий:

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

### 3.8. Задание первичного ключа

Понятие ключевого поля дано в разделе 2.8. теоретической части. Для задания ключевого поля 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 2007/2010**: выбрать пиктограмму **Ключевое поле** в группе **Сервис** на вкладке ленты **Конструктор**.

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

### 3.9. 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 автоматический 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 2007/2010 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.

**Условие:** разработать схему данных для созданных таблиц путём установления связей между ними (рис. 16).

#### Порядок выполнения:

Следует обратить внимание на то, при вводе данных в поля *Покупатель* и *Товар* таблицы *Учет продаж* значения берутся из соответствующих полей *КодПокупателя* таблицы *Покупатели* и *КодТовара* таблицы *Товары*. Аналогично, при заполнении поля *Группа товара* таблицы *Товары* используются данные поля *Код группы товара* из

таблицы **Группы товаров**. Для того, чтобы заполнение указанных полей было удобным, необходимо воспользоваться инструментом **Мастер подстановок**.

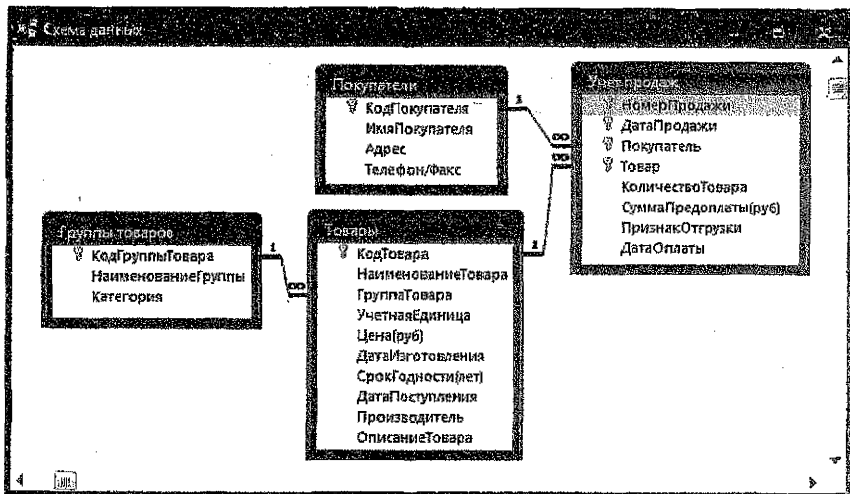


Рисунок 16 – Схема данных в БД Магазин "Строитель"

### 3.10. Использование Мастера подстановки для создания связей между таблицами

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

Последовательно следовать 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. 17). Нажать кнопку **Далее**.

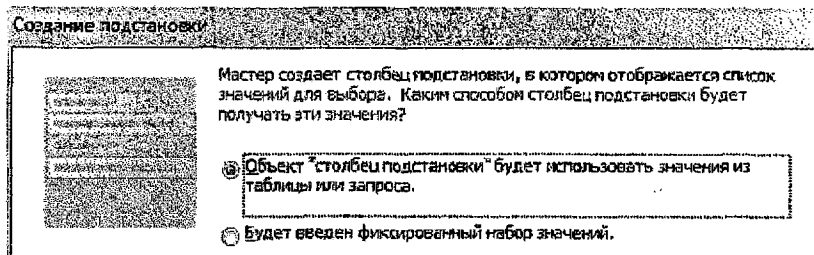


Рисунок 17 – Первый шаг Мастера подстановок

2. Во втором диалоговом окне **Создание подстановки** следует выбрать конкретную таблицу **Покупатели** (рис. 18). Нажать кнопку **Далее**.

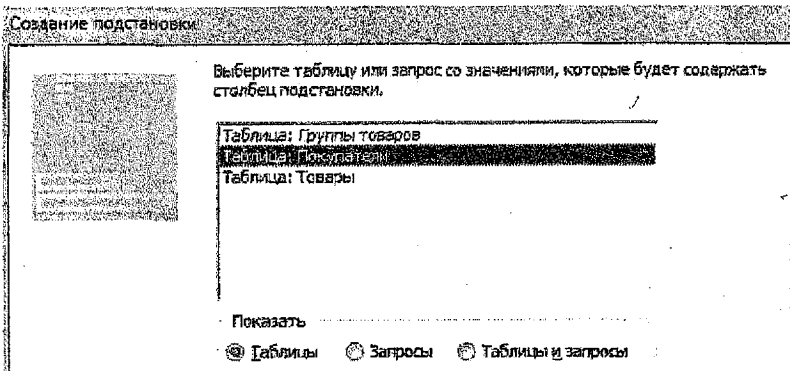


Рисунок 18 – Второй шаг Мастера подстановок

3. В третьем диалоговом окне необходимо отобразить отдельные поля таблицы с помощью кнопки  (либо все поля таблицы с помощью кнопки ). В данном случае выбираются поля **КодПокупателя**, **ИмяПокупателя** и **Адрес** (рис. 19). Нажать кнопку **Далее**.

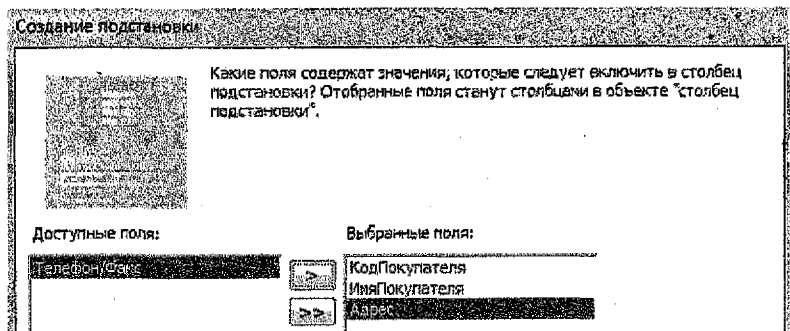


Рисунок 19 – Третий шаг Мастера подстановок

4. В следующем диалоговом окне можно указать способ сортировки элементов списка выбираемых значений (рис. 20). Нажать кнопку **Далее**.

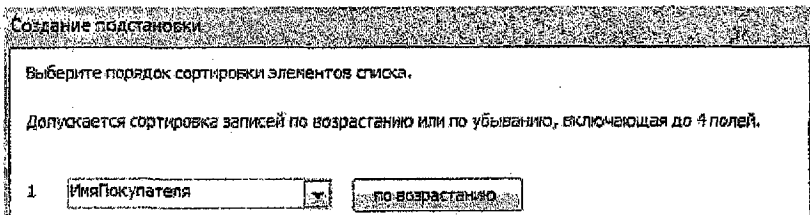


Рисунок 20 – Четвертый шаг Мастера подстановок

5. Затем можно подобрать ширину столбцов подстановки для того, чтобы данные поля просматривались при выборе из списка полностью (рис. 21). Нажать кнопку **Далее**.



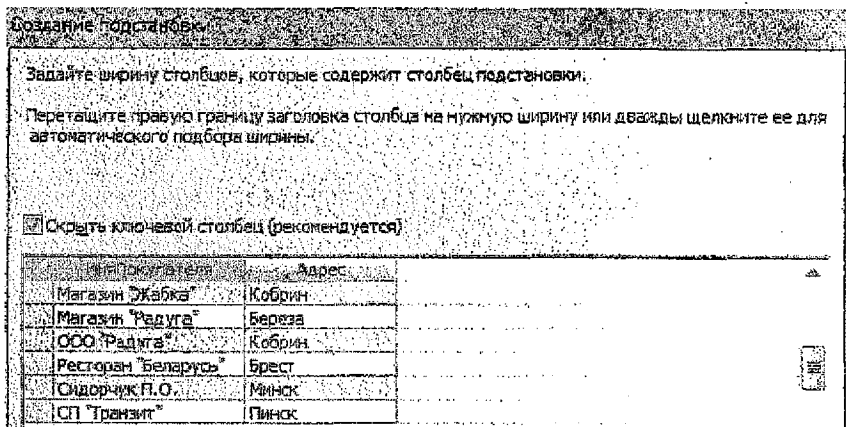


Рисунок 21 – Пятый шаг Мастера подстановок

6. В предпоследнем диалоговом окне можно оставить всё без изменений, соглашаясь с заданной подписью столбца подстановки (рис. 22), нажать кнопку **Готово**.

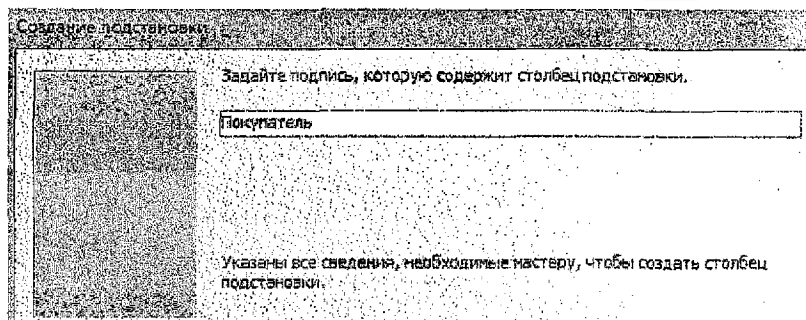


Рисунок 22 – Шестой шаг Мастера подстановок

7. В последнем диалоговом окне утвердительно ответить – **Да** (рис. 23).

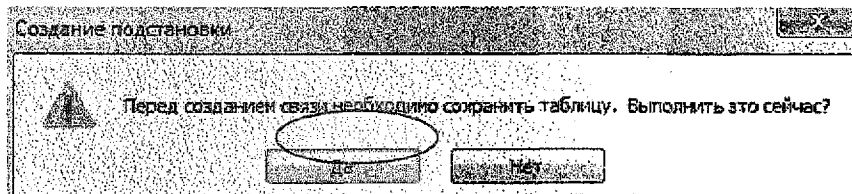



Рисунок 23 – Последний шаг Мастера подстановок

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

Аналогично используется Мастер подстановок для связи таблиц **Группа товаров** и **Товары** по полю **Группа Товара**.

### 3.11. Установка связей между таблицами

В MS Access межтабличные связи создаются и просматриваются с помощью окна «Схема данных», которое вызывается с помощью кнопки **Схема данных** :

в версии ACCESS 2003: на панели инструментов или соответствующей команде в п.м. Сервис,

в версии ACCESS 2007/2010: 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. 24).

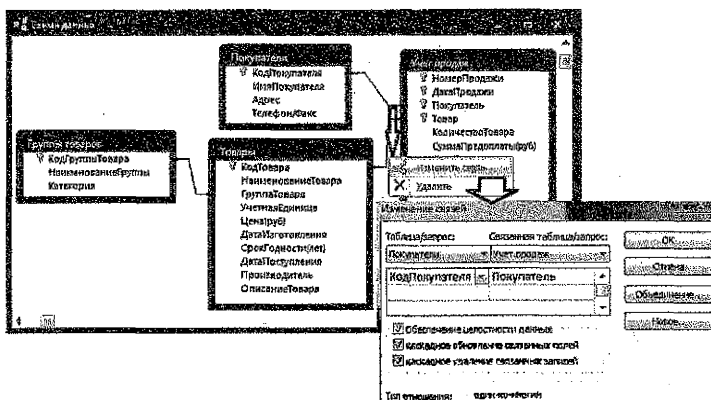


Рисунок 24 – Изменение линий связи в окне **Схема данных**

3. В диалоговом окне **Изменение связей** (рис. 24) установить флажок  **Обеспечение целостности данных**. При необходимости, в этом же окне можно установить каскадные операции:  **каскадное обновление связанных полей** и  **каскадное удаление связанных полей**.

4. Нажать кнопку **Создать** (или **OK**), после чего связи будут представлены в виде линий, соединяющих **первичный ключ** основной (или главной) таблицы с полем **внешнего ключа** связанной (или подчинённой) таблицы (рис. 25).

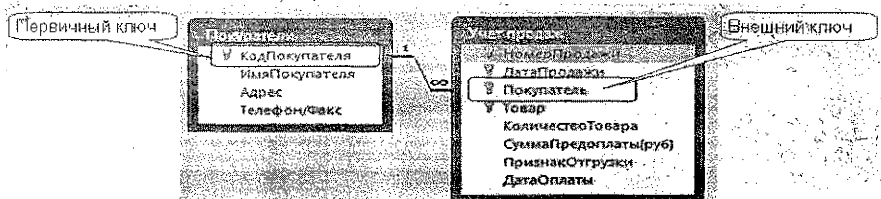


Рисунок 25 – Связь между таблицами, обеспеченная целостностью данных.

5) Закрыть 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. 26).

КодПокупателя	ИмяПокупателя	НомерПродаж	ДатаПродаж	Покупатель	Товар
1162	Дедушкин П.И.	1	13.05.2016	Магазин "Берес"	Паркет TEMPА
1103	Леонович С.В.	2	13.06.2016	Золотерев И.И.	Обои GRAND
1204	Кривизин А.Р.	3	14.06.2016	ИП Сидоров С.С.	Краска для пола KRAS
1303	Сидорчук П.О.	3	14.06.2016	ИП Сидоров С.С.	Валик ГРЕЙС с ручкой
1201	ИП Сидорова С.С.	4	13.07.2016	Сидорчук П.О.	Паркет CLASSIC
1202	ИП Петренко П.П.	5	15.07.2016	СП "Гранит"	Обои WIL TAN
1283	ИП Иванов И.И.	5	15.07.2016	СП "Гранит"	Линолеум ТАКТ
1801	ООО "Радуга"	6	17.07.2016	Ресторан "Беларусь"	Краска интерьерная SWET
1902	СП "Гранит"	6	17.07.2016	Ресторан "Беларусь"	Валик ГРЕЙС с ручкой
1402	Ресторан "Беларусь"	7	09.08.2016	Леонович С.В.	Диск записной по металлу
1402	Кафе "Прайм"	8	10.08.2016	СП "Гранит"	Краска для потолка СНЕЖОК
		9	11.08.2016	ООО "Радуга"	Ламинат ЭЗАР

Рисунок 26 – 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 окне **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 **Учет продаж**). Откроется диалоговое окно **Изменение связей**, в котором надо выполнить пункты 3-4 текущего раздела.

### 3.12. 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 данными.

**Условие:** заполнить данными сначала главные таблицы, а затем подчиненные таблицы.

### 3.15.1. 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: команду **Удалить столбец** из пункта меню **Правка**;

в версии ACCESS 2007/2010: команду **Удалить столбцы** на вкладке ленты **Конструктор** в группе инструментов **Настройка запроса**.


### 3.15.2. 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**.

### 3.15.3. Запуск запроса

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

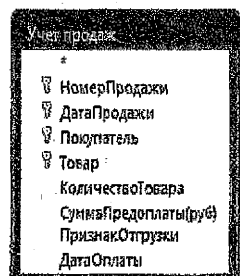
в версии ACCESS 2003: на панели инструментов нажать кнопку **«Запуск»**  или выполнить команду **Запуск** из пункта меню **Запрос**;

в версии ACCESS 2007/2010: команду **Выполнить** на вкладке ленты **Конструктор** в группе инструментов **Результаты**.

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

**Условие ЗАПРОСА1:** вывести список клиентов по алфавиту, которым товар не отгружен.

Бланк Конструктора запроса представлен на рис. 29.



Поле:	НомерПродажи	ДатаПродажи	Покупатель	Товар	ПризнакОтгрузки
Имя таблицы:	Учет продаж	Учет продаж	Учет продаж	Учет продаж	Учет продаж
Сортировка:			по возрастанию		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:					Да
или:					

Рисунок 29 – Запрос1 в режиме Конструктора

### 3.16. 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

кнопку  **Построить** на панели инструментов.

в версии ACCESS 2007/2010: выбрать

пиктограмму  **Построитель** на вкладке ленты **Конструктор**: для задания свойств полей – в группе инструментов **Сервис**, для использования в бланке запроса – в группе **Настройка запроса**.

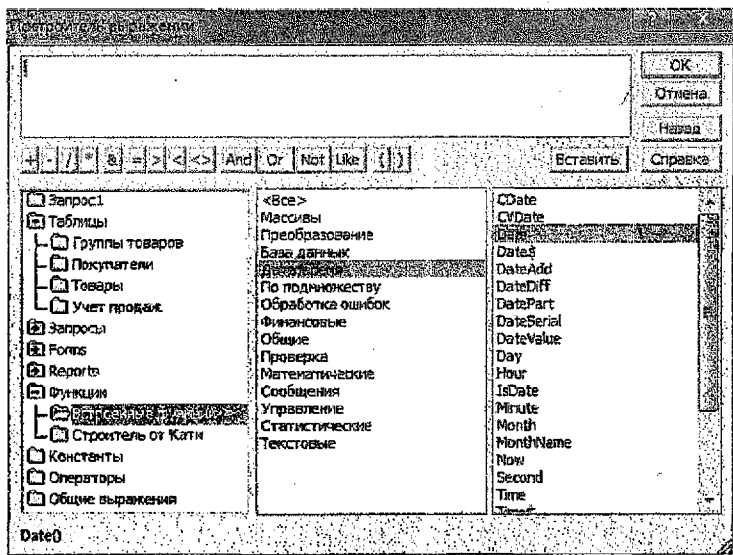


Рисунок 30 – Окно Построителя выражений

Окно построителя выражений состоит из трех разделов (рис. 30):

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.

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. 30).

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 **ОК**.

Пример выбора поля таблицы рассмотрен на рисунке 35 при создании вычисляемого запроса, а пример использования **встроенной функции** приведен на рисунке 30.

#### **Замечания:**

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

### **3.17. Формирование условий отбора**

Список операторов, используемых при задании выражений следующий:

#### **> операторы сравнения:**

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

**BETWEEN** – позволяет задать диапазон значений.

Синтаксис: **Between «Выражение» And «Выражение»** (например: **BETWEEN 10 And 20** означает то же, что и логическое выражение **>=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 конъюнкция, т. е. логическое умножение. Объединяет два и более условий и возвращает истинное значение только при выполнении всех условий (например, от 10 и до 20: **>=10 AND <=20**).

**OR** – логическое **ИЛИ**, эквивалент логического сложения – дизъюнкции. Связывает два или больше условий, но возвращает истинный результат при выполнении хотя бы одного условия (например, до 50 или выше 100: **<50 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](#).

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

Для того, чтобы сообщить Access, что вводится 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](#).

### 3.19. Сложные критерии выборки

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.

**Условие ЗАПРОСА2:** выбрать записи по продажам обоев GRAND в количестве более 5-ти рулонов.

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 них выбираются те, количество продаж которых *более 5-ти*. Бланк Конструктора запроса представлен на рис. 31.

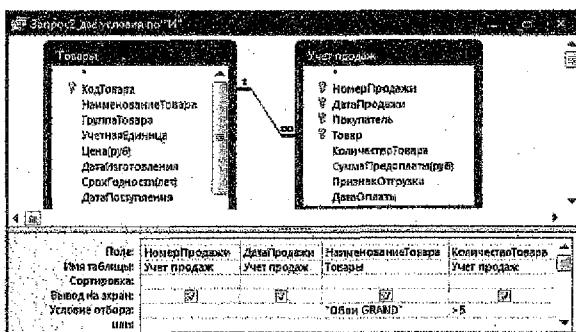


Рисунок 31 – Запрос2 в режиме Конструктора



**Условие ЗАПРОСА3:** выбрать информацию по ламинату 32-го и 33-го классов.

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

В сформированном запросе использована возможность вывода всех полей таблицы *Товары* с помощью символа «\*». Бланк Конструктора запроса представлен на рис. 32.

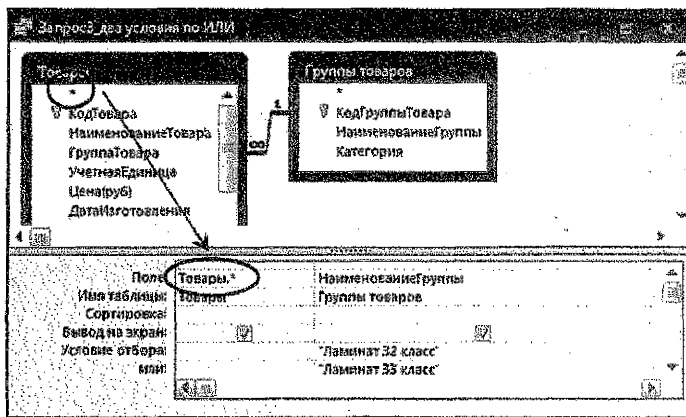


Рисунок 32 – Запрос3 в режиме Конструктора

**Замечание:** Возможен другой вариант задания двух критериев в одной строке *Условие отбора* для поля *Наименование группы*: "Ламинат 32 класс" **Or** "Ламинат 33 класс".

## Практическое задание 6. Параметрические запросы.

### Порядок выполнения:

#### 3.20. Проектирование параметрических запросов

Запрос с параметрами выводит одно или несколько predeterminedных диалоговых окон, в которых пользователь должен ввести значения параметров (условия отбора).

Параметрический запрос каждый раз при выполнении требует ввода определенных параметров. Преимущества данного вида запросов:

- не нужно постоянно модифицировать запрос в режиме Конструктора;
- удобно использовать в формах и отчетах, т. к. каждый раз при их открытии Access запрашивает у пользователя требуемый параметр.

Чтобы установить параметр, нужно вместо конкретных данных в Бланк запроса в строку *Условие отбора* ввести имя или фразу, заключенную в квадратные скобки, т. е. то, что заключено в [ ], Access рассматривает как параметр.

Например, для поля, которое выводит даты, можно ввести приглашения следующего вида: «Введите начальную дату.» и «Введите конечную дату.», чтобы задать границы

диапазона значений. В ячейке *Условие отбора* поля типа Дата/время будет стоять выражение **Between** [Введите начальную дату:] **And** [Введите конечную дату:].

**Условие Запроса4:** *Выбрать продажи за временной интервал, границы которого вводятся с клавиатуры.*

Данный запрос относится также к виду «И-запросы» с двумя критериями отбора по одному полю таблицы. Бланк Конструктора запроса представлен на рис. 33.

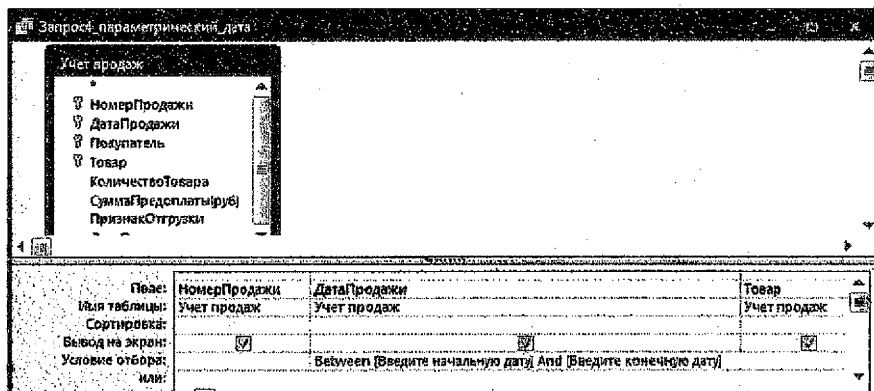


Рисунок 33 – Запрос4 в режиме Конструктора

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. 34).

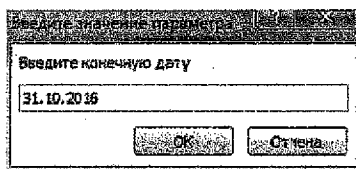
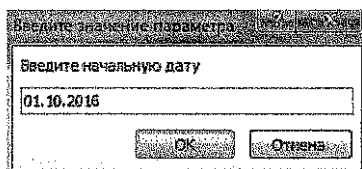


Рисунок 34 – Ввод границ временного интервала при выполнении запроса4

## Практическое задание 7. Вычисляемые запросы.

### Порядок выполнения:

#### 3.21. Вычисляемые поля в запросах

Существует ряд вычислений, которые можно выполнить в запросе. Например, найти сумму или среднее по значениям одного поля, перемножить значения двух полей или вычислить дату, отстоящую от текущей на три месяца.

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* бланка с помощью **Построителя выражений** и начинается с названия поля и двоеточия.

**Условие ЗАПРОСА5:** *Посчитать стоимость товаров в рублях. Определить окончательный размер оплаты покупки покупателем с учетом предоплаты.*

В данном запросе создаются последовательно два вычисляемых поля:  
Сначала создается поле для вычисления стоимости покупки в рублях:

Стоимость товара (руб): [Товары]![Цена(руб)] \* [Учет продаж]![КоличествоТовара]

Затем запрос сохраняется для того, чтобы иметь возможность в следующем вычисляемом поле сослаться на уже созданное поле.

Далее формируется следующее поле с помощью **Построителя**, в котором поле *Стоимость покупки (руб)* выбирается в нижней части окна в левой области списка источников данных в папке текущего запроса (рис. 35):

Сумма доплаты (руб): [Стоимость товара (руб)] – [Учет продаж]![СуммаПредоплаты(руб)]

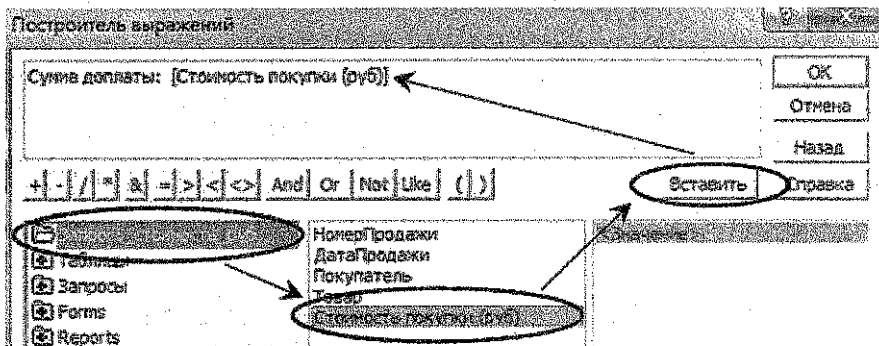


Рисунок 35 – Ввод границ временного интервала при выполнении запроса

Бланк Конструктора запроса представлен на рис. 36.

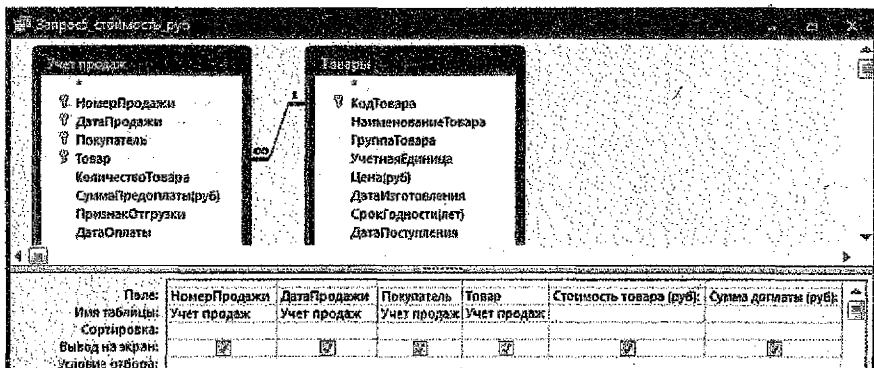


Рисунок 36 – Запрос в режиме Конструктора

**Условие ЗАПРОСА6:** Посчитать стоимость товаров в валюте, курс которой вводится как параметр с клавиатуры.

Для создания данного запроса можно использовать предыдущий запрос, в котором рассчитана стоимость в рублях. Для ввода курса валюты каждый раз при выполнении запроса он вводится в выражение для вычисляемого поля как параметр (см. раздел 3.22 в практическом задании 3):

Стоимость товара в валюте: [Запрос3\_стоимость]!([Стоимость покупки (руб)] / [Введите курс валюты])

Бланк Конструктора запроса представлен на рис. 37.

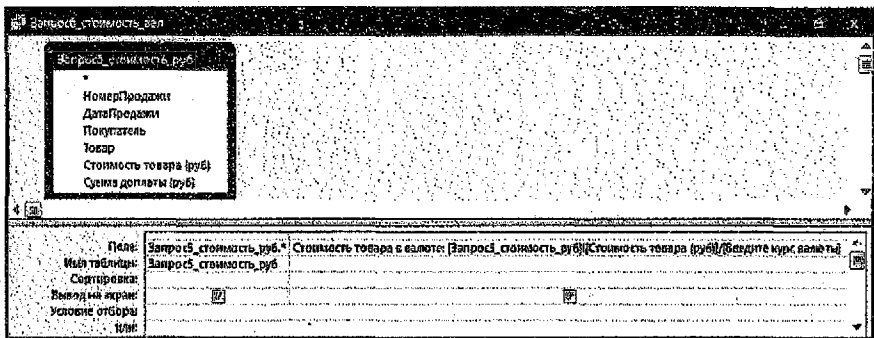


Рисунок 37 – Запрос в режиме Конструктора

**Условие Запроса7:** Определить: а) дату истечения срока годности товара, исходя из даты его изготовления и срока хранения; б) сколько месяцев осталось до истечения срока годности товара.

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

**Замечания:** 1) для того 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 для формирования выражений следующих полей; 2) синтаксис использованных функций см. в Приложении 3 и Приложении 4.

Выражения, построенные в Построителе для вычисляемых полей, следующие:

Дата истечения срока годности: `DateAdd("yyyy"; [Товары]![СрокГодности(лет)]; [Товары]![ДатаИзготовления])`

До истечения годности осталось месяцев: `([Дата истечения срока годности]-Date())/ 30`

Списание: `IF([До истечения годности осталось месяцев]<0; "списать")`

Бланк Конструктора запроса представлен на рис. 38.

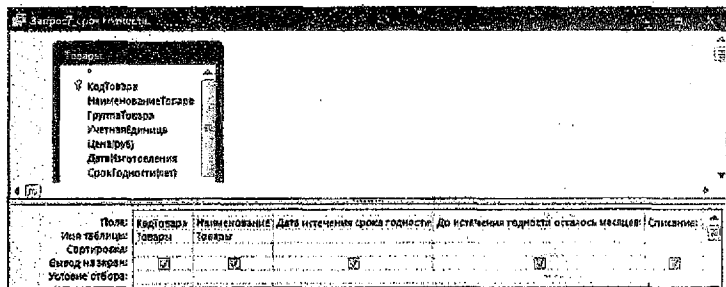


Рисунок 38 – Запрос7 в режиме Конструктора

**Условие Запроса8:** Вывести список товаров, до истечения срока годности которых осталось менее месяца, и вычислить для них цену со скидкой в размере 50%.

Для создания данного запроса можно использовать предыдущий запрос, в котором рассчитано количество месяцев до окончания срока хранения товара.

Выражение, построенное в Построителе для вычисляемого поля, следующее:

Цена со скидкой: `[Товары]![Цена(руб)] * 0,5`

Бланк Конструктора запроса представлен на рис. 39.

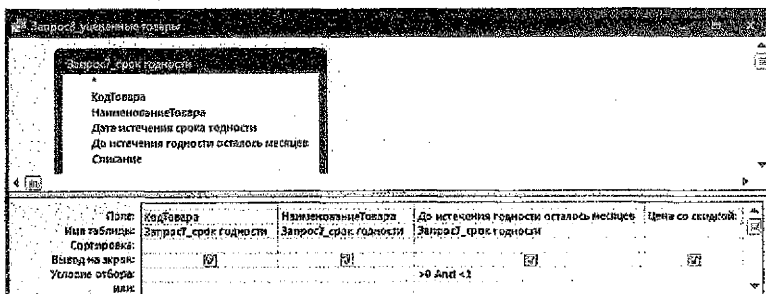


Рисунок 39 – Запрос8 в режиме Конструктора

**Условие Запроса9:** Для покупателей, не оплативших за товар в день покупки, определить, через сколько дней от них поступила оплата.

Выражение, построенное в Построителе для вычисляемого поля, следующее:

Кол-во дней до полной оплаты: [Учет продаж]![ДатаОплаты] – [Учет продаж]![ДатаПродажи]

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

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

DateDiff("d";[Учет продаж]![ДатаПродажи];[Учет продаж]![ДатаОплаты])

Бланк Конструктора запроса представлен на рис. 40.



Рисунок 40 – Запрос9 в режиме Конструктора

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

Сначала создается поле, в котором определяется название дня недели продаж:

День недели продаж: WeekdayName(Weekday([Учет продаж]![ДатаПродажи]);0;1)

После сохранения запроса создается поле для вычисления скидки:

Скидка: IIf([День недели продаж]="Воскресенье";  
0,05 \* [Запрос5\_стоимость\_руб]![Стоимость товара (руб)]; IIf([День недели продаж]="суббота"; 0,03 \* [Запрос5\_стоимость\_руб]![Стоимость товара (руб)]))

Бланк Конструктора запроса представлен на рис. 41.

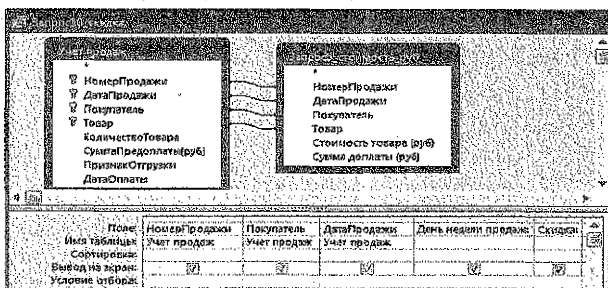


Рисунок 41 – Запрос10 в режиме Конструктора


## Практическое задание 8. Итоговые запросы.


### Порядок выполнения:

#### 3.22. Проектирование итоговых запросов

Запросы, которые выполняют вычисления в группах записей, называются итоговыми. Можно вычислить сумму, среднее, минимальное и максимальное значения поля.

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

в версии ACCESS 2003: вызвать команду **Групповые операции** из пункта меню **Вид** или нажать на кнопку  на панели инструментов;

в версии ACCESS 2007/2010: выбрать пиктограмму  **Итоги** на вкладке ленты **Конструктор** в группе инструментов **Показать или скрыть**.

После этого в Бланке запроса появится новая строка **Групповые операции**. В ней для соответствующего поля указывается тип выполняемого вычисления из раскрывающегося списка установок групповых операций:

<b>Групповая функция</b>	<b>Результат</b>
Sum	Сумма значений поля
Avg	Среднее значение поля
Min	Минимальное значение поля
Max	Максимальное значение поля
Count	Число значений в поле (не считая пустые)
StDev	Среднеквадратичное отклонение значений поля
Var	Дисперсия значений поля
First	Возвращают значение поля из первой записи результирующего набора запроса
Last	Возвращают значение поля из последней записи результирующего набора запроса

<b>Групповая установка</b>	<b>Результат</b>
Группировка	Определяет группы записей, для которых рассчитываются итоговые значения.
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»).

**Условие Запроса11:** Определить, на какую сумму, в каком объеме и сколько раз продавался каждый товар.

Можно воспользоваться ранее созданным запросом, в котором была рассчитана стоимость товара.

Бланк Конструктора запроса представлен на рис. 42.

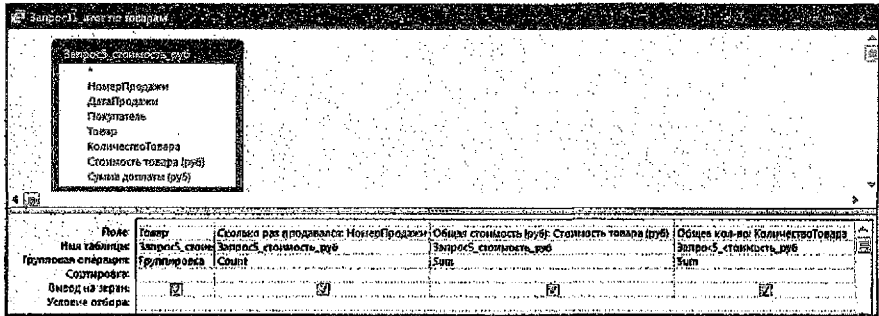


Рисунок 42 – Запрос11 в режиме Конструктора

**Условие Запроса12:** Определить количество проданных товаров по каждой группе в разрезе производителей.

Бланк Конструктора запроса представлен на рис. 43.

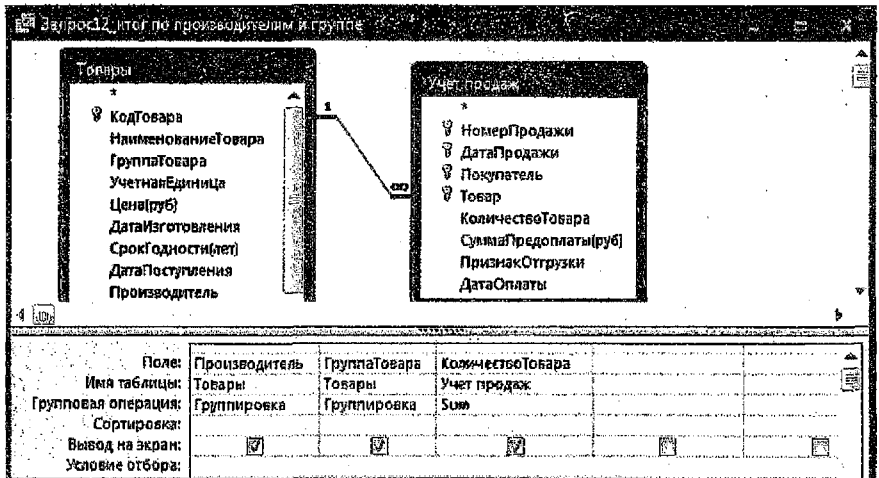


Рисунок 43 – Запрос12 в режиме Конструктора



**Условие Запроса13:** Определить, на какую сумму продано товаров по каждой группе товаров из категории "Напольные покрытия".

В данном случае использовалась групповая установка **Выражение** для обозначения вычисления значения стоимости товара непосредственно в текущем запросе и обработки его групповой функцией **Sum**:

Общая сумма:  $Sum([Товары].[Цена(руб)] * [Учет продаж].[КоличествоТовара])$

Бланк Конструктора запроса представлен на рис. 44.

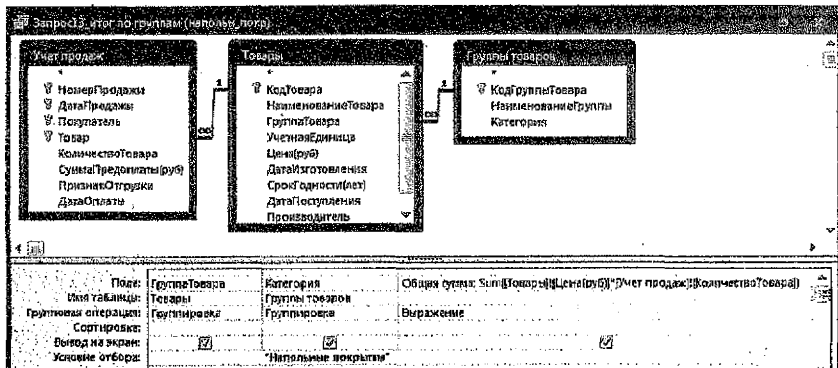


Рисунок 44 – Запрос13 в режиме Конструктора

**Условие Запроса14:** Определить сумму продаж за каждый день недели.

В данном запросе используется ранее созданный запрос, в котором была рассчитана стоимость товара, и создается поле, в котором определяется название дня недели продаж:

День недели:  $Format([Учет продаж].[ДатаПродажи], "dddd")$

Бланк Конструктора запроса представлен на рис. 45.

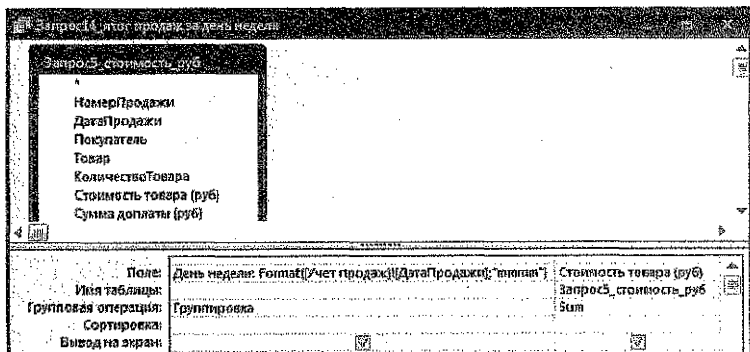


Рисунок 45 – Запрос14 в режиме Конструктора

**Условие Запроса15:** Определить количество наименований товаров, у которых год изготовления – 2015.

В данном случае использовалась групповая установка **Условие** для обозначения критерия отбора по данному полю, которое при этом не отображается в результате запроса. Выражение, построенное в Построителе для извлечения значения года поля *Дата изготовления*, следующее:

Год изготовления: Year([Товары].[ДатаИзготовления])

**Замечание 1:** можно использовать другие встроенные функции для извлечения значения года поля с типом *Дата/Время*, 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 запроса представлен на рис. 46.

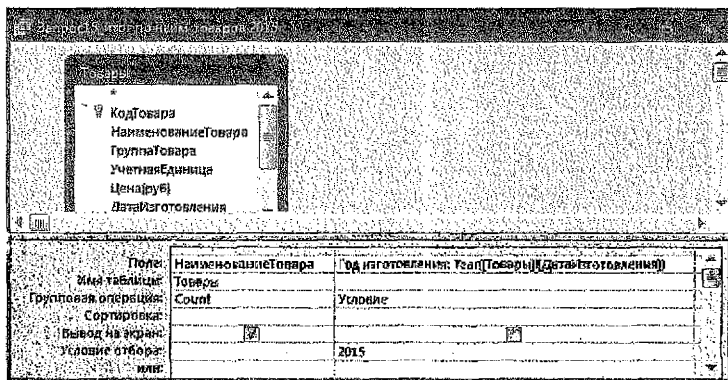


Рисунок 46 – Запрос15 в режиме Конструктора

**Условие Запроса16:** Определить, на какую сумму было продано товаров за каждый месяц определенного квартала, номер которого вводится с клавиатуры как параметр.

Выражения, построенные в Построителе для преобразования поля *ДатаПродажи*, следующие:

Месяц: MonthName(Month([Учет продаж].[ДатаПродажи]))

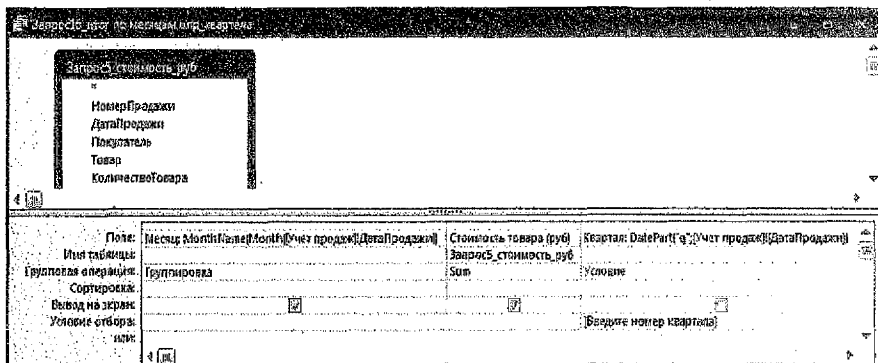
Квартал: DatePart("q";[Учет продаж].[ДатаПродажи])

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

Месяц: Format([Учет продаж].[ДатаПродажи];"mmmm")

Квартал: Format([Учет продаж].[ДатаПродажи];"q")

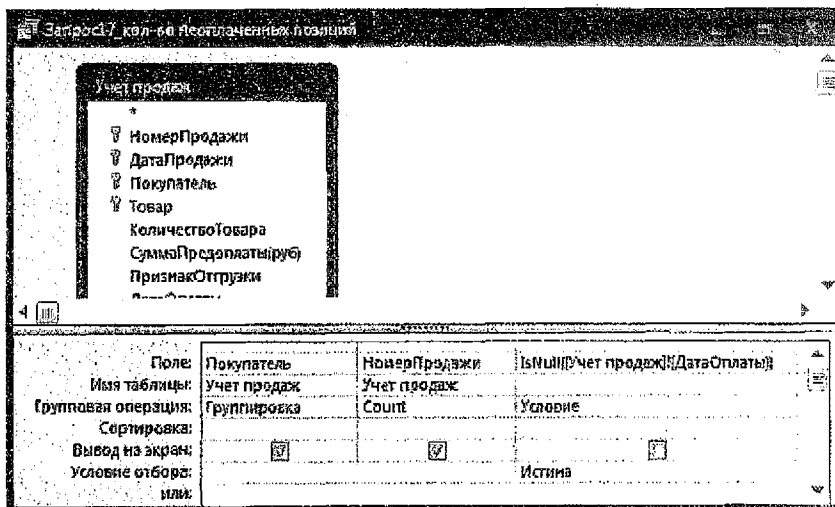
Бланк Конструктора запроса представлен на рис. 47.



**Условие Запроса17:** Подсчитать количество неоплаченных позиций продаж по каждому покупателю

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 запроса представлен на рис. 48.



## Практическое задание 9. Перекрестные запросы.

### Порядок выполнения:

#### 3.23. Проектирование перекрестных запросов


**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.

Перекрестные запросы можно создавать как в режиме *Конструктора*, так и с помощью *Мастера*.

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

- ✓ сначала создать обычный запрос-выборку (см. пп. 1 – 4 раздела 3.15. практической части);
- ✓ далее необходимо преобразовать тип данного запроса на **Перекрестный** следующим образом:

в версии ACCESS 2003: находясь в окне Конструктора, из пункта меню **Запрос** выбрать команду **Перекрестный**;


в версии ACCESS 2007/2010: выбрать пиктограмму  Перекрестный на вкладке ленты **Конструктор** в группе инструментов **Тип запроса**;

- ✓ 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 выбирается необходимая статистическая функция.

**Условие Запроса18**: Посчитать среднюю цену товаров в каждой группе в разрезе производителей. Результат представить в виде перекрестной таблицы. Добавить итоговый столбец, в котором вывести среднюю цену товаров в каждой группе.

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

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

в версии ACCESS 2007/2010: выбрать пиктограмму  Мастер запросов на вкладке ленты **Создать** в группе инструментов **Другие** (или в группе **Запросы**);

- ✓ в появившемся диалоговом окне **Новый запрос** выбрать опцию **Перекрестный запрос** (рис. 49) и нажать кнопку **ОК**;

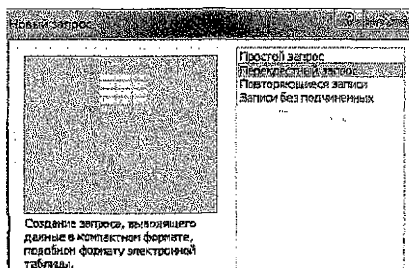


Рисунок 49 – Окно выбора "Новый запрос"

- ✓ в появившемся окне **Создание перекрестных таблиц** выбрать из списка таблиц источник **Товары** (рис. 50). Нажать кнопку **Далее**.

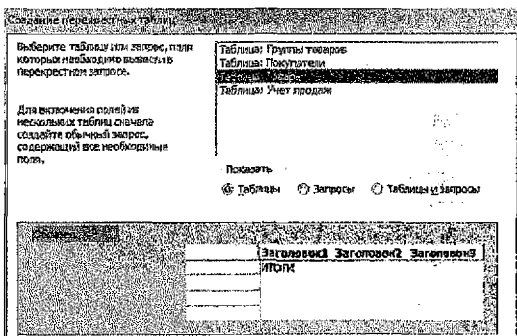


Рисунок 50 – Первый шаг Мастера создания перекрестного запроса

- ✓ на втором шаге **Мастера** выбрать с помощью кнопки **>** поле **ГруппаТовара**, значения которого будут использоваться в качестве заголовков строк (рис. 51). Нажать кнопку **Далее**.

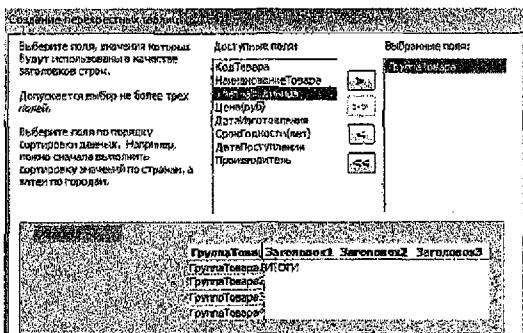


Рисунок 51 – Второй шаг Мастера создания перекрестного запроса

- ✓ на третьем шаге **Мастера** выбрать поле **Производитель**, значения которого будут использованы в качестве заголовков столбцов (рис. 52). Нажать кнопку **Далее**.

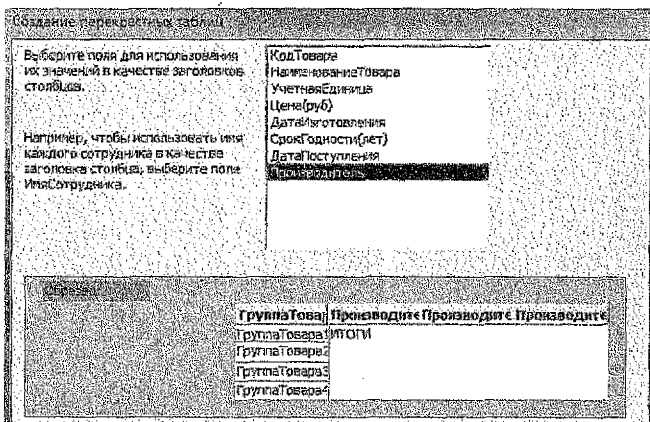


Рисунок 52 – Третий шаг Мастера создания перекрестного запроса

- ✓ 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. 53). Нажать кнопку **Далее**.

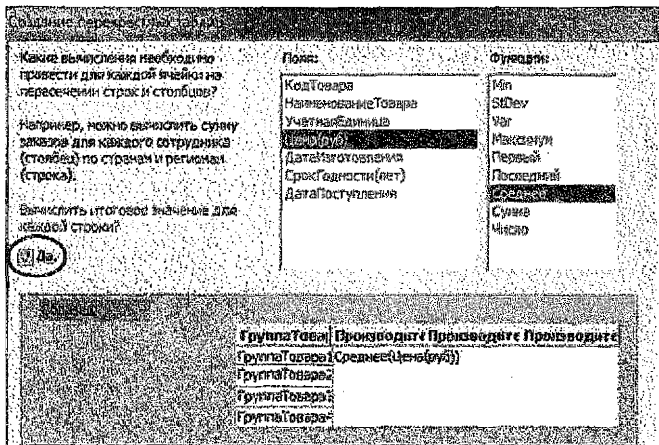


Рисунок 53 – Четвертый шаг Мастера создания перекрестного запроса

- ✓ 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. 54). Нажать кнопку **Готово**.

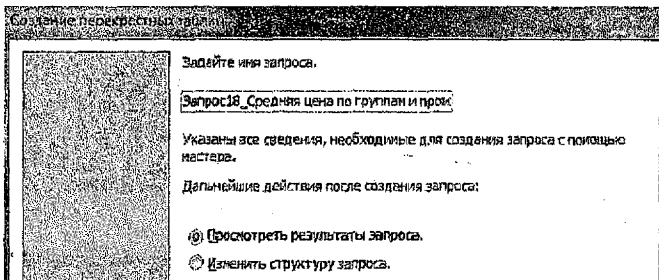


Рисунок 54 – Последний шаг Мастера создания перекрестного запроса

Вид таблицы в результате запроса представлен на рис. 55.

ГруппаТовара	Итоговое значение	Цена(руб)	Беларусь	Германия	Италия	Китай	Россия	Украина
Ламинат 32 класс	13,00р.							
Ламинат 33 класс	78,00р.	78,00р.						
Перьевая доска	114,00р.	150,00р.						
Дюймовик	10,20р.	10,20р.						
Краска водо-дисперсион	21,30р.	21,30р.						
Краска-эмаль	98,30р.				189,00р.		5,00р.	
Лак для паркета	70,00р.							
Обои Бумажные	0,00р.							
Обои Флизелиновые	33,30р.				53,50р.		0,00р.	
Инструмент слесарный	3,50р.							3,50р.
Инструмент малярный	0,25р.							

Бланк Конструктора запроса представлен на рис. 55.

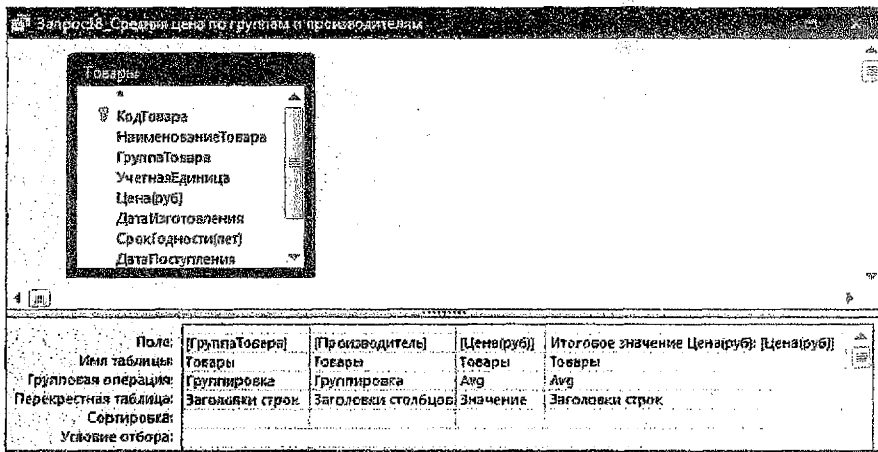


Рисунок 55 – Запрос18 в режиме Конструктора

**Условие Запроса19:** Подсчитать сумму продаж по каждому товару за каждый месяц. Результат представить в виде перекрестной таблицы. Добавить итоговый столбец, в котором вывести максимальное количество проданного товара.

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

Месяц: `MonthName(Month([Учет продаж][ДатаПродажи]))`

Общая стоимость: `[Учет продаж][КоличествоТовара] * [Товары][Цена(руб)]`

Бланк Конструктора запроса представлен на рис. 56.

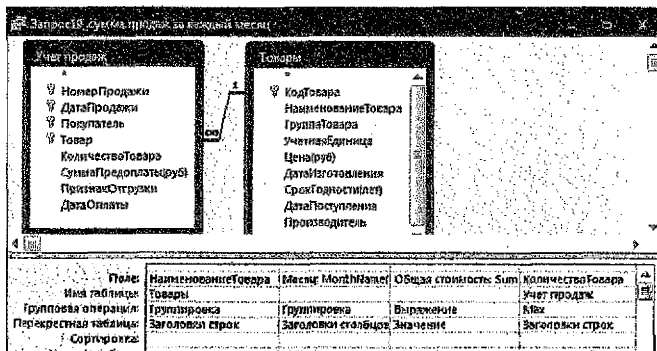


Рисунок 56 – Запрос19 в режиме Конструктора

**Условие Запроса20:** Посчитать, сколько раз продано каждого товара за каждый день недели определенного месяца, название которого вводится с клавиатуры как параметр. Результат представить в виде перекрестной таблицы.

Выражения, построенные в Построителе для преобразования поля *ДатаПродажи*, следующие:

День недели: `WeekdayName(Weekday([Учет продаж][ДатаПродажи];2))`

Месяц: `MonthName(Month([Учет продаж][ДатаПродажи]))`

Замечание: для определения типа данных параметра необходимо его добавить в окно **Параметры запроса** (рис. 57), которое открывается следующим образом:

в версии ACCESS 2003: выбрать команду **Параметры** из пункта меню **Запрос**.

в версии ACCESS 2007/2010: выбрать пиктограмму **Параметры** на вкладке ленты **Создание** в группе инструментов **Показать или скрыть** (или в группе **Запросы**).

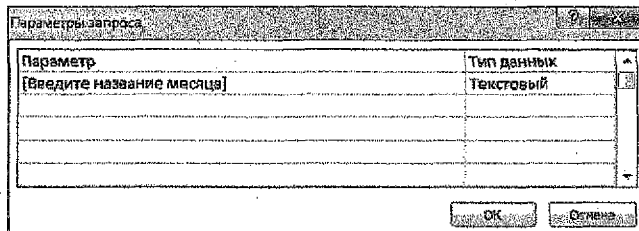


Рисунок 57 – Окно Параметры запроса



Бланк Конструктора запроса представлен на рис. 58.

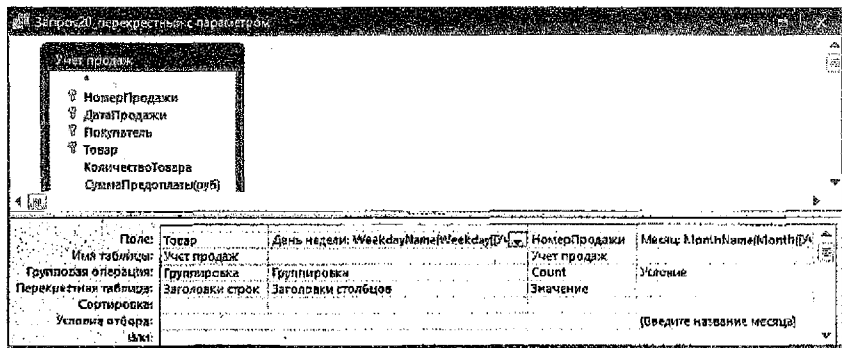


Рисунок 58 – Запрос20 в режиме Конструктора

## Практическое задание 10. Проектирование форм.

### 3.24. Способы создания форм

**Формы** – это диалоговый графический интерфейс для работы пользователя с БД. Любая форма, с помощью которой есть необходимость 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.

### 3.25. 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 пункта 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 2007/2010:

находясь в окне БД, в списке объектов **Таблицы** (в области переходов – левая часть окна приложения) выделить таблицу, для которой создаётся форма, и вы-

брать пиктограмму  **Формы** на вкладке ленты **Создание** в группе инструментов **Формы**.


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

### Порядок выполнения:

#### 3.26. 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 2007/2010: выбрать пиктограмму  **Мастер форм** на вкладке ленты **Создание** в группе инструментов **Формы** (может находиться в дополнительном меню **Другие формы**);

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. 59), нажать кнопку **Далее**;

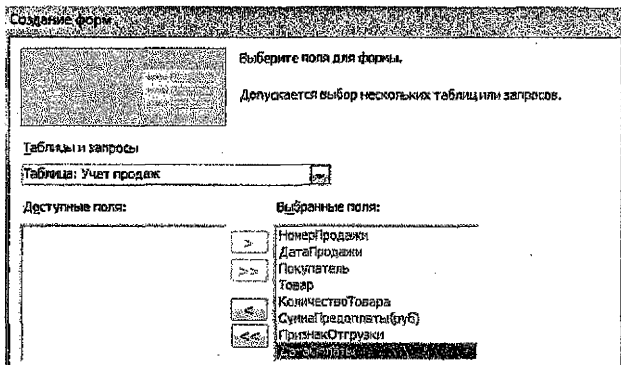


Рисунок 59 – Первый шаг Мастера форм

**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 рис. 60.

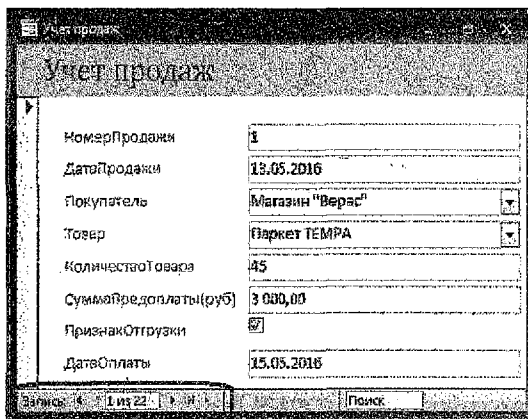



Рисунок 60 – Вид формы в режиме просмотра данных

Просматривать записи в форме можно, используя кнопки внизу окна формы.

### 3.27. Создание формы в режиме Конструктора

Для создания формы в режиме Конструктора необходимо:

в версии ACCESS 2003: находясь в окне БД, на вкладке **Формы** нажать кнопку **Создать**; в окне **Новая форма** задать режим **Конструктор**.

в версии ACCESS 2007/2010: выбрать пиктограмму  **Конструктор форм** на вкладке ленты **Создание** в группе инструментов **Формы**).

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.).

### 3.28. Структура формы

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

Области окна формы (рис. 61):

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

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

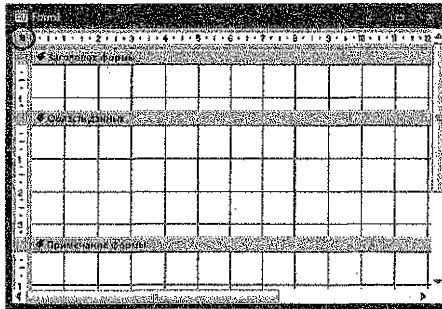



Рисунок 61 – Области окна формы в режиме Конструктора

Объекты 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 2007/2010 располагается на вкладке ленты **Конструктор** (рис. 62).



версия ACCESS 2003



версия ACCESS 2007/2010

Рисунок 62 – 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 к формам и отчетам ещё некоторые функциональные возможности.

### 3.29. 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 элементом 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 2007/2010: выбрать пиктограмму  **Страница свойств** на вкладке ленты **Конструктор** в группе инструментов **Сервис**.

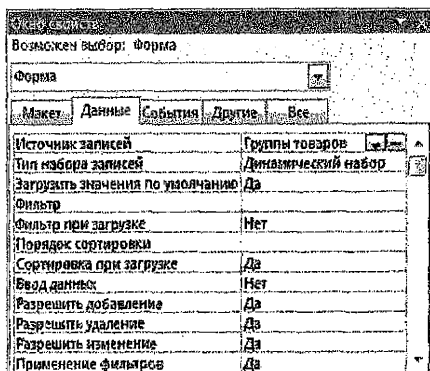
**Замечание 1:** для отображения окна свойств выбирается команда **Свойства** из контекстного меню, вызываемого щелчком правой кнопки мыши по выделенному объекту.

**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.

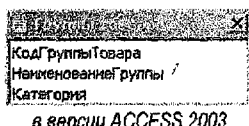
В окне **Свойств** все свойства разбиты по группам на вкладках **Макет**, **Данные**, **События**, **Другие**.

### 3.30. Размещение элемента управления "Поле" в форме

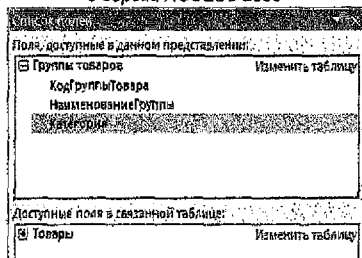
При создании формы в режиме Конструктора необходимо связать её с источником данных. Для этого в свойстве формы (см. **замечание 2** предыдущей темы) на вкладке **Данные** в строке **Источник записей** с помощью элемента управления "список" () выбрать источник данных, т. е. таблицу или запрос (рис. 63-а).



а)



в версии ACCESS 2003




в версии ACCESS 2007/2010


б)

Рисунок 63 – Окно свойств формы и список полей источника данных

Далее в окне **Список полей** (рис. 63-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 соответствующая команда из пункта меню **Вид** или пиктограмма  на панели инструментов.

в версии **ACCESS 2007/2010** выбирается пиктограмма  **Добавить поля**, которая располагается на вкладке ленты **Конструктор** в группе инструментов **Сервис**.





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





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

### Порядок выполнения:


Открыть форму **Учет продаж** в режиме Конструктора. Переключение между режимами работы рассмотрены в п. 3.2. практической части (рис. 5-в).

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

✓ изменить размеры окна Конструктора с помощью стрелок , , ,  на границах окна;

- ✓ изменить размеры области формы (область заголовка, данных, примечания) с помощью стрелок  на границах области;
- ✓ передвинуть 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 ) , нажать кнопку мыши, перетащить в другое место и отпустить;
- ✓ передвинуть 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:

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. 64).

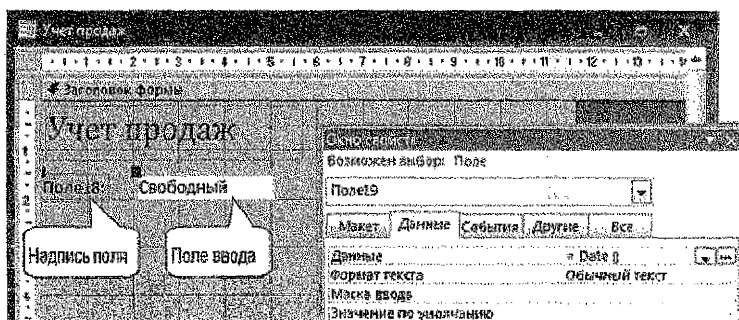



Рисунок 64 – Создание вычисляемого поля в форме

2. Далее для выделенного нового **поля ввода** необходимо открыть окно **Свойства** (см. п. 3.28. практической части), в котором на вкладке **Данные** выбрать свойство **Данные** (рис. 64). Для добавления выражения с использованием встроенных функций целесообразно использовать **Построитель выражений**, который можно открыть кнопкой  в правой части строки свойства **Данные** (использование Построителя см. в п. 3.16. 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.

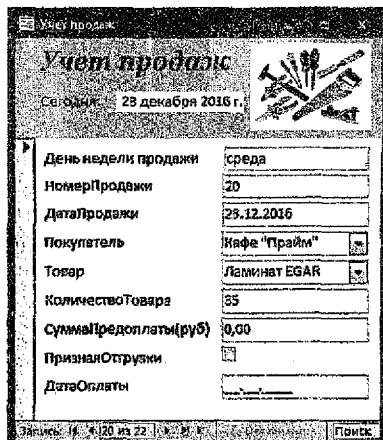
4. Закрыть окно свойств и откорректировать ширину поля и надписи по содержанию.

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 рис. 65.



Режим Конструктора



Режим формы

Рисунок 65 – Отредактированная форма Учет продаж

**Условие:** Создать многотабличную форму для просмотра и ввода данных таблиц Товары и Учет продаж.


### Порядок выполнения:

#### 3.31. Разработка многотабличных (составных) форм

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 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 2007/2010: выбрать пиктограмму  **Мастер форм** на вкладке ленты **Создание** в группе инструментов **Формы** (может находиться в дополнительном меню **Другие формы**);

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 в на этом же шаге выбрать таблицу **Учет 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. 66), нажать кнопку **Далее**

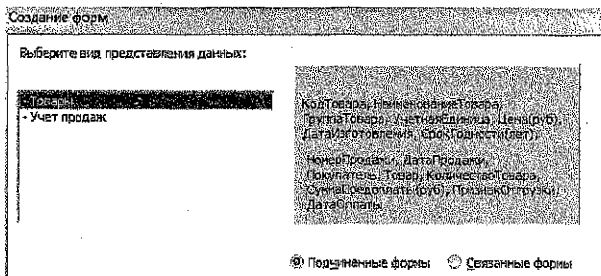


Рисунок 65 – Второй шаг Мастера создания многотабличной формы

**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. 60).

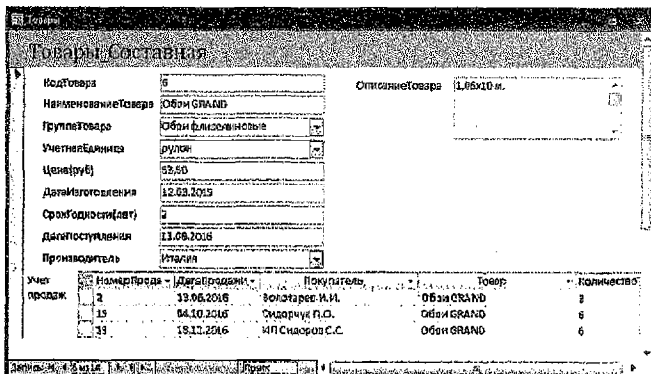



Рисунок 60 – Вид многотабличной формы в режиме просмотра данных

Просматривать записи в форме можно, используя кнопки внизу окна формы.

**Замечание:** в версии ACCESS 2007/2010 составную форму для главных таблиц (участвующих в связях в отношении "ОДИН") можно создавать автоматически, выделив её в списке объектов БД и выбрав пиктограмму  **Формы** на вкладке ленты **Создание** в группе инструментов **Формы**.

**Условие:** Создать отчет по продажам в разрезе покупателей.

### Порядок выполнения:

#### 3.3.2. Проектирование отчетов

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**; 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 2007/2010: выбрать пиктограмму **Мастер отчетов** на вкладке ленты **Создание** в группе инструментов **Отчеты**;

2) следовать шагам Мастера 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 по продажам в разрезе покупателей приведен на рис. 61.

### Отчет по покупателям

Имя покупателя	Дата отгрузки	Склад	Товар	Количество	Сумма
Магазин "Вираж"					
	1	13.05.2016	Паркет TEMPА	45	6 750,00р.
Итого для 'ПодПокупателя' = 1020 (1 запись)					
Sum				45	6 750,00р.
Залогарев И.И.					
	2	13.06.2016	Обои GRAND	3	160,50р.
	17	13.11.2016	Обои WILTON	4	35,00р.
Итого для 'ПодПокупателя' = 1101 (2 записей)					
Sum				7	195,50р.

Рисунок 61 – Отчет по продажам в разрезе покупателей

**Условие:** Создать кнопочную форму для управления объектами и самой БД.


### Порядок выполнения:

#### 3.33. Создание управляющей (кнопочной) формы

Порядок создания кнопочной формы:

1. Создать пустую (т.е. без указания источника данных) форму в режиме Конструктора (см. п. 3.26. практической части).

2. Добавить в область данных текст заголовка с помощью ЭУ **Надпись** *As*, 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. 62), нажать кнопку **Далее**;

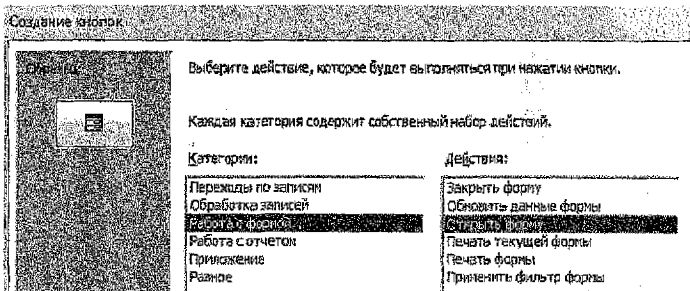


Рисунок 62 – Первый шаг Мастера создания кнопок

на 2-м шаге выбрать открываемый объект, например, **составную (многотабличную) форму** (рис. 63), нажать кнопку **Далее**;

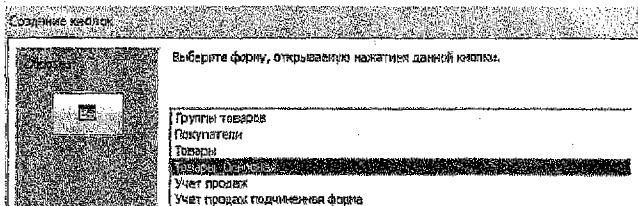


Рисунок 63 – Второй шаг Мастера создания кнопок

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 выделив 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 можно найти в контекстном меню для выделенных объектов.

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

Пример управляющей формы для БД Магазин "Строитель" приведен на рис. 64.

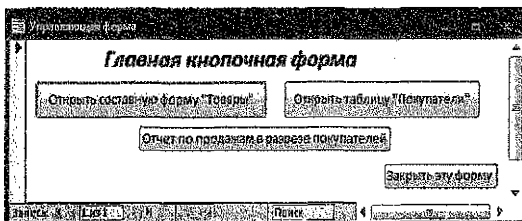



Рисунок 64 – Управляющая кнопочная форма БД Магазин "Строитель"

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

### 3.34. Использование макросов

**Макрос** – это средство для автоматизации задач и добавления функциональных возможностей формы, отчеты и элементы управления. В приложении Access макросы можно рассматривать как упрощенный язык программирования, на котором программа записывается в виде списка макрокоманд для выполнения. Макрос создается с помощью **Конструктора макросов**, который открывается следующим образом:

в версии ACCESS 2003: находясь в окне БД, на вкладке **Макросы** нажать кнопку **Создать**.

в версии ACCESS 2007/2010: выбрать пиктограмму  **Макрос** на вкладке ленты **Создание** в группе инструментов **Другие** (или в группе **Макросы и код**).

Пример создания макроса для открытия таблицы **Покупатели** приведен на рис. 65.

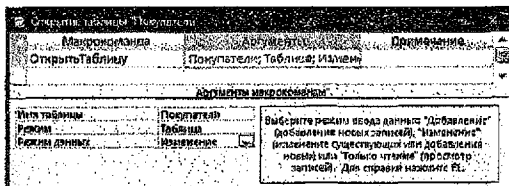


Рисунок 65 – Окно Конструктора макроса

**ПРИЛОЖЕНИЕ 1. Описание структур таблиц БД Магазин "Строитель"**

**Таблица «Группы товаров»**

<i>Имя поля</i>	<i>Тип данных</i>	<i>Размер</i>	<i>Другие свойства</i>
КодГруппыТовара	Числовой	Длинное целое	Ключ
Наименование группы	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

**Таблица «Товары»**

<i>Имя поля</i>	<i>Тип данных</i>	<i>Размер*</i>	<i>Другие свойства</i>
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		

**Таблица «Покупатели»**

<i>Имя поля</i>	<i>Тип данных</i>	<i>Размер</i>	<i>Другие свойства</i>
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)" "9991-991-99;0;*

**Таблица «Учет продаж»**

<i>Имя поля</i>	<i>Тип данных</i>	<i>Размер</i>	<i>Другие свойства</i>	
НомерПродажи	Составной ключ	Числовой	Длинное целое	Обязательное поле; Индексированное – Да (Допускаются совпадения)
ДатаПродажи		Дата/время		Обязательное поле; Маска ввода – краткий формат даты (00.00.0000;0;_); Значение по умолчанию – Date( ) Индексированное – Да (Допускаются совпадения)
КодПокупателя		Числовой	Длинное целое	Обязательное поле; Индексированное – Да (Допускаются совпадения); Подстановка: значения из таблицы «Клиенты»
КодТовара		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; ". " .

1                      2                      3

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 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 не обязателен).
&	Любой символ или пробел (ввод обязателен).
C	Любой символ или пробел (ввод не обязателен).
<	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, которые вводят условия обора для даты и времени:

**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 с 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
ууу	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 ≤ дата2.



#### ПРИЛОЖЕНИЕ 4. Функции Iif( ) и Format( )

Функция Iif(условие; еслиИстина; еслиЛожь) – возвращает один из двух аргументов в зависимости от результата вычисления выражения. Находится в категории "Управление".

<b>Аргумент</b>	<b>Назначение</b>
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. Находится в категории "Текстовые".

Для выражений даты/времени можно применять следующие символы в инструкции форматирования:

<b>Символ</b>	<b>Описание</b>
c	Полный формат даты
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)
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-значное значение года (от 100 до 9999)

### ПРИЛОЖЕНИЕ 5. Примеры условий отбора

Поле	Выражение	Заказы, отобранные в запросе
Город получателя	"Москва"	Заказы, отправленные в Москву
Город получателя	"Москва" Or "Васюки"	Заказы, отправленные в Москву или Васюки
Дата исполнения	=#2.02.16#	Заказы, отправленные 2-фев-2016 г.
Дата исполнения	Between #5-январь-16# And #10-январь-16#	Заказы отправленные не раньше 5-январь-16 и не позже 10-январь-16
Страна получателя	In ("Чили", "Перу", "Россия")	Заказы, отправленные в Чили, или Перу, или Россию
Страна получателя	Not "США"	Заказы, отправленные во все страны, кроме США
Дата размещения	< Date() - 30	Заказы с давностью более 30 дней
Дата размещения	Year([Дата размещения])=2016	Заказы за 2016 г.
Дата размещения	DatePart("q", [Дата размещения])=4	Заказы за четвертый квартал
Дата размещения	Year([Дата размещения]) = Year(Now()) And Month([Дата размещения]) = Month(Now())	Заказы за текущий месяц текущего года
Название получателя	Like "A*"	Заказы получателей, названия которых начинаются на букву "А"
Название получателя	Like "*рулетка"	Заказы получателей, названия которых заканчиваются словом "рулетка"
Название получателя	Like "[А-Г]*"	Заказы получателей, названия которых начинаются с букв от А до Г включительно
Заказ	Right([Заказ], 2)="99"	Заказы с последними цифрами кода в поле "Заказ" 99

## СПИСОК ВОПРОСОВ К ЭКЗАМЕНУ

1. Понятие базы данных (БД). Виды моделей организации данных.
2. Понятие нормализации РБД. Правила нормализации (построения) РБД.
3. Системы управления базами данных (СУБД). Реляционные СУБД. Правила создания реляционных СУБД.
4. Общая характеристика СУБД Microsoft Access. Объекты Microsoft Access. Их назначение и особенности.
5. Окно Базы данных. Общие принципы и способы работы с объектами.
6. Создание БД в Microsoft Access. Таблица как объект БД. Способы создания таблиц в Microsoft Access. Переименование, удаление, копирование таблиц.
7. Режимы работы с объектами в СУБД Microsoft Access
8. Режим Конструктора таблиц. Создание и изменение структуры таблицы. Ограничения для названия полей.
9. Типы данных и их особенности.
10. Формирование значений поля при помощи таблицы подстановки из списка.
11. Ключевые поля в таблицах. Назначение и использование. Способы задания и изменения.
12. Понятие записи и поля. Свойства полей. Условие на значение. Примеры.
13. Свойства полей. Значение по умолчанию. Примеры.
14. Работа с данными в Режиме таблицы. Модификация макета таблицы. Сортировка данных.
15. Особенности проектирования многотабличных БД.
16. Схема данных. Определение связей между таблицами. Изменение существующих связей. Целостность данных.
17. Запрос как объект БД. Типы и возможности запросов. Способы создания запроса в Microsoft Access.
18. Режим Конструктора запроса. Вид бланка QBE. Способы задания полей. Выполнение и сохранение запроса. Просмотр результатов.
19. Режим Конструктора запроса. Свойства запроса.
20. Запрос-выборка. Назначение. Виды. Способы создания.
21. Задание условия отбора в бланке QBE. Комбинация нескольких условий отбора. Примеры.
22. Вычисляемое поле. Назначение. Правила и способы создания. Свойства вычисляемого поля.
23. Выражения. Построитель выражений. Полное имя поля в выражении.
24. Использование встроенных функций в Microsoft Access. Группы функций.
25. Функции даты/времени. Примеры.
26. Функция Format(). Функция Iif(). Примеры.
27. Порядок создание функции пользователя в VBA. Использование пользовательской функции в запросах.
28. Запрос с параметром. Назначение. Особенности. Правила создания. Примеры.
29. Создание итоговых запросов. Установки Групповых операций. Назначение группировки. Примеры.

30. Подведение частных итогов в запросах. Выбор записей, формирующих группы. Примеры.

31. Групповые функции. Особенности их использования в выражениях для вычисляемых полей.

32. Вычисление общих итогов в запросах. Примеры.

33. Особенности создания многотабличных запросов. Объединение таблиц в многотабличных запросах. Примеры.

34. Перекрестный запрос. Назначение. Правила и способы создания.

35. Запросы-действия. Виды и назначение. Особенности создания и выполнения.

36. Запрос на обновление данных. Запрос на удаление данных. Особенности создания и выполнения.

37. Запрос на создание таблицы. Особенности создания и выполнения. Просмотр результатов.

38. Форма как объект БД. Назначение. Режимы просмотра. Способы проектирования форм в Microsoft Access.

39. Виды форм (простая, ленточная, табличная). Формы, создаваемые мастером.

40. Режим Конструктора форм. Вид окна конструктора форм. Области формы.

41. Этапы создания формы в Режиме Конструктора формы. Связь формы с данными таблиц или запросов. Элементы вычислений в формах.

42. Бланк свойств формы. Форма для добавления и редактирования данных. Форма для просмотра данных.

43. Элементы управления, их виды и назначение.

44. Ввод элементов управления в форму и их настройка (выравнивание, размер, палитра). Свойства элементов управления.

45. Размещение текстовой информации в форме. Размещение полей ввода. Окно Списка полей.

46. Создание формы для ввода новых записей с помощью ЭУ «список» и «поле со списком».

47. Создание кнопок управления в форме. Пример.

48. Отчет как объект БД. Назначение. Отличия отчетов от форм. Просмотр и печать отчетов.

49. Способы проектирования отчетов в Microsoft Access. Окно Конструктора Отчетов. Области окна. Общее между отчетами и формами.

50. Создание отчетов с вычисляемыми полями, с частными и общими итогами.

51. Улучшение внешнего вида форм и отчетов (вставка линий, прямоугольников, рисунков, диаграмм и др. OLE-объектов).

## ЛИТЕРАТУРА

1. Карпова, Т.С. Базы данных: модели, разработка, реализация: учеб. для вузов. – СПб.: Питер, 2001.
2. Федоров, А.Г. Базы данных для всех. – М.: ООО "Компьютер-Пресс", 2001.
3. Коннолли, Т. Базы данных. Проектирование, реализация, сопровождение. Теория и практика / Т. Коннолли, К. Бегг, А. Страчан. – М.: Вильямс, 2001.
4. Оскерко, В.С. Технологии организации, хранения и обработки данных. Система дистанционного обучения: учеб.-практ. пособие. – Минск: БГЭУ, 2002.
5. Глушаков, С.В. Базы данных: учеб. курс. / С.В. Глушаков, Д.В. Ломотыко. – Харьков: Фолио; Ростов-н/Д: Феникс; Киев: Абрис, 2002.
6. Когаловский, М.Р. Энциклопедия технологий баз данных. – М.: Финансы и статистика, 2002.
7. Базы данных: учеб. для вузов / А.Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев; под ред. А.Д. Хомоненко. – СПб.: КОРОНА принт, 2004.
8. Малыгина, М.П. Базы данных: учеб. пособие. – СПб.: БХВ-Петербург, 2004.
9. Оскерко, В.С. Практикум по технологиям баз данных: учеб. пособие / В.С. Оскерко, З.В. Пунчик. – Мн.: БГЭУ, 2004.
10. Кузин, А.В. Разработка баз данных в системе Microsoft Access: учебник / А.В. Кузин, В.М. Демин. – М.: ФОРУМ: ИНФРА-М, 2005.
11. Левчук, Е.А. Технологии организации, хранения и обработки данных. – Минск: Высш. шк., 2005.
12. Кренке, Д. Теория и практика построения баз данных. – СПб.: Питер, 2005.
13. Тимошок, Т.В. Microsoft Office Access 2007: самоучитель / Т.В. Тимошок. – М.: Вильямс, 2008.
14. Смирнова, О.В. Access 2007 на практике. М.: Феникс, 2009.
15. Гурвиц, Г. Microsoft Access 2010. Разработка приложений на реальном примере. – СПб.: БХВ-СПб, 2010.
16. Сеннов, А. Access 2010. – СПб.: Питер, 2010.
17. Одиночкина, С.В. Разработка баз данных в Microsoft Access 2010. – СПб.: НИУ ИТМО, 2012.
18. Сурядный, А.С. Microsoft Access 2010. Лучший самоучитель. – М.: Астрель, ВКТ, 2012.
19. Кошелёв, В.Е. Базы данных Access 2007. – М.: Бином, 2013.

Учебное издание

*Составители:*  
**Ирина Михайловна Гучко**  
**Екатерина Николаевна Рубанова**

**Методические указания**  
по курсу «Компьютерные информационные технологии»  
2-й раздел: «Технологии баз данных и знаний»  
для студентов экономических специальностей

Ответственный за выпуск: Рубанова Е.Н.  
Редактор: Боровикова Е.А.  
Компьютерная верстка: Митлошук М.А.  
Корректор: Никитчик Е.В.

---

Подписано в печать 06.03.2017 г. Формат 60x84 1/16. Бумага «Performer».  
Гарнитура «Arial Narrow». Усл. печ. л. 4,42. Уч. изд. л. 4,75. Заказ № 1306. Тираж 50 экз.  
Отпечатано на ризографе учреждения образования «Брестский государственный  
технический университет». 224017, г. Брест, ул. Московская, 267.