

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

М.М. КРИКУНОВ, А.Н. ПОРУЧИКОВ

ОСНОВЫ БАЗ ДАННЫХ

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве учебного пособия для обучающихся по основной образовательной программе высшего образования по направлению подготовки 01.03.03 Механика и математическое моделирование

САМАРА

Издательство Самарского университета

2021

УДК 004.6(075)

ББК 32.973.2-018.2я7

К820

Рецензенты: д-р техн. наук, проф. Н. Н. В а с и н,
канд. техн. наук, доц. И. В П о т а п о в

Крикунов, Михаил Михайлович

К820 **Основы баз данных:** учебное пособие / *М.М. Крикунов, А.Н. Поручиков.* – Самара: Издательство Самарского университета, 2021. – 84 с.

ISBN 978-5-7883-1671-0

В пособии кратко и в доступной форме излагаются основы теории баз данных. Основное внимание уделяется вопросам создания и использования реляционных баз данных, являющихся на сегодняшний день наиболее распространёнными.

Предназначено для обучающихся по направлению подготовки 01.03.03 Механика и математическое моделирование при изучении дисциплины «Базы данных», а также может быть полезно по другим специальностям, при изучении дисциплин «Базы данных» и «Управление данными».

Подготовлено на кафедре теоретической механики Самарского университета.

УДК 004.6(075)

ББК 32.973.2-018.2я7

ISBN 978-5-7883-1671-0

© Самарский университет, 2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. ВВЕДЕНИЕ В СУБД.....	5
2. ИНФОРМАЦИОННО-ЛОГИЧЕСКАЯ МОДЕЛЬ БАЗЫ ДАННЫХ.....	9
3. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ	17
4. ЗАПРОСЫ НА ВЫБОРКУ	27
5. ЗАПРОСЫ НА ИЗМЕНЕНИЕ.....	45
6. ФОРМЫ.....	62
7. ОТЧЕТЫ.....	69
ЗАКЛЮЧЕНИЕ	79
СПИСОК ЛИТЕРАТУРЫ.....	80
ПРИЛОЖЕНИЕ А. ТИПЫ И РАЗМЕРЫ ПОЛЕЙ.....	81
ПРИЛОЖЕНИЕ Б. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ФУНКЦИЙ	83

ВВЕДЕНИЕ

Развитие средств вычислительной техники и информационных технологий обеспечило возможности для создания и широкого применения автоматизированных информационных систем (АИС) разнообразного назначения. Разрабатываются и внедряются информационные системы управления хозяйственными и техническими объектами, модельные комплексы для научных исследований, системы автоматизации проектирования и производства, всевозможные тренажеры и обучающие системы. Различают АИС, основанные на знаниях, и АИС, основанные на данных. К первым можно отнести, например, экспертные системы (ЭС), интеллектуальные системы поддержки принятия решений (СППР) и т.п. Ко вторым – всевозможные прикладные системы, которые сейчас активно используются и на предприятиях, и в учреждениях. Такие прикладные системы применяются очень широко, и в рамках данного курса наше внимание будет сосредоточено именно на системах, которые основаны на данных. База данных (БД) является моделью предметной области информационной системы. В пособии кратко и в доступной форме излагаются основы теории баз данных; основное внимание уделяется вопросам создания и использования реляционных баз данных, являющихся на сегодняшний день наиболее распространёнными. В качестве основной СУБД для реализации обучения выбрана MS Access 2013.

Пособие предназначено для студентов высших учебных заведений, обучающихся по направлению подготовки бакалавров «Механика и математическое моделирование» при изучении дисциплины «Базы данных»; пособие может быть полезно для студентов, обучающихся по другим специальностям, при изучении дисциплин «Базы данных» и «Управление данными».

Учебное пособие является частью учебно-методического комплекса по дисциплине «Базы данных», включающего в себя методические указания к лабораторным работам, а также тестовые материалы.

1. ВВЕДЕНИЕ В СУБД

БАЗА ДАННЫХ - ОСНОВА ИНФОРМАЦИОННОЙ СИСТЕМЫ

Информационная система – совокупность определенным образом организованных данных (база данных – БД) и комплекса аппаратно-программных средств хранения и манипулирования данными (система управления – СУ).

Назначение базы данных

База данных является хранилищем данных в виде файла данных на внешних носителях.

Назначение системы управления

Системы управления базами данных (СУБД, DBMS Database Management System) обеспечивают:

1. Создание БД.
2. Хранение БД.
3. Защиту данных.
4. Обновление БД:
 - добавление данных;
 - редактирование данных;
 - удаление данных.
5. Манипулирование данными:
 - упорядочение данных;
 - отбор данных;
 - выдача отчетов.
6. Создание приложений.

Структура СУБД

Структура СУБД показана ниже (Рисунок 1). Минимальной единицей информации в базе данных является байт, система управления может быть представлена в виде оболочки (компилятор или интерпретатор). Исполняемый файл может быть реализован с помощью специализированных языков программирования.

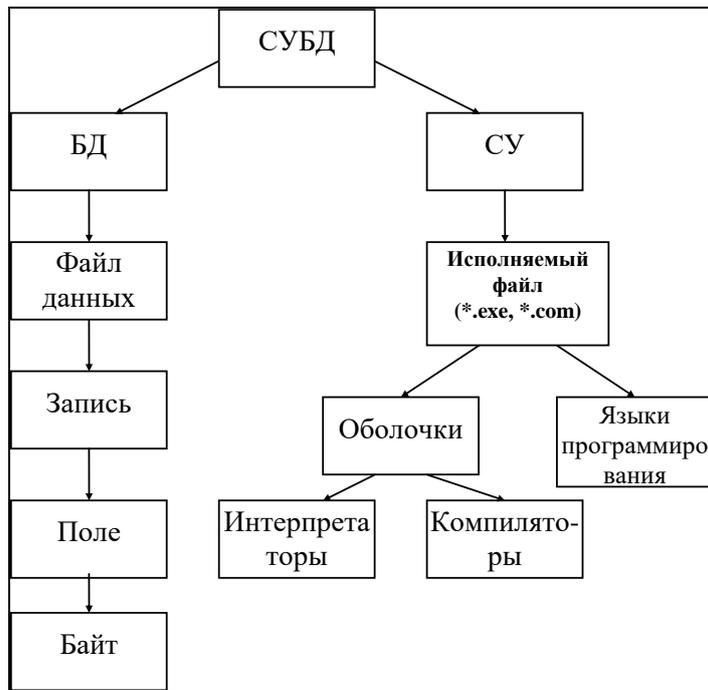


Рисунок 1 – Структура СУБД

Типы БД в зависимости от модели данных

Используемый тип базы данных зависит от модели данных, которые мы хотим в ней разместить (Рисунок 2).

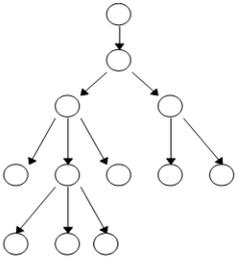
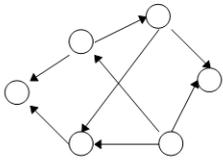
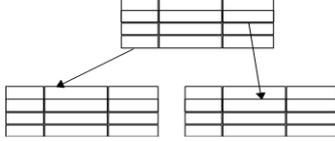
Модель данных		
<p>дерево родительская и дочерняя запись отношение один-ко-многим</p> 	<p>сеть каждая запись может иметь нескольких родителей отношение многие-ко-многим</p> 	<p>набор связанных таблиц отношения один-ко-многим, один-к-одному</p> 
Тип базы данных		
Иерархические	Сетевые	Реляционные

Рисунок 2 – Типы БД в зависимости от модели данных

РЕЛЯЦИОННАЯ БАЗА ДАННЫХ

Реляционная база данных (РБД) – это набор связанных таблиц. К базам данных реляционного типа относится СУБД Access.

Основные термины и понятия РБД

В реляционных базах данных используется следующая терминология:

1. Таблица – хранилище данных.
2. Записи – строки таблицы.
3. Поля – столбцы таблицы.
4. Имена полей – имена столбцов таблицы.
5. Ключ – уникальное поле или совокупность полей таблицы, позволяющее однозначно идентифицировать запись. Каждая таблица РБД должна иметь ключ, который называется первичным ключом.

Отличия таблицы реляционной базы данных от электронной таблицы

Таблицы реляционной базы данных похожи на электронные таблицы, однако есть и существенные отличия:

- база данных состоит из нескольких связанных таблиц;
- все имена полей в таблице уникальные;
- для каждого поля заранее определен тип и размер данных, а также другие свойства;
- нет полей с вычисляемыми значениями;
- все записи одинаковы по структуре.

Объекты базы данных

База данных состоит из объектов, взаимодействующих между собой:

1. Таблицы – каждая таблица описана своей структурой и связями с другими таблицами.
2. Запросы – позволяют выполнять любую обработку данных.

3. Формы – позволяют организовать просмотр, ввод, редактирование информации в нужном виде, а также обеспечить работу в системе за счет элементов управления.
4. Отчеты – позволяют выдать на печать информацию из БД в нужном виде.
5. Макросы – позволяют автоматизировать работу с базой данных.

Взаимодействие объектов базы данных

Исходные данные хранятся в таблицах, затем они обрабатываются в запросах, которые создают новые виртуальные таблицы. Результаты обработки исходных данных отображаются на экране в формах или на экране и бумаге в отчетах (Рисунок 3).

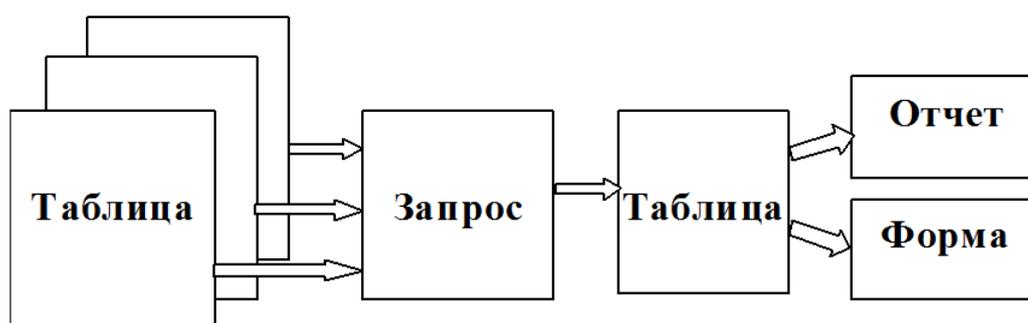


Рисунок 3 – Взаимодействие объектов базы данных

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите правильную последовательность основных составляющих базы данных.
2. Укажите основные типы баз данных.
3. Перечислите основные объекты базы данных Access.
4. Какая модель данных соответствует СУБД Access?
5. Назовите правильный порядок создания базы данных.

2. ИНФОРМАЦИОННО-ЛОГИЧЕСКАЯ МОДЕЛЬ БАЗЫ ДАННЫХ

ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ

1. Построение информационно-логической модели (ИЛМ).
2. Разработка БД средствами СУБД:
 - создание базы данных;
 - разработка приложения.

ПОСТРОЕНИЕ ИНФОРМАЦИОННО-ЛОГИЧЕСКОЙ МОДЕЛИ

Определение предметной области

Предметная область – это множество взаимосвязанных объектов, представляющих интерес с точки зрения решаемой задачи.

Предметная область отражается в названии БД.

Например, предметной областью может являться покупка товаров.

Объекты предметной области

Предметная область состоит из взаимосвязанных объектов.

Объектами нашей предметной области являются товары и покупатели. Отдельные элементы объекта называют *экземплярами*, например, экземплярами объекта **Товары** являются отдельные товары.

Объектам предметной области в базе данных соответствуют таблицы. В нашем примере в базе данных **Покупка товаров** будут таблицы **Товары** и **Покупатели**.

Определение атрибутов

Каждый объект описывается набором атрибутов. *Атрибут объекта* – значимая с точки зрения решаемой задачи характеристика (свойство, параметр) объекта. В таблице базы данных атрибуты размещаются в полях, имена полей соответствуют названию атрибута.

Атрибутами наших объектов являются:

1. **ТОВАРЫ**: Наименование, Категория, Описание, Производитель, Дата поставки, Количество, Цена поставки, Единица измерения, Скидка %.
2. **ПОКУПАТЕЛИ**: Покупатель, Адрес, Телефон, Физическое лицо.

Выделение задач, решаемых в предметной области

В нашей предметной области *Покупка товаров* необходимо решать следующие задачи:

- учет товара;
- учет покупателей;
- учет покупок.

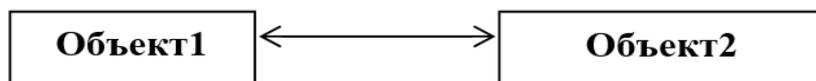
Информационно-логическая модель

ИЛМ изображается в виде схемы, на которой прямоугольники – это объекты, а линии, связывающие объекты – это отношения между объектами. Одинарная стрелка на линии связи обозначает один экземпляр, двойная стрелка – несколько экземпляров. В нашем случае имеется два объекта:



Связи (отношения) между объектами ИЛМ могут быть разных типов:

- *один к одному* – одному экземпляру первого объекта соответствует один экземпляр второго объекта.



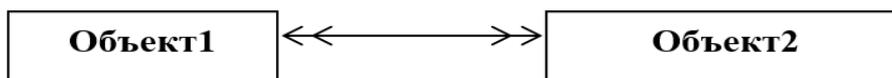
- *один ко многим* – одному экземпляру первого объекта соответствует несколько экземпляров второго объекта.



- *многие к одному* – нескольким экземплярам первого объекта соответствует один экземпляр второго объекта.

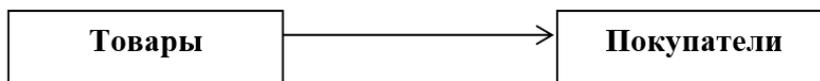


- *многие ко многим* – нескольким экземплярам первого объекта соответствует несколько экземпляров второго объекта.

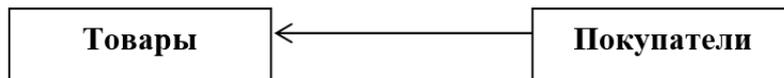


Рассмотрим отношения между объектами нашей предметной области:

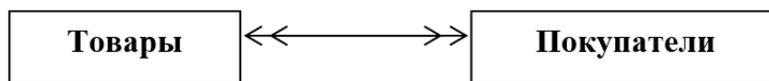
- отношении Товары-Покупатели. Один и тот же товар могут купить несколько покупателей – связь *один ко многим*:



- отношение Покупатели-Товары. Один и тот же покупатель может купить несколько товаров – связь *многие к одному*:



- совместим две схемы и получим ИЛМ предметной области. Получится связь *многие ко многим*:



Такая схема отношений означает, что каждый покупатель в объекте **Покупатели** встречается столько раз, сколько товаров он купил. Точно также, каждый товар в объекте **Товары** встречается столько раз, сколько покупателей купили этот товар. Таким образом, один и тот же товар описан в объекте **Товары** много раз, как и каждый покупатель в объекте **Покупатели**. Реализация таких объектов в реляционной базе данных недопустима, так как приводит к повторяемости одних и тех же значений атрибутов в каждом экземпляре объектов **Товары** и **Покупатели**.

Преобразование ИЛМ в соответствии с требованиями реляционной СУБД

В реляционной базе данных недопустимы отношения *многие ко многим*, поэтому ликвидируем такие отношения за счет введения дополнительного объекта **Покупки**:



Обратим внимание на то, что объекты **Товары** и **Покупатели** – это реально существующие объекты, а объект **Покупки** – это описание взаимодействия между покупателями и товарами, т.е. описание процесса взаимодействия покупателя с товаром.

Определим атрибуты объекта **Покупки**:

- наименование товара;
- покупатель;
- количество;
- цена покупки;
- дата покупки.

Пример отношений в предметной области **Сессия**, где студенты и преподаватели – это реально существующие объекты, а сессия – это процесс взаимодействия студентов и преподавателей:



Результаты построения ИЛМ

Результатом построения ИЛМ является:

1. Определение предметной области.
2. Определение задач.
3. Определение объектов.
4. Определение атрибутов объектов.
5. Построение отношений между объектами.

Терминология ИЛМ

При описании информационно-логической модели используется следующая терминология:

- предметная область;
- объект;
- атрибуты;
- задачи;
- отношения.

ПЕРЕНОС ИЛМ В РЕЛЯЦИОННУЮ БАЗУ ДАННЫХ

Необходимо реализовать ИЛМ средствами выбранной СУБД.

В базе данных требуется поставить в соответствие объектам информационно-логической модели таблицы реляционной базы данных, построить отношения (связи) между таблицами и описать структуру каждой таблицы.

Соответствие терминов ИЛМ и РБД:

ИЛМ	РБД
Предметная область	Название базы данных
Имя объекта	Имя таблицы
Объект	Таблица
Экземпляр объекта	Запись
Атрибут	Поле
Имя атрибута	Имя поля
Отношения	Связи
	Ключ

Пример: база данных **Покупка товаров**.

Объекты и атрибуты ИЛМ:

- **ТОВАРЫ:** Наименование, Категория, Описание, Производитель, Дата поставки, Количество, Цена поставки, Единица измерения, Скидка %, Фото товара;
- **ПОКУПАТЕЛИ:** Покупатель, Адрес, Телефон, Физическое лицо;
- **ПОКУПКИ:** Наименование, Покупатель, Количество, Цена покупки, Дата покупки.

Одна и та же ИЛМ может быть представлена в БД по-разному.

Хранение данных в одной таблице

Создадим одну таблицу для хранения атрибутов всех объектов:

Покупка товаров: Наименование товара, Категория, Производитель, Модель, Характеристика, Цена, Покупатель, Адрес, Реквизиты, Количество, Цена продажи, Дата продажи.

При этом исключаются дублирующие атрибуты разных объектов.

Однако хранение всех объектов в одной таблице приводит к избыточности данных, а неизбежные ошибки при вводе не позволяют правильно извлекать данные.

Хранение данных в нескольких таблицах

Выполним декомпозицию данных, т.е. распределим данные по таблицам:

Таблица Товары	Таблица Покупки	Таблица Покупатели
Наименование	Наименование товара	Покупатель
Категория	Покупатель	Адрес
Описание	Количество	Телефон
Производитель	Цена покупки	Физическое лицо
Дата поставки	Дата покупки	
Количество		
Цена поставки		
Единица измерения		
Скидка %		
Фото товара		

Структура таблицы

Структура каждой таблицы в базе данных представляет собой описание полей таблицы через их свойства. Основными свойствами поля являются: *Имя поля*, *Тип данных* в поле, *Размер поля*, *Формат*. Множество других свойств поля зависит от типа данных в этом поле. На первом этапе определимся с именами полей и типами данных.

Таблица Товары

Имя поля	Тип данных
Наименование товара	Текст
Категория	Текст
Описание товара	Текст
Производитель	Текст
Дата поставки	Дата/время
Количество	Целое число
Цена поставки	Денежный
Единица измерения	Текст
Скидка %	Вещественное число
Фото товара	Изображение

Таблица Покупатели

Имя поля	Тип данных
Покупатель	Текст
Адрес	Текст
Телефон	Текст
Физическое лицо	Логический

Таблица Покупки

Имя поля	Тип данных
Наименование товара	Текст
Покупатель	Текст
Количество	Целое число
Цена покупки	Денежный
Дата покупки	Дата/Время

Результаты представления ИЛМ в базе данных

- каждому объекту поставлена в соответствие таблица;
- описана структура каждой таблицы (имена полей, типы и размеры данных);
- в каждой таблице определен первичный ключ;
- добавлены (при необходимости) дополнительные таблицы-справочники;
- таблицы связаны между собой, создана схема данных.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите типы связей (отношений) между объектами ИЛМ.
2. Какие отношения недопустимы в реляционной базе данных?
3. Что является результатом построения ИЛМ?
4. Какие свойства входят в описание полей таблицы?

3. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ

Созданию базы данных всегда предшествует этап проектирования. Проектирование базы данных состоит из нескольких этапов:

1. Определение предметной области.
2. Декомпозиция данных.
3. Описание структур таблиц базы данных.
4. Определение ключей в таблицах.
5. Связывание таблиц.

Определение предметной области

Предметная область – это множество взаимосвязанных объектов, представляющих интерес с точки зрения решаемой задачи. Предметная область отражается в названии БД. Например, предметной областью может являться покупка товаров.

Объекты предметной области

Предметная область состоит из взаимосвязанных объектов. Объектами нашей предметной области являются товары и покупатели. Отдельные элементы объекта называют *экземплярами*, например, экземплярами объекта **Товары** являются отдельные товары.

Объектам предметной области в базе данных соответствуют таблицы. В нашем примере в базе данных **Покупка товаров** будут таблицы **Товары** и **Покупатели**

Определение атрибутов

Каждый объект описывается набором атрибутов. *Атрибут объекта* – значимая с точки зрения решаемой задачи характеристика (свойство, параметр) объекта. В таблице базы данных атрибуты размещаются в полях, имена полей соответствуют названию атрибута.

Атрибутами наших объектов являются:

1. Таблица **ТОВАРЫ**: Наименование товара, Дата поступления, Количество, Цена, Производитель, Описание, Характеристика.
2. Таблица **ПОКУПАТЕЛИ**: Покупатель, Адрес, Реквизиты.

Выделение задач, решаемых в предметной области

В нашей предметной области **Покупка товаров** необходимо решать следующие задачи:

- учет товара;
- учет покупателей;
- учет покупок.

ДЕКОМПОЗИЦИЯ ДАННЫХ

Размещение данных в одной таблице

Создадим одну таблицу **Покупка товаров** для хранения атрибутов всех объектов с полями *Наименование, Категория, Описание, Производитель, Дата поставки, Количество, Цена поставки, Цена покупки, Покупатель, Адрес, Телефон, Физическое лицо, Количество, Цена покупки, Дата покупки* (Таблица 1).

Таблица 1 – Размещение данных в одной таблице*

Наименование	Категория	Описание	Производитель	Количество	Цена поставки	Цена покупки	Покупатель	Адрес	Физическое лицо
дверь	стройматериалы	яшень 2,05*0,84	ОАО "Строим сами"	2	1 820,00р.	2 002,00р.	Пилгогина М.С.	пр. Калинина 22-2	Да
утеплитель	хозтовары	для окон и дверей	ОАО "Строим сами"	4	150,00р.	172,50р.	Андреев А.Л.	Комсомоль- ская 25-1	Да
сланцы муж.	обувь	Тайвань, разм. 42-43	АО "По- дошва"	2	1 200,00р.	1 416,00р.	ООО "Альберт"	пер. Длинный 45-12	Нет
утеплитель	хозтовары	для окон и дверей	ОАО "Строим сами"	6	150,00р.	172,50р.	ЧП Поно- марев	Тракторная 113-24	Нет
плитка керам.	стройматериалы	Италия, 30x30	ОАО "Строим сами"	10	800,00р.	880,00р.	ЧП Поно- марев	Тракторная 113-24	Нет
сахар рафинад	продукты	белгородс- кий	ООО "Анта- рес"	2	60,50р.	78,65р.	ООО "Альберт"	пер. Длинный 45-12	Нет
сервиз чайный	посуда	45 предметов	ЧП "Фар- фор"	1	3 800,00р.	4 484,00р.	ЧП Поно- марев	Тракторная 113-24	Нет
ветровка	одежда	Тайвань	ИП Рубаш- кин	1	980,00р.	1 078,00р.	Андреев А.Л.	Комсомоль- ская 25-1	Да
плитка керам.	стройматериалы	Италия, 30x30	ОАО "Строим сами"	8	800,00р.	880,00р.	ЧП Поно- марев	Трактор- ная113-24	Нет

* Для экономии места в таблице показаны не все поля и часть записей.

Хранение всех объектов в одной таблице приводит к избыточности данных (повторение одинаковых данных в полях Адрес, Физическое лицо, Телефон и др.), а неизбежные ошибки при вводе повторяющихся данных не позволят правильно извлекать данные.

Размещение данных в нескольких таблицах

Декомпозиция данных предполагает распределение набора данных предметной области по таблицам. Так как предметная область состоит из объектов и процессов их взаимодействия, то при декомпозиции появляются таблицы, описывающие каждый объект и каждый процесс.

Например, предметная область **Продажи товара** содержит две таблицы, описывающие объекты этой предметной области – **Товары**, **Покупатель**, и одну таблицу, которая описывает процесс взаимодействия этих объектов – **Покупки** (Рисунок 4).

Товары	Покупки	Покупатель
Наименование	Наименование	Покупатель
Категория	Покупатель	Адрес
Описание	Количество	Телефон
Производитель	Цена покупки	Физическое лицо
Дата поставки	Дата покупки	
Количество		
Цена поставки		
Единица измерения		
Скидка %		
Фото товара		

Рисунок 4 – Декомпозиция предметной области **Продажи товаров**

Декомпозиция позволяет исключить дублирование данных и значительно уменьшает ошибки при вводе данных.

ОПИСАНИЕ СТРУКТУРЫ КАЖДОЙ ТАБЛИЦЫ

Структура каждой таблицы представляет собой описание полей таблицы через их свойства (Таблица 2 – Таблица 4). Основными свойствами поля являются: *Имя поля*, *Тип данных* в поле, *Размер поля*, *Формат*. Множество других свойств поля зависит от типа данных в этом поле. Типы данных и размеры полей приведены в приложении А.

Таблица 2 – Свойства полей таблицы «Товары»

Имя	Тип	Размер
Код товара	Длинное целое	4
Наименование	Короткий текст	50
Категория	Короткий текст	25
Описание	Короткий текст	150
Производитель	Короткий текст	100
Дата поставки	Дата и время	8
Количество	Целое	2
Цена поставки	Денежный	8
Единица измерения	Короткий текст	5
Скидка %	Одинарное с плавающей	4
Фото товара	Поле объекта OLE	-

Таблица 3 – Свойства полей таблицы Покупки

Имя	Тип	Размер
Код покупки	Длинное целое	4
Код товара	Длинное целое	4
Покупатель	Короткий текст	100
Количество	Целое	2
Цена покупки	Денежный	8
Дата покупки	Дата и время	8

Таблица 4 – Свойства полей таблицы Покупатели

Имя	Тип	Размер
Покупатель	Короткий текст	100
Адрес	Короткий текст	100
Телефон	Короткий текст	12
Физическое лицо	Логический	1

Особенности использования различных типов данных

Рекомендации по использованию различных типов данных представлены ниже (Таблица 5).

Таблица 5 – Использование различных типов данных

Данные	Примеры данных	Тип данных	Размер, формат
Любой набор символов – букв, цифр, знаков; данные, состоящие из цифр, но не являющиеся числами	Адрес, паспортные данные, номер телефона, номер группы, номер зачетки, год, месяц, день недели	Короткий текст	До 255 символов
Любые числовые данные (кроме денежных): <ul style="list-style-type: none"> • целые числа; • дробные числа; • проценты 	Количество, рост, вес, оценка (числовая), год, месяц. Проценты: <ul style="list-style-type: none"> • скидки, • наценки, • налоги, • надбавки 	Числовой	Для целых чисел байт, целое, длинное целое. Для вещественных чисел одинарное с плавающей точкой, дополнительно задать число десятичных знаков. Для процентов одинарное с плавающей точкой, дополнительно задать процентный формат и число десятичных знаков
Денежные данные	Любые денежные данные: цены, стоимости, надбавки, скидки	Денежный	Задать формат поля денежный и число десятичных знаков
Дата и/или время	Точная дата в формате дд.мм.гггг. Точное время в формате чч:мм:сс Дата вместе со временем дд.мм.гггг чч:мм:сс	Дата и время	Выбрать формат отображения даты, времени или одновременно даты и времени
Двухзначные данные типа Да/Нет, Есть/Нет	Скидка есть/нет, наценка есть/нет	Логический	
Фотографии, текстовые файлы, электронные таблицы и любые другие объекты	Фотографии объектов: люди, товары, недвижимость (*.jpg). Файлы Word. Электронные таблицы Excel	Поле объекта OLE	

ОПРЕДЕЛЕНИЕ ПЕРВИЧНЫХ КЛЮЧЕЙ В КАЖДОЙ ТАБЛИЦЕ

Первичный ключ – поле или группа полей, однозначно идентифицирующий запись.

Если поле (или группа полей) является ключом, то ввести повторяющиеся значения в это поле (или группу полей) невозможно. Поэтому, любому значению в ключе соответствует только одна запись.

Простой ключ – первичный ключ, состоящий из одного поля.

Составной ключ – первичный ключ, состоящий из нескольких полей.

Теория реляционных баз данных требует наличия первичного ключа в каждой таблице, однако, на практике, в некоторых таблицах можно не создавать ключи. Если в таблице нет поля, которое можно сделать ключом, то можно создать ключ с помощью поля типа СЧЕТЧИК, в котором каждой записи будет присвоен уникальный номер. В таблице может быть только один первичный ключ!

В таблице **Товары** поле **Наименование** не может быть ключевым, т.к. товары с одинаковым наименованием могут быть от разных производителей, с разной ценой, в разной упаковке. Другие поля этой таблицы также не могут быть ключами, т.к. значения в любом из них могут повторяться. Поэтому добавляем искусственное ключевое поле **Код товара**, тип данных – счетчик. Теперь товар однозначно определяется не наименованием, а кодом товара, поэтому в таблице **Покупки** поле **Наименование** заменяем на **Код товара** (числовой, длинное целое).

В таблице **Покупки** добавляем ключевое поле **Номер покупки**, тип данных – счетчик.

В таблице **Покупатели** ключевым может быть поле **Покупатель**.

Ключевые поля в таблицах выделим жирным шрифтом (Рисунок 5).

Таблица Товары	Таблица Покупки	Таблица Покупатели
Код товара	Номер покупки	Покупатель
Наименование	Код товара	Адрес
Категория	Покупатель	Телефон
Описание	Количество	Физическое лицо
Производитель	Цена покупки	
Дата поставки	Дата покупки	
Количество		
Цена поставки		
Единица измерения		
Скидка %		
Фото товара		

Рисунок 5 – Определение ключевых полей

ДОБАВЛЕНИЕ ТАБЛИЦ-СПРАВОЧНИКОВ

Добавим дополнительную таблицу-справочник **Производители** и определим ее структуру, первичным ключом здесь является поле **Производитель** (Таблица 6).

Таблица 6 – Свойства полей таблицы Производители

Имя	Тип	Размер
Производитель	Короткий текст	100
Адрес	Короткий текст	100
Телефон	Короткий текст	12
Руководитель	Короткий текст	100

СВЯЗЫВАНИЕ ТАБЛИЦ

Так как база данных реляционного типа по определению является набором связанных таблиц, то займемся связыванием таблиц в нашей базе данных.

Связывание таблиц происходит через поля, одинаковые по смыслу, имени, типу и размеру. Между двумя таблицами можно создать только одну связь (Рисунок 6).



Рисунок 6 – Связи между таблицами

Средства для создания

В реляционной базе данных допустимы типы связей *один ко многим* и *один к одному*.

Связь один ко многим

Отношения *один-ко-многим* в ИЛМ преобразуются в реляционной базе данных в связи *один-ко-многим*.

На стороне *один* в связи поле является *первичным ключом*, на стороне *много* поле называют *внешним ключом*.

В нашей базе данных таблицы **Товары** и **Покупки** связаны через поле **Код товара**, таблицы **Покупки** и **Покупатели** связаны через поле **Покупатель**, таблицы **Товары** и **Производители** связаны через поле **Производитель** (Рисунок 7).

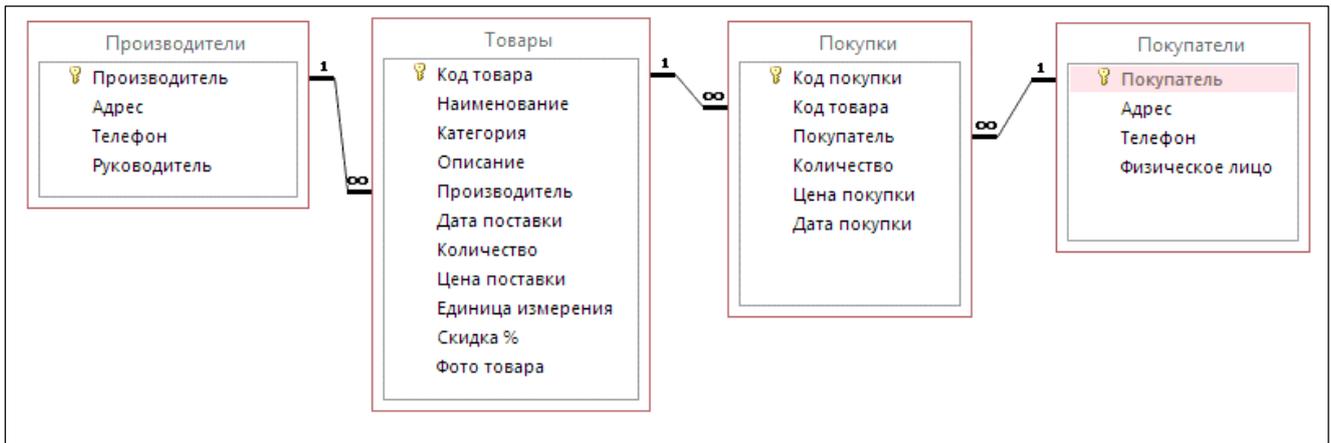


Рисунок 7 – Схема данных со связями один ко многим

Связь один к одному

Эта связь возникает, когда на обеих сторонах связи поля являются первичными ключами, т.е. одной записи первой таблицы соответствует только одна запись второй таблицы (Рисунок 8).

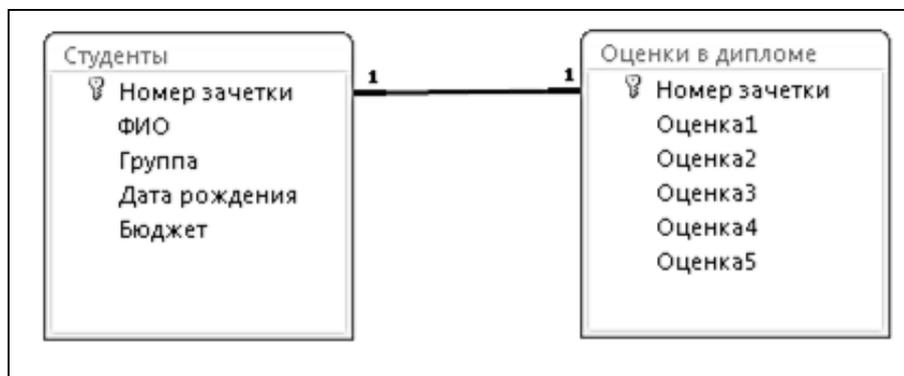


Рисунок 8 – Схема данных со связью один к одному

Свойства связей

На каждой из связей можно использовать следующие свойства:

- обеспечение целостности данных;
- каскадное обновление связанных полей;
- каскадное удаление связанных записей.

Включенное свойство *обеспечение целостности данных* не позволит в поле таблицы на стороне *много* этой связи вводить значения, которых нет в поле таблицы на стороне *один*. Таким образом, в базе данных поддерживается *ссылочная целостность* – на стороне много нельзя вводить значения, которых нет на стороне один. Это очень полезное свойство, поэтому его обычно включают.

Включенное свойство *каскадное обновление связанных полей* позволяет при изменении данных в поле на стороне *один* автоматически изменять его в поле таблицы на стороне *много*. Так как изменение данных в одной записи на стороне *один* приводит к изменению данных во всех связанных записях на стороне *много*, то неконтролируемое использование этого свойства может привести к потере данных.

Включенное свойство *каскадное удаление связанных записей* позволяет при удалении записи на стороне *один* автоматически удалять связанные записи в другой таблице с таким же значением на стороне *много*. Неконтролируемое использование этого свойства может привести к потере данных в связанных таблицах (Рисунок 9).

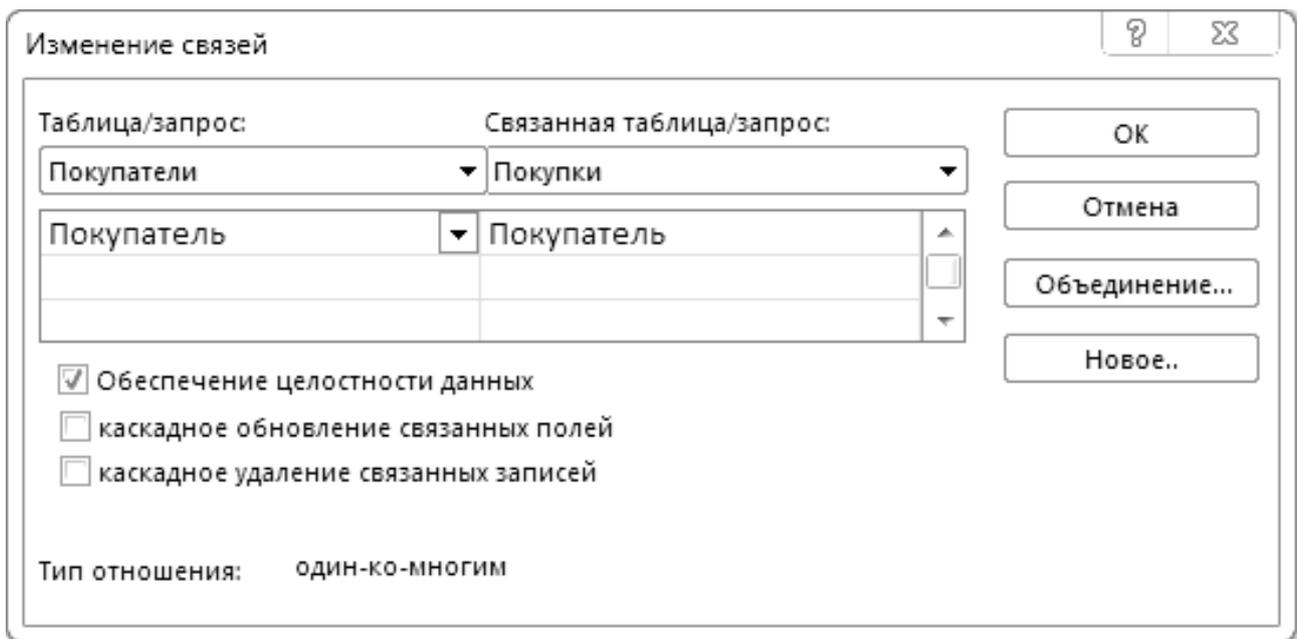


Рисунок 9 – Свойства связей

Порядок заполнения таблиц

Таблицы необходимо заполнять, начиная со стороны *один* связей (Рисунок 10).

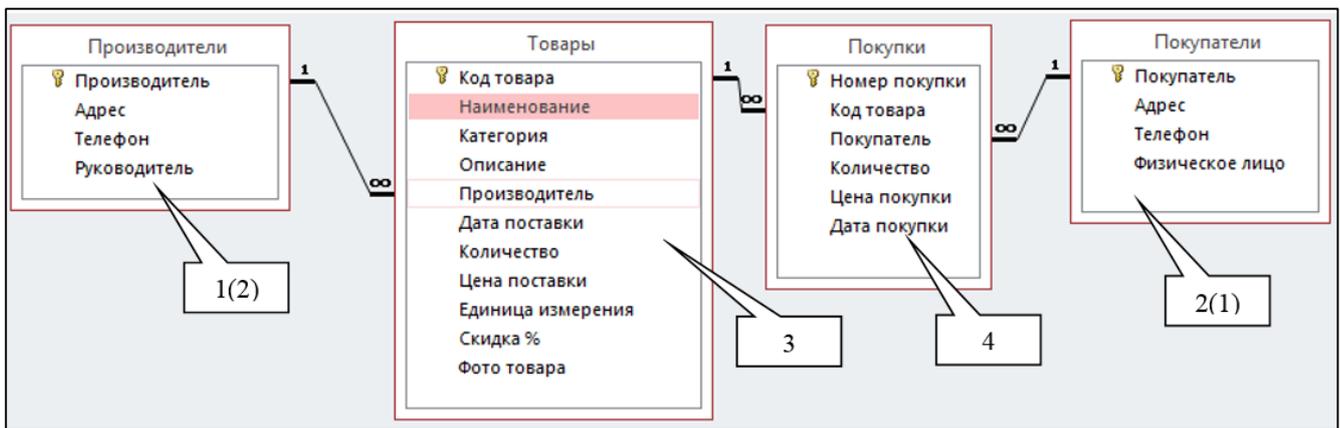


Рисунок 10 – Порядок заполнения таблиц

Таким образом, возможны два варианта порядка заполнения таблиц:

- Производители – Покупатели – Товары – Покупки;
- Покупатели – Производители –Товары – Покупки.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите этапы проектирования базы данных.
2. Что такое предметная область?
3. Перечислите преимущества декомпозиции данных.
4. Перечислите различия между первичным и внешним ключами.
5. Назовите типы связей в реляционной базе данных.

4. ЗАПРОСЫ НА ВЫБОРКУ

ЧТО ТАКОЕ ЗАПРОС. ВИДЫ ЗАПРОСОВ

С помощью запроса можно выполнить любую обработку данных из таблиц или запросов.

Запросы делятся на два вида: запросы на выборку и запросы на изменение:

- запросы на выборку позволяют отбирать поля одного или нескольких источников, сортировать, отбирать записи, создавать новые поля, группировать данные для подведения итогов и создания перекрестных таблиц. Эти запросы не изменяют исходные данные;
- запросы на изменение позволяют создавать новые таблицы как объекты, удалять, добавлять записи, обновлять данные в полях. Эти запросы изменяют исходные данные.

ИСТОЧНИКИ И РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЗАПРОСА

Источниками запроса могут быть таблицы и/или запросы.

Результатом выполнения запроса является виртуальная таблица с обработанными записями. Эта таблица не сохраняется как объект в базе данных. Ее можно использовать для построения отчетов и форм. Результаты выполнения запроса представляются пользователю в виде форм или отчетов (Рисунок 11).

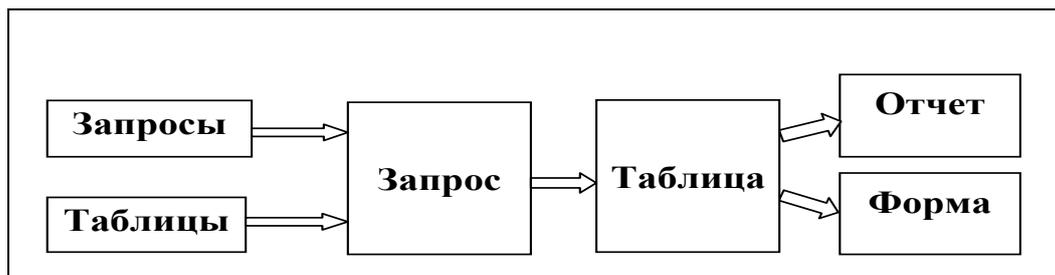


Рисунок 11 – Источники запроса

ОСОБЕННОСТИ ЗАПРОСОВ

Запросы имеют следующие особенности:

- в качестве источника запроса могут быть одна или несколько таблиц и запросов;
- запрос может содержать любое количество полей;

- поля запроса могут участвовать в запросе, но не выводиться на экран;
- в запросе могут быть созданы новые вычисляемые поля;
- запрос динамически связан с источником – при изменении и добавлении данных в запросе меняются или добавляются соответствующие данные в таблице, при изменении и добавлении данных в таблице меняются или добавляются соответствующие данные в запросе;
- запросы могут изменять данные.

ТИПЫ ЗАПРОСОВ И СРЕДСТВА ДЛЯ ИХ СОЗДАНИЯ

Типы запросов

В Access используются следующие типы запросов.

1. Запросы на выборку. Эти запросы не изменяют данные, однако с их помощью можно создавать новые поля:
 - отбор полей одной или нескольких таблиц;
 - сортировка записей;
 - отбор записей по критериям (запрос на выборку);
 - запрос с параметром;
 - запрос с вычисляемыми полями;
 - запрос с итогами;
 - перекрестный запрос.
2. Запросы на изменение. Эти запросы изменяют данные (добавляют, удаляют, обновляют, создают новые таблицы). Такие запросы требуют особого внимания при выполнении:
 - запрос на добавление;
 - запрос на удаление;
 - запрос на обновление;
 - запрос на создание таблицы.

Средства для создания запросов

Access предлагает следующие средства для создания запросов:

1. Мастер запросов. Позволяет создавать лишь некоторые типы запросов в режиме диалога. Редактирование запроса с помощью мастера невозможно.
2. Конструктор запросов. Позволяет создавать и редактировать любые типы запросов. В конструкторе Access запросы формируются путем заполнения бланка запроса. Эта технология называется «создание запроса по образцу» – QBE (Query By Example).
3. Язык программирования запросов SQL (Structured Query Language – структурированный язык запросов). Универсальное средство создания и редактирования любых запросов. Является стандартом любой системы поддержки реляционных СУБД.

Использование средств создания запросов

Запросы могут быть созданы разными средствами (Таблица 7).

Таблица 7 – Средства создания запросов

Тип запроса	Средство создания
Запросы на выборку	
Отбор полей одной или нескольких таблиц	Мастер запросов "Простой запрос", конструктор запросов
Сортировка записей	Конструктор запросов
Отбор записей по критериям (запрос на выборку)	
Запрос с параметром	
Запрос с вычисляемыми полями	
Запрос с итогами	Мастер запросов "Простой запрос", конструктор запросов
Перекрестный запрос	Мастер запросов "Перекрестный запрос", конструктор запросов
Запросы на изменение	
Запрос на удаление	Конструктор запросов
Запрос на создание таблицы	
Запрос на обновление	
Запрос на добавление	

ОПРЕДЕЛЕНИЕ НЕОБХОДИМЫХ ЗАПРОСОВ

Количество и тип запросов определяется количеством необходимых форм и отчетов. Практически все формы и отчеты строятся на запросах. Исключение составляют формы для ввода и редактирования данных, которые строятся непосредственно на таблицах. Анализируя необходимые формы для просмотра и отчеты, мы определяем, запросы какого типа нам нужны. Необходимо определить источники для каждой формы или отчета и описать запросы в таблице.

СОЗДАНИЕ ЗАПРОСА С ПОМОЩЬЮ МАСТЕРА

Для вызова мастера запросов выполните команду:

Вкладка **Создание** → Группа **Запросы** → Команда **Мастер запросов**

Далее предлагается выбрать мастер запросов:

- простой запрос;
- перекрестный запрос;
- повторяющиеся записи;
- записи без подчиненных.

Мы будем использовать только два первых мастера.

Затем по шагам определяют различные параметры создаваемого запроса.

СОЗДАНИЕ ЗАПРОСА С ПОМОЩЬЮ КОНСТРУКТОРА

Конструктор запросов можно вызвать несколькими способами:

- для создания нового запроса выполнить команду Вкладка **Создание** → Команда **Конструктор запросов**;
- для редактирования существующего запроса щелкните по нему в области навигации правой кнопкой мыши и выберите **Конструктор**.

Бланк запроса

При создании нового запроса в конструкторе сначала появляется окно выбора источников запроса – это могут быть одна или несколько таблиц (запросов).

Затем выводится окно бланка запроса на выборку. В конструкторе Access запросы формируются путем заполнения бланка запроса. Эта технология называется «создание запроса по образцу» – QBE (Query By Example).

В верхней части бланка отображаются источники запроса, в нижней части – структура таблицы результатов выполнения запроса. Выбрав источники запроса, формируем таблицу результата выполнения запроса, добавляя туда в нужном порядке поля источников. По каждому полю можно выполнить сортировку, отбор данных и вывод поля на экран. Набор строк бланка меняется в зависимости от типа запроса (Рисунок 12).

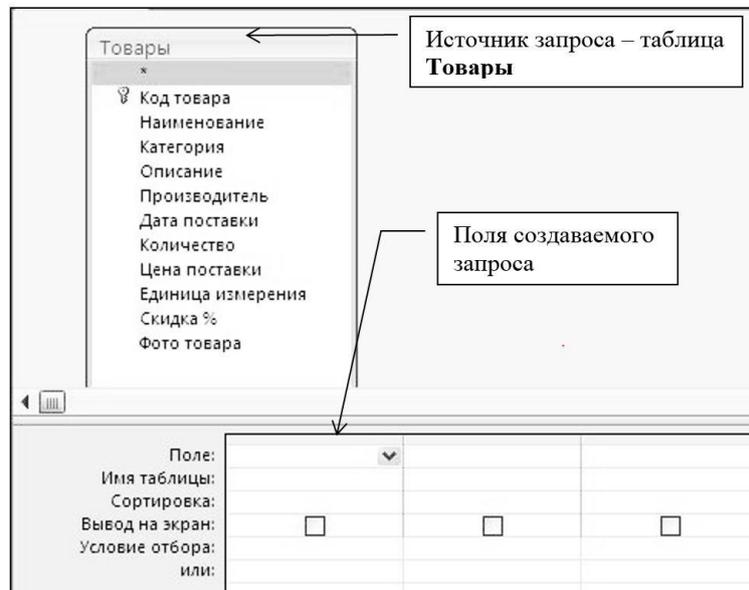


Рисунок 12 – Окно конструктора запросов с выбранным источником

Назначение строк бланка запроса

В строке **Поле** можно выбрать любое поле источника или создать новое поле с именем, отличающимся от имен полей источников.

Строка **Имя таблицы** заполняется автоматически именем таблицы-источника, из которой взято поле. Если создается новое поле, то имени источника для него не существует и строка **Имя таблицы** остается для него пустой.

В строке **Сортировка** можно из списка выбрать способ сортировки в поле – **отсутствует, по возрастанию, по убыванию**. Если сортировка выбрана в нескольких полях, то сначала сортируется по значениям в самом левом поле, затем среди одинаковых значений этого поля по значениям в следующем поле сортировки и т.д.

В строке **Вывод на экран** указывается, показывать или нет это поле в таблице-результате выполнения запроса. Обычно не показывают результаты поля различных промежуточных расчетов.

В строку **Условие отбора** помещают критерии отбора записей по этому полю. Для составных критериев заполняют последующие строки **ИЛИ**.

Пример заполнения бланка запроса

В запросе (Рисунок 13) источником является таблица **Товары**. Из неё в запрос выбраны поля *Наименование*, *Категория*, *Цена*, *Количество*. По поле *Наименование* выполнена сортировка по возрастанию (т.е. по алфавиту), в строке *Условие отбора* введен критерий для отбора товаров категории **Продукты**.

The screenshot shows a query builder interface. A dropdown menu is open for the table 'Товары', listing fields: Код товара, Наименование, Категория, Описание, Производитель, Дата поставки, Количество, Цена поставки, Единица измерения, Скидка %, and Фото товара. Below the menu is a table for query configuration:

Поле:	Наименование	Категория	Цена поставки	Количество
Имя таблицы:	Товары	Товары	Товары	Товары
Сортировка:	по возрастанию			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		'продукты'		
или:				

Рисунок 13 – Пример заполнения бланка запроса

Результат выполнения запроса показан ниже (Таблица 8).

Таблица 8 – Результат выполнения запроса

Наименование	Категория	Цена поставки	Количество
рис	продукты	95,00р.	100
сахар	продукты	41,00р.	200
сахар рафинад	продукты	60,50р.	500
сахар тростниковый	продукты	80,50р.	300

ЗАПРОСЫ НА ВЫБОРКУ

Запрос с отбором записей

Для отбора записей в бланке в строках **Условие отбора** и **ИЛИ** вводят критерии отбора. Они могут состоять из следующих компонентов: констант (числовых, текстовых, дата/времени), знаков операций, скобок, имен полей, символов шаблона (*, ?) и имён функций.

Простые условия отбора

Условия отбора по текстовым полям:

- по точному текстовому значению

Поле:	Категория
Условие отбора:	“продукты”

- по символам с начала строки

Поле:	Наименование
Условие отбора:	“сах*”

- по подстроке

Поле:	Наименование
Условие отбора:	“*сах*”

Условия отбора по числовым полям:

- отбор по точному значению

Поле:	Цена
Условие отбора:	200

- отбор с операциями сравнения

Поле:	Цена
Условие отбора:	<200

Условие отбора по логическим полям:

- логическое значение «Да» (физическое лицо)

Поле:	Покупатель	Физическое лицо
Условие отбора:		Да

- логическое значение «Нет» (не физическое лицо)

Поле:	Покупатель	Физическое лицо
Условие отбора:		Нет

Условие отбора по полям даты/время отличается от чисел только тем, что даты (и время) необходимо заключать в решетки (#):

- отбор по точной дате

Поле:	Дата покупки
Условие отбора:	#01.03.2013#

- отбор по диапазону дат (февраль 2013)

Поле:	Дата покупки
Условие отбора:	>=#01.02.2013# And <=#28.02.2013#

При построении условий отбора по части даты нужно иметь ввиду, что дату можно рассматривать как текст из восьми символов в формате ДД.ММ.ГГ:

- выбор дня по дате

Поле:	Дата покупки
Условие отбора:	Like "01*"

- выбор месяца по дате

Поле:	Дата покупки
Условие отбора:	Like "??.02*"

- выбор года по дате

Поле:	Дата поставки
Условие отбора:	Like "*13"

Условие отбора для пустых и непустых значений.

В Microsoft Access различаются два типа пустых значений: отсутствие значения в поле и пустые строки для текстовых полей (“”). Отсутствие значения означает, что данные в поле не введены независимо от типа поля.

Например, если таблица «Товары» содержит поле «Дата поставки», это поле может остаться не заполненным, когда неизвестна точная дата поставки товара. В этом случае поле имеет значение Null, указывающее, что точное значение неизвестно.

Критерий для пустых и непустых значений не зависит от типа данных в поле:

- отсутствует дата поставки – поле **Дата поставки** не заполнено

Поле:	Дата поставки
Условие отбора:	Is Null

- поле **Дата поставки** заполнено

Поле:	Дата поставки
Условие отбора:	Is Not Null

- по незаполненным значениям в текстовом поле (поле **Адрес** не заполнено)

Поле:	Адрес
Условие отбора:	Is Null

В текстовое поле может быть введено пустое значение ""

- по пустым значениям в текстовом поле (в поле **Телефон** введено пустое значение "")

Поле:	Телефон
Условие отбора:	""

- по непустым значениям в текстовом поле

Поле:	Телефон
Условие отбора:	<>""

Составные (сложные) условия отбора

Условие отбора типа ИЛИ

Для формирования условий отбора типа ИЛИ заполняют строку **Условие отбора** и строку **ИЛИ:**

- отбор товаров категорий **обувь** и **продукты**

Поле:	Категория
Условие отбора:	"обувь"
Или:	"продукты"

Условие отбора типа И:

- по значению в одном поле (диапазон значений, отображены товары с ценой от 30 до 100)

Поле:	Цена поставки
Условие отбора:	Between 30 And 100
Или:	

- для формирования условий отбора типа И в разных полях заполняют строку. **Условие отбора** в этих полях, будут отображены товары категории **продукты** с ценой не более 31.

Поле:	Категория	Цена поставки
Условие отбора:	"продукты"	<=31
Или:		

Запрос с вычисляемым полем

Для создания вычисляемого поля необходимо в пустой колонке бланка конструктора запросов в строку «Поле» ввести имя нового поля (оно не должно совпадать с именами полей источников запроса) и после двоеточия вычисляемое выражение. Оно может содержать константы, знаки операций, скобки, имена функций и полей. Примеры использования функций в вычисляемых полях приведены в Приложении Б. Имена полей нужно вводить в прямоугольных скобках (Рисунок 14).



Рисунок 14 – Запрос с вычисляемым полем

Примеры вычисляемых полей:

- вычисления с числовыми полями – стоимость товара

Стоимость: [Цена поставки]*[Количество]

- вычисления с числовыми полями - *стоимость товара со скидкой*

Стоимость товаров: [Цена покупки]*[Покупки]*[Количество]*(1-[Скидка %])

- вычисления с текстовыми полями – *полные фамилия, имя, отчество*

ФИОП: [Фамилия] &” “&[Имя]&” “&[Отчество]

- вычисления с текстовыми полями – *фамилия с инициалами*

ФИО: [Фамилия] &” “&Left([Имя];1)&”.“& Left([Отчество];1)&”.“

- вычисления с полями дат

вычисление возраста

Возраст: Int((Date()-[Дата рождения])/365)

вычисление года по дате

Год: Year([Дата рождения])

вычисление месяца по дате

Год:Month([Дата рождения])

- преобразование одних значений в другие

получение текстовой оценки из числовой (числовая оценка в поле [оценка]) с использованием функции Switch:

Текстовая оценка: Switch([Оценка]=5;"отлично";[Оценка]=4;"хорошо";
[Оценка]=3;"удовлетворительно";[Оценка]=2;"неудовлетворительно";[Оценка]=0;"неявка")

получение названия месяца по его номеру (номер месяца в поле [Месяц]) с использованием функции Choose:

Название месяца:Choose([Месяц];"январь";"февраль";"март";"апрель";"май";"июнь";"июль";"август";"сентябрь";"октябрь";"ноябрь";"декабрь")

Запрос с параметром

Параметр – вводимое пользователем в режиме диалога значение в условии отбора или в вычисляемом поле. Позволяет пользователю изменять условие отбора или вычисляемое выражение не используя конструктор запросов.

Параметр в условии отбора

Параметр в условии отбора – это текст подсказки пользователю, который вводится в конструкторе в строку условий отбора в квадратных скобках. Он не должно совпадать с именами полей.

Пример заполнения условия отбора с параметром и окна ввода параметра приведены ниже (Рисунок 15,Рисунок 16).

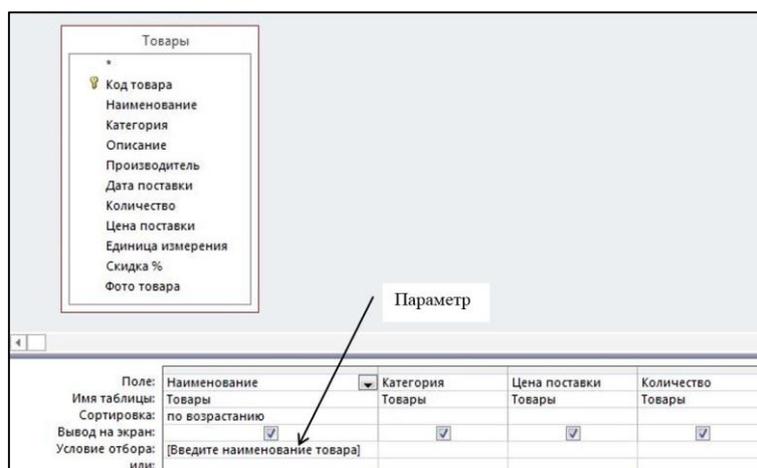


Рисунок 15 – Запрос с параметром

При выполнении запроса сначала необходимо ввести значение параметра, затем с ним будет подсчитано значение поля в каждой записи. Например, вычисление цены со скидкой, величина скидки вводится как параметр (Рисунок 17).

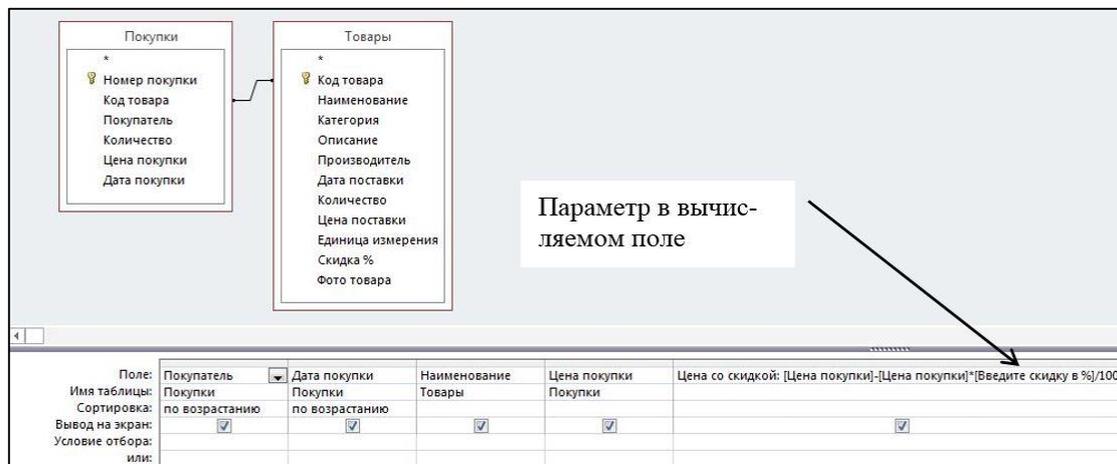


Рисунок 17 – Использование параметра в вычисляемом поле

Запрос с итогами

Запрос с итогами позволяет сгруппировать данные по полю, в котором есть повторяющиеся значения, а по другим полям подвести итоги. Для числовых полей итогами является вычисление максимума, минимума, среднего, суммы. Для текстовых полей итогами может быть первое или последнее значение в группе записей. Запрос с итогами можно создать с помощью мастера простых запросов или в конструкторе запросов, включив строку **Группировка** командой **Итоги** (Рисунок 18).

В запросах с итогами полей группировки и полей для подведения итогов может быть несколько. Должно быть, как минимум, одно поле группировки и одно поле итогов. Если группировка происходит по нескольким полям, то сначала группируются одинаковые значения в самом левом поле, затем среди одинаковых значений в этом поле происходит группировка по значениям в следующем поле и т.д. Группировка по ключевым полям не имеет смысла – в них значения не повторяются и группу для подведения итогов образовать невозможно (Рисунок 18).

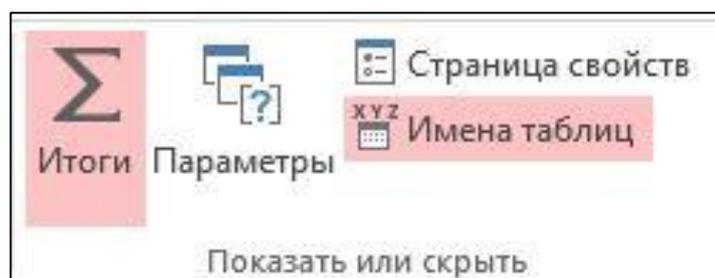


Рисунок 18 – Команда **Итоги** в конструкторе запросов

Заполнение бланка запроса с итогами (Рисунок 19).

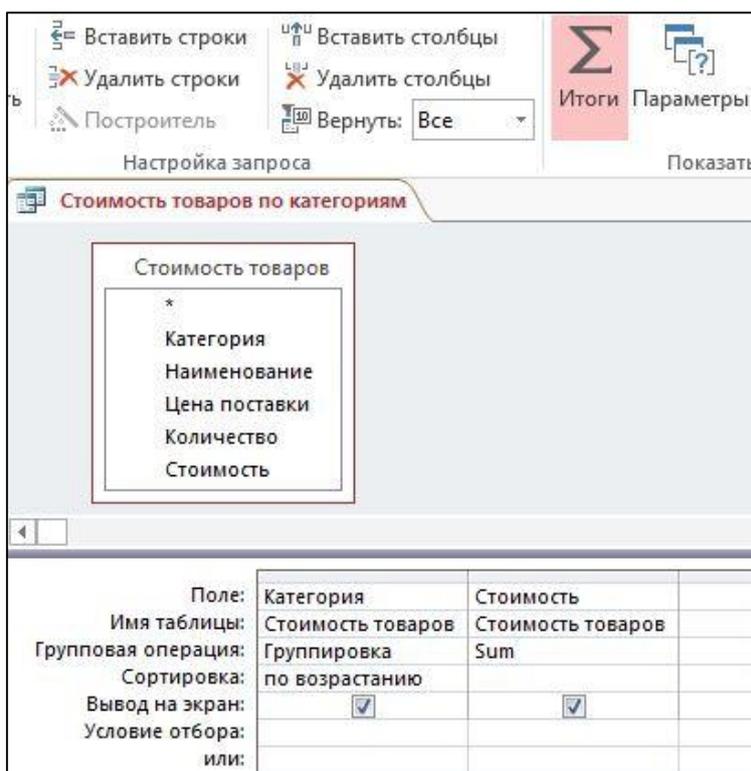


Рисунок 19 – Заполнение бланка запроса с итогами

Результат выполнения запроса с итогами представлен ниже (Рисунок 20).

Категория	Sum-Стоимость
инструмент	2 900,00 Р
обувь	150 000,00 Р
одежда	18 830,00 Р
посуда	33 400,00 Р
продукты	72 100,00 Р
сантехника	33 300,00 Р
стройматериалы	436 400,00 Р
хозтовары	30 000,00 Р

Рисунок 20 – Результат выполнения запроса с итогами

Варианты заполнения строки **Групповая операция**:

1. **Группировка** – для поля группировки.
2. **Функция** – для поля подведения итогов.
3. **Выражение** – для вычисляемых полей.
4. **Условие** – для полей, по которым заполняется условие отбора.

Перекрестный запрос

Перекрестный запрос предполагает группировку по двум полям и подведение итогов по одному полю. Результат выполнения перекрестного запроса похож на сводную таблицу в Excel. Создать новый перекрестный запрос можно с помощью мастера «Перекрестный запрос» или в конструкторе. В конструкторе создают запрос на выборку, определяют источники запроса и затем в группе команд «Тип запроса» кнопкой выбирают **Перекрестный**. В бланк добавляются строки **Групповая операция** и **Перекрестная таблица**. Пример заполнения бланка перекрестного запроса показан ниже (Рисунок 21).

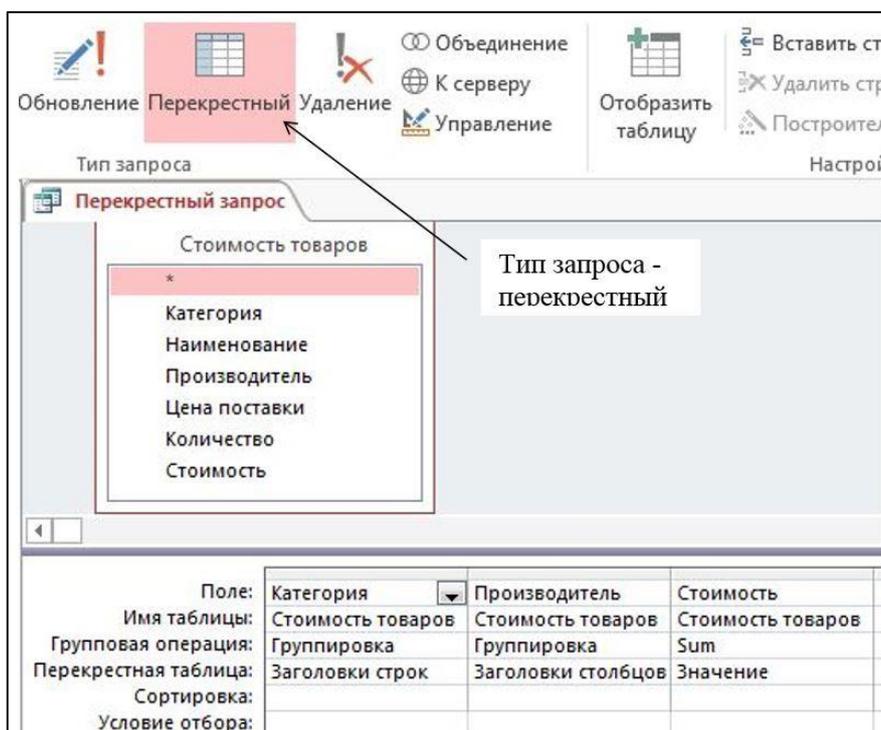


Рисунок 21 – Заполнение бланка перекрестного запроса

Результат выполнения перекрестного запроса представлен ниже (Рисунок 22).

Категория	АО "Подол"	ЗАО "Инстр"	ИП Рубашки	ОАО "Боро"	ОАО "Стро"	ООО "Анта"	ЧП "Фарфор"
инструмент					2 900,00 ₺		
обувь	150 000,00 ₺						
одежда		4 130,00 ₺	14 700,00 ₺				
посуда				3 000,00 ₺			30 400,00 ₺
продукты						72 100,00 ₺	
сантехника					33 300,00 ₺		
стройматериаль					436 400,00 ₺		
хозтовары					30 000,00 ₺		

Рисунок 22 – Результат выполнения перекрестного запроса

Варианты заполнения строки **Групповая операция**:

1. **Группировка** – для поля группировки.
2. **Название функции** – для поля подведения итогов.
3. **Выражение** – для вычисляемых полей.
4. **Условие** – для полей, в которых заполнено условие отбора.

Варианты заполнения строки **Перекрестная таблица**:

1. **Заголовки строк** – для поля группировки по строкам.
2. **Заголовки столбцов** – для поля группировки по столбцам.
3. **Значение** – для подведения итогов.

РЕКОМЕНДАЦИИ ПО СОЗДАНИЮ ЗАПРОСОВ

При создании запросов необходимо руководствоваться следующими рекомендациями:

- не включать в качестве источников таблицы, поля которых не используются в запросе;
- включать в запрос только необходимые поля источников;
- выводить поля в нужном порядке;
- кодовые поля (счетчики) обычно не включают в запрос;
- обязательно выполнять сортировку в запросах, сортировка по нескольким полям должна быть оправданной;
- сложную обработку данных лучше выполнять в нескольких последовательных запросах;
- давать запросам осмысленные имена.

ИСПОЛЬЗОВАНИЕ СВОЙСТВ ЗАПРОСА

Свойства запроса

Свойства запроса можно изменить, открыв в конструкторе запросов окно свойств из контекстного меню (правая кнопка мыши) или командой **Страница свойств** (Рисунок 23):

Вкладка Конструктор → **Группа команд Показать или скрыть** → **Команда Страница свойств**.

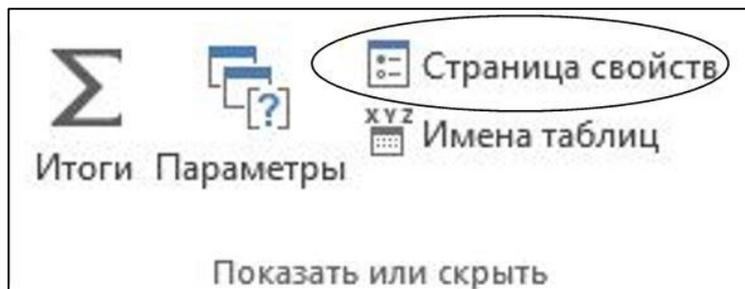


Рисунок 23 – Команда Страница свойств

Свойства запроса представлены ниже (Рисунок 24).

Окно свойств	
Тип выделенного элемента: Свойства запроса	
Общие	
Описание	
Режим по умолчанию	Режим таблицы
Вывод всех полей	Нет
Набор значений	Все
Уникальные значения	Нет
Уникальные записи	Нет
База данных-источник	(текущая)
Строка подключения-источник	
Блокировка записей	Отсутствует
Тип набора записей	Динамический набор
Время ожидания ODBC	60
Фильтр	
Порядок сортировки	
Максимальное число записей	
Ориентация	Слева направо
Имя подтаблицы	
Подчиненные поля	
Основные поля	
Высота подтаблицы	0 см
Развернутая подтаблица	Нет
Фильтр при загрузке	Нет
Сортировка при загрузке	Да

Рисунок 24 – Свойства запроса

Свойства поля в запросе

Для изменения свойств поля в конструкторе запросов необходимо выделить нужное поле и открыть окно свойств из контекстного меню (правая кнопка мыши) или командой **Страница свойств**. Обычно свойства поля изменяют в вычисляемых полях. Например, в вычисляемом поле **Стоимость** выбран формат поля **Денежный** и число десятичных знаков **2** (Рисунок 25).

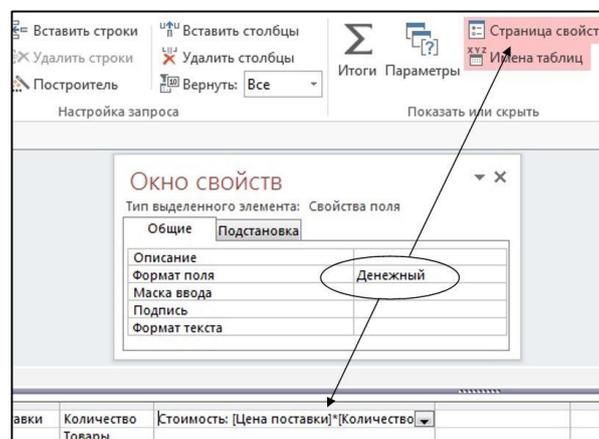


Рисунок 25 – Свойства поля Стоимость

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите универсальный язык программирования запросов.
2. Назовите источники запроса.
3. Перечислите запросы на изменение.
4. Что подсчитает выражение [Цена]*[Скидка в %] в вычисляемом поле запроса?

5. ЗАПРОСЫ НА ИЗМЕНЕНИЕ

ОСОБЕННОСТИ ЗАПРОСОВ НА ИЗМЕНЕНИЕ

Эти запросы изменяют данные (добавляют, удаляют, обновляют, создают новые таблицы). Неправильное применение запросов на изменение может привести к потере данных, поэтому такие запросы требуют особого внимания при выполнении. Перед выполнением таких запросов необходимо создавать копии таблиц-источников запроса или всей базы данных.

ЗАДАЧИ, РЕШАЕМЫЕ ЗАПРОСАМИ НА ИЗМЕНЕНИЕ

С помощью запросов на изменение можно решать следующие задачи (Таблица 9).

Таблица 9 – Типы запросов

Тип запроса	Решаемая задача
Запрос на создание таблицы	Создание копий таблиц (архивов)
	Отбор записей в новую таблицу
	Создание таблиц из нескольких таблиц-источников
	Добавление новых полей в создаваемую таблицу
	Создание пустых таблиц
Запрос на удаление	Удаление всех данных в таблице (очистка таблицы)
	Удаление определенных записей в таблице
Запрос на обновление	Обновление данных в полях таблицы
Запрос на добавление	Добавление новых данных из других источников (баз данных, электронных таблиц)

В качестве примера рассмотрим базу данных **Студенты**, схема данных которой представлена ниже (Рисунок 26).

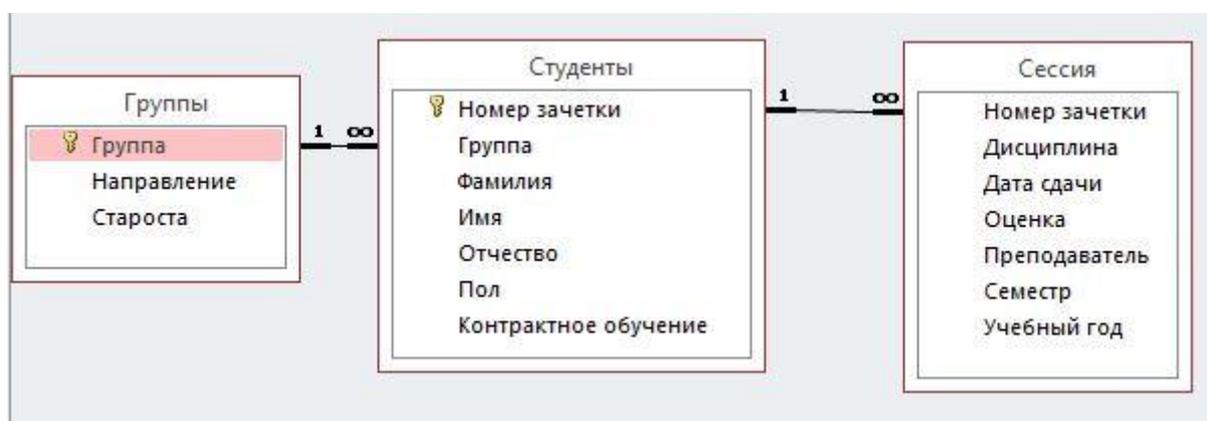


Рисунок 26 – Схема базы данных Студенты

База данных содержит сведения о студентах, обучавшихся в 2019/20 году. Необходимо в конце 2019/20 учебного года подготовить таблицы базы данных для нового 2020/21 учебного года. Последовательность операций, которые необходимо выполнить при преобразовании базы данных, представлена ниже (Таблица 10).

Таблица 10 – Последовательность выполняемых запросов

№ п/п	Операция	Тип запроса на изменение	Периодичность выполнения
1	Создание таблицы Студенты 2020 из таблицы Студенты	Запрос на создание таблицы	один раз в конце учебного года
2	Удаление данных о закончивших в 2020 году обучение в таблице Студенты с условием отбора по номеру курса (5 курс)	Запрос на удаление	
3	Перевод студентов на следующий курс в 2014/15 учебном году с увеличением номера курса на единицу.	Запрос на обновление	
4	Добавление данных о 1 курсе из таблицы Excel в таблицу Студенты	Запрос на добавление	

СОЗДАНИЕ ЗАПРОСОВ НА ИЗМЕНЕНИЕ

Запросы на изменение создаются в конструкторе. Для этого выполняют команду создания запроса на выборку с помощью конструктора, определяют источники запроса, затем на вкладке **Конструктор** в группе команд **Тип запроса** выбирают нужный тип запроса на изменение (Рисунок 27). В бланк добавляются дополнительные строки, соответствующие типу запроса.



Рисунок 27 – Команды выбора типа запроса

ЗАПРОС НА СОЗДАНИЕ ТАБЛИЦЫ

В запросе на создание таблицы участвуют таблицы-источники (одна или несколько) и таблица-получатель. Таблиц-источников может быть несколько, таблица-получатель всегда одна.

Запрос на создание копии таблицы

Сначала создаем обычный запрос на выборку, в качестве источника берем таблицу **Студенты** (Рисунок 28).

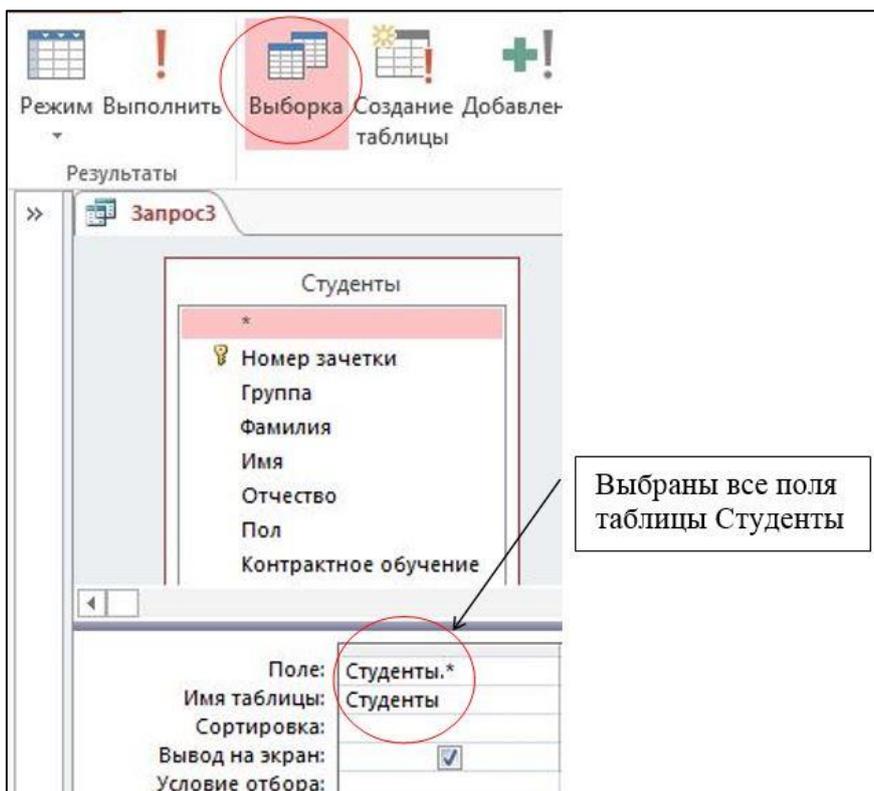


Рисунок 28 – Базовый запрос для создания полной копии таблицы

Изменяем тип запроса – запрос на создание таблицы и указываем имя создаваемой таблицы-получателя (Рисунок 29).

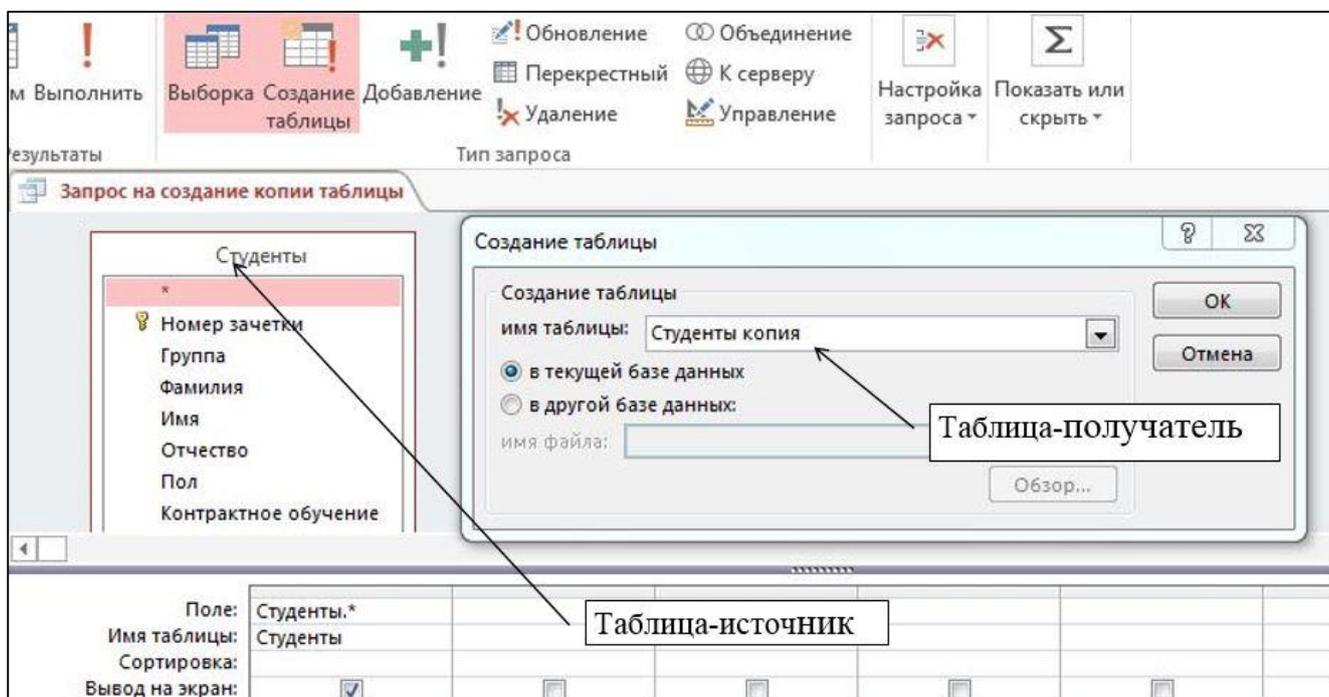


Рисунок 29 – Заполнение бланка запроса на создание полной копии таблицы

Имя таблицы-получателя можно увидеть и изменить в свойствах запроса (Рисунок 30).

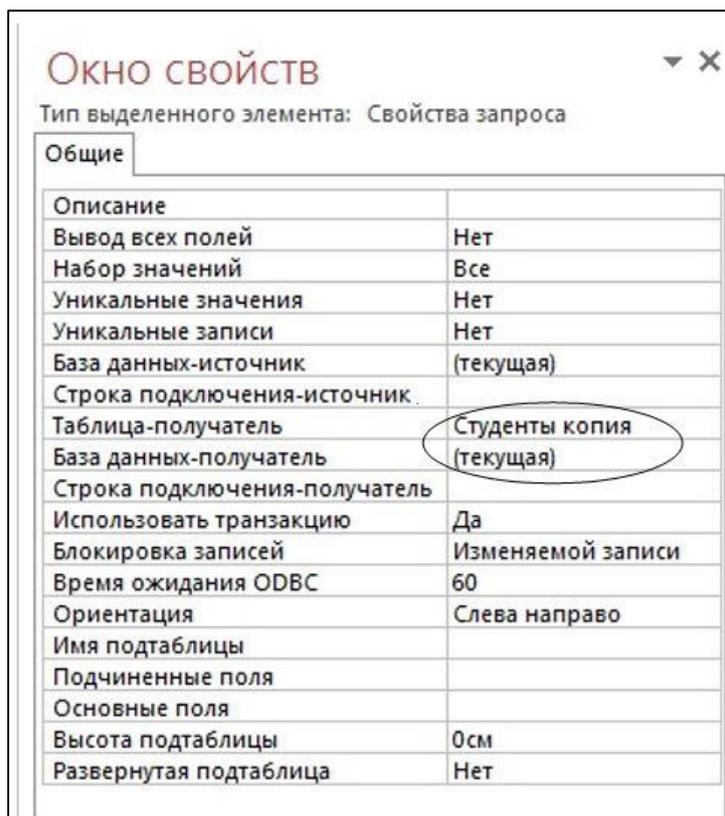


Рисунок 30 – Свойства запроса на создание копии

На каждом шаге выполнения запроса на создание таблицы (и других запросов на изменение) требуется подтверждение или отмена его выполнения (Рисунок 31).

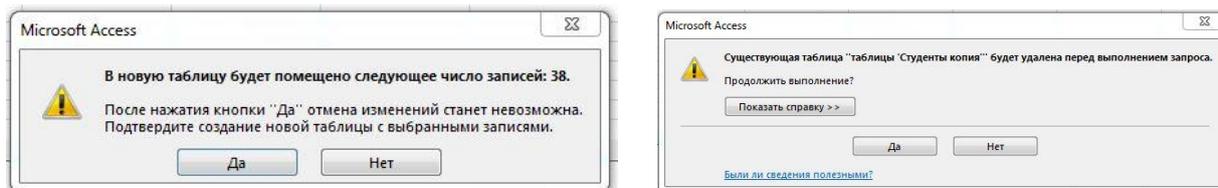


Рисунок 31 – Предупреждения при выполнении запроса на изменение данных

Запрос на создание таблицы с отбором записей

Создадим таблицу результатов сдачи сессии одной из групп – Рисунок 32.

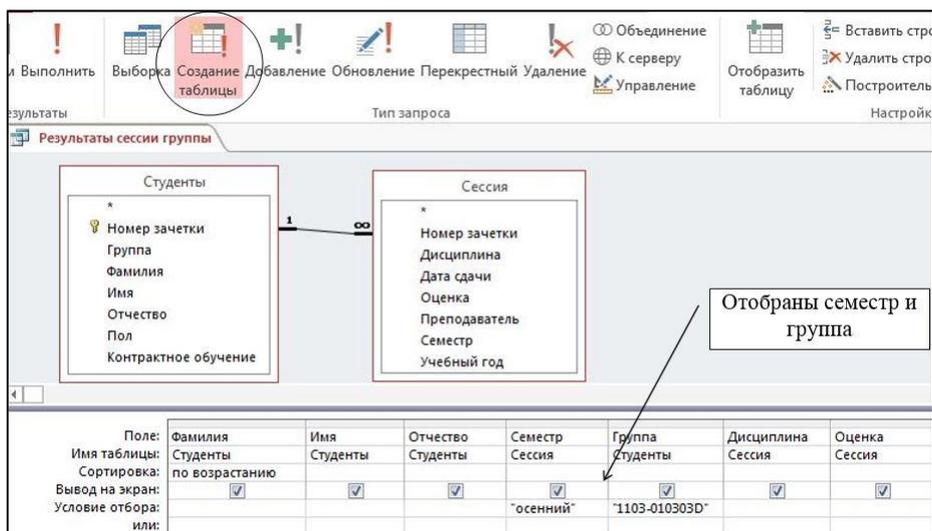


Рисунок 32 – Создание таблицы результатов сессии группы

Запрос на создание таблицы с отбором и добавлением полей

Создадим таблицу **Ведомость передачи**, добавив новое поле ФИО (фамилия с инициалами) и отбором студентов, не сдавших экзамен (Рисунок 33).

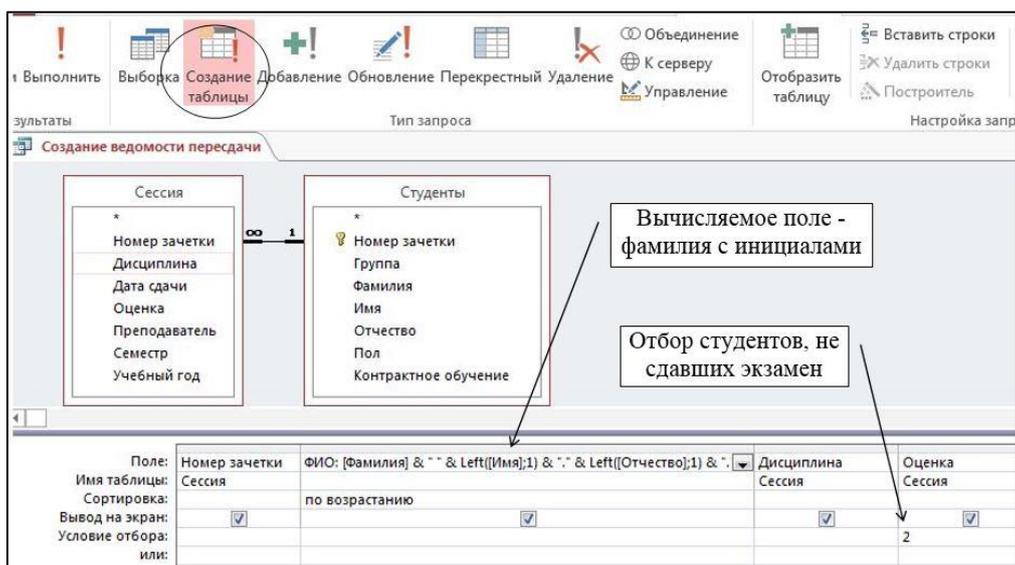


Рисунок 33 – Запрос на создание таблицы ведомости передачи

Создание пустой таблицы

С помощью запроса создадим пустую таблицу **Студент 2020**. Условие отбора для создания пустой таблицы **Студенты** показано ниже. Так как студентов с отсутствующим номером зачетки нет, то с критерием IsNull (отбор пустых значений) не будет отобрано ни одной записи – таблица будет пустой. Т.к. нельзя выводить одно и тоже поле **Номер зачетки** (Рисунок 34) дважды – в общем списке полей **Студенты**. * и отдельно, поэтому вывод на экран поля Номер зачетки отключен.

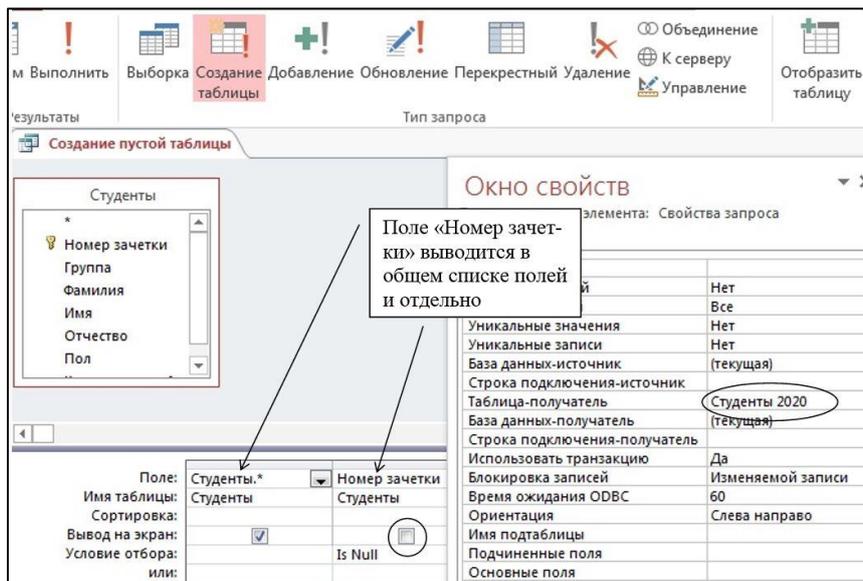


Рисунок 34 – Создание пустой таблицы

Создание таблицы уникальных значений в поле

Очень часто требуется из повторяющихся значений в поле создать список уникальных (неповторяющихся) значений. Например, можно создать таблицу **Список групп** взяв их номера из поля **Группа** таблицы **Студенты**, где номера групп многократно повторяются. Включим свойство запроса **Уникальные значения** (Рисунок 35).

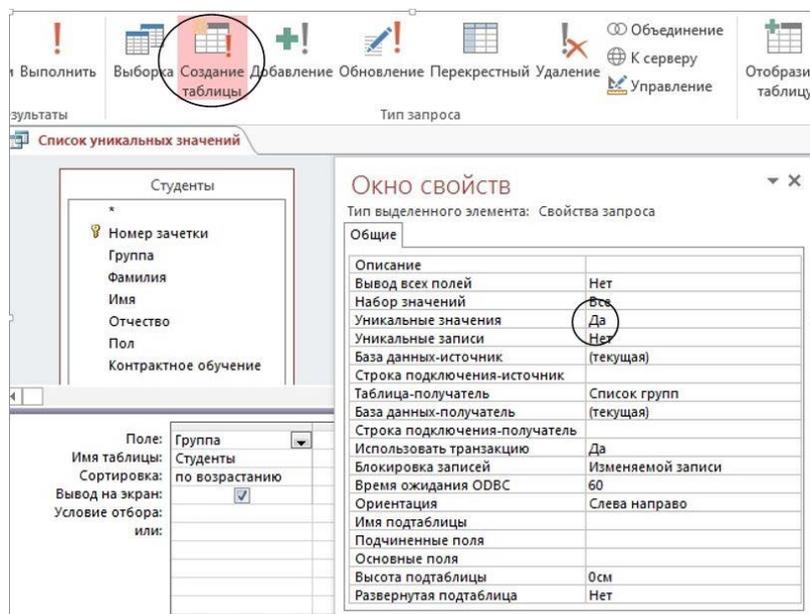


Рисунок 35 – Создание таблицы уникальных значений

После выполнения запроса получим таблицу списка групп:

Список групп
Группа
1103-010303D
1104-150303D
1105-010303D
1113-240301D
1114-240301D

Возможности запроса на создание таблицы

Запрос на создание таблицы позволяет:

1. Создавать копию таблицы.
2. Создавать таблицу из нескольких таблиц-источников.
3. Отбирать записи в новую таблицу.
4. Добавлять новые поля в создаваемую таблицу.
5. Создавать пустые таблицы.
6. Создавать таблицы уникальных значений полей.

ЗАПРОС НА УДАЛЕНИЕ

Удаление отобранных записей

Создадим запрос, удаляющий результаты сдачи сессии осеннего семестра из таблицы **Сессия** (Рисунок 36).

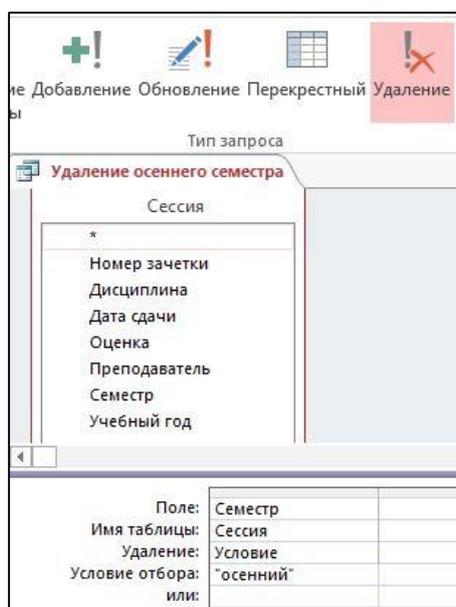


Рисунок 36 – Запрос на удаление данных осеннего семестра

В запросе на удаление указывают только те поля, в которых имеется условие отбора (Рисунок 37)!

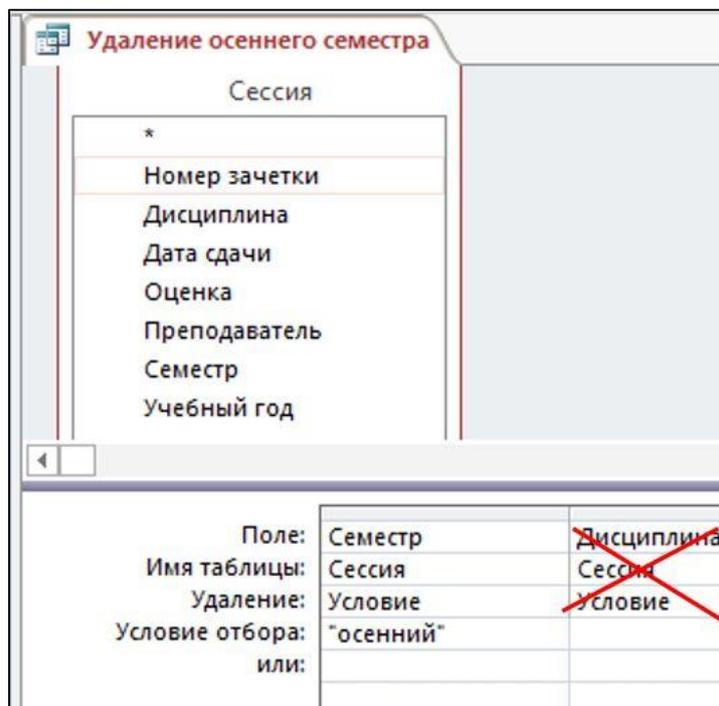


Рисунок 37 – Лишнее поле в запросе на удаление

Предупреждение при выполнении запроса на удаление показано ниже (Рисунок 38).

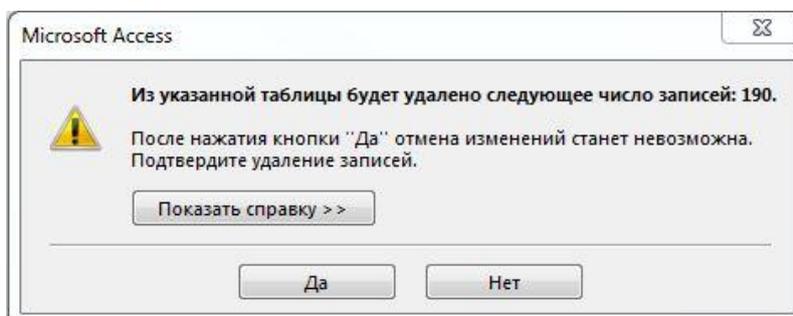


Рисунок 38 – Предупреждение о количестве удаляемых записей

Удаление всех записей из таблицы

Если нужно удалить все записи из таблицы (очистить таблицу), то используем следующий запрос (Рисунок 39 – Удаление всех записей из таблицы).

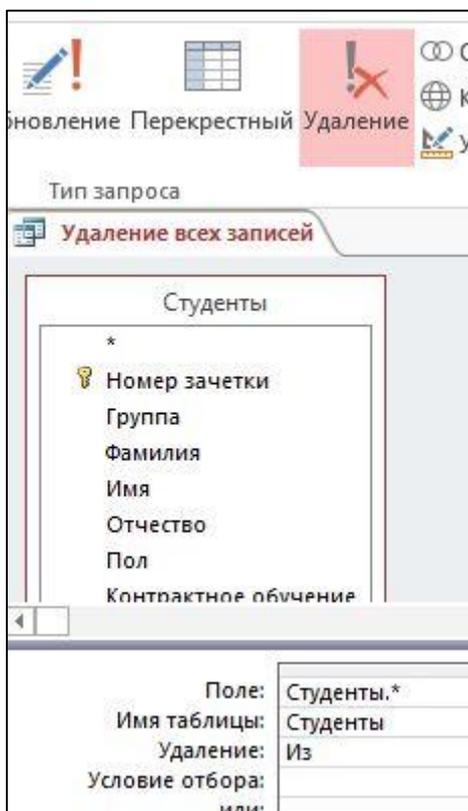


Рисунок 39 – Удаление всех записей из таблицы

Особенности выполнения операции удаления через запрос

При выполнении запросов на удаление необходимо ориентироваться на схему данных (Рисунок 40).

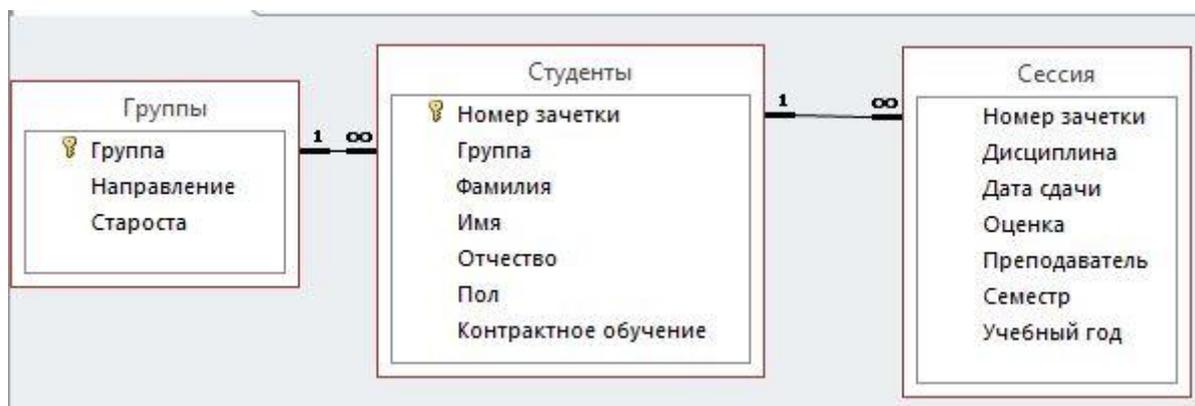


Рисунок 40 – Схема данных базы данных Студенты

Рекомендации по созданию и выполнению запросов на удаление:

1. Перед выполнением запроса на удаление создавайте резервные копии таблиц, из которых происходит удаление.
2. В запрос на удаление включают только те поля, в которых записывают условия отбора
3. Удаление записей в таблице, у которой нет ни одной стороны связи "один" происходит без проблем. В нашем случае это таблица **Сессия**.
4. Удаление записей в таблице на стороне "один" связи "один-ко-многим" (таблица **Студенты** по отношению к таблице **Сессия**, таблица **Группы** по отношению к таблице **Студенты**) возможно, если включено свойство связи "Каскадное удаление". В этом случае происходит удаление записей на стороне "один" и соответствующих записей (если они есть) в другой таблице на стороне "много".

Возможности запроса на удаление

Запрос на удаление записей из таблицы позволяет:

1. Удалять отобранные записи.
2. Удалять все записи.

ЗАПРОС НА ОБНОВЛЕНИЕ

Обновление поля во всех записях

Создадим запрос на обновление, переводящий всех студентов на следующий курс учитывая структуру номера группы. Номер курса – это второй символ в номере группы (Рисунок 41).



Рисунок 41 – Структура номера группы

Номер группы присутствует в связанных таблицах **Группы** (сторона *один*) и в таблице **Студенты** (сторона *много*).

Обновление номера группы на стороне *много* невозможно, т.к. новых групп нет на стороне *один*. Поэтому обновление групп выполняем в таблице **Группы**, а за счет включенного свойства связи *Каскадное обновление* (Рисунок 42) произойдет изменение номеров групп на стороне *много* в таблице **Студенты**.

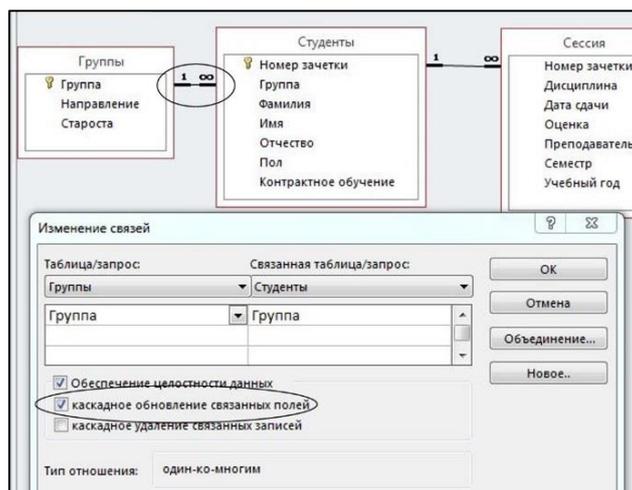


Рисунок 42 – Использование свойства каскадного обновления

Вид запроса в конструкторе представлен ниже (Рисунок 43).

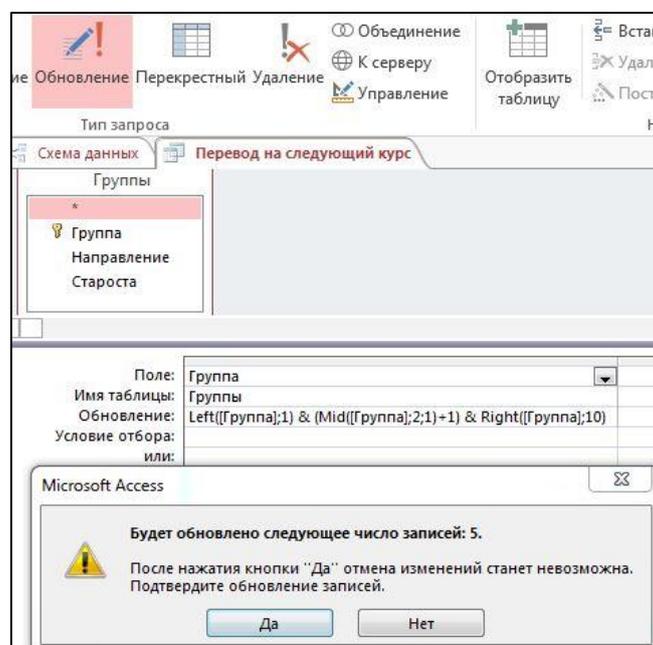


Рисунок 43 – Запрос на обновление номера группы

В строку **Обновление** поля **Группа** введем выражение для увеличения курса на единицу (второй символ). Поле **Группа** текстовое, поэтому собираем новый номер группы с помощью операции & (слияние) как текст из номера института, номера курса, увеличенного на единицу, и оставшихся десяти символов (порядкового номера группы на курсе, направления подготовки):

`Left([Группа];1) & (Mid([Группа];2;1)+1) & Right([Группа];10),`

где:

- Left([Группа];1) – функцией Left() вырезаем первый символ слева (институт);
- (Mid([Группа];2;1)+1) – функцией Mid() вырезаем второй символ (номер курса), добавляем единицу (увеличиваем номер курса);
- Right([Группа];10) – функцией Right() вырезаем десять символа справа (порядковый номер группы и направление подготовки).

Особенности выполнения запроса на обновление

1. При выполнении запросов на обновление необходимо ориентироваться на схему данных и связи между таблицами.
2. Перед выполнением запроса на обновление создают резервные копии таблиц, в которых происходит обновление.
3. В запрос включают только обновляемые поля и поля, по которым производится отбор.
4. Обновление данных в поле на стороне «много» возможно, если в связанном поле на стороне «один» есть новые значения. Например, если мы изменяем номера групп в таблице **Студенты**, то новые номера групп должны быть в таблице **Группы**;
5. Обновление данных в поле на стороне «один» невозможно, если в связанной таблице в поле на стороне «много» есть такие значения. Для обновления записей в таких случаях включите свойство связи *Каскадное обновление*. Например, если мы изменяем номер группы в таблице **Группы** (сторона *один*), то в таблице **Студенты** старый номер группы (сторона *много*) изменится на новый во всех записях, если включено свойство связи *Каскадное обновление* на этой связи.

ЗАПРОС НА ДОБАВЛЕНИЕ

Для добавления (импорта) новых данных в таблицы базы данных используется запрос на добавление. Данные можно добавить из таблиц текущей базы данных, другой базы данных или таблиц Excel.

Добавление данных из таблицы Excel

Необходимо добавить сведения о новых студентах в таблицу **Студенты** из таблицы Excel **Новые студенты** (Таблица 11).

Таблица 11 – Фрагмент данных таблицы Excel Новые студенты

Номер зачетки	Группа	Фамилия	Имя	Отчество
097026	1103-010303D	КОЛОСОВ	ВЛАДИМИР	ВЛАДИМИРОВИЧ
097019	1103-010303D	КОНОВАЛЮК	ВЛАДИСЛАВ	ВЛАДИМИРОВИЧ
097016	1103-010303D	КОТКИН	ГРИГОРИЙ	ВЛАДИМИРОВИЧ
097017	1103-010303D	ЛЕХИН	ДМИТРИЙ	ВЯЧЕСЛАВОВИЧ
097029	1103-010303D	МОРОЗОВ	ЕВГЕНИЙ	ВЛАДИМИРОВИЧ
097028	1103-010303D	МИТИН	ЕВГЕНИЙ	ВЛАДИМИРОВИЧ
097008	1104-150303D	МАКСИМЕНКОВА	ЕЛЕНА	ВЛАДИМИРОВНА
097009	1104-150303D	МАКСИМОВА	ЕЛЕНА	ВЛАДИМИРОВНА
097020	1104-150303D	ВИТВИНИНА	ЕЛЕНА	ВИКТОРОВНА
097006	1104-150303D	БЕЗДВЕРНАЯ	ЗОЯ	ВЛАДИМИРОВНА
097012	1104-150303D	АНДРЕЕВА	ИННА	ВЛАДИМИРОВНА

При выполнении запросов на добавление нужно ориентироваться на схему данных. В нашем случае таблицы **Группы** и **Студенты** связаны по полю *Группа*, поэтому добавление новых записей сразу в таблицу **Студенты** невозможно, т.к. в связанной таблице **Группы** нет новых групп и возникнет ошибка «Нарушение уникальности ключа».

Необходимо сначала добавить новые группы в таблицу **Группы**, а затем добавлять новых студентов в таблицу **Студенты**.

Связывание таблицы Excel с базой данных

Сначала необходимо связать таблицу Excel **Новые студенты** с базой данных. Для связывания используем команду:

Вкладка **Внешние данные** → Группа команд **Импорт и связи** → Команда **Excel**.

По шагам выполняем связывание (Рисунок 44 – Рисунок 47).

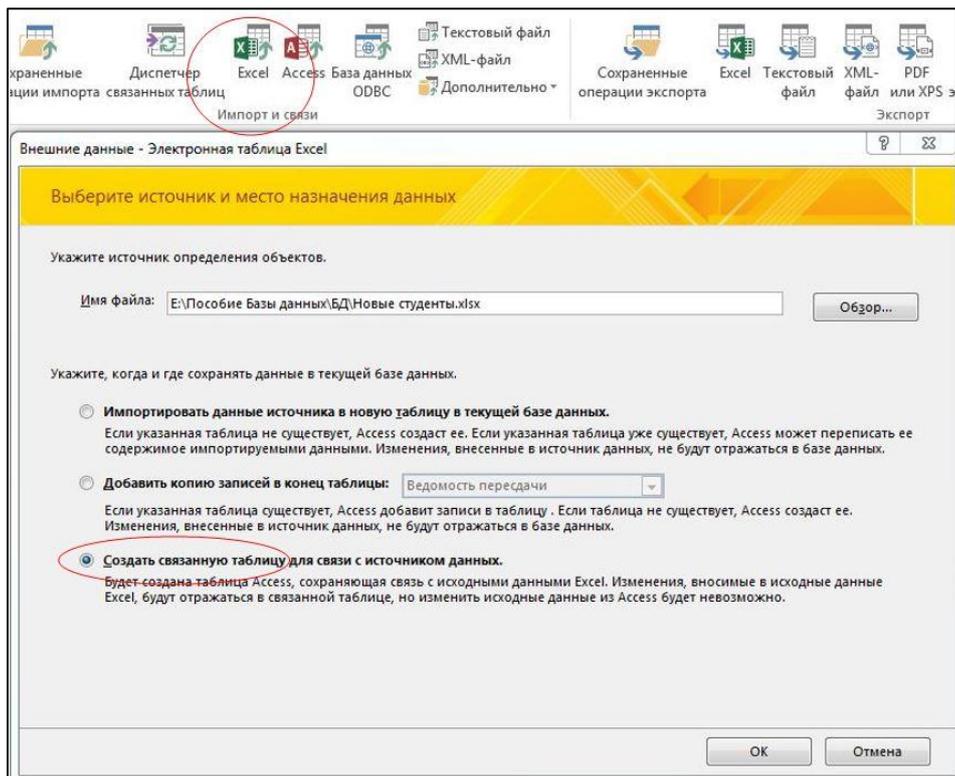


Рисунок 44 – Шаг 1. Выбор источника и способа связывания

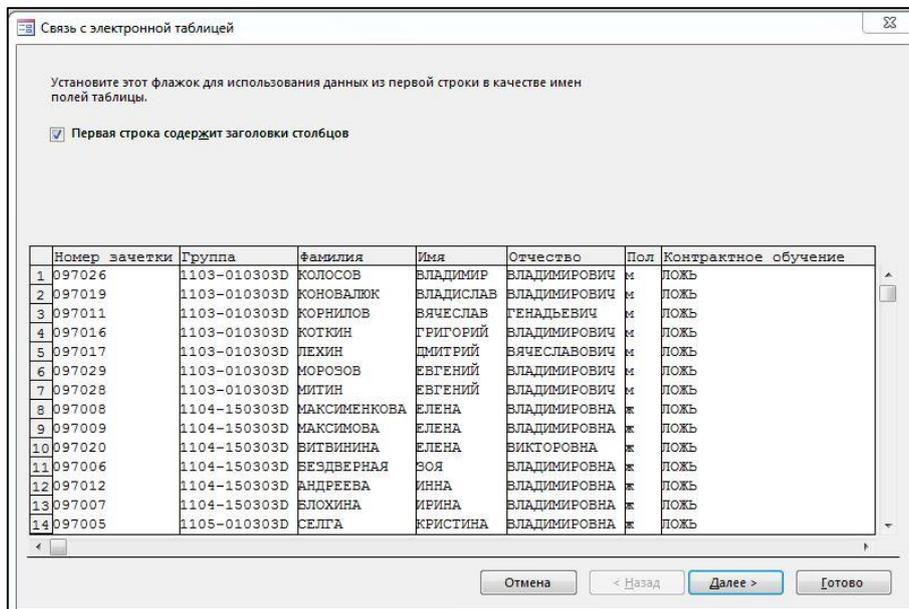


Рисунок 45 – Шаг 2. Заголовки столбцов

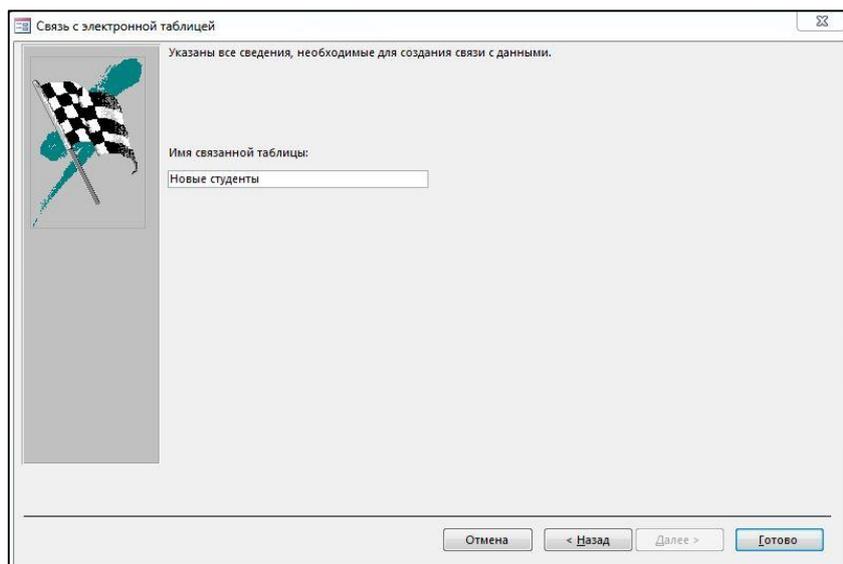


Рисунок 46 – Шаг 3. Имя таблицы Excel в базе данных

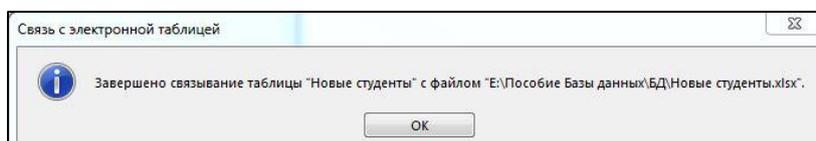


Рисунок 47 – Шаг 4. Завершение связывания

После окончания связывания в области навигации появится объект – связанная с базой данных таблица Excel (Рисунок 48). В отличие от обычной таблицы базы данных в связанной таблице Excel нельзя изменять данные и структуру, однако использовать данные этой таблицы можно в любых запросах. Если мы изменим данные в связанной таблице непосредственно в Excel, то эти данные изменятся и в связанной таблице в базе данных.

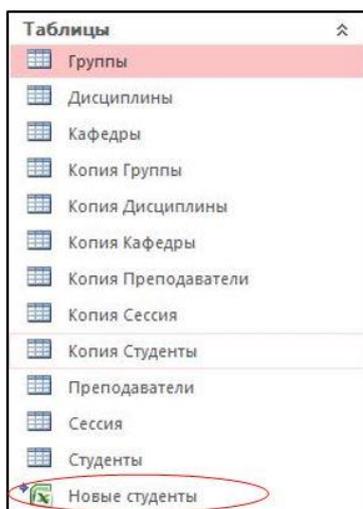


Рисунок 48 – Объект – связанная с базой данных таблица Excel

Запрос на добавление данных из таблицы Excel

После связывания создаем запрос на добавление данных из таблицы **Новые студенты** в таблицу **Студенты** (Рисунок 49). Получателем является таблица базы данных **Студенты**. Значения добавляются не во все поля таблицы **Студенты**.

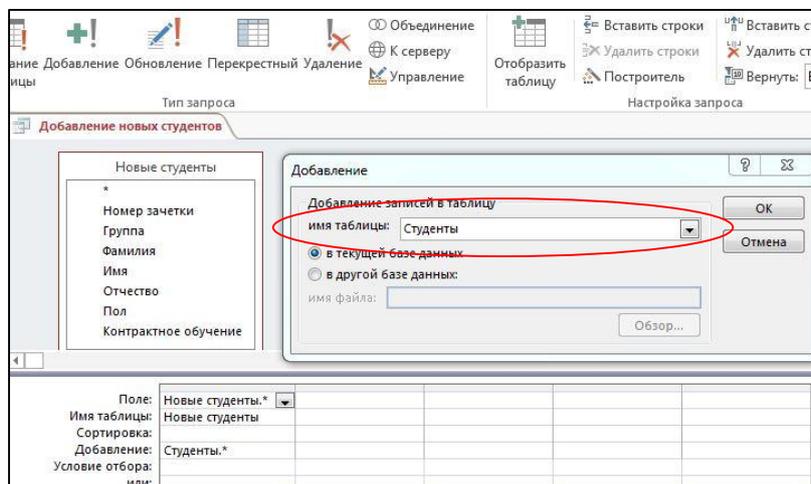


Рисунок 49 – Запрос на добавление данных из связанной таблицы Excel

В запросах на добавление в качестве таблиц-источников можно таким же образом использовать обычные таблицы базы данных и условия отбора.

Особенности добавления записей через запрос

При выполнении запросов на добавление необходимо ориентироваться на схему данных:

Если в добавляемых записях дублируются значения ключевых полей таблицы-получателя, то записи не будут добавлены из-за нарушения уникальности ключа. Не включайте в запрос ключевое поле-счетчик таблицы-источника. Если ключевое поле таблицы-источника смысловое (заполняется путем ввода), то проверьте, нет ли таких же значений ключевого поля в таблице-получателе.

Записи не удастся добавить в четырех случаях: ошибка преобразования типа, нарушение уникальности ключа, нарушение блокировки, нарушение условий на значение.

Если это происходит, то появляется предупреждающее сообщение. В нем указывается причина и количество не добавленных записей.

В нашем случае повторное добавление записей из таблицы Excel невозможно, так как будут повторяться номера зачетов, а номер зачетки – ключевое поле в таблице **Студенты**.

ТРЕБОВАНИЯ К ЗАПРОСАМ

При создании запросов необходимо выполнять следующие требования:

1. Осмысленность запросов: выводить необходимые поля в нужном порядке.
2. Обязательная сортировка в запросах, сортировка по нескольким полям должна быть оправданной.
3. В запросах с итогами обычно поле группировки одно, и несколько полей для подведения итогов. Группировка по ключевым полям не имеет смысла – в них значения не повторяются и группу для подведения итогов образовать невозможно.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Задачи, решаемые запросами на изменение.
2. Перечислите средства для создания запросов на изменение в Access.
3. Перечислите типы запросов на изменение в Access.
4. Каким образом на выполнение запросов на изменение влияют связи между таблицами?

6. ФОРМЫ

НАЗНАЧЕНИЕ И ИСТОЧНИКИ ФОРМ

Формы для ввода и редактирования данных

Эти формы построены на исходных таблицах и предназначены для ввода и редактирования данных в таблицах (Рисунок 50).

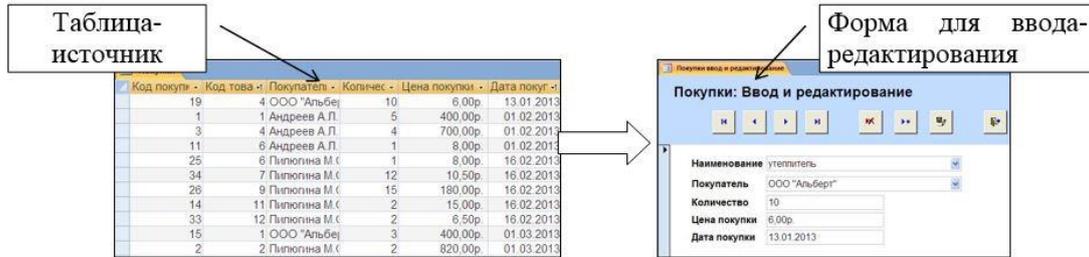


Рисунок 50 – Форма для ввода и редактирование

Формы для просмотра

Эти формы предназначены для просмотра данных, поэтому операции редактирования в них запрещены. Источником таких форм чаще всего является запрос (Рисунок 51).

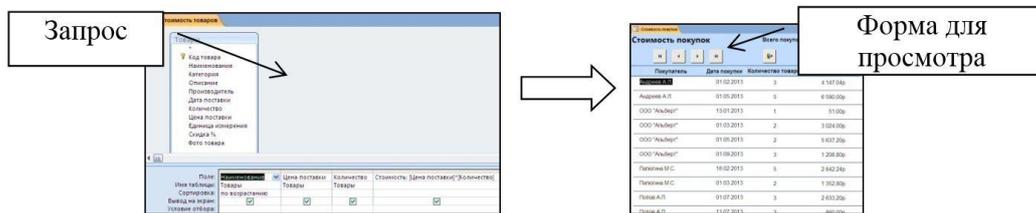


Рисунок 51 – Форма для просмотра

Особым видом формы являются подчиненные формы. У них два источника – таблицы или запросы. Данные из двух источников отображаются на одном экране (Рисунок 52).

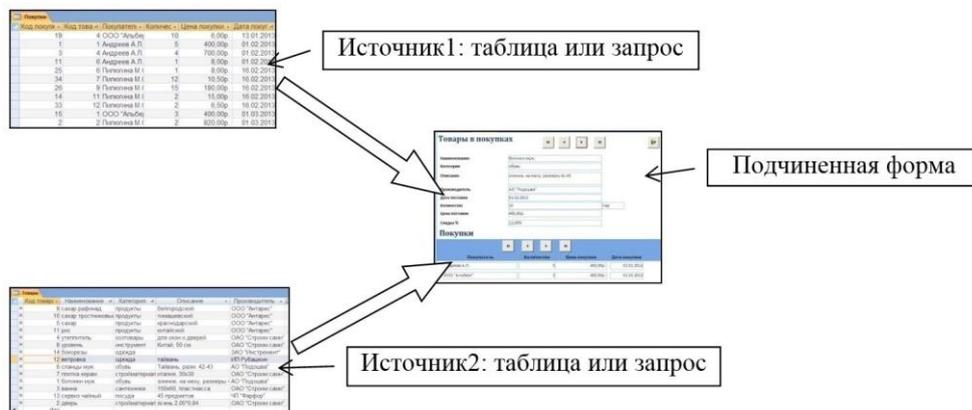


Рисунок 52 – Подчиненная форма

Формы – диаграммы

Источником формы – диаграммы может быть таблица или запрос (Рисунок 53).

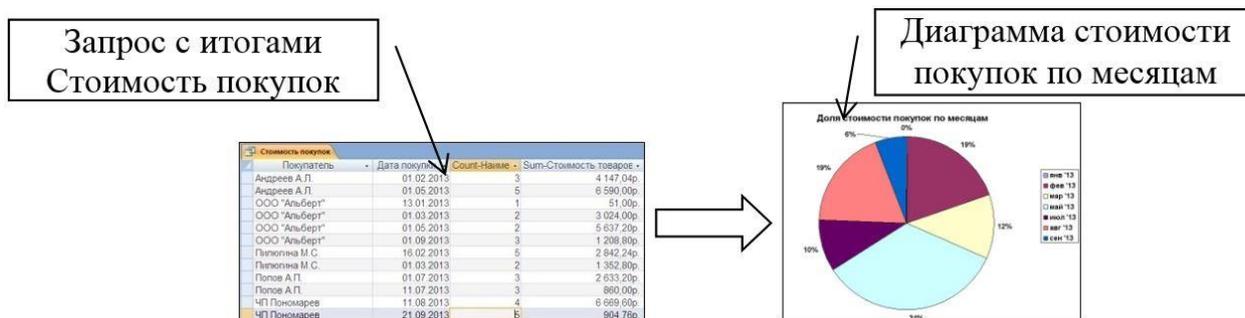


Рисунок 53 – Форма-диаграмма

СРЕДСТВА ДЛЯ СОЗДАНИЯ ФОРМ

Все средства для создания форм отображены на вкладке **Создание** в группе команд **Формы** (Рисунок 54).

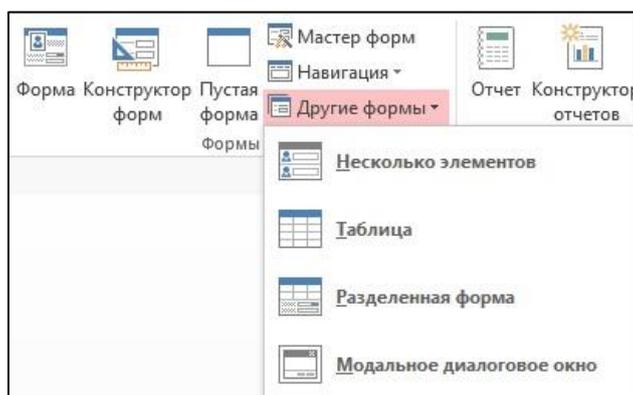


Рисунок 54 – Команды для создания форм

Мастера форм

Мастера позволяют быстро создать заготовку будущей формы, которую затем дорабатывают в конструкторе. С помощью мастера создают формы для ввода и редактирования, формы для просмотра (в том числе подчиненные), диаграммы. Возможности мастера ограничены, кроме того, мастера не позволяют редактировать формы.

В группе команд **Формы** доступны следующие мастера:

1. Форма (форма в один столбец).
2. Пустая форма.
3. Мастер форм – позволяет создать формы в один столбец, ленточные, табличные форму, подчиненные.

В команде **Другие формы** предусмотрены следующие мастера:

1. Несколько элементов (ленточная форма).
2. Таблица (табличная форма).
3. Разделенная форма (форма в один столбец совмещенная с табличной формой).

Конструктор форм

В конструкторе можно создать и отредактировать любую форму. Чтобы не начинать создание формы «с нуля», обычно сначала создают форму с помощью мастера, а затем редактируют ее в конструкторе. Конструктор форм вызывается командой:

Вкладка **Создание** → Группа команд **Формы** → Команда **Конструктор форм**.

Форма в конструкторе состоит из разделов **Заголовок формы**, **Примечание формы**, **Область данных**, **Верхний колонтитул**, **Нижний колонтитул**. В разделах формы размещают данные (поля источника, свободные поля), элементы управления (кнопки, поля со списками), подписи (Рисунок 55).

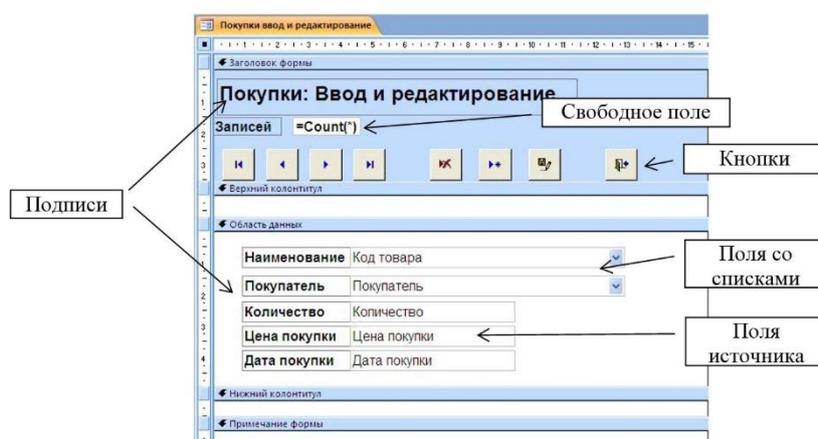


Рисунок 55 – Форма для ввода-редактирования в конструкторе форм

Команды для работы с формой в конструкторе находятся в группе вкладок **Инструменты конструктора форм**, состоящей из двух вкладок **Конструктор** и **Упорядочить**.

ПОРЯДОК СОЗДАНИЯ ФОРМ

Сначала с помощью соответствующего мастера форм создаётся заготовка формы нужного вида, затем в конструкторе форм в неё добавляются необходимые надписи, кнопки, списки и другие элементы управления (ЭУ) и оформления (Рисунок 56).

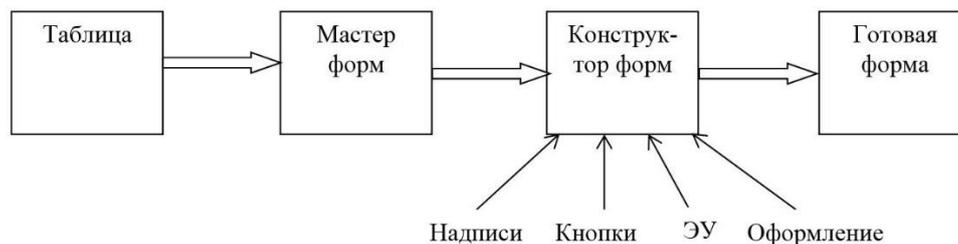


Рисунок 56 – Порядок создания формы

Результат преобразования формы в один столбец, созданной мастером, в форму для ввода и редактирования показан ниже (Рисунок 57).

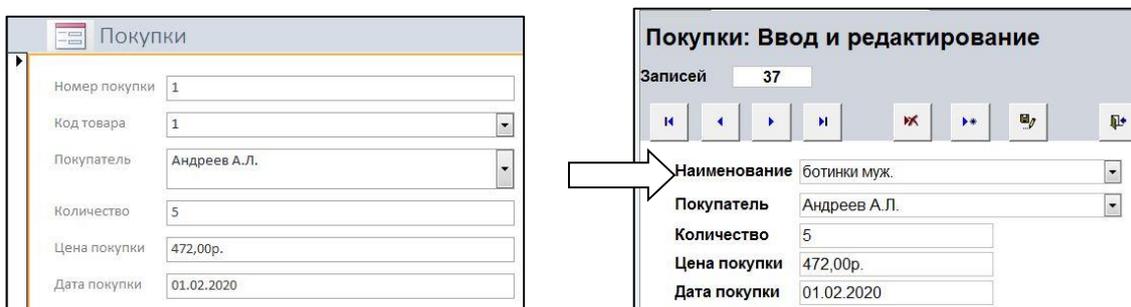


Рисунок 57 – Форма для ввода и редактирование покупок

В этих формах недопустимы кодовые поля, если они не имеют смысла.

На стороне «один» кодовые поля в форме для ввода и редактирования должны быть удалены. На стороне «много» кодовые поля заменяются полем со списком, источником для которого является смысловое поле на стороне «один». На стороне «много» смысловые поля заменяются полем со списком, источником для которого является смысловое поле на стороне «один».

Для логических полей ставится элемент управления «флажок».

Для полей, имеющих небольшой набор фиксированных значений (2-5), используют элемент управления «переключатель».

Для полей, имеющих несколько фиксированных значений, используют элемент управления «поле со списком фиксированных значений».

Результат преобразования ленточной формы в форму для просмотра показан ниже (Рисунок 58).

Покупатель	Дата покупки	Count-Наименование	Sum-Стоимость товаров
Андреев А.Л.	01.02.2020	1	2 076,80р.
Андреев А.Л.	01.05.2020	3	12 700,16р.
Андреев А.Л.	21.05.2020	1	668,73р.
Андреев А.Л.	01.09.2020	2	1 832,58р.
Андреев А.Л.	21.09.2020	1	11 211,20р.
ООО "Альберт"	01.03.2020	2	6 050,88р.

Покупатель	Дата покупки	Количество товаров	Стоимость покупки
Андреев А.Л.	01.02.2020	1	2 076,80р.
Андреев А.Л.	01.05.2020	3	12 700,16р.
Андреев А.Л.	21.05.2020	1	668,73р.
Андреев А.Л.	01.09.2020	2	1 832,58р.
Андреев А.Л.	21.09.2020	1	11 211,20р.
ООО "Альберт"	01.03.2020	2	6 050,88р.

Рисунок 58 – Форма для просмотра Стоимость покупок

В этих формах недопустимы кодовые поля, если они не имеют смысла. На стороне «много» эти поля заменяют смысловыми полями со стороны «один», на стороне «один» такие поля просто удаляют, недопустимы также поля со списками – они заменяются простыми полями.

В формах для просмотра запрещается добавление, изменение и удаление записей.

Подчиненная форма создается из двух связанных таблиц

Результат преобразования подчиненной формы из мастера в подчиненную форму для просмотра показан ниже (Рисунок 59).

Наименование	Ботинки муж.	Скидка %	
Категория	обувь		
Описание	зимние, на меху, размеры 41-43		
Производитель	АО "Подорова"		
Дата поставки	01.02.2013		
Количество	15		пар
Цена поставки	400,00р.		
Единица измере	кг		

Покупатель	Количество	Цена покупки	Дата покупки
Андреев А.Л.	5	400,00р.	01.02.2013
ООО "Альберт"	3	400,00р.	01.03.2013

Рисунок 59 – Подчиненная форма Товары в покупках

Для создания формы-диаграммы сначала с помощью мастера **Пустая форма (Создание → Формы → Пустая форма)** создаётся пустая форма, затем в конструкторе в неё добавляются элемент управления **Диаграмма**. Далее в мастере диаграмм выбираем источник, поля источника, вид и имя диаграммы (Рисунок 60).

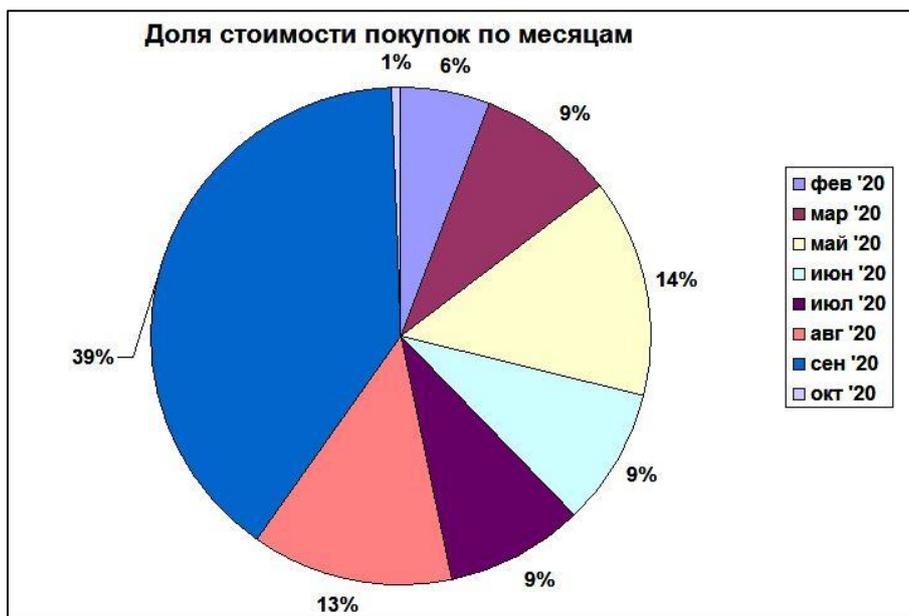


Рисунок 60 – Форма-диаграмма

СВОЙСТВА ФОРМЫ

Свойства формы позволяют управлять многими её параметрами.

Для доступа к свойствам формы нужно в конструкторе выделить всю форму и использовать правую кнопку мыши или команду **Страница свойств**. Получим список свойств формы (Рисунок 61).

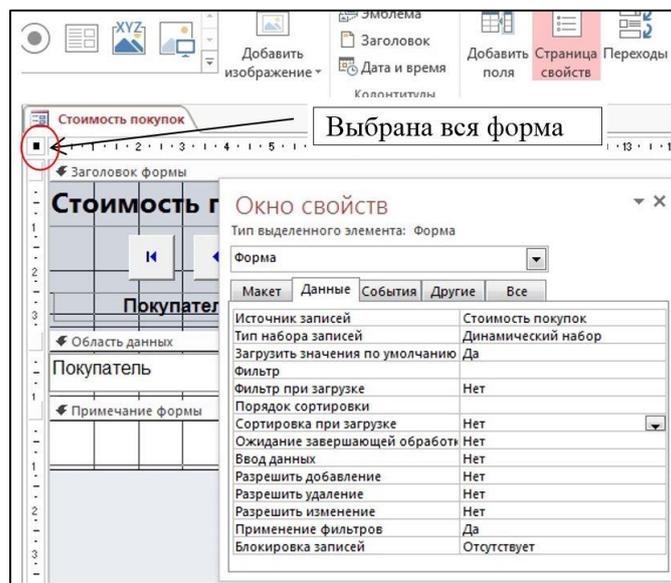


Рисунок 61 – Свойства формы

Набор свойств меняются в зависимости от вида формы.

На вкладке **Данные** окна свойств мы можем изменить источник данных формы (свойство **Источник записей**), использовать при загрузке фильтр (свойства **Фильтр** и **Фильтр при загрузке**), задавать сортировку (свойства **Порядок сортировки** и **Сортировка при загрузке**) а также определить режим работы с данными в форме через свойства **Разрешить добавление**, **Разрешить удаление**, **Разрешить изменение**. В форме можно также выполнить фильтрацию, используя свойства **Фильтр** и **Применение фильтров**.

На вкладке **Макет** окна свойств мы можем управлять видом формы на экране через свойства **Область выделения**, **Кнопки перехода** и другие.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите источники форм.
2. Перечислите средства для создания форм в Access.
3. Из каких разделов состоит форма?
4. Как можно использовать свойства форм?

7. ОТЧЕТЫ

НАЗНАЧЕНИЕ ОТЧЕТОВ

Отчет служит для представления сведений из базы данных в нужном виде на экране или печати.

В отчет могут включаться только необходимые записи и колонки (поля), а также добавляться промежуточные и окончательные итоги. Можно также использовать самые разнообразные способы размещения элементов отчета: названий, записей, колонок, итогов.

Отчет - один из объектов БД, один раз созданное описание отчета может многократно использоваться для получения отчета.

Отчет создается в два этапа:

- создается структура отчета, т.е. описываются все элементы отчета - поля, сортировки, итоги, названия и прочее;
- выдается сам отчет в режиме предварительного просмотра, затем готовый отчет можно распечатать или направить в Word или Excel.

Отчет можно получить следующими способами:

- ленточный отчет с помощью команды **Отчет**;
- использовать **Мастер отчетов** для построения типовых отчетов различного вида;
- создавать отчет любого вида с помощью конструктора.

Первый способ самый быстрый и позволяет формировать только очень простые отчеты, второй способ позволяет формировать отчеты только определенного вида, третий способ значительно сложнее, но позволяет создавать и редактировать отчеты любого вида.

ИСТОЧНИК ОТЧЕТА

Источником отчета может быть таблица или запрос (Рисунок 62). Желательно иметь один источник отчета. Если исходные данные для отчета находятся в разных таблицах или запросах, то сначала необходимо собрать эти данные в одном запросе, а затем использовать его в качестве источника отчета.

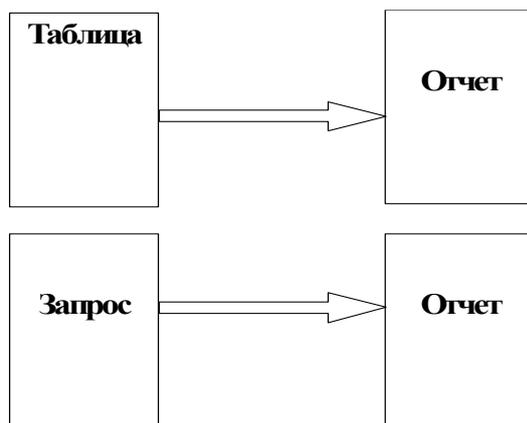


Рисунок 62 – Источники отчета

ВИДЫ ОТЧЕТОВ

Существует несколько видов отчета в зависимости от способа размещения данных (записей) в них.

Отчет в один столбец

В таком отчете все поля каждой записи источника располагаются в одну колонку, друг под другом (Рисунок 63).

Товары	
Наименование	бокорезы
Категория	одежда
Описание	
Производитель	ЗАО "Инструмент"
Дата поставки	02.04.2020
Количество	14
Цена поставки	295,00р.
Единица измерения	шт
Скидка %	0,00%
Наименование	ботинки муж.
Категория	обувь
Описание	зимние, на меху, размеры 41-43
Производитель	АО "Подосва"
Дата поставки	01.02.2020
Количество	15
Цена поставки	400,00р.
Единица измерения	пар
Скидка %	12,00%

Рисунок 63 – Отчет в один столбец

Отчет в один столбец можно разделить на страницы таким образом, чтобы на каждой странице размещалась только одна запись.

Ленточный отчет

В таком отчете все поля каждой записи источника располагаются в одну строчку, друг за другом (Рисунок 64).

Товары						
Наименование	Категория	Производитель	Дата поставки	Количество	Единица измерения	Цена поставки
уровень	инструмент	ОАО "Строим сами"	01.01.2020	10	шт	290,00р
ботинки муж.	обувь	АО "Подосва"	01.02.2020	15	пар	400,00р
сланцы муж.	обувь	АО "Подосва"	01.07.2020	120	пар	1 200,00р
ветровка	одежда	ИП Рубашкин	01.01.2020	15	шт	980,00р
бокореzy	одежда	ЗАО "Инструмент"	02.04.2020	14	шт	295,00р
стакан	посуда	ОАО "Бородинский стекольный завод"	12.04.2020	120	шт	25,00р
сервиз чайный	посуда	ЧП "Фарфор"	01.05.2020	8	шт	3 800,00р
сахар рафинад	продукты	ООО "Ангарес"	01.01.2020	500	кг	60,50р
сахар	продукты	ООО "Ангарес"	12.04.2020	200	кг	41,00р

Рисунок 64 – Ленточный отчет

Отчет с группировкой и итогами

По виду такой отчет напоминает ленточный, но в нем записи сгруппированы по одному или нескольким полям, подведены итоги в каждой группе и общие итоги (Рисунок 65).

Стоимость поставленных товаров по производителям			
Наименование	Цена поставки	Количество	Стоимость
Производитель АО "Подосва"			
ботинки муж.	400,00р.	15	6 000,00 Р
сланцы муж.	1 200,00р.	120	144 000,00 Р
		Всего	150 000,00 Р
Производитель ЗАО "Инструмент"			
бокореzy	295,00р.	14	4 130,00 Р
		Всего	4 130,00 Р
Производитель ИП Рубашкин			
ветровка	980,00р.	15	14 700,00 Р
		Всего	14 700,00 Р
Производитель ОАО "Бородинский стекольный"			
стакан	25,00р.	120	3 000,00 Р
		Всего	3 000,00 Р

Рисунок 65 – Отчет с группировкой и итогами

Необходимыми элементами оформления отчетов являются:

- название отчета;
- сортировка строк отчета (чаще всего по первой колонке);
- нумерация строк отчета;
- при необходимости, расчерчивание строк отчета.

СРЕДСТВА ДЛЯ СОЗДАНИЯ ОТЧЕТОВ

Все средства для создания отчетов отображены на вкладке **Создание** в группе команд **Отчеты** (Рисунок 66).



Рисунок 66 – Команды для создания отчетов

Мастера для создания отчетов

Команда **Отчет** создает ленточный отчет (Рисунок 67).

Стоимость товаров		22 апреля 2021 г. 12:53:58		
Наименование	Производитель	Цена поставки	Количество	Стоимость
ботинки муж.	АО "Подолва"	400,00р.	15	6 000,00 Р
дверь	ОАО "Строим сами"	1 820,00р.	20	36 400,00 Р
ванна	ОАО "Строим сами"	3 700,00р.	9	33 300,00 Р
утеплитель	ОАО "Строим сами"	150,00р.	200	30 000,00 Р
сахар	ООО "Антарес"	41,00р.	200	8 200,00 Р
сланцы муж.	АО "Подолва"	1 200,00р.	120	144 000,00 Р
плитка керам.	ОАО "Строим сами"	800,00р.	500	400 000,00 Р
уровень	ОАО "Строим сами"	290,00р.	10	2 900,00 Р
сахар рафинад	ООО "Антарес"	60,50р.	500	30 250,00 Р
сахар тростниковый	ООО "Антарес"	80,50р.	300	24 150,00 Р
рис	ООО "Антарес"	95,00р.	100	9 500,00 Р

Рисунок 67 – Ленточный отчет, созданный мастером

Команда **Пустой отчет** создает пустой отчет, он может применяться для создания диаграмм.

Команда **Мастер отчетов** позволяет создавать отчеты в один столбец, ленточные отчеты и отчеты с группировкой и итогами. При этом можно выбрать источник отчета и поля, задать группировку, вид итогов в отчете с группировкой, сортировку, изменить стиль и название отчета.

Ниже приведен пример отчета в один столбец (Рисунок 68) и отчета с группировкой и итогами (Рисунок 69), созданные этим мастером.

Покупатели	
Покупатель	Андреев А.Л.
Адрес	Комсомольская 25-1
Телефон	89992346572
Физическое лицо	<input checked="" type="checkbox"/>
Покупатель	ООО "Альберт"
Адрес	пер. Длинный 45-12
Телефон	88126532745
Физическое лицо	<input type="checkbox"/>

Рисунок 68 – Отчет в один столбец, созданный мастером

Стоимость товаров по производителям				
Производитель	Наименование	Цена поставки		Стоимость
АО "Подосва"	ботинки муж.	400,00р.	15	6 000,00 Р
	сланцы муж.	1 200,00р.	120	144 000,00 Р
ЗАО "Инструмент"	бокореzy	295,00р.	14	4 130,00 Р
	ветровка	980,00р.	15	14 700,00 Р
ИП Рубашкин	ветровка	980,00р.	15	14 700,00 Р
ОАО "Бородинский стеко"	стакан	25,00р.	120	3 000,00 Р

Рисунок 69 – Отчет с группировкой и итогами (фрагмент)

Конструктор отчетов

В конструкторе можно создать и отредактировать любой отчет. Чтобы не начинать создание отчета «с нуля», обычно сначала создают отчет с помощью мастера, а затем редактируют его в конструкторе. Конструктор отчетов вызывается командой:

Вкладка **Создание** → Группа команд **Отчеты** → Команда **Конструктор отчетов**.

Отчет в конструкторе состоит из разделов **Заголовок и примечание отчета**, **Область данных**, **Верхний и нижний колонтитулы**, **Заголовок и примечание группы**. В разделах отчета размещают данные (поля источника, свободные поля), подписи (Рисунок 70). Обычно в отчетах не размещают такие элементы управления, как кнопки, поля со списками, т.к. в отчете они не работают, а выводить их на печать бессмысленно.

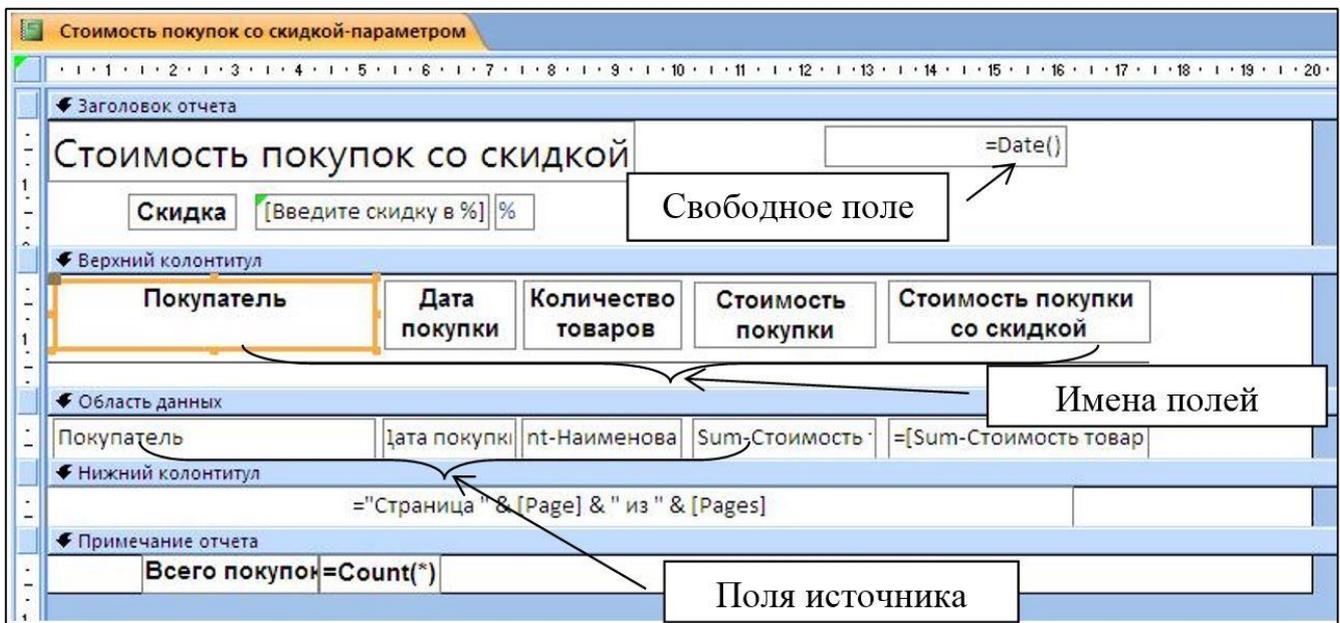


Рисунок 70 – Отчет с группировкой и итогами в конструкторе

Команды для работы с отчетом в конструкторе находятся в группе вкладок **Инструменты конструктора отчетов**, состоящей из четырех вкладок **Конструктор**, **Упорядочить**, **Формат** и **Параметры страницы**.

ПОРЯДОК СОЗДАНИЯ ОТЧЕТОВ

Отчет в один столбец

Сначала с помощью **Мастера отчетов** (**Создание** → **Отчеты** → **Мастер отчетов**) создаётся по шагам базовый отчет, затем в конструкторе отчетов в него добавляются необходимые надписи, свободные поля, линии, подбирается оптимальная ширина колонок, меняют, при необходимости, оформление (Рисунок 71).

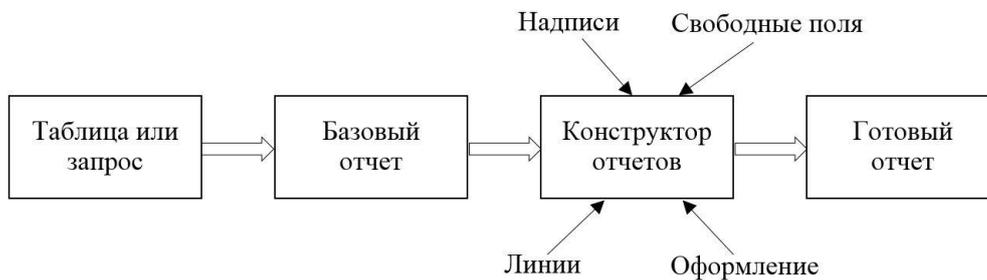


Рисунок 71 – Порядок создания отчета

Результат преобразования в конструкторе отчета в один столбец, созданного **Мастером отчетов**, показан ниже (Рисунок 72).



Рисунок 72 – Отчет в один столбец, преобразованный в конструкторе
 Результат преобразования ленточного отчета в конструкторе показан ниже (Рисунок 73).

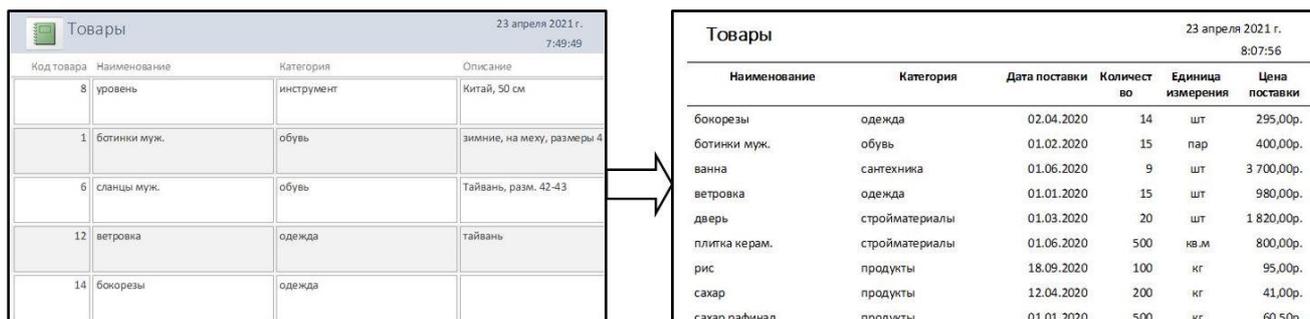


Рисунок 73 – Ленточный отчет, созданный мастером и преобразованный в конструкторе

Отчет с группировкой и итогами создается **Мастером отчетов**. В отчете данные сгруппированы по одному или нескольким полям и подведены итоги в группе и общие итоги. Порядок создания отчета с группировкой и итогами такой же, как и ленточного отчета. На втором шаге мастера выбирается поле группировки, на третьем шаге – сортировку и итоговые функции. Затем в конструкторе отчетов в него добавляются необходимые надписи, свободные поля, линии, подбирается оптимальная ширина колонок, меняют, при необходимости, оформление (Рисунок 74).

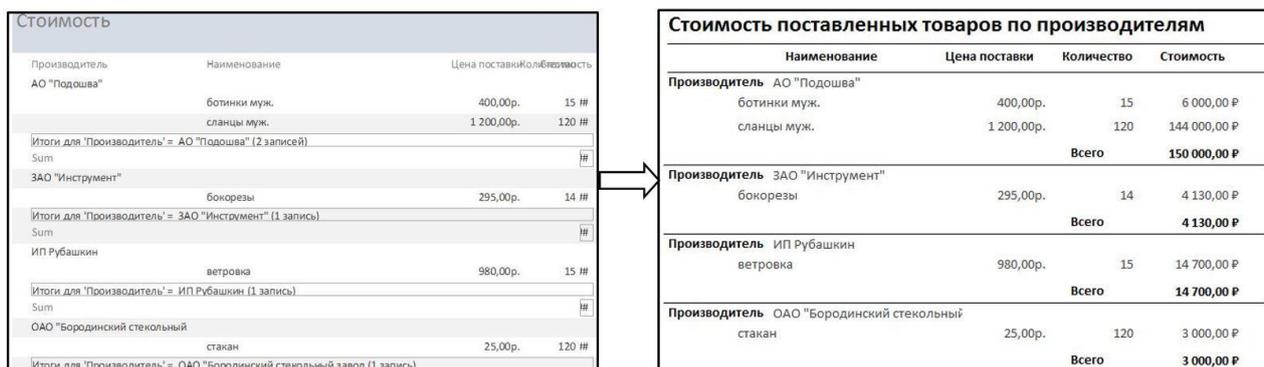


Рисунок 74 – Отчет с группировкой и итогами, исправленный в конструкторе

ОТЧЕТ С ВЫЧИСЛЯЕМЫМИ ПОЛЯМИ

Сначала с помощью мастера создаём ленточный отчет по таблице **Товары**. затем в конструкторе добавляем свободные поля для подсчета стоимости в каждой записи в области данных (формула $=[\text{Цена поставки}] * [\text{Количество}] * (1 - [\text{Скидка \%}])$) и общей стоимости в примечании

отчета (формула $=\text{Sum}([\text{Цена поставки}]*[\text{Количество}]*(1-[\text{Скидка \%}])))$ в примечании отчета (Рисунок 75).

The screenshot shows a report builder interface with the following sections:

- Заголовок отчета:** Contains the title "Стоимость поставленных товаров" and a calculated field $=\text{Now}()$.
- Верхний колонтитул:** Contains the title "Стоимость поставленных товаров".
- Область данных:** Contains a table with columns: №, Наименование, Категория, Производитель, Количество, Цена поставки, and Стоимость поставки. The "Цена поставки" column has a calculated field $=[\text{Цена поставки}]$.
- Нижний колонтитул:** Contains a page number field $=\text{"Стр. " & [Page] & " из " & [Pages]}$.
- Примечание отчета:** Contains a total row with the formula $=\text{Sum}([\text{Цена пос}])$.

Callouts point to the $=\text{Now}()$ field and the $=[\text{Цена поставки}]$ field, both labeled "Вычисляемое поле в области данных". Another callout points to the total row formula, labeled "Вычисляемые итоги в примечании отчета".

Рисунок 75 – Отчет с добавленными вычисляемыми полями в конструкторе
Получим отчет о стоимости товаров с вычисляемыми полями (Рисунок 76).

Стоимость поставленных товаров						23 апреля 2021 г.
№	Наименование	Категория	Производитель	Количество	Цена поставки	Стоимость поставки
1	бокорезы	одежда	ЗАО "Инструмент"	14	295,00р.	4 130,00 Р
2	ботинки муж.	обувь	АО "Подошва"	15	400,00р.	6 000,00 Р
3	ванна	сантехника	ОАО "Строим сами"	9	3 700,00р.	33 300,00 Р
4	ветровка	одежда	ИП Рубашкин	15	980,00р.	14 700,00 Р
5	дверь	стройматериалы	ОАО "Строим сами"	20	1 820,00р.	36 400,00 Р
6	плитка керам.	стройматериалы	ОАО "Строим сами"	500	800,00р.	400 000,00 Р
7	рис	продукты	ООО "Ангарес"	100	95,00р.	9 500,00 Р
8	сахар	продукты	ООО "Ангарес"	200	41,00р.	8 200,00 Р
9	сахар рафинад	продукты	ООО "Ангарес"	500	60,50р.	30 250,00 Р
10	сахар тростниковый	продукты	ООО "Ангарес"	300	80,50р.	24 150,00 Р
11	сервиз чайный	посуда	ЧП "Фарфор"	8	3 800,00р.	30 400,00 Р
12	сланцы муж.	обувь	АО "Подошва"	120	1 200,00р.	144 000,00 Р
13	стакан	посуда	ОАО "Бородинский"	120	25,00р.	3 000,00 Р
14	уровень	инструмент	ОАО "Строим сами"	10	290,00р.	2 900,00 Р
15	утеплитель	хозтовары	ОАО "Строим сами"	200	150,00р.	30 000,00 Р
Итого						776 930,00 Р

Рисунок 76 – Отчет с вычисляемыми полями

ОТЧЕТ С ПАРАМЕТРОМ

Если в элемент управления **Поле** ввести текст подсказки в прямоугольных скобках, не являющийся именем поля источника отчета, то этот текст становится параметром, т.е. именем, которое получает значение в результате ввода параметра.

Используем параметр для построения отчета о цене товаров с учетом скидки. В заголовке отчета создадим свободное поле и впишем в него параметр $=[\text{Введите скидку в \%}]$

Используем тот же параметр для вычисления стоимости покупок со скидкой в области данных $=[\text{Цена поставки}]*(\text{1}-[\text{Введите скидку в \%}]/100)$ (Рисунок 77):

Рисунок 77 - Отчет с параметром в конструкторе

При открытии отчета сначала появляется окно ввода параметра, а после ввода параметра открывается отчет (Рисунок 78).

Товары со скидкой		Скидка, %	10	23 апреля 2021 г.		
№	Наименование	Категория	Количество	Единица измерения	Цена поставки	Цена со скидкой
1	бокорезы	одежда	14	шт	295,00р.	265,50 Р
2	ботинки муж.	обувь	15	пар	400,00р.	360,00 Р
3	ванна	сантехника	9	шт	3 700,00р.	3 330,00 Р
4	ветровка	одежда	15	шт	980,00р.	882,00 Р
5	дверь	стройматериалы	20	шт	1 820,00р.	1 638,00 Р
6	плитка керам.	стройматериалы	500	кв.м	800,00р.	720,00 Р
7	рис	продукты	100	кг	95,00р.	85,50 Р
8	сахар	продукты	200	кг	41,00р.	36,90 Р
9	сахар рафинад	продукты	500	кг	60,50р.	54,45 Р
10	сахар тростниковый	продукты	300	кг	80,50р.	72,45 Р
11	сервиз чайный	посуда	8	шт	3 800,00р.	3 420,00 Р
12	сланцы муж.	обувь	120	пар	1 200,00р.	1 080,00 Р
13	стакан	посуда	120	шт	25,00р.	22,50 Р
14	уровень	инструмент	10	шт	290,00р.	261,00 Р
15	утеплитель	хозтовары	200	пог.м	150,00р.	135,00 Р

Рисунок 78 - Отчет с параметром

ИСПОЛЬЗОВАНИЕ СВОЙСТВ ОТЧЕТА

Свойства отчета позволяют управлять многими параметрами любого элемента отчета, раздела отчета и всего отчета. Для доступа к свойствам элемента отчета, раздела отчета или всего

отчета нужно в конструкторе выделить необходимую часть отчета и использовать правую кнопку мыши или выполнить команду **Страница свойств**.

В свойствах отчета можно увидеть источник отчета и при необходимости изменить его, включить сортировку, выполнить фильтрацию.

Управляя свойствами раздела отчета можно разделить отчет на страницы, используя свойство **Конец страницы** на вкладке **Макет**.

В свойствах элемента отчета используя свойство **Расширение** можно разрешить расширение данных в поле, которое позволит длинные тексты отображать в несколько строк (высота поля автоматически подстраивается под размер данных в этом поле), изменить формат поля, выполнить нумерацию строк в отчете. Для нумерации строк в отчете создаем свободное поле в начале строки отчета и вводим туда выражение $\boxed{=1}$. Затем в свойстве этого поля **Сумма с накоплением** выбираем **Для всего** если в отчете нет группировки или **Для группы** в отчете с группировкой.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите источники отчета.
2. Перечислите виды отчетов.
3. Какие средства для создания отчетов существуют в Access?
4. Как можно использовать свойства отчета?

ЗАКЛЮЧЕНИЕ

Появление реляционных СУБД стало важным шагом вперед по сравнению с иерархическими и сетевыми СУБД, в этих системах стали использоваться непроцедурные языки манипулирования данными и была достигнута значительная степень независимости данных от обрабатывающих программ. В то же время, существует ряд недостатков реляционных систем, обусловленных самой моделью:

- Реляционная модель данных не допускает естественного представления данных со сложной (иерархической) структурой, поскольку в ее рамках возможно моделирование лишь с помощью плоских отношений (таблиц). Все отношения принадлежат одному уровню, многие значимые связи между данными либо теряются, либо их поддержку приходится осуществлять в рамках конкретной прикладной программы.
- По определению в реляционной модели поля кортежа могут содержать лишь атомарные значения. Однако, в таких приложениях как САПР (системы автоматизированного проектирования), ГИС (геоинформационные системы), искусственный интеллект системы оперируют со сложно – структурированными объектами.
- Кроме того, даже в том случае, когда сложный объект удастся «уложить» в реляционную базу данных, его данные распределяются, как правило, по многим таблицам. Соответственно, извлечение каждого такого объекта требует выполнения многих операций соединения (join), что значительно замедляет работу СУБД. Обойти это и предыдущее ограничения можно было бы в том случае, если бы реляционная модель допускала: возможность определения новых типов данных и определение наборов операций, связанных с данными определенного типа.

Тема баз данных довольно необъятна и, несмотря на отмеченные недостатки, реляционные базы данных имеют широкое применение, и для изучения основ этого предмета данное пособие может быть использовано студентами, так как даёт представление о предмете и позволяет в короткие сроки освоить создание простого, но законченного продукта в форме информационной системы и облегчить дальнейшее изучение данной области.

СПИСОК ЛИТЕРАТУРЫ

ОСНОВНАЯ

1. Голицына, О.Л. Базы данных: учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - Изд. 2-е, испр. и доп. - М. : ФОРУМ : ИНФРА-М, 2007. - 399 с. - (Профессиональное образование). - ISBN 978-5-91134-098-8, ISBN 978-5-16-002966-5.
2. Хомоненко, А.Д. Базы данных: учебник для вузов / А.Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев ; под ред. А.Д. Хомоненко. - 6-е изд. - М.: Бином-Пресс ; СПб. : КОРОНА-Век, 2007. - 736 с. - ISBN 5-7931-0349-х.
3. Базы данных: методические указания к лабораторным работам / сост. А.Н. Поручиков; Федер. агентство по образованию, Самар. гос. аэрокосм. ун-т им. академика С.П. Королева. - Самара: Изд-во СГАУ, 2008. - 28 с.
4. Базы данных: методические указания к курсовому проекту / сост. А.Н. Поручиков; Федер. агентство по образованию, Самар. гос. аэрокосм. ун-т им. академика С.П. Королева. - Самара: Изд-во СГАУ, 2008. - 50 с.
5. Проектирование и реализация баз данных средствами СУБД ACCESS: методические указания / сост. Е.И. Чигарина ; Федер. агентство по образованию, Самар. гос. аэрокосм. ун-т им. академика С.П. Королева. - Самара: Изд-во СГАУ, 2009. - 43 с.
6. Общие требования к учебным текстовым документам. Стандарт предприятия СТО СГАУ 02068410-004-2018.

ДОПОЛНИТЕЛЬНАЯ

1. Проектирование баз данных. СУБД Microsoft Access: учебное пособие для вузов / Н.Н. Гринченко, Е.В. Гусев, Н.П. Макаров [и др.]. - М.: Горячая линия-Телеком, 2004. - 240 с.
2. Степанов, В.В. Microsoft Access 2003 для начинающих / В.В. Степанов. - М.: ООО «Аквариум-Принт», 2004. - 128 с.
3. Золотова, С.И. Практикум по Access : подготовительный курс, предваряющий более глубокое изучение технологии баз данных / С.И. Золотова. - М. : Финансы и статистика, 2007. - 143 с. - (Диалог с компьютером). - ISBN 978-5-279-02284-7.
4. Диго, С.М. Access: учебно-практическое пособие / С.М. Диго. - М.: ТК Велби, Изд-во Проспект, 2006. - 240 с.

ПРИЛОЖЕНИЕ А. ТИПЫ И РАЗМЕРЫ ПОЛЕЙ

Тип поля	Описание	Размер поля
Текстовый	Текст и числа, например, имена и адреса, номера телефонов и почтовые индексы	Текстовое поле может содержать до 255 символов.
Числовой	Общий тип для числовых данных, допускающих проведение математических расчетов, за исключением расчетов для денежных значений. Свойство Размер поля позволяет указать различные типы числовых данных	<ol style="list-style-type: none"> 1. Байт (числа от 0 до 255 без дробной части) 2. Целое (Числа от -32 768 до 32 767 без дробной части). 3. Длинное целое (Числа от -2 147 483 648 до 2 147 483 647 без дробной части). 4. С плавающей точкой (4 байт). Числа от $-3,402823 \cdot 10^{38}$ до $3,402823 \cdot 10^{38}$, до 7 значащих цифр 5. С плавающей точкой (8 байт). Числа от $-1,797 \cdot 10^{308}$ до $1,797 \cdot 10^{308}$, до 15 значащих цифр
Дата/время	Значения даты и времени. Пользователь имеет возможность выбрать один из многочисленных стандартных форматов или создать специальный формат	
Денежный	Денежные значения. Не рекомендуется использовать для проведения денежных расчетов значения, принадлежащие к числовому типу данных, так как последние могут округляться при расчетах. Значения типа "Денеж-	

Тип поля	Описание	Размер поля
	ный" всегда выводятся с указанным числом десятичных знаков после запятой	
Счетчик	Автоматически вставляющиеся последовательные номера. Нумерация начинается с 1. Поле счетчика удобно для создания ключа. Это поле является совместимым с полем числового типа, для которого в свойстве Размер поля "Длинное целое"	
Логический	Значения "Да"/"Нет", "Истина"/"Ложь", "Вкл"/"Выкл", т.е. одно из двух возможных значений	
Поле Мемо	Длинный текст и числа, например, комментарии и пояснения	Мемо-поле может содержать до 64 000 символов
Поле объекта OLE	Объекты, созданные в других программах, поддерживающих протокол OLE. Объекты связываются или внедряются в базу данных Microsoft Access через элемент управления в форме или отчете	

ПРИЛОЖЕНИЕ Б. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ФУНКЦИЙ

Функции используются в вычисляемых полях в запросах, формах, отчетах.

1. Использование функций для данных типа дата/время:

- получение номера дня недели из даты: `№ дня недели: Weekday([Дата покупки];0)` Здесь второй аргумент функции 0 означает, что нумерация дней в неделе начинается с понедельника;
- получение названия дня недели из его номера: `День месяца: Choose([№ дня недели]; "понедельник"; "вторник"; "среда"; "четверг"; "пятница"; "суббота"; "воскресенье")`. Функция Choose по номеру дня выбирает текстовое значение из списка;
- получение номера месяца из даты `№ месяца: Month([Дата покупки])`;
- получение названия месяца из его номера `Месяц: Choose([№ месяца]; "январь"; "февраль"; "март"; "апрель"; "май"; "июнь"; "июль"; "август"; "сентябрь"; "октябрь"; "ноябрь"; "декабрь")`;
- получение года из даты `Year([Дата покупки])`.

2. Использование функций для данных логического типа:

- преобразование значения логического поля (да/нет, истина/ложь, флажок) в текст
`Статус лица: If([Физическое лицо];"физическое лицо";"юридическое лицо")`, `Посещение занятий: If([Посещение];"+";"н")`. Здесь значения логических полей [Физическое лицо], [Посещение] преобразуются в соответствующие текстовые значения).
- Обратите внимание на то, что сортировка в запросах, формах, отчетах по дням недели или месяцам производится по номеру, а не по названию!

Учебное издание

*Крикунов Михаил Михайлович,
Поручиков Алексей Николаевич*

ОСНОВЫ БАЗ ДАННЫХ

Учебное пособие

Редактор А.В. Ярославцева
Компьютерная верстка А.В. Ярославцевой

Подписано в печать 13.12.2021. Формат 60×84/8.

Бумага офсетная. Печ. л. 10,5.

Тираж 25 экз. Заказ № . Арт. – 7(РЗУ)/2021.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
443086, Самара, Московское шоссе, 34.

Издательство Самарского университета.
443086, Самара, Московское шоссе, 34.