



Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский государственный аграрный университет – Московская сельскохозяйственная академия имени К. А. Тимирязева»

Институт мелиорации, водного хозяйства и строительства имени А. Н. Костякова

М. В. Петухова, Н. В. Гавриловская, О. Н. Ивашова, Е. В. Щедрина

ПРАКТИКУМ ПО ДИСЦИПЛИНЕ «БАЗЫ ДАННЫХ»

Учебно-методическое пособие



Москва, 2023

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА
РОССИЙСКОЙ ФЕДЕРАЦИИ
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ –
МСХА ИМЕНИ К. А. ТИМИРЯЗЕВА

Институт мелиорации, водного хозяйства и строительства
имени А. Н. Костякова

Петухова Мария Владиславовна
Гавриловская Надежда Владимировна
Ивашова Ольга Николаевна
Щедрина Елена Владимировна

ПРАКТИКУМ ПО ДИСЦИПЛИНЕ «БАЗЫ ДАННЫХ»

Учебно-методическое пособие

Москва, 2023

УДК 004.65 (076.5)
ББК 32.973-018.2я73-5
П 31

Рецензенты:

Худякова Е. В., д-р экон. наук, профессор кафедры прикладной информатики
ФГБОУ ВО РГАУ-МСХА им. К. А. Тимирязева

Соболева Е. В., канд. пед. наук, доцент кафедры цифровых технологий
в образовании ФГБОУ ВО «Вятский государственный университет»

Петухова М. В., Гавриловская Н. В., Ивашова О. Н., Щедрина Е. В.
П 31 Практикум по дисциплине «Базы данных» : учебно-методическое
пособие / М. В. Петухова, Н. В. Гавриловская, О. Н. Ивашова,
Е. В. Щедрина. // ФГБОУ ВО РГАУ-МСХА имени К. А. Тимирязева. –
Москва; Саратов : Амирит, 2023. – 50 с.

ISBN 978-5-00207-393-1

Учебно-методическое пособие «Практикум по дисциплине “Базы данных”» составлено в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 20.03.01 «Техносферная безопасность». Целью пособия является поддержка изучения учебной дисциплины «Базы данных», предусмотренной учебным планом указанного направления подготовки в плане освоения принципов проектирования реляционных баз данных, их создания и эффективного использования на примере системы управления базами данных MS Access.

УДК 004.65 (076.5)
ББК 32.973-018.2я73-5

ISBN 978-5-00207-393-1

© Петухова М. В., 2023
© ФГБОУ ВО РГАУ-МСХА
имени К. А. Тимирязева, 2023

Оглавление

ВВЕДЕНИЕ	5
ЛАБОРАТОРНАЯ РАБОТА № 1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ. СОЗДАНИЕ БАЗЫ ДАННЫХ В СУБД MS ACCESS	6
ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ.....	6
<i>Общая постановка задачи.....</i>	6
<i>Описание предметной области.....</i>	6
<i>Логическая модель</i>	6
ФИЗИЧЕСКАЯ МОДЕЛЬ. СОЗДАНИЕ БАЗЫ ДАННЫХ В СУБД ACCESS	11
<i>Создание таблиц данных</i>	11
<i>Установление связей</i>	14
<i>Заполнение таблиц.....</i>	15
<i>Оптимизация ввода значений внешних ключей.....</i>	15
ЛАБОРАТОРНАЯ РАБОТА № 2 ФОРМЫ В MS ACCESS	18
РЕЖИМЫ ПРЕДСТАВЛЕНИЯ ФОРМ.....	18
СОЗДАНИЕ ФОРМ	20
<i>Создание простой формы</i>	20
<i>Создание формы, в которой отображается сразу несколько записей.....</i>	21
<i>Создание разделённой формы</i>	21
<i>Создание формы при помощи Мастера форм</i>	22
<i>Создание формы при помощи инструмента Пустая форма</i>	23
<i>Конструктор форм.....</i>	23
ЛАБОРАТОРНАЯ РАБОТА № 3 ЗАПРОСЫ В MS ACCESS	29
СОЗДАНИЕ ЗАПРОСОВ С ПОМОЩЬЮ МАСТЕРОВ	29
СОЗДАНИЕ ЗАПРОСОВ В РЕЖИМЕ КОНСТРУКТОРА	31
<i>Запросы на выборку</i>	31
<i>Запросы с параметром</i>	33
<i>Запросы с вычисляемыми полями.....</i>	34
<i>Аналитические запросы</i>	35
<i>Перекрестные запросы</i>	37
ЛАБОРАТОРНАЯ РАБОТА № 4 ОТЧЕТЫ В ACCESS.....	39
ИЗМЕНЕНИЕ ОТЧЕТОВ	39
СОЗДАНИЕ ПРОСТОГО ОТЧЕТА	40
ИСПОЛЬЗОВАНИЕ МАСТЕРА ОТЧЕТОВ.....	40
СОЗДАНИЕ ОТЧЕТОВ В РЕЖИМЕ КОНСТРУКТОРА.....	41
<i>Группировка и сортировка в отчетах</i>	41
<i>Вычисляемые поля в отчетах</i>	44
СОЗДАНИЕ ОТЧЕТА ПРИ ПОМОЩИ ИНСТРУМЕНТА ПУСТОЙ ОТЧЕТ	45
ЛАБОРАТОРНАЯ РАБОТА № 5 ГЛАВНАЯ КНОПОЧНАЯ ФОРМА.....	46
СОЗДАНИЕ КНОПОЧНОЙ ФОРМЫ И СТРАНИЦ КНОПОЧНОЙ ФОРМЫ.....	46
СОЗДАНИЕ ЭЛЕМЕНТОВ (КНОПОК) НА СТРАНИЦЕ КНОПОЧНОЙ ФОРМЫ	47
ОФОРМЛЕНИЕ КНОПОЧНОЙ ФОРМЫ	48
ЗАДАНИЕ ПО ЗАЧЕТНОЙ БАЗЕ ДАННЫХ.....	49
СПИСОК ЛИТЕРАТУРЫ.....	50

Введение

Целью дисциплины «Базы данных» является формирование у студентов направления подготовки 20.03.01 «Техносферная безопасность» компетенций, обеспечивающих знание подходов к проектированию, созданию и использованию баз данных в своей профессиональной деятельности при реализации поиска, обработки, обобщения и анализа данных.

В настоящее время практически любой человек является пользователем многих баз данных, как в своей повседневной жизни, так и в профессиональной деятельности, поэтому важно представлять внутреннюю структуру, способы создания и поддержки базы данных профессиональной направленности, методы эффективного взаимодействия с ними.

В пособии рассматриваются базовые вопросы реляционных баз данных, этапы разработки и создание реляционной базы данных в среде системы управления базами данных на примере MS Access, ее управление и настройка, а также вопросы эффективного использования баз данных.

По учебному плану на изучение дисциплины отведено 16 часов лекционных и 16 часов лабораторных занятий, форма контроля – зачет. В пособии предлагается пять лабораторных работ, из которых на первую и третью предполагается отвести по два занятия, на остальные – по одному занятию. В завершении пособия приводится обобщенное зачетное задание.

Лабораторная работа № 1.

Проектирование базы данных.

Создание базы данных в СУБД MS Access

Проектирование базы данных

Проектирование базы данных (БД) в общем случае состоит из этапов **логического** и **физического** проектирования. Во время логического проектирования разрабатывается модель базы данных, не зависящая от конкретной системы управления базами данных (СУБД), но в терминологии определенной даталогической модели, например, реляционной. В процессе физического проектирования создается структура базы данных в СУБД.

Напомним, что под **реляционной БД** понимается БД, в которой данные организованы в виде таблиц, связанных между собой. Таблицы и связи между ними организованы по определенным правилам, о которых будут пояснения по ходу лабораторных работ.

Общая постановка задачи

Организация «ЭКСПЕРТ» занимается проведением независимых экспертиз. Необходимо создать базу данных для автоматизации учета информации по следующим процессам:

- 1) поступление и оформление заказов на проведение экспертиз от заказчиков;
- 2) распределение заказов между сотрудниками на проведение экспертиз;
- 3) выполнение сотрудниками заказов.

Описание предметной области

Организация «ЭКСПЕРТ» принимает и выполняет заказы на проведение независимых экспертиз, включающих экспертизу охраны труда, экспертизу противопожарной безопасности, экспертизу промышленной безопасности, экологическую экспертизу. Организация может оказывать консультирование, а также составлять полное экспертное заключение. У организации имеются филиалы в трех городах.

Штат сотрудников включает директоров филиалов, экспертов, бухгалтеров и секретарей.

От заказчиков поступают и оформляются заказы. В выполнении каждого заказа могут участвовать несколько сотрудников в одном из следующих качеств: руководитель, исполнитель либо консультант. Каждый сотрудник может одновременно участвовать в выполнении нескольких заказов.

Логическая модель

Разработаем логическую структуру реляционной базы данных ЭКСПЕРТ для решения поставленной задачи. Для этого обычно выполняются следующие шаги.

1 шаг – определение таблиц

Таблицы могут соответствовать:

- процессам, информация о которых должна учитываться;
- объектам, которые участвуют в этих процессах.

Точнее говоря, каждая **таблица** описывает **класс объектов или процессов**. А каждая заполненная строка, которая называется **записью**, описывает один **конкретный объект или процесс**.

Имя таблицы должно быть уникально в пределах базы данных. Принято называть таблицу по названию объекта/процесса в единственном числе, например, «Студент» – если таблица содержит информацию о студентах, «Экзамен» – если таблица содержит информацию о проводимых экзаменах).

Для решения поставленной задачи в базе данных ЭКСПЕРТ нужны таблицы: Филиал, Сотрудник, Вид экспертизы, Заказчик, Заказ.

2 шаг – определение полей таблиц

Для определения столбцов таблицы требуется установить, какие факты надо отслеживать для этих объектов или процессов, какие ограничения накладываются на данные, например, формат, принадлежность значения некоторому диапазону и др.

Таким образом, **поля** (столбцы) таблицы описывают **атрибуты (характеристики, свойства)** объектов и процессов.

Для каждого поля необходимо задать:

- имя, уникальное в пределах таблицы,
- тип данных (число, текст, дата и др.),

Кроме этого, можно задать дополнительные свойства и ограничения.

В каждой таблице реляционной базы данных должен быть задан первичный ключ. **Первичный ключ** – это минимальное множество полей (часто – одно поле) таблицы, позволяющее однозначно идентифицировать записи таблицы. То есть, значения первичного ключа не должны повторяться и должны быть не пустые. Как правило, в качестве первичного ключа выбирают поля, содержащие числа или короткий текст. В каждой таблице реляционной БД должен быть один и только один первичный ключ.

Логические модели БД удобно строить в виде схем, на которых таблицы принято обозначать в виде прямоугольников с указанием имени таблицы и перечнем полей, где первичный ключ выделяется каким-либо образом, например, подчеркивается (см. рис. 1).



Рисунок 1 – Пример изображения таблицы базы данных

Для построения проектирования и создания базы данных ЭКСПЕРТ в таблице 1 описаны поля ее таблиц, а также требования к их значениям.

Таблица 1 – Описание таблиц базы данных ЭКСПЕРТ

Имя таблицы	Имена полей	Описание и требования к значениям полей
Филиал	Код филиала	Содержит уникальный код филиала согласно внутренней кодировке организации. Поле обязательно должно быть заполнено *
	Город	Содержит название города, в котором расположен филиал
	Адрес	Содержит адрес филиала
	Телефон	Содержит телефон филиала
Сотрудник	Табельный №	Содержит табельный номер сотрудника. Поле обязательно должно быть заполнено
	Фамилия	Поле обязательно должно быть заполнено
	Имя	Поле обязательно должно быть заполнено
	Отчество	Поле может быть не заполнено
	Дата рождения	Содержит дату в формате 00.00.0000 (короткий формат даты). Возраст сотрудника должен быть не менее 18 лет
	Пол	Предполагает выбор из двух возможных значений: мужской, женский
	Адрес	Содержит адрес проживания сотрудника
	Телефон	Содержит телефон сотрудника. Записывается в формате 8(000)000-00-00 (где 0 – любая цифра)
	Дата найма	Содержит дату приема сотрудника на работу в организацию. Не может быть больше текущей даты. По умолчанию устанавливается текущая дата
	Должность	Содержит название должности сотрудника в организации. Предполагает выбор из списка возможных должностей (см. описание предметной области)
Филиал	Содержит код филиала, в котором работает сотрудник, значение берется из таблицы Филиал	
Вид экспертизы	Код вида экспертизы	Содержит уникальный код вида экспертизы согласно внутренней кодировке организации. Поле обязательно должно быть заполнено
	Наименование экспертизы	Содержит одно из значений: Экспертиза охраны труда, Экспертиза противопожарной безопасности, Экспертиза промышленной безопасности, Экологическая экспертиза

* Если для поля не указано такое требование, то оно может оставаться незаполненным

Имя таблицы	Имена полей	Описание и требования к значениям полей
	Результат	Содержит описание результата экспертизы: консультация эксперта, экспертное заключение
	Стоимость типовая	Содержит типовую стоимость за описанный вид экспертизы. Указывается в рублях
Заказчик	Код заказчика	Содержит уникальный код заказчика согласно внутренней кодировке организации, коды. Поле обязательно должно быть заполнено
	Наименование заказчика	Содержит наименование организации-заказчика, если это юридическое лицо
	Контактное лицо	Содержит ФИО представителя заказчика
	Адрес	Содержит адрес заказчика
	Телефон	Содержит телефон заказчика
Заказ	№ заказа	Содержит № заказа, номера уникальны. Поле обязательно должно быть заполнено
	Заказчик	Содержит код заказчика, который будет браться из таблицы Заказчик . Поле обязательно должно быть заполнено
	Вид экспертизы	Содержит код вида экспертизы, который будет браться из таблицы Вид экспертизы . Поле обязательно должно быть заполнено
	Дата заказа	Содержит дату оформления заказа. Не может быть больше текущей даты. По умолчанию устанавливается текущая дата. Поле обязательно должно быть заполнено
	Дата выполнения	Содержит плановую дату выполнения заказа. Поле обязательно должно быть заполнено
	Коэффициент стоимости	Содержит действительное число, на которое умножается типовая стоимость экспертизы в зависимости от сложности и объема заказа

3 шаг – определение связей

При проектировании БД стараются минимизировать избыточность данных в ней. Избыточные данные приводят к увеличению размера базы, а также к нарушению целостности данных при обновлении. Одним из средств устранения избыточности являются **связи**, которые устанавливаются между таблицами.

Необходимо определить, по каким полям таблиц установить связь. При этом вводится понятие внешнего ключа. **Внешним ключом** таблицы называется поле (или множество полей), ссылающееся на первичный ключ другой таблицы. Связываемые поля должны иметь одинаковый тип данных.

Таблица, первичный ключ которой участвует в связи, называется **ссылочной (главной)**, а таблица, содержащая соответствующий внешний ключ

– **ссылающейся (подчиненной)**.

Пример связи показан на рис. 2 – в таблице **Студент** для каждого студента может быть указана группа, в которой он учится, точнее, код группы, и этот код группы ссылается на соответствующий код группы в таблице **Учебная группа**, в которой уже содержится более полная информация об этой группе. То есть, в БД не придется дублировать данные о направлении подготовки, профиле и курсе для каждого студента одной и той же группы.



Рисунок 2 – Пример связи таблиц в БД (связь «один-ко-многим»).

В реляционной БД можно реализовать три типа связей между таблицами:

- **один-к-одному**: каждая запись одной таблицы связана строго с единственной записью другой таблицы. Такой тип связей встречается редко, поскольку данные из одной таблицы можно включить в другую. Пример см. на рис. 3.

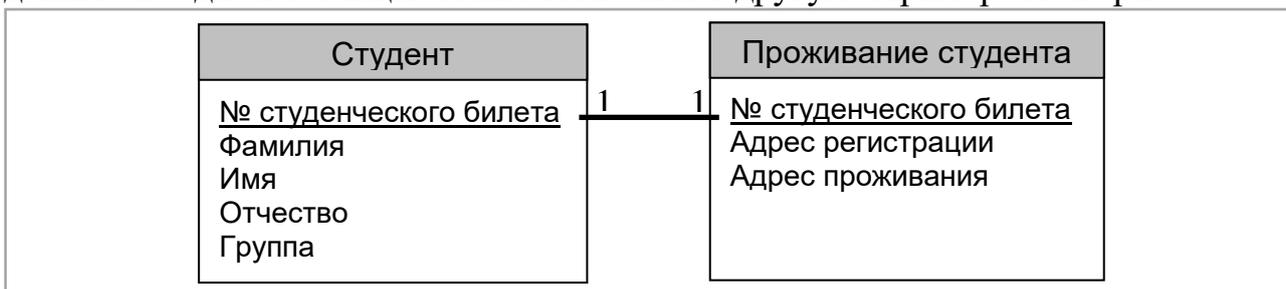


Рисунок 3 – Пример связи «один-к-одному»

- **один-ко-многим**: каждая запись главной таблицы может быть связана с несколькими записями подчиненной таблицы, а каждая запись подчиненной таблицы связана только с одной записью главной таблицы. Это наиболее распространенный тип связи. Пример см. выше на рис. 2 – для каждого студента может быть указана только одна группа (по линии связи от любого значения поля **Группа** таблицы **Студент** «приходим» только к одному коду группы в таблице **Учебная группа**), а в каждой группе учатся многие студенты (по линии связи от любого значения поля **Код группы** таблицы **Учебная группа** «приходим» к многим студентам в таблице **Студент**, для которых указан этот код группы).
- **многие-ко-многим**: каждая запись одной таблицы может быть связана с несколькими записями второй таблицы и наоборот. В реляционной СУБД такая связь реализуется разбиением ее на две связи один-ко-многим. Пример см. на рис. 4 – здесь связь описывает ситуацию, когда любой студент может принимать участие во многих студенческих организациях, и в любой студенческой организации участвует много студентов; такая связь реализована созданием связующей таблицы **Участие в организациях**, поля которой

ссылаются на первичные ключи связываемых таблиц **Студент** и **Студенческая организация**; важно, что первичный ключ связующей таблицы представляет собой множество из двух полей (важно, что это один составной первичный ключ, ведь более одного первичного ключа в таблице быть не может), то есть, неповторяющейся должна быть пара значений, а не каждое по отдельности, что продемонстрировано на рис. 4 в таблице после схемы: например, студент с номером студенческого билета 112233 является участником двух организаций, а в организации с кодом ТК есть трое участников-студентов.

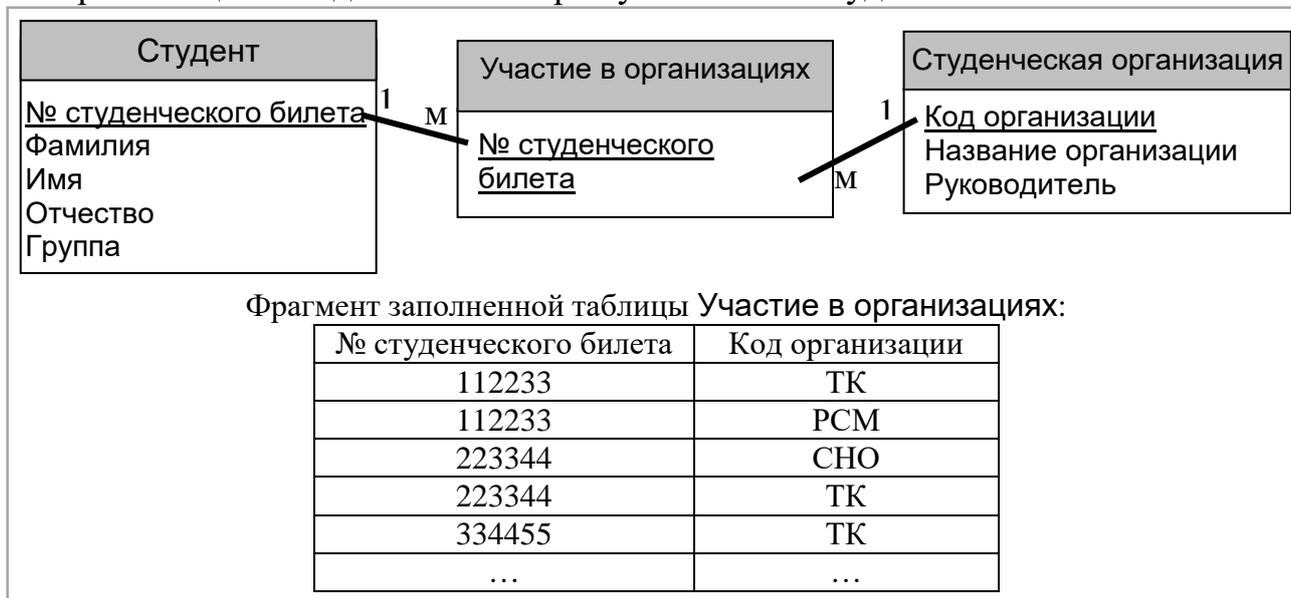


Рисунок 4 – Пример связи «многие-ко-многим».

Связи для нашей задачи проектирования БД ЭКСПЕРТ можно выявить из описания предметной области и табл. 1.

Задание 1.1. Постройте на листе схематично структуру базы данных ЭКСПЕРТ аналогично примерам, представленным на рис. 1–4. Для каждой таблицы подчеркните первичные ключи и линиями покажите связи между таблицами (концы связей пометьте 1 или М).

Обратите внимание на реализацию связи между таблицами **Сотрудник** и **Заказ** – по описанию предметной области это связь многие-ко-многим, нужна связующая таблица, которую можно назвать **Выполнение заказов**.

Физическая модель. Создание базы данных в СУБД Access

При запуске СУБД MS Access предлагает создать новую БД, запустить мастеров или открыть готовую БД. При выборе пункта **Новая база данных** система предлагает определить папку и имя файла новой БД. В процессе дальнейшей работы с БД все созданные объекты будут сохраняться в этом файле.

Создание таблиц данных

Для создания таблицы необходимо перейти на вкладку **Создание** и в группе **Таблицы** выбрать один из способов (режимов) создания таблицы: **Таблица**, **Конструктор таблиц** или **Списки SharePoint**. Можно импортировать

таблицы из других файлов – на вкладке **Внешние данные** в группе **Импорт** есть возможность выбрать источник данных.

В **режиме таблицы** пользователь сразу вводит имена полей и данные, но не может задать типы и свойства полей – они задаются по умолчанию исходя из вводимых значений.

Режим **Списки SharePoint** предполагает создание таблицы с использованием сайта, созданного в программе SharePoint.

Импорт таблиц предусматривает введение в текущую базу данных таблицы, созданной в другой БД или в другом приложении (например, в табличном процессоре Excel).

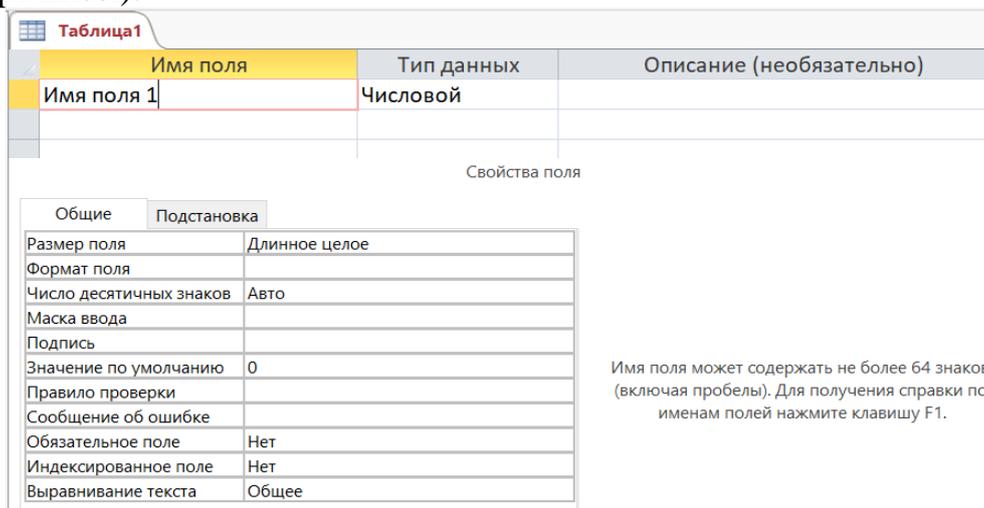


Рисунок 5 – Создание таблицы в режиме Конструктора

Наиболее детальным является режим **Конструктора** (см. рис. 5). Здесь пользователь сам осуществляет процесс определения структуры таблицы: задает имена и типы полей, их свойства (в нижней части экрана). В необязательном столбце **Описание** можно внести краткие сведения о данных. Эти сведения отображаются потом в строке состояния при выделении поля в таблице.

Для задания **первичного ключа** все поля, которые войдут в его состав, нужно выделить и определить их в качестве первичного ключа командой **Ключевое поле** на вкладке **Конструктор**, либо аналогичной командой в контекстном меню. После этого слева от имени поля появится желтый ключик.

Примечание 1. При определении полей, содержащих значения строго определенного формата, можно применить подходящий шаблон в свойстве **Маска ввода**. Например, для семизначного телефонного номера можно задать маску: 000-00-00, где 0 обозначает любую цифру.

Примечание 2. Если значения поля будут выбираться из заранее известного не слишком большого множества значений, удобно создать поле со списком, из которого при заполнении таблицы можно будет выбрать нужное значение. Для этого в разделе свойств поля на вкладке **Подстановка** нужно установить следующие значения параметров: **Тип элемента управления** – «Поле со списком» или «Список»; **Тип источника строк** – «Список значений», в параметре **Источник строк** перечислить все необходимые значения через точку с запятой (но не через запятую!). Например, для поля **Пол** можно задать список из двух значений: мужской; женский – см. рис. 6.

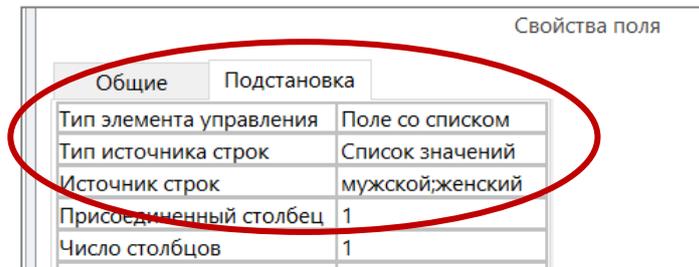


Рисунок 6 – Пример задания списка значений для поля

Примечание 3. Для задания ограничений на значения поля в виде логических выражений используют свойство **Правило проверки** (или **Условие на значение**). Выражение может включать имена полей, функции языка SQL, значения, знаки сравнения. Например, одно из условий для поля **Дата заказа** сформулировано так: «Не может быть больше текущей даты». Для задания этого условия в свойстве **Правило проверки** (**Условие на значение**) нужно вписать выражение **[Дата заказа]<=Date()**, см. рис. 7. Имена полей в выражениях записывается в квадратных скобках, функция **Date()** возвращает текущую дату, в некоторых версиях Access эта функция выглядит как **Дата()**, пустые скобки после имени функции обязательны, они указывают, что это функция, а не просто слово.

Примечание 4. Задание значения по умолчанию проиллюстрируем опять на поле **Дата заказа**, для которого сказано: «По умолчанию устанавливается текущая дата». Для реализации этого ограничения нужно в свойстве **Значение по умолчанию** записать выражение **Date()** (для некоторых версий Access – **Дата()**), см. рис. 7.

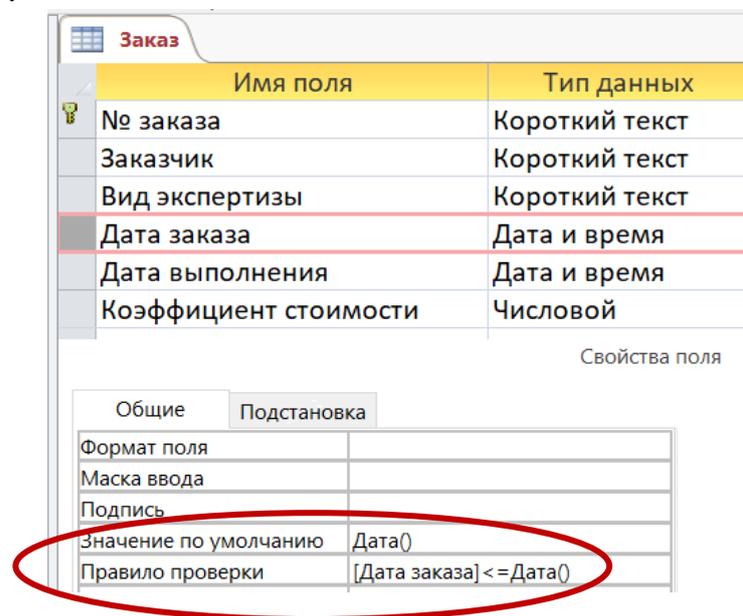


Рисунок 7 – Задание ограничения на значения поля и установка значения по умолчанию

Сохранение структуры таблицы – меню **Файл / Сохранить**, или аналогичная команда в контекстном меню созданной таблицы, или просто закрыть окно конструктора, подтвердив сохранение.

Задание 1.2. Создайте все таблицы базы данных **ЭКСПЕРТ**, самостоятельно определите и задайте типы полей, а также свойства полей согласно описанию. Обязательно назначьте для всех таблиц первичные ключи.

Установка связей

Важный этап – установка связей между таблицами. Связи устанавливаются в окне **Схема данных** (соответствующая команда на вкладке **Работа с базами данных**).

Выбирают и добавляют таблицы БД в схему в окне **Добавление таблицы** (кнопка **Отобразить таблицу** на вкладке **Конструктор** или через контекстное меню окна **Схема данных**).

Напомним, что связь может быть создана только между полями с одинаковыми типами данных.

Создание связи осуществляется в окне **Изменение связей**, которое автоматически открывается при протягивании мышью от одного связываемого поля к другому. В окне **Изменение связей** (см. рис. 8) необходимо проверить имена связываемых таблиц и полей, а также задать свойства связи.

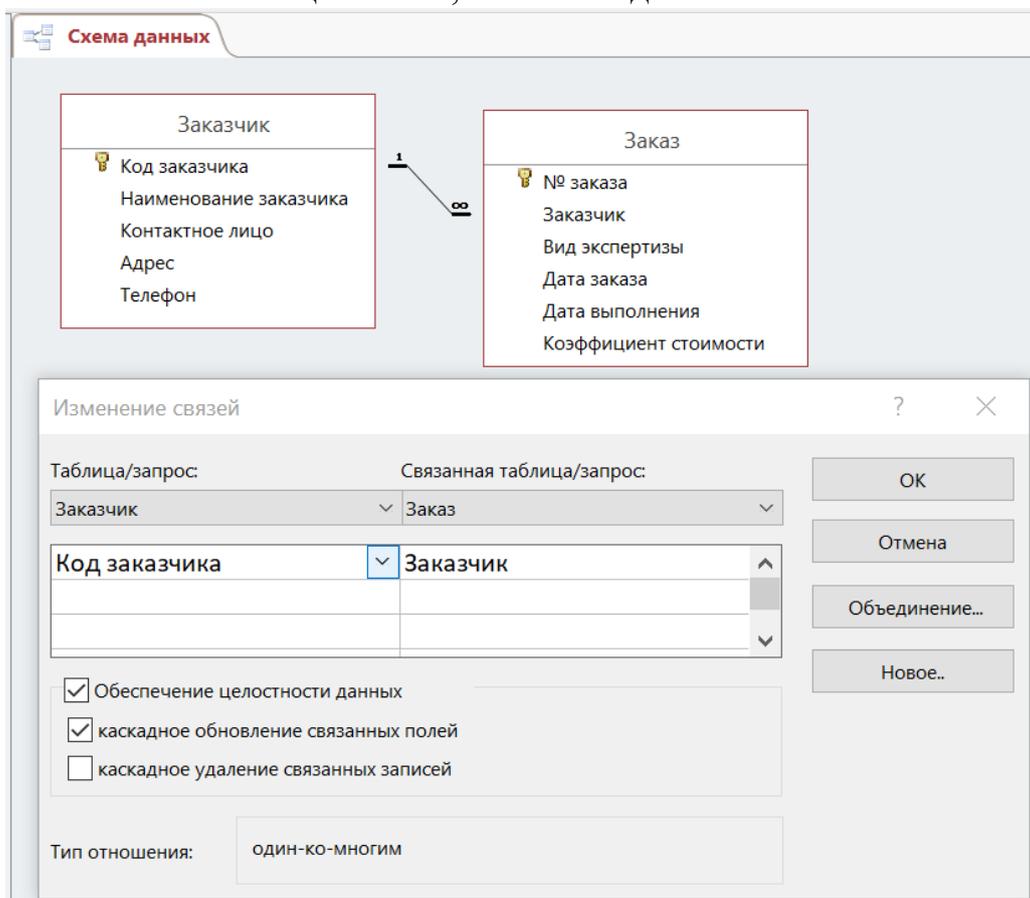


Рисунок 8 – Создание связи

Установка флажка **Обеспечение целостности данных** в окне **Изменение связей** обеспечивает запрет на ввод значений внешнего ключа в подчиненной таблице, которых нет среди значений первичного ключа в главной таблице. Для примера на рис. 8 в поле **Заказчик** таблицы **Заказ** нельзя будет ввести код заказчика, которого нет в таблице **Заказчик**.

Каскадное обновление связанных полей обеспечивает автоматическое изменение ссылающегося значения внешнего ключа, если изменяется соответствующее ему значение первичного ключа в главной таблице. Для примера на рис. 8 изменение кода заказчика в таблице **Заказчик**, повлечет за собой такие же изменения кода этого заказчика в таблице **Заказ**.

При **каскадном удалении связанных записей** после удаления записи из

главной таблицы происходит удаление из подчиненной таблицы всех записей, содержащих ссылающееся значение внешнего ключа. Для примера на рис. 8 удаление записи о каком-либо заказчике из таблицы **Заказчик** повлечет за собой удаление тех записей из таблицы **Заказ**, где в поле **Заказчик** содержится код этого заказчика.

Для **изменения связи** необходимо выделить линию связи и вызвать окно **Изменение связи** (через основное или контекстное меню), а для **удаления связи** достаточно выделить связь и воспользоваться клавишей **DELETE** или соответствующей командой контекстного меню.

Задание 1.3. Создайте все необходимые связи между таблицами БД **ЭКСПЕРТ**, для каждой связи обеспечив соблюдение целостности данных и задав каскадное обновление связанных полей.

Примечание. Помните, что связь можно установить только между полями одинаковых типов, т. е. если система не создает связь, прежде проверьте, одинаковы ли типы данных и основные свойства связываемых полей. Для этого можно прямо из окна **Схема данных** открыть таблицу в режиме Конструктора, выбрав соответствующую команду в контекстном меню таблицы.

Заполнение таблиц

При заполнении таблиц важна последовательность – сначала лучше заполнить те таблицы, от первичных ключей которых идут связи. Либо временно не заполнять значения внешних ключей, если не внесены значения соответствующих первичных ключей (например, не заполнять поле **Заказчик** таблицы **Заказ**, пока не заполнена таблица **Заказчик**).

Таблицы БД можно заполнять непосредственно в режиме таблицы, но часто это делают через формы, о которых – в следующем разделе.

Задание 1.4. Внесите по две записи в каждую таблицу БД **ЭКСПЕРТ**, продумав предварительно последовательность заполнения.

Оптимизация ввода значений внешних ключей

При внесении данных в таблицы, которые ссылаются на другие таблицы (т.е. имеют внешние ключи), важно знать, какие значения можно вносить. Вы убедились в этом при выполнении задания 1.4. Система «не позволит» внести, например, для заказа код заказчика, которого нет в таблице **Заказчик**.

Можно оптимизировать ввод значений внешнего ключа, обеспечив пользователю возможность выбирать его из списка допустимых значений, то есть таких, которые имеет первичный ключ ссылающейся таблицы.

Для этого в режиме **Конструктора** для поля внешнего ключа на вкладке **Подстановка** в качестве источника строк можно указать другую таблицу (**Тип источника строк** – «Таблица или запрос»). В этом случае достаточно указать имя таблицы, из которой будут взяты значения.

Например, для поля **Заказчик** таблицы **Заказ** будут использоваться коды заказчиков из таблицы **Заказчик** (см. рис. 9)

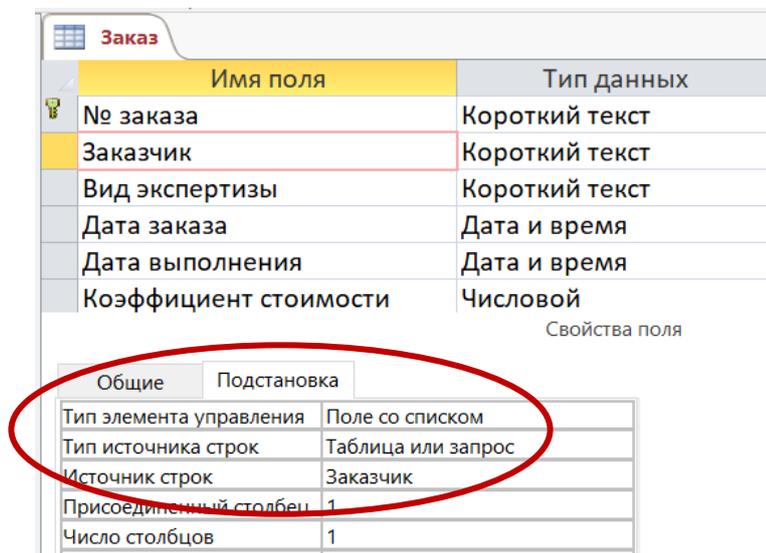


Рисунок 9 – Применение подстановки значений внешнего ключа

В некоторых случаях при выборе из списка значений внешнего ключа важно видеть не только значения первичного ключа, но и значения некоторых неключевых полей. Например, при выборе сотрудника в таблице **Выполнение заказов** важно видеть не только табельный номер сотрудника, но и его должность и фамилию. Можно настроить список так, чтобы при выборе видно было несколько полей из ссылочной таблицы: вызвать построитель запросов (кнопка с многоточием в свойстве **Источник строк**), выбрать нужные поля, и в свойстве **Число столбцов** указать нужное количество полей (см. рис. 10).

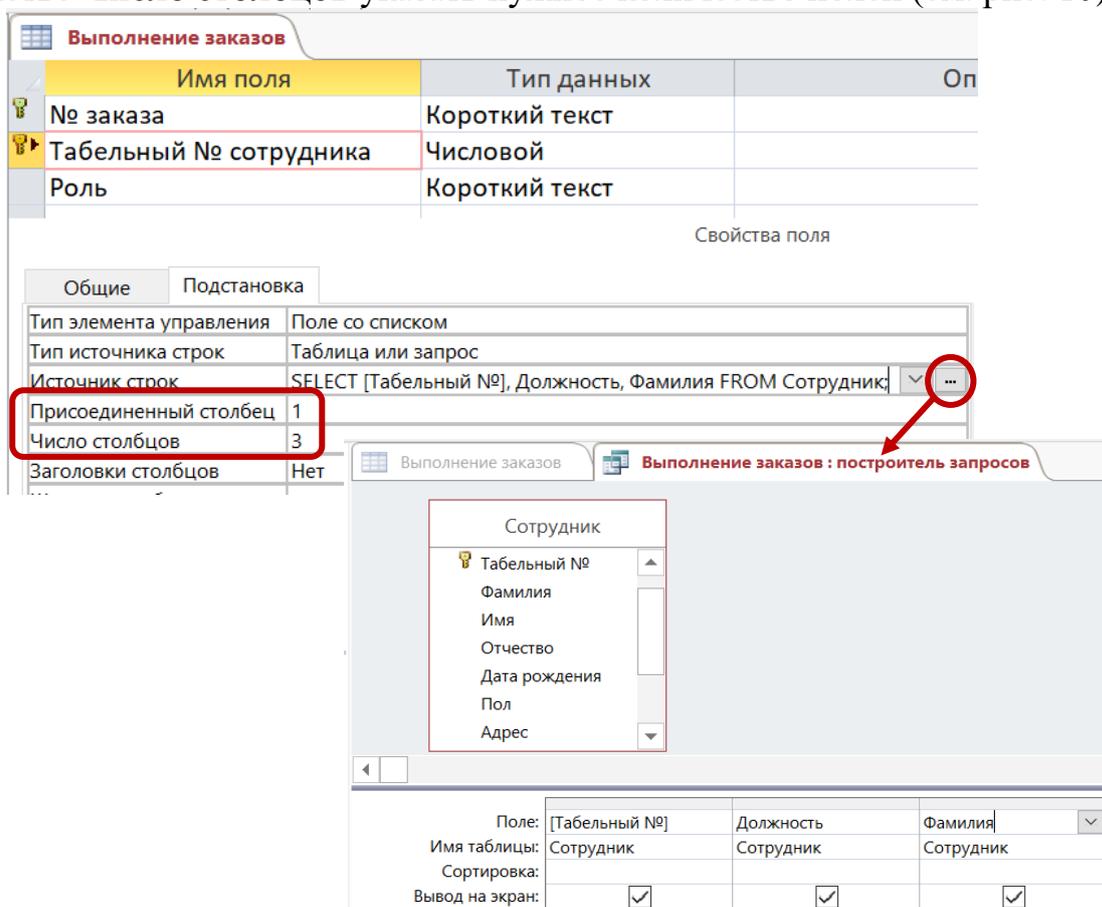
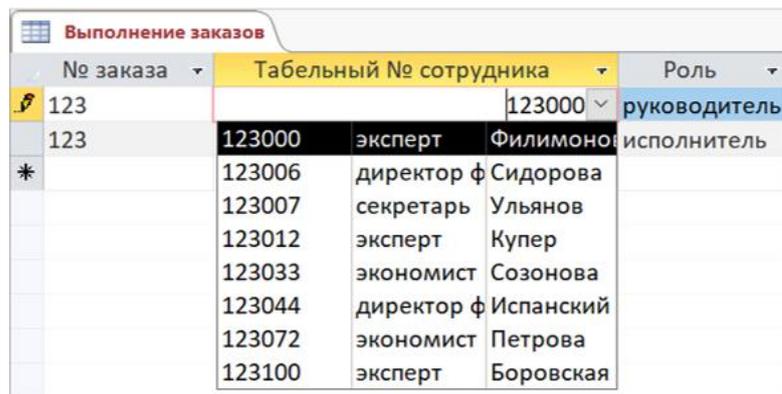


Рисунок 10 – Применение подстановки с выбором нескольких полей

В результате при заполнении таблицы в поле со списком будет открываться список, показывающий значения трех полей ссылочной таблицы (см. рис. 11).



№ заказа	Табельный № сотрудника	Роль
123	123000	руководитель
123	123000	эксперт Филимонов
*	123006	директор ф Сидорова
	123007	секретарь Ульянов
	123012	эксперт Купер
	123033	экономист Созонова
	123044	директор ф Испанский
	123072	экономист Петрова
	123100	эксперт Боровская

Рисунок 11 – Список из нескольких полей для выбора значения внешнего ключа

Задание 1.5. Организуйте ввод значений всех внешних ключей через элемент **Поле со списком**.

Задание 1.6. Внесите еще несколько записей в таблицы БД ЭКСПЕРТ:

- в таблицу **Филиал** внесите данные еще об одном филиале;
 - в таблицу **Сотрудник** внесите данные еще о четырех сотрудниках;
 - в таблицу **Вид экспертизы** внесите сведения об оставшихся видах экспертиз;
 - в таблицу **Заказчик** внесите данные еще о двух заказчиках;
 - в таблицу **Заказ** внесите сведения еще о двух заказах;
 - в таблицу **Выполнение заказов** внесите еще шесть записей.
-

Лабораторная работа № 2

Формы в MS Access

Форма — это объект базы данных, который используют для создания интерфейса пользователя базы данных.

Формы бывают привязанные и свободные.

Привязанная форма напрямую соединена с источником данных (например, с таблицей или запросом) и может использоваться для *ввода, изменения и отображения* данных из этого источника данных. Привязанные формы можно представить как окна, через которые пользователи могут просматривать и изменять данные в базе данных.

Свободная форма не связана напрямую с источником данных, но может содержать кнопки, надписи и другие элементы управления, необходимые для работы приложения.

Преимущественно создают и используют привязанные формы: с их помощью можно определять, какие поля или записи источника данных будут отображаться. Например, некоторым пользователям достаточно видеть лишь несколько полей большой таблицы. Если предоставить им форму, содержащую только эти нужные поля, это облегчит для них работу с базой данных. Для автоматизации часто выполняемых действий в форму можно добавить кнопки и другие функциональные элементы.

Режимы представления форм

Прежде чем описать способы создания форм, рассмотрим различные режимы представления формы: режим формы, режим макета и конструктор (*в отличие от таблиц (см. лаб. работу № 1), для которых имеется только два режима представления: режим таблицы и конструктор*).

Режим формы позволяет просматривать, вносить и изменять данные, но не позволяет менять структуру формы.

Для изменения структуры предназначены другие два режима.

Режим макета (рис. 12) представляет собой наиболее наглядный режим изменения форм. Здесь данные отображаются практически так же, как при реальном использовании формы. Но при этом можно изменять структуру формы. Поскольку при изменении формы отображаются реальные данные, в режиме макета удобно, например, задавать размер элементов управления. Однако вносить и изменять данные в режиме макета невозможно.

В режиме **макета** можно также изменить свойства формы, ее разделов и элементов управления с помощью **окна свойств** (рис. 12), которое вызывается командой **Страница свойств** (вкладка **Конструктор**, группа **Сервис**) или клавишей F4.

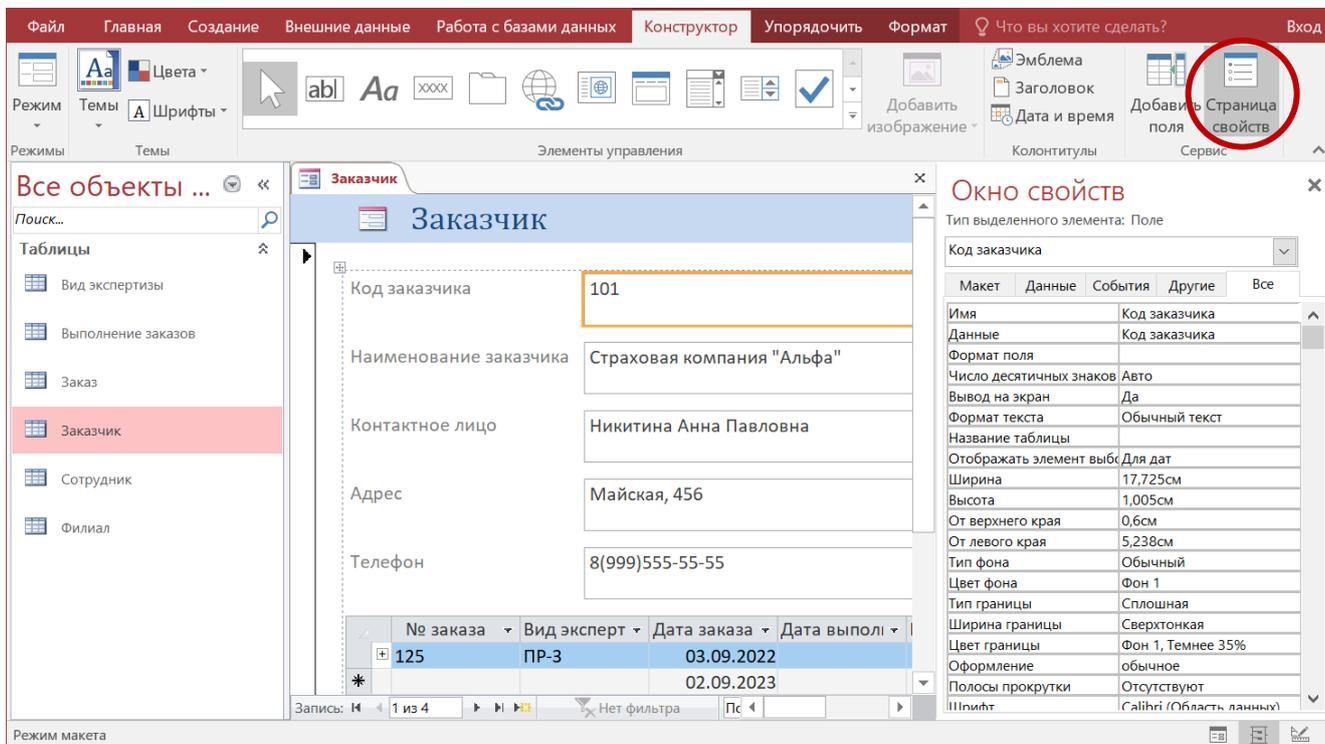


Рисунок 12 – Форма в режиме Макета, окно свойств элемента

Можно добавить в макет формы поля из таблиц – из области **Список полей** (рис. 13), который вызывается командой **Добавить поля** (вкладка **Конструктор**, группа **Сервис**) или клавишами ALT+F8.

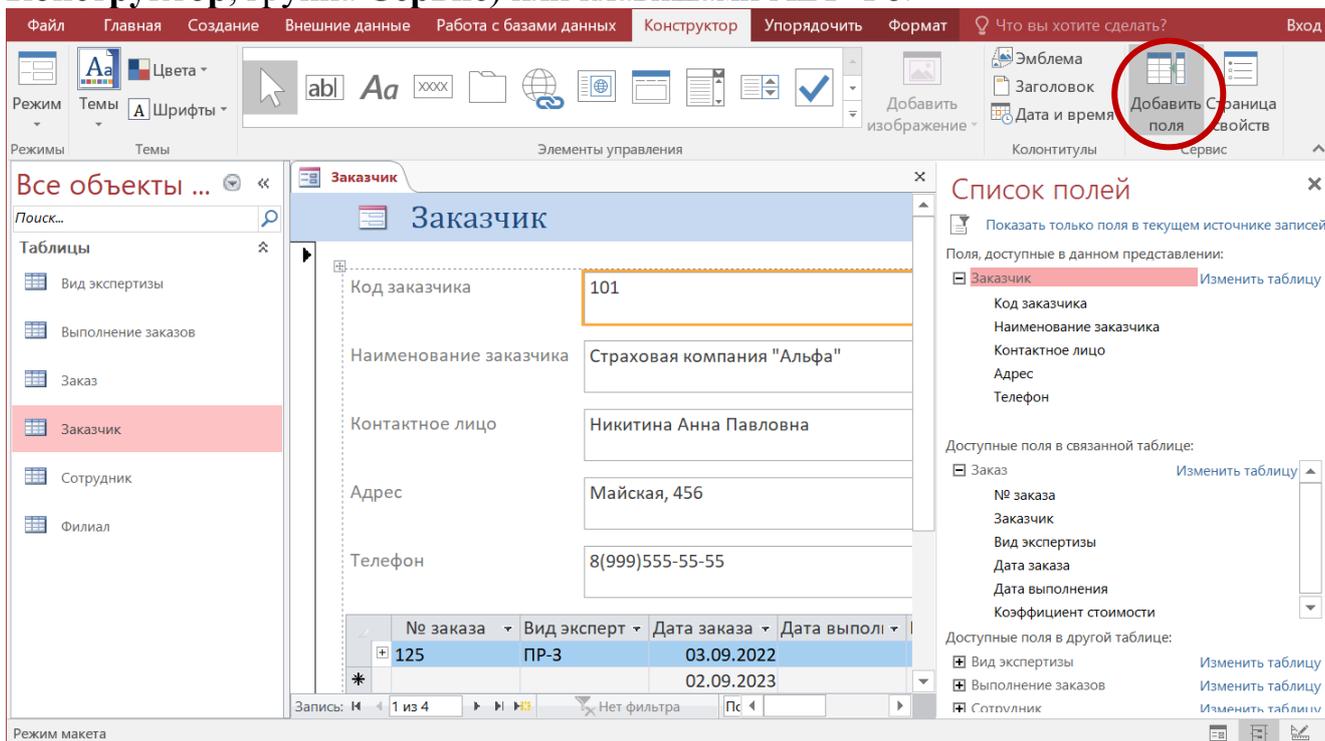


Рисунок 13 – Форма в режиме Макета, окно списка полей для добавления

Режим конструктора (рис. 14) обеспечивает более подробное представление структуры формы. В нем отображаются разделы заголовка, примечания, области данных, колонтитулов. В этом режиме видна только структура формы, но не видны сами данные. Многие задачи удобнее выполнять в режиме конструктора, а не макета, например, добавление в форму дополнительных элементов управления, изменение размеров разделов формы,

изменение некоторых свойств формы. Вызов окна свойств и списка полей осуществляется точно так же, как и в режиме макета.

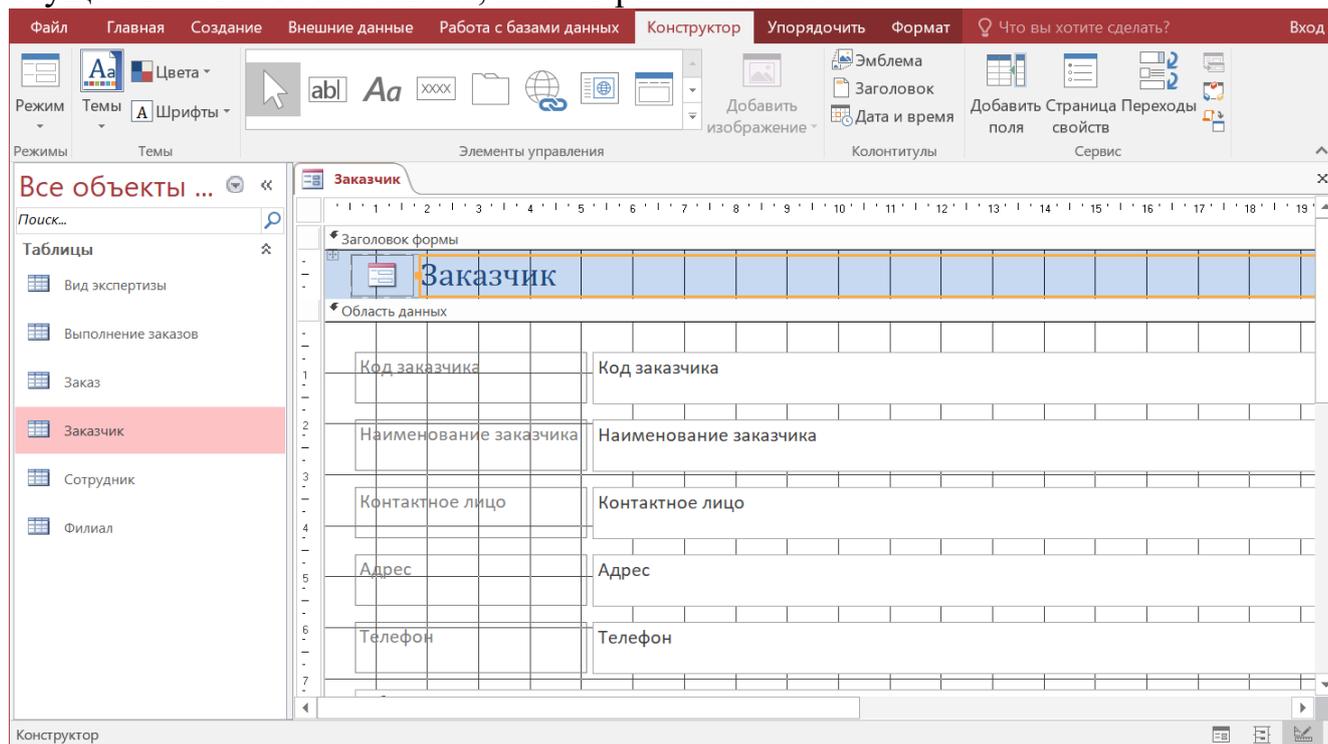


Рисунок 14 – Форма в режиме Конструктора

Для переключения между режимами можно воспользоваться контекстным меню, вызванным на названии формы в области навигации или на ярлыке открытой формы, кроме этого есть команда **Режим** – самая первая на вкладке **Конструктор**.

Создание форм

Для создания формы необходимо перейти на вкладку **Создание** и в группе **Формы** выбрать один из способов создания формы: Форма, Конструктор форм, Пустая форма, Мастер форм, Навигация, Другие формы.

Создание простой формы

Самый простой способ создания формы – с помощью инструмента **Форма**:

- в области навигации слева выделить таблицу;
- на вкладке **Создание** в группе **Формы** выбрать команду **Форма**.

Будет создана новая форма и отображена в режиме макета. При этом все поля выделенной таблицы размещаются в форме. Можно сразу же перейти в режим формы и начать использование новой формы либо при необходимости изменить ее в режиме макета или конструктора.

Задание 2.1. Создайте простую форму по таблице **Заказчик**. Внесите через эту форму сведения еще о двух заказчиках (информацию об их заказах пока вносить не надо). Сохраните форму и проверьте, занеслись ли новые записи в таблицу **Заказчик**.

Примечание: для перехода между записями в форме используют кнопки перехода в нижней части формы, для добавления новой записи можно использовать кнопку , расположенную там же.

Создание формы, в которой отображается сразу несколько записей

В форме, созданной с помощью инструмента **Форма**, в каждый момент времени отображается только одна запись. Если нужна форма, в которой отображается сразу несколько записей, и при этом требуются более широкие возможности настройки, чем у таблицы, можно воспользоваться инструментом **Несколько элементов**:

- в области навигации выделить таблицу;
- вкладка **Создание**, группа **Формы**, команда **Другие формы** → **Несколько элементов**.

Будет создана новая форма и отображена в режиме макета. Эта форма включает также все поля выделенной таблицы и внешне напоминает саму таблицу – данные расположены в строках и столбцах, одновременно отображается несколько записей. Однако форма **Несколько элементов** предоставляет больше возможностей настройки, чем таблица. Например, в нее можно добавлять графические элементы, кнопки и другие элементы управления.

Задание 2.2. Создайте форму по таблице Вид экспертизы с помощью инструмента **Несколько элементов**. В режиме Макета откорректируйте высоту и ширину элементов.

Создание разделённой формы

Разделенная форма позволяет одновременно отображать данные в двух представлениях – в режиме формы и в режиме таблицы. Причем эти два представления связаны с одним источником данных и всегда синхронизированы друг с другом. При выделении поля в одной части формы выделяется то же поле в другой части. Данные можно добавлять, изменять или удалять в любой части.

Разделенная форма позволяет использовать преимущества обоих типов форм в одной. Например, через табличную часть формы можно быстро найти запись, а затем просмотреть или изменить запись в другой части формы.

Создание разделенной формы выделенной таблицы: на вкладке **Создание** в группе **Формы** выбрать команду **Другие формы** → **Разделенная форма**.

Будет создана новая разделенная форма и отображена в режиме Макета. Эта форма также отображает все поля выделенной таблицы.

Задание 2.3. Создайте разделенную форму по таблице Сотрудник. В режиме Макета уменьшите высоту полей так, чтобы все поля были видны. Внесите через эту форму (в режиме Формы) сведения еще об одном сотруднике.

Создание формы при помощи Мастера форм

Для получения большей свободы выбора полей, отображаемых на форме, можно воспользоваться **Мастером форм**. При этом можно также указать способ группировки и сортировки данных, а также включить в форму поля из нескольких связанных таблиц.

Мастер форм запускается на вкладке **Создание** в группе **Формы**. Мастер предлагает выполнить ряд шагов:

- выбор таблиц и полей, которые будут представлены в форме; если в форму нужно включить поля из нескольких таблиц, то сначала выбирается одна таблица, из нее «перекидываются» поля, затем другая и так далее;
- выбор вида представления данных (если выбраны поля из нескольких таблиц): т.е. нужно указать, каким образом в форме будут соотноситься выбранные таблицы – как главная и подчиненная или как просто связанные; причем, если выбрано представление «Подчиненные формы», то в левой части окна выделяется главная таблица (например, если в форме предполагается отобразить данные о филиалах с перечислением списка сотрудников, работающих в каждом филиале, то главной таблицей будет **Филиал**, а подчиненной – **Сотрудник**, см. рис. 15);

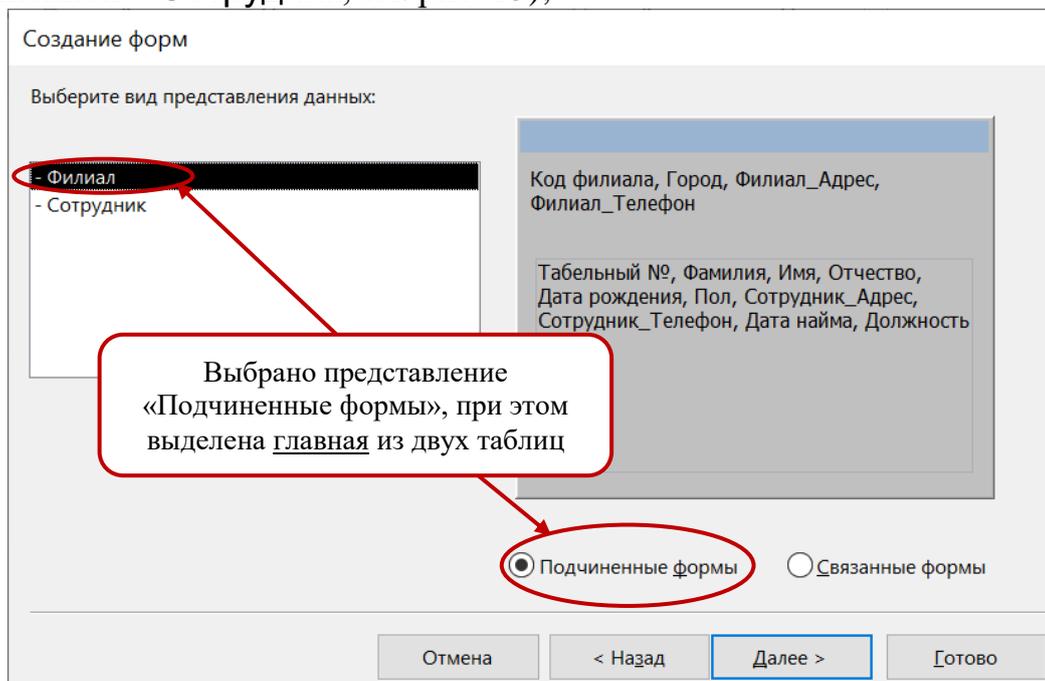


Рисунок 15 – Мастер форм, шаг выбора представления данных

- если в форме было выбрано представление «Подчиненные формы», то на следующем шаге выбирается вид подчиненной формы;
- назначение имени форме.

Задание 2.4. Используя Мастер форм, создайте форму для ввода основных данных о сотрудниках филиалов (по таблицам **Филиал** и **Сотрудник**, главной является таблица **Филиал**). Внесите через эту форму сведения еще о двух сотрудниках разных филиалов. Назовите форму **Филиалы с сотрудниками**.

Создание формы при помощи инструмента Пустая форма

Для создания формы можно воспользоваться инструментом **Пустая форма** (вкладка **Создать**, группа **Формы**). При запуске этого инструмента не надо выделять никакого источника данных, поскольку в любом случае будет открыта совершенно пустая новая форма в режиме макета. Также на экране откроется список полей, откуда можно и перетаскивать нужные поля на пустую форму. Перетаскивать надо именно поля, а не всю таблицу целиком.

Задание 2.5. Используя инструмент **Пустая форма**, создайте форму, показывающую информацию обо всех заказах (по таблице **Заказ**).

Конструктор форм

Конструктор предусматривает более детальное, «ручное» проектирование макета формы. Добавление нужных полей и вызов свойств элементов формы описано выше в пункте «Режимы представления форм».

В Конструкторе можно добавить на форму различные элементы управления (вкладка **Конструктор**, группа **Элементы управления**), такие, как наборы вкладок, кнопки, переключатели, поля со списком и др., которые позволяют оптимизировать ввод данных.

Рассмотрим основные элементы управления.



Надпись. Используется для размещения в форме текста (заголовков, поясняющих надписей). Надписи бывают присоединенными и свободными. Присоединенные служат заголовками полей, они автоматически создаются при вводе в форму поля и присоединяются к этому полю. Свободные надписи создаются указанной кнопкой и служат для оформления формы, они не выводятся в самой таблице.



Поле. Служит для ввода и отображения данных, обычно к нему прикреплена поясняющая надпись. В поле пользователем вводится значение или формула. Например, на рис. 16 вставлено поле, отображающее текущую дату.

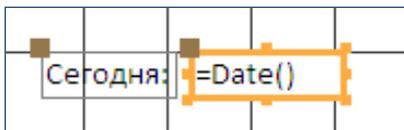


Рисунок 16 – Поле, выводящее текущую дату, в режиме Конструктор форм



Кнопка. Именно с кнопками связаны многие действия, выполняемые пользователем в приложении (сохранение введенных данных, вызов другой формы, вывод документа на печать и др.). Создание кнопки оптимально производить с помощью мастера создания кнопок. Можно настроить свойства кнопки с помощью макроса.



Выключатель, переключатель и флажок предоставляют пользователю возможность выбирать значения. Все они используются для

представления данных логического типа, т. е. могут принимать значения 0 (нет, выключен) или 1 (да, включен).

 **Группа** используется для того, чтобы дать возможность пользователю выбрать единственное значение из короткого списка. Сами элементы управления в группе не должны присоединяться к полям таблицы, но для каждого из них необходимо задать значение параметра. Оно должно быть числовым и обязательно разным для каждого элемента группы (чаще всего это значения ключевого поля). Когда пользователь выбирает переключатель в группе, значение элемента становится равным значению параметра выбранного переключателя. Это значение сохраняется в соответствующем поле таблицы.

 **Список.** Используется для представления на экране поля, значения которого ограничиваются не слишком длинным списком.

 **Поле со списком.** Этот элемент во многом аналогичен **Списку**. Применяется в формах для той же цели, т. е. когда требуется, чтобы значение поля не вводилось вручную, а выбиралось из известного списка. Однако поле со списком может включать большее количество значений, так как в нем отображается только текущее значение, а остальные выводятся, когда пользователь открывает список.

 **Рисунок, Свободная рамка объекта и Присоединенная рамка объекта** используются для вставки в форму объектов, созданных в других приложениях. Эти элементы различаются тем, будут ли объекты, вставленные в эти рамки, храниться в таблицах Access или не будут. **Свободная рамка** содержит объект, который хранится либо в самой форме, либо во внешнем файле. **Присоединенная рамка** содержит объект, который хранится в поле таблицы (в этом случае тип поля – Поле объекта OLE). Элемент управления **Рисунок** близок по свойствам к **Свободной рамке объекта**, но работает быстрее, так как требует меньше ресурсов.

 Инструменты **Линия** и **Прямоугольник** используются для оформления внешнего вида форм. Изменить цвет и тип фона, цвет и тип линии можно с помощью команд в правой части группы Элементы управления (на вкладке Конструктор).

 **Набор вкладок** позволяет разнести поля формы на разные вкладки (например, по тематике) и используется для создания многостраничных форм. Вкладки можно добавлять и убирать. Все параметры каждой вкладки – в окне свойств. При работе с вкладками необходимо помнить, что если элемент первоначально располагался на форме, то для того, чтобы переместить его на вкладку его нужно вырезать, а затем вставить на нужную вкладку.

 **Разрыв страницы.** Используется в многостраничных формах и отчетах

для того, чтобы определить место конца одной страницы и начала другой. В режиме конструктора этот элемент отображается в виде пяти горизонтально расположенных точек и имеет всего четыре свойства: *Имя*, *От левого края* и *От верхнего края* (координаты относительно верхнего левого угла формы), *Дополнительные сведения*. Для переключения между страницами готовой формы используются клавиши Page Up, Page Down.

Принципы создания описанных элементов управления одинаковы – их можно добавлять в режиме Конструктора и настраивать их свойства при помощи мастера или же путем прямой настройки свойств.

При вставке в форму элементов управления удобно использовать мастера. Чтобы при выборе элемента запускались мастера, кнопка **Мастера** в группе Элементы управления (все на то же на вкладке Конструктор) должна быть нажата (рис. 17).

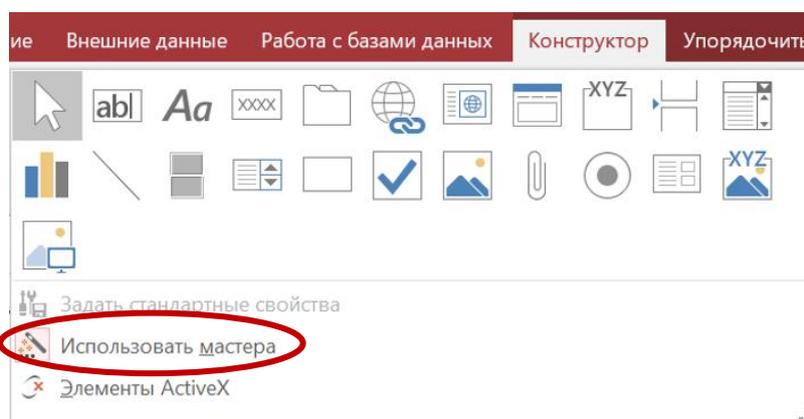


Рисунок 17 – Включен режим использования мастера в режиме Конструктора форм

В дальнейшем можно корректировать свойства созданных элементов: изменить размеры элементов и подписи, задать дополнительные параметры с помощью окна свойств, которое вызывается командой **Страница свойств** (группа **Сервис** на вкладке **Конструктор**) или клавишей F4.

Корректировать размеры полей и их расположение на форме можно с помощью мыши (рис. 18) либо клавиш CTRL и SHIFT и стрелок управления курсором на клавиатуре.

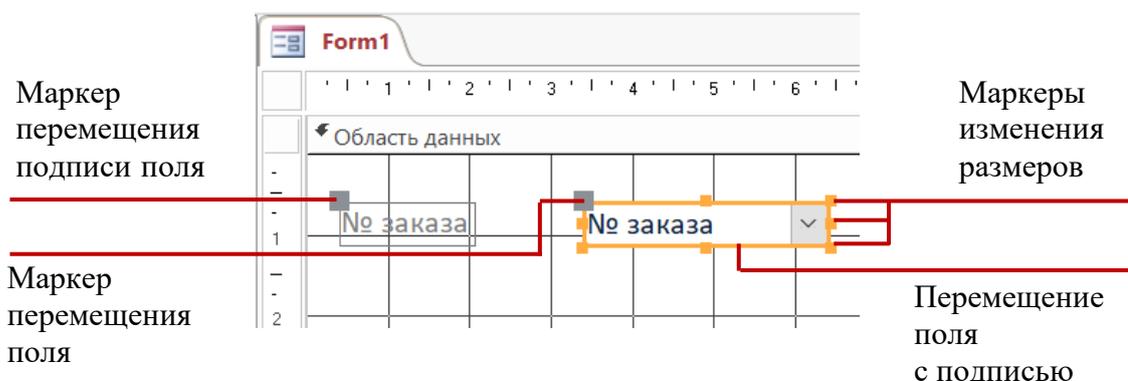


Рисунок 18 – Корректировка размеров и положения элементов

При создании формы в режиме Конструктор в этой форме создается единственный раздел – *Область данных*. Чтобы добавить разделы заголовка и примечания формы можно выбрать из контекстного меню области данных

команду **Заголовок / примечание формы**. Аналогично можно добавить области **колонтитулов**. Удаление областей заголовка, примечания и колонтитулов – тоже через контекстное меню.

Может возникнуть необходимость перенести подпись поля из Области данных в Заголовок (а само поле оставить в Области данных) – это невозможно сделать перетаскиванием мышью, можно только вырезать элемент и затем вставить его в другой раздел.

Команды группы Колонтитулы (рис. 19) исследуйте самостоятельно при выполнении задания 2.6.

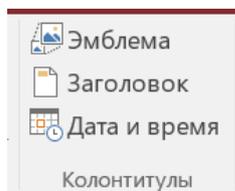


Рисунок 19 – Группа команд Колонтитулы в Конструкторе форм

Задание 2.6. Откройте в режиме Конструктора форму **Заказ**, созданную в предыдущем задании, вставьте название формы, произвольный рисунок и поле, в котором будет выводиться текущая дата.

Задание 2.7. В форме **Заказ** рядом с полем **Заказчик** добавьте элемент управления **Поле со списком**, в котором будет наименование заказчика (из таблицы **Заказчик**).

*Это позволит не только видеть названия заказчиков, но и при вводе новых данных выбирать заказчика по названию, а не по коду, хотя в таблице **Заказ** будет сохраняться только код заказчика – как ссылочное значение первичного ключа таблицы **Заказчик**.*

Один из способов выполнению задания 2.7:

- Откройте форму **Заказ** в режиме конструктора.
- Убедитесь, что включены мастера  и виден список полей.
- Выберите (включите) элемент **Поле со списком** в группе **Элементы управления**.
- Из списка полей таблицы **Заказ** перетащите поле **Заказчик** на форму рядом с уже находящимся там полем **Заказчик**. После этого откроется окно «Создание полей со списком».
- На первом шаге Мастера предлагается выбрать способ получения значений – выберите вариант «Объект «поле со списком» получит значения из другой таблицы».
- Далее выбирается определяется тип источника строк – Таблица, и выбирается таблица **Заказчик**, поскольку наименования заказчиков будут браться из таблицы **Заказчик**.
- На следующем шаге в качестве источника данных выбирается поле **Наименование заказчика**.
- Затем можно выбрать вариант сортировки (это необязательный шаг).

- На следующем шаге показан список наименований заказчиков, который будет показан, здесь же по умолчанию отмечен параметр «Скрыть ключевой столбец», оставьте его отмеченным.
- Важным является следующий шаг, когда необходимо задать действие, которое будет выполняться при выборе значения из списка. Поскольку выбранного заказчика (его код) необходимо сохранить в таблице **Заказ** (в поле **Заказчик**, которое является *внешним ключом*, ссылающимся на таблицу **Заказчик**), то определите действие **Сохранить в поле** и выберите в списке предлагаемых полей поле **Заказчик** (рис. 20).

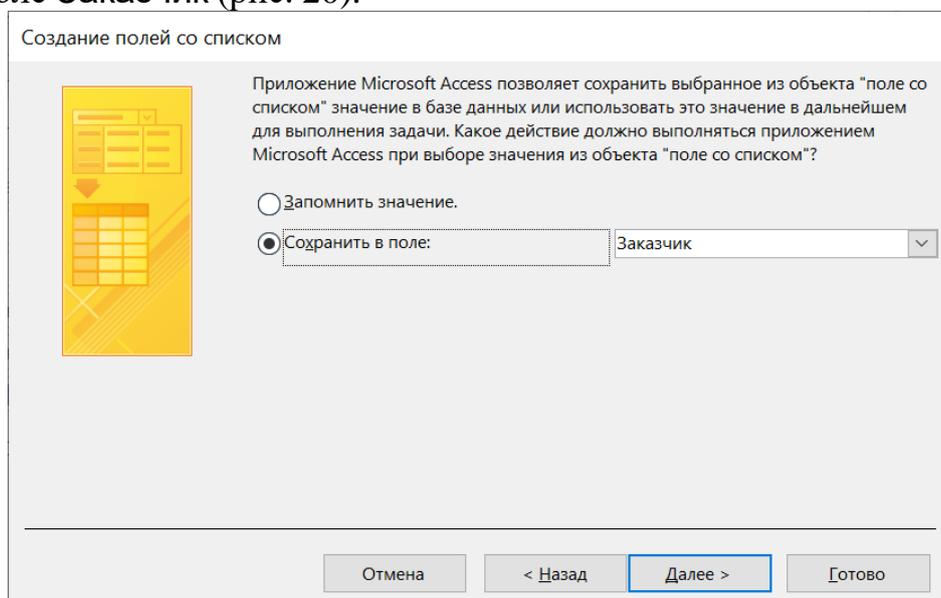


Рисунок 20 – Задание параметра сохранения значения поля со списком

- На завершающем шаге можно задать подпись поля, например, «Наименование заказчика»
- После завершения работы мастера подкорректируйте размеры поля со списком. Сохраните изменения. Просмотрите данные в режиме формы.

Задание 2.8. Создайте простую форму по таблице **Выполнение заказов**. Рядом с полем **Табельный №** сотрудника добавьте элемент управления **Поле со списком**, содержащий фамилию, имя и отчество сотрудника.

Задание 2.9. В созданной ранее форме **Выполнение заказов** в область заголовка добавьте кнопку, при нажатии на которую открывается форма **Заказ**, показывающая только выполняемый заказ.

Подсказки по выполнению задания:

- Поскольку для решения этой задачи нужно использовать связь между таблицами **Заказ** и **Выполнение заказов**, сперва необходимо уточнить, по каким полям они связаны – в **Схеме данных** видно, что связаны они по полям **№ заказа**.
- Откройте форму **Выполнение заказов** в режиме конструктора. Включите мастеров.
- Добавьте кнопку на форму (с помощью соответствующего элемента группы **Элементы управления**). Откроется окно «Создание кнопок».

- Выберите категорию **Работа с формой**, действие **Открыть форму** и на следующем шаге название нужной формы – **Заказ**.
- Поскольку требуется выполнить отбор записей (т. е. вывести не все записи, а только ту, которая содержит сведения о выполняемом заказе), то на следующем шаге задайте **Открыть форму для отобранных записей**, а на следующем шаге отметьте поля, по которым таблицы связаны.
- На следующем шаге можно ввести надпись или выбрать рисунок для кнопки.
- На заключительном шаге предлагается задать внутреннее имя кнопки, его можно не менять.

выбирается таблица **Заказчик** и поля, которые необходимо отобразить (**Наименование заказчика, Телефон, Контактное лицо**), а на втором шаге запросу присваивается имя, например, **Телефоны Заказчиков**.

Задание 3.2. Создайте запрос **Повтор даты найма**, выводящий фамилии, имена и отчества сотрудников, принятых на работу в один день.

Пояснения по выполнению задания 3.2: примените мастер создания запроса на **повторяющиеся записи**. Работа с ним состоит из следующих шагов:

1-й шаг – выбрать таблицу, по которой делается запрос;

2-й шаг – выбрать поле, по которому будет идти поиск повторяющихся значений (**Дата найма**);

3-й шаг – выбрать те поля, которые должны выводиться в запросе (**Фамилия, Имя, Отчество**);

4-й шаг – ввести имя запроса (**Повтор даты найма**).

Задание 3.3. Создайте запрос **Заказчики без заказов**, выводящий сведения о заказчиках, от которых пока не оформлены заказы.

Пояснения по выполнению задания 3.3: воспользуйтесь мастером создания запросов на **записи без подчиненных**. Запросы такого типа могут быть использованы для контроля согласованности введенных пользователями данных.

1-й шаг – выбрать таблицу, записи из которой нужно получить (**Заказчик**);

2-й шаг – выбрать связанную таблицу с подчиненными записями (**Заказ**);

3-й шаг – указать поля, по которым связаны эти таблицы (**Код заказчика** и **Заказчик**);

4-й шаг – выбрать поля из основной таблицы, которые должны войти в запрос (все поля);

5-й шаг – ввести имя запроса (**Заказчики без заказов**).

Задание 3.4. Создайте запрос **Должности**, показывающий распределение сотрудников компании по должностям и полу (рис. 22).



Должность	Итоговое значение	ж	м
директор филиала	2	1	1
секретарь	1		1
экономист	2	2	
эксперт	3	1	2

Рисунок 22 – Примерный результат запроса «Должности»

Пояснения по выполнению задания 3.4: для создания такого запроса удобно применить мастер **перекрестных запросов**. Перекрестные запросы

используют для расчетов и представления данных в виде, облегчающем их анализ. Перекрестный запрос может подсчитывать сумму, количество значений или выполнять другие статистические операции, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой – заголовки строк.

Работа с мастером состоит из следующих шагов (на примере выполнения задания 3.4):

1-й шаг – выбрать таблицу (**Сотрудник**), причем в качестве источника записей можно выбрать только одну таблицу или запрос;

2-й шаг – выбрать поля, значения которых будут заголовками строк (**Должность**);

3-й шаг – выбрать поля, значения которых будут заголовками столбцов (**Пол**);

4-й шаг – выбрать поле и функцию для вычисления результатов (поле – **Табельный номер**, так как подсчитать сотрудников удобно по ключевому полю; а функция – **Число**, так как нужно подсчитать их число, т. е. количество);

5-й шаг – ввести имя запроса **Должности**.

Создание запросов в режиме Конструктора

Запросы на выборку

Режим Конструктора является наиболее общим способом при создании запросов. Вызов окна для создания запроса в режиме Конструктора осуществляется командой **Конструктор запросов** (лента **Создание**, группа **Запросы**).

Открыть созданный ранее запрос в режиме Конструктора можно через контекстное меню, вызванное на имени этого запроса в области переходов (в левой части экрана).

Составление запроса в режиме Конструктора включает в себя следующие этапы:

1) Выбор нужных **таблиц** через окно **Добавление таблицы**, которое открывается сразу при запуске конструктора (это окно можно открыть также командой **Отобразить таблицу** на ленте **Конструктор** в группе **Настройка запроса** либо через контекстное меню). Если в результирующем наборе данных будут присутствовать поля таблиц, напрямую не связанных между собой, то в запрос должны быть включены все связывающие их таблицы в соответствии со **Схемой данных**.

2) Указание **полей**, которые будут использоваться в запросе, – в нижней области конструктора (в строке «**Поле:**») (см. рис. 23). Поля можно «перетащить» из таблиц, либо выполнить двойной щелчок мыши по их именам, либо выбрать из списков, открывающихся в строке **Поле:**

3) Описание **вычисляемых полей** и **групповых операций** над записями. При создании запросов допускается создание вычисляемых полей, значения которых формируются в процессе выполнения арифметических операций над

значениями других полей и использования набора специальных встроенных функций. Значения вычисляемых полей формируются каждый раз при выполнении запроса.

4) Установление **условий отбора** записей в результирующем наборе.

При записи условия отбора допустимо использование:

- Операций сравнения: $>$, $<$, $>=$, $<=$, $<>$, $=$ (предполагается по умолчанию).
- Логических операций: **Not**, **And**, **Or** (можно использовать строку **или**, которая находится после строки **Условие отбора** в окне конструктора запроса).
- Функции **Between**, которая используется для указания диапазона значений в следующем формате: *between значение1 and значение2*. Например: **between 01.12.2022 and 15.12.2022** – значения даты с 1 по 15 декабря 2022 года.
- Функции **Like**, которая определяет соответствие строкового выражения маске, для записи которой используются символы $*$ (любое количество произвольных символов) и $?$ (один произвольный символ). Например, **Like «А*»** – текстовые значения, начинающиеся с «А», **Like «*в»** – текстовые значения, оканчивающиеся на букву «в», **Like «Д???»** – текстовые значения, начинающиеся с «Д» и содержащие еще три символа.

5) Определение **параметров отображения** результатов выполнения запроса (направление сортировки полей, показ полей).

В режиме конструктора запросов направление сортировки поля указывается в строке **Сортировка**, а показ полей – установкой или снятием флажка в строке **Вывод на экран**.

6) **Сохранение** и ввод имени созданного запроса.

Пример. На рис. 23 показано окно Конструктора запросов, в котором создан запрос, выводящий информацию о заказчиках из Москвы.

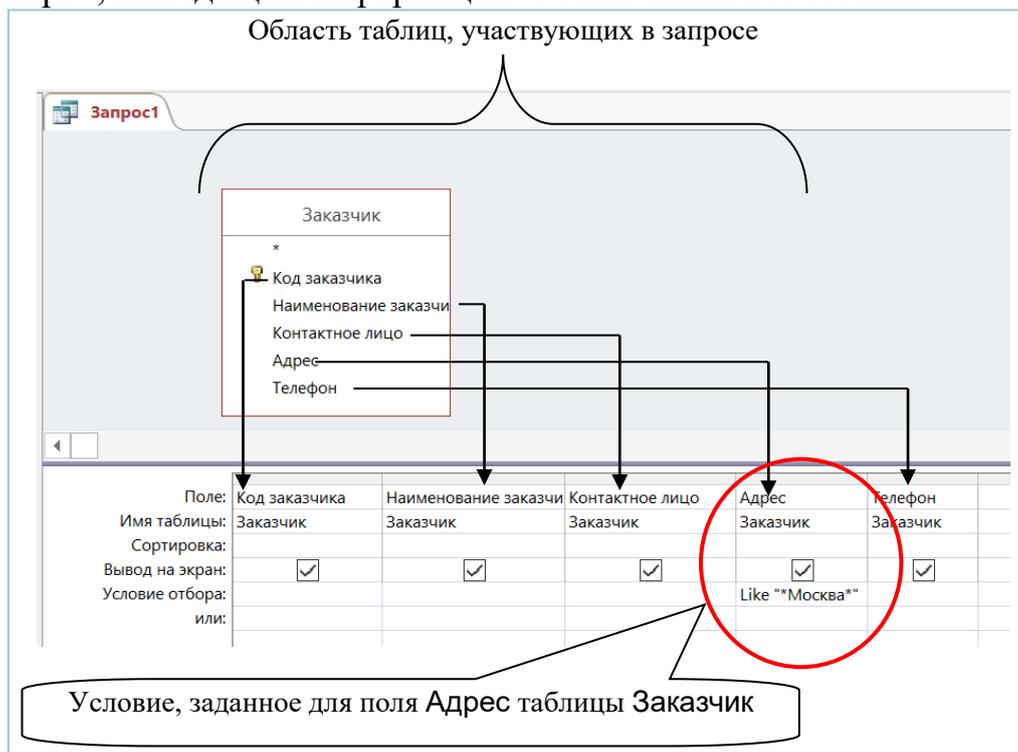


Рисунок 23 – Конструктор запросов

Здесь в качестве условия отбора для поля **Адрес** введено выражение **Like «*Москва*»**, поскольку перед названием города и после него могут быть еще символы (например, «г. Москва, ул. Тимирязева, 113-42»), а может и не быть ничего.

Задание 3.5. Создайте запрос **Дорогие услуги**, показывающий описание видов экспертизы, стоимость которых свыше 50000 руб.

Задание 3.6. Выведите список заказчиков, которые являются ОАО, указав их наименования, телефоны, контактные лица. Запрос должен также выводить номера и даты всех заказов этих заказчиков. Сохраните запрос под именем **Заказчики ОАО**.

Запросы с параметром

В таких запросах при запуске у пользователя запрашивается конкретное значение поля-параметра (например, даты, наименования и т. п.). В этом случае в строке **Условие отбора** в режиме Конструктора нужно в качестве условия отбора задать запрос **параметра**: вместо конкретного значения поля указывается в квадратных скобках приглашение на ввод значения: [Введите ...] (например, [Введите дату] или [Введите наименование]).

Задание 3.7. Создайте запрос **Сотрудники по должности**, выводящий информацию о сотрудниках указанной пользователем должности (при запуске запроса).

На рисунке 24 показано, как будет выглядеть окно для ввода значения параметра при запуске запроса.

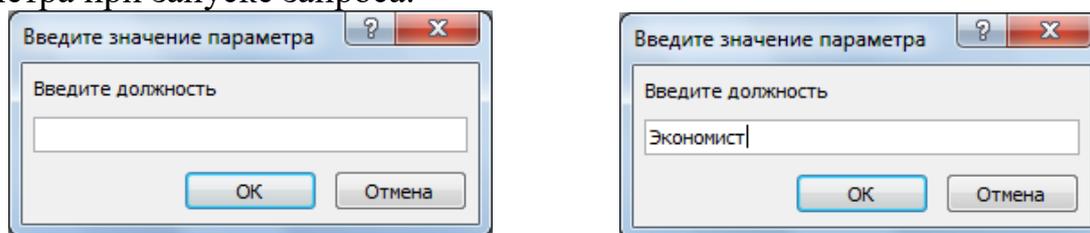


Рисунок 24 – Инструменты создания запроса

Пояснения по выполнению задания 3.7.

Поскольку необходимо, чтобы пользователь вводил нужную ему должность при запуске запроса, то простое указание конкретной должности в строке **Условие отбора** в режиме конструктора не решит задачу. Здесь нужно в качестве условия отбора добавить запрос параметра.

Для этого в строке **Условие отбора** вместо конкретного значения поля указывается в квадратных скобках приглашение на ввод значения: [Введите должность].

Запросы с вычисляемыми полями

В запросе может быть создан новый столбец, значения которого будут получены из значений полей таблиц. Для этого в свободном столбце запроса (в конструкторе запросов) в строке **Поле** вводится имя нового поля, ставится двоеточие, а затем вводится формула вручную или через **Построитель выражений**, который вызывается командой **Построитель**  в группе **Настройка запроса** (на ленте **Конструктор**).

В окне **Построителя выражений** (рис. 25), раскрывая структуру в разделе **Элементы выражений**, можно выбрать поля таблиц и необходимые функции и поместить их в выражение двойным щелчком мыши. Во избежание ошибок старайтесь выбирать нужные элементы из списка объектов базы данных, а не прописывать вручную их имена.

Примечание. Если имя вычисляемого поля сразу не было указано, то по умолчанию оно будет определено как *Выражение1*, по усмотрению пользователя может быть переименовано.

Задание 3.8. Для каждого заказа рассчитайте его стоимость с учетом коэффициента стоимости. Сохраните запрос под именем **Стоимость заказов**.

На рис. 25 показано выражение для вычисления стоимости заказа.

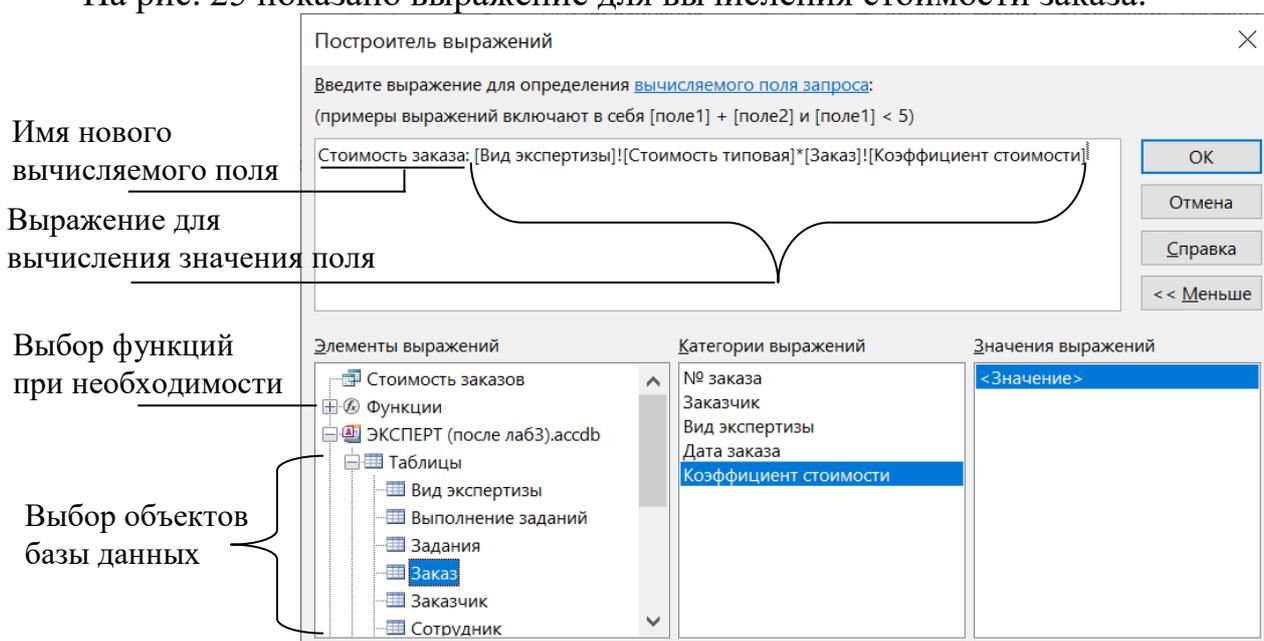


Рисунок 25 – Построитель выражений в Конструкторе запросов

Задание 3.9. Создайте запрос **Длительность заказов**, содержащий данные обо всех заказах с указанием для каждого заказа его длительность в днях (разность между датой выполнения и датой заказа).

Пояснения по выполнению задания 3.9: Здесь необходимо в режиме Конструктора добавить в запрос вычисляемое поле, содержащее формулу, вычисляющую разность дат.

Задание 3.10. Создайте запрос **Дни рождения по месяцу**, выводящий список сотрудников, у которых день рождения в месяце, указанном пользователем.

Пояснения по выполнению задания 3.10:

Во-первых, здесь необходимо создать вычисляемое поле, отображающее месяц рождения (т. к. в таблице **Сотрудник** есть только дата рождения целиком). Чтобы получить месяц нужной даты можно применить функцию **Month(дата)**, которая выделяет месяц из указанной в аргументе даты. То есть итоговое выражение будет иметь вид: **Месяц: Month([Дата рождения])**.

Во-вторых, нужно сделать запрос параметра (месяца) для этого нового вычисляемого поля.

Примечание: при запуске запроса месяц вводится в виде числа (например, май – 5).

***Дополнение к заданию 3.10.** Дополните выражение для вычисления месяца так, чтобы результатом было название месяца, а не его номер. **Подсказка:** найдите функцию, преобразующую номер месяца в его название.

Задание 3.11. Измените запрос **Сотрудники по должности** так, чтобы вместо трех отдельных полей **Фамилия**, **Имя**, **Отчество** выводилось поле, содержащее фамилии с инициалами.

Пояснения по выполнению задания 3.11:

В вычисляемом поле нужно построить выражение, которое «склеит» следующие фрагменты: фамилию полностью, затем пробел (для отделения инициалов от фамилии), далее первую букву имени, за ним точку с пробелом, далее первую букву отчества и снова точку (чтобы получилось, например, так: «Иванов И. И.»). Склеивание осуществляется знаком «+» или «&», а взять первую букву текста можно с помощью функции **Left(текст; количество символов)**, которая «берет» из указанного текста (или поля, содержащего текст) указанное количество символов слева. **Например**, результатов формулы **Left(«экспертиза»; 4)** будет значение «эксп». Вместо конкретного текста «экспертиза» может стоять имя поля, в котором содержатся текстовые значения, например, **Left(Имя; 1)** – будет выведена первая буква значений поля **Имя**.

Аналитические запросы

Аналитическая выборка данных из БД неразрывно связана с группировкой записей и использованием агрегатных функций, применяемых к сгруппированным записям.

Вычисления такого рода, как и вычисляемые поля, рассмотренные выше, не хранятся в базовой таблице, а производятся всякий раз, когда выполняется запрос, тем самым позволяют всегда представить актуальное содержимое базы данных.

Аналитическая выборка данных позволяет производить такие вычисления в запросе, с помощью которых по группам записей или по всем записям можно найти сумму, среднее арифметическое, количество значений, минимальное или максимальное значение, стандартное отклонение или дисперсию, – используя соответствующую агрегатную функцию:

Sum	Сумма значений поля
Avg	Среднее арифметическое значений поля
Min	Наименьшее значение поля
Max	Наибольшее значение поля
Count	Количество значений поля без учета пустых (Null) значений
StDev	Среднеквадратичное отклонение от среднего значения поля
Var	Дисперсия значений поля

Рассмотрим построение аналитических запросов на примере задания 3.12.

Задание 3.12. Создайте запрос **Количество заказов по заказчикам**, выводящий для каждого заказчика количество сделанных им заказов.

Пояснения по выполнению задания 3.12.

Результат может выглядеть так, как на рис. 26:

Наименование заказчика	Количество заказов
ОАО "Бета"	2
ОАО "Гамма"	2
Страховая компания "Альфа"	1
Торговая компания "Дельта"	1

Рисунок 26 – Примерный результат запроса «Количество заказов по заказчикам»

- Для решения поставленной задачи создайте запрос по таблицам **Заказ** и **Заказчик**, содержащий поля **Наименование заказчика** и **№ заказа**.

- Далее необходимо в запрос добавить групповую операцию, вычисляющую количество заказов для каждого заказчика: для применения групповых операций необходимо в табличной части окна Конструктора запросов вызвать новую строку **Групповая операция** – командой **Итоги**  (группа **Показать или скрыть** ленты **Конструктор**). В этой строке содержится список команд группировки и агрегатных функций (см. рис. 27):

Группировка определяет группы, для которых выполняются вычисления.

Конкретная **агрегатная функция** (перечислены выше) создает вычисляемое поле, в котором размещается результат агрегирования группы записей заданного поля.

Условие указывается в том случае, когда поле используется для проведения отбора до проведения групповых операций и вычислений.

- Для того чтобы переименовать столбец с примененной агрегатной функцией, необходимо в режиме конструктора в строке **Поле** соответствующего столбца ввести выражение **ИмяСтолбца: ИмяПоля**, например, **Количество заказов: № заказа** (рис. 27).

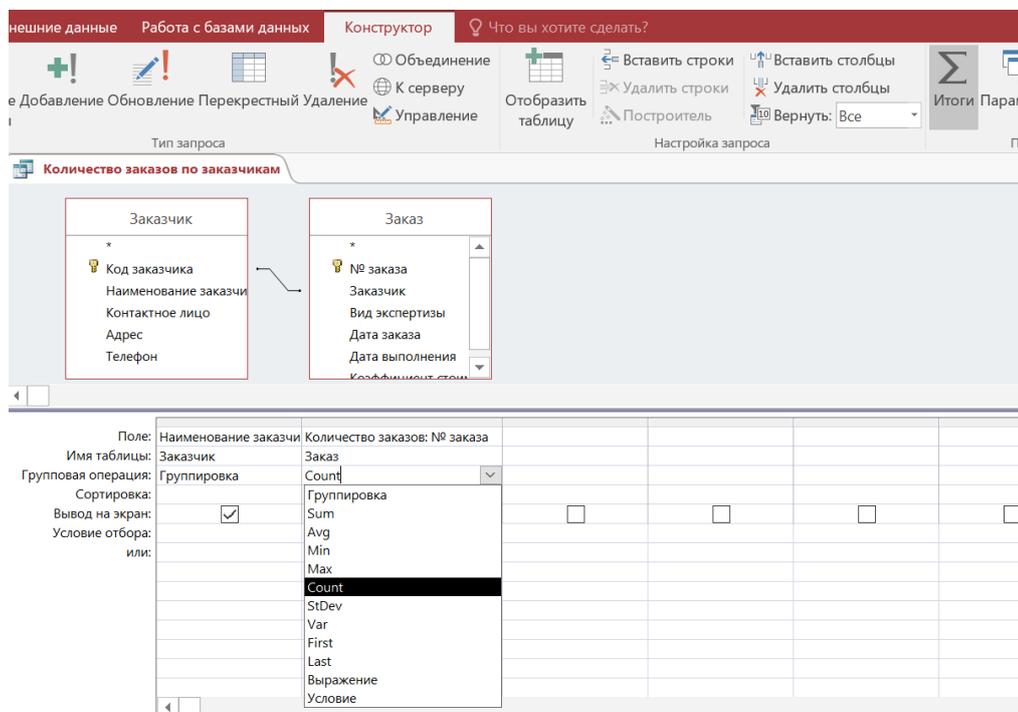


Рисунок 27 – Выбор групповой операции в аналитическом запросе

Задание 3.13. Создайте запрос **Количество заказов по сотрудникам**, определяющий количество заказов, выполняемых каждым сотрудником (вывести фамилии, имена и отчества сотрудников, а не их табельные номера).

Перекрестные запросы

Ранее мы уже рассматривали создание перекрестных запросов с помощью мастера. Существенным недостатком использования мастера является невозможность включения данных из нескольких таблиц (только если предварительно создать вспомогательный запрос). Режим Конструктора позволяет устранить этот недостаток.

В режиме Конструктора необходимо указать, что запрос будет особого типа – перекрестный, это делается командой **Перекрестный** в группе **Тип запроса** ленты **Конструктор** (см. рис. 28).

При этом в табличной части окна появляются строки **Групповая операция** и **Перекрестная таблица**, которые являются отличительными чертами запросов такого вида. Строка **Перекрестная таблица** предусматривает следующие значения:

Заголовки строк/столбцов – поля, значения которых группируются по строкам и столбцам соответственно и являются их заголовками.

Значение – поле, подвергающееся анализу, т. е. к значениям которого применяется указанная в групповой операции агрегатная функция.

(не отображается) – поля, которые применяются для установки условий.

Рассмотрим создание перекрестного запроса на примере задания 3.14.

Задание 3.14. Создайте запрос **Количество сотрудников** по должностям в филиалах, выводящий количество сотрудников по каждой должности в каждом из городов.

Пояснения по выполнению задания 3.14:

- В Конструкторе создайте запрос, включив в него поля **Город**, **Должность**, **Табельный №** из таблиц **Филиал** и **Сотрудник** соответственно.
- Задайте тип запроса – **перекрестный**.
- В строке **Перекрестная таблица** установите значение **Заголовки строк** для поля **Город** и **Заголовки столбцов** для поля **Должность** (можно наоборот).
- Определите групповую операцию **Count** для поля **Табельный №** для вычисления количества сотрудников по их табельным номерам.

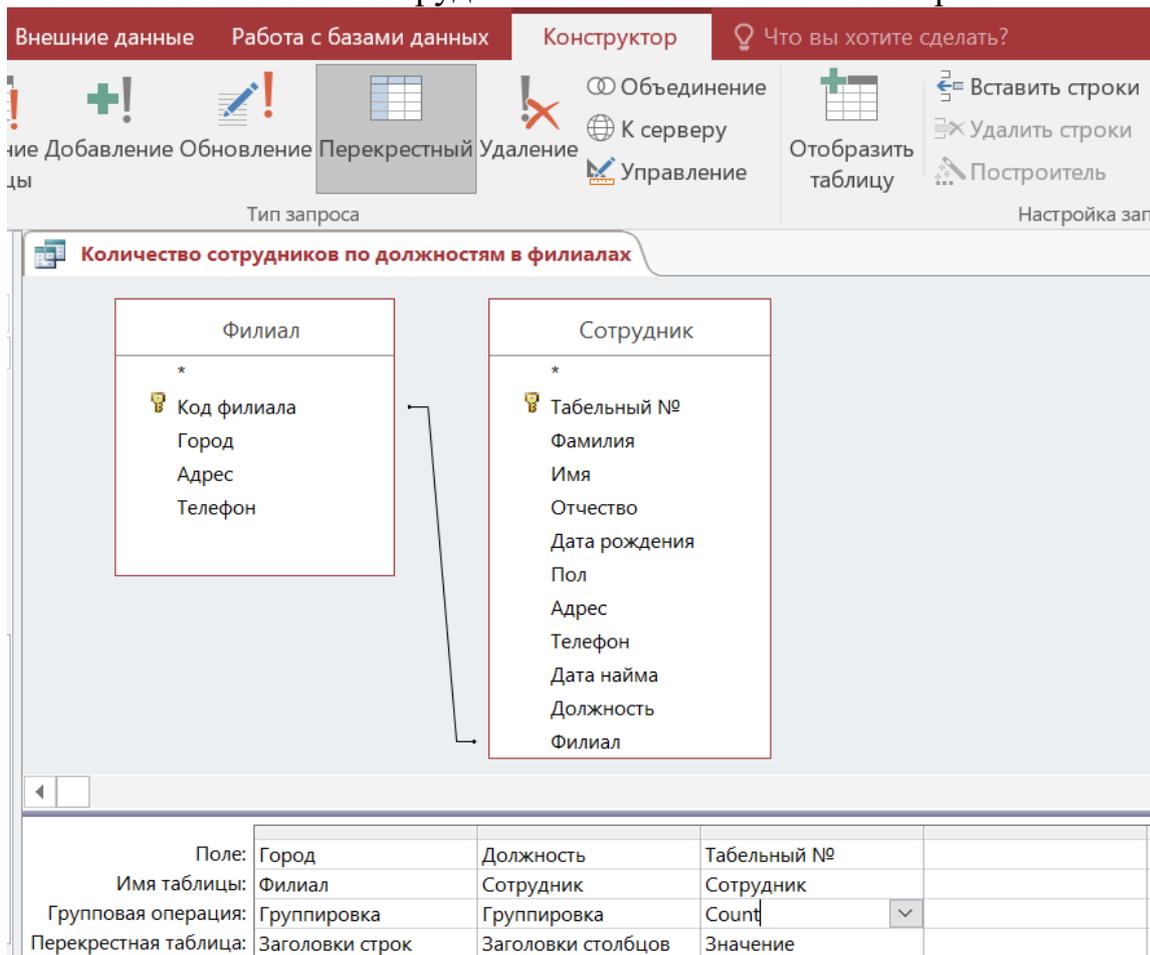


Рисунок 28 – Пример перекрестного запроса в режиме Конструктора

Задание 3.15. Создайте перекрестный запрос **Суммы заказов** по месяцам и видам экспертиз, показывающий общую сумму стоимости всех заказов по месяцу даты заказа для каждого вида экспертизы.

Пояснения по выполнению задания 3.15: для создания этого запроса используйте созданный ранее запрос **Стоимость заказов**. То есть создавайте новый запрос, но вместо таблиц выбирайте имеющийся запрос.

Лабораторная работа № 4

Отчеты в Access

Отчет – это зафиксированный результат анализа и обработки данных, поэтому он не используется как средство ввода или редактирования данных. Отчеты Access – это оптимальное средство для подготовки данных к печати. Конечно, печать данных можно осуществить непосредственно из таблицы или формы, но лишь при помощи отчетов можно удобно представить и структурировать данные, увидеть их в том виде, в котором они будут выведены на лист бумаги посредством принтера.

Для создания отчета необходимо перейти на вкладку **Создание** и в группе **Отчеты** выбрать один из способов создания отчета: **Отчет**, **Конструктор отчетов**, **Пустой отчет**, **Мастер отчетов**.

Режимы представления отчета аналогичны режимам представления форм: **режим представления отчета**, **макета** и **конструктора**, плюс дополнительный режим **предварительный просмотр**.

Режимы представление отчета и **предварительного просмотра** позволяют просматривать данные, но не позволяют менять структуру отчета.

Для изменения структуры предназначены другие два режима.

Режим Макета представляет собой наиболее наглядный режим изменения структуры отчета. Поскольку в режиме макета отображаются реальные данные, то в этом режиме удобно, например, задавать размер элементов управления.

Режим Конструктора обеспечивает более подробное представление структуры отчета. В нем отображаются разделы заголовка, примечания, области данных, колонтитулов. В этом режиме видна только структура отчета, но не видны сами данные. Многие задачи удобнее выполнять в режиме Конструктора, а не Макета, например, добавление в отчет дополнительных элементов управления, группировок и сортировок данных, изменение размеров разделов отчета, изменение некоторых свойств отчета.

Переключение между режимами полностью аналогично приемам работы с другими типами объектов в Access.

Изменение отчетов

В **режиме макета** изменить свойства отчета, его разделов и элементов управления можно с помощью стандартного **окна свойств**, которое вызывается командой **Страница свойств** (вкладка **Конструктор**, группа **Сервис**) или клавишей F4.

Можно добавить в макет отчета поля из таблиц – из области **Список полей**, который вызывается командой **Добавить поля** (вкладка **Конструктор**, группа **Сервис**) или клавишами ALT+F8.

В **режиме Конструктора** можно более детально доработать отчет: добавить новые элементы управления и поля, разместив их на сетке макета. В окне свойств доступны все свойства настройки отчета.

Вызов окна свойств и списка полей осуществляется точно так же, как и в режиме макета.

Создание простого отчета

Самый простой способ создания отчета – с помощью инструмента **Отчет**:

- в области навигации выделить нужную таблицу (запрос);
- на вкладке **Создание** в группе **Отчеты** выбрать команду **Отчет**.

Будет создан новый отчет и отображен в режиме **Макета**. При этом все поля выделенной таблицы размещаются в отчете. Можно сразу же перейти в режим представления отчета или предварительного просмотра и начать использование нового отчета, либо при необходимости изменить его в режиме **Макета** или **Конструктора**.

Задание 4.1. Создайте простой отчет по таблице **Филиал**. В режиме **Макета** приведите его к удобному виду.

Использование Мастера отчетов

Мастер отчетов позволяет проконтролировать создание отчета более подробно. Для его вызова можно воспользоваться командой **Мастер отчетов** на вкладке **Создание** группа **Отчеты**.

В общем случае создание отчета с помощью Мастера содержит следующие шаги:

1-й шаг – выбор таблиц и полей, которые нужно включить в отчет (можно выбрать поля из разных таблиц или запросов).

2-й шаг – выбор полей группировки. Отчеты становятся более понятными и структурированными, если данные в них организованы в группы. Группировка – это разделение всех записей по группам, определяемым значением какого-либо поля или выражения. Здесь нужно выбрать одно или несколько полей для группировки, а затем можно определить вид группировки (кнопка **Группировка**).

3-й шаг – выбор полей и порядка сортировки.

4-й шаг – выбор макета отчета из предложенного списка и выбор ориентации страницы отчета. Список макетов может быть различным в зависимости от того, производилась группировка записей или нет.

5-й шаг – выбор стиля отчета.

6-й шаг – задание имени отчета.

Задание 4.2.

А) Создайте отчет **Виды экспертных услуг** с использованием мастера по таблице **Вид экспертизы**, в котором будут отражены наименования видов экспертиз, их стоимости и результаты. Сгруппируйте услуги по наименованиям и отсортируйте по стоимости.

Б) В режиме **Макета** доработайте отчет **Виды экспертных услуг**, настроив ширину полей и их расположение таким образом, чтобы все данные были полностью видны на экране. Наименования видов экспертиз выделите более крупным шрифтом.

Создание отчетов в режиме Конструктора

Конструктор отчетов предусматривает детальную проработку создания отчетов. Процесс создания отчета в этом режиме очень похож на создание форм с помощью конструктора форм. Конструктор отчетов вызывается командой **Конструктор отчетов**, расположенной на вкладке **Создание** в группе **Отчеты**.

Откройте отчет **Виды экспертных услуг** в режиме Конструктора (см. рис. 29) для исследования его структуры.

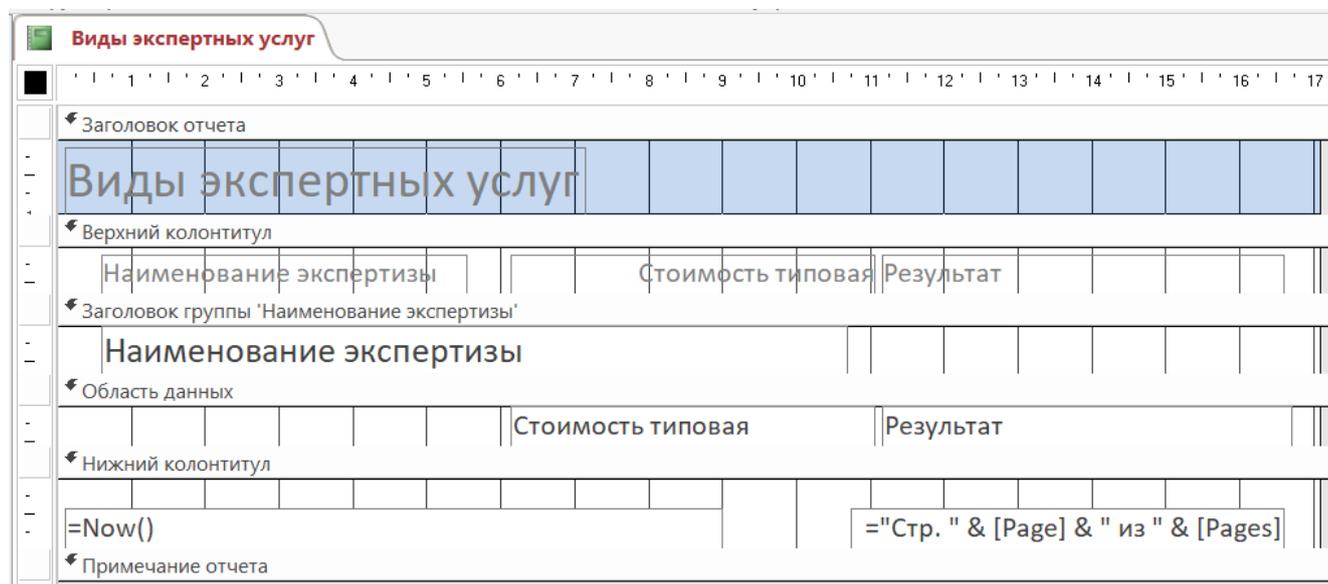


Рисунок 29 – Отчет в режиме Конструктора

Любой отчет состоит из нескольких разделов (см. табл. 2), количество которых может меняться в зависимости от структуры отчета. На рис. 29 присутствуют не все возможные разделы.

Таблица 2 – Описание разделов отчета в режиме Конструктора

Название раздела	Содержимое раздела	Вывод содержимого раздела
Заголовок отчета	Название отчета, пояснения к отчету и др.	Только в начале первой страницы отчета
Верхний колонтитул	Номера страниц, пояснения к страницам и др.	Вверху каждой страницы отчета
Заголовок группы	Поле группировки, пояснения к группе, поля с групповыми операциями и др.	В начале каждой группы записей
Область данных	Поля, значения которых будут выведены в основной части отчета	В основной части отчета
Примечание группы	Поля с групповыми операциями, пояснения к группе и др.	В конце каждой группы записей
Нижний колонтитул	Номера страниц, пояснения к страницам и др.	Внизу каждой страницы отчета
Примечание отчета	Поля с групповыми операциями по всему отчету, пояснения к отчету и др.	Только в конце последней страницы отчета

Группировка и сортировка в отчетах

Операцию группировки можно осуществить, находясь только в режимах Конструктора или Макета отчета через область **Группировка, сортировка и**

итоги (см. рис. 30), вызываемую с помощью команды  расположенной на вкладке **Конструктор** в разделе **Группировка и итоги**.

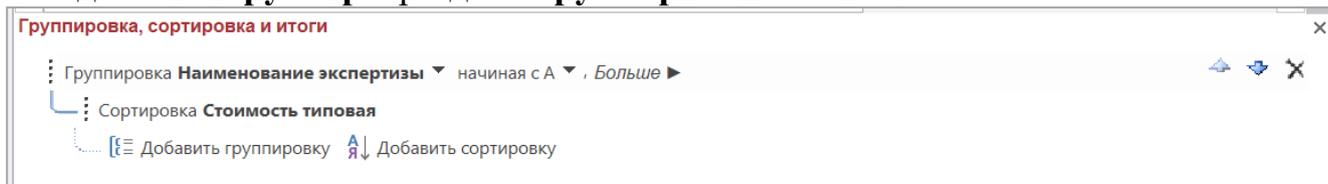


Рисунок 30 – Область группировки, сортировки и итогов в Конструкторе отчетов

В области **Группировка, сортировка и итоги** перечисляются поля, по которым уже осуществлена группировка и сортировка данных, а также отображаются кнопки, с помощью которых можно добавить новые уровни группировки и поля для сортировки.

Группировку можно производить по нескольким полям. Кроме того, для группировки можно ввести выражение, содержащее функции.

Для сортировки достаточно выбрать поле и порядок сортировки.

Задание 4.3. Создайте отчет **Заказы по датам** в режиме Конструктора таким образом, чтобы выводился список заказов, сгруппированный по датам заказов. В основной части отчета должны содержаться номера заказов и наименования заказчиков. Оформите отчет наиболее наглядно.

Примечание (ВАЖНО!): в MS Access начиная с версии 2010 есть такая особенность: когда в отчетах делается группировка по дате, по умолчанию устанавливается группировка по кварталам, поэтому необходимо сразу изменить параметры группировки.

Примерный вид отчета в режиме предварительного просмотра – на рис. 31:

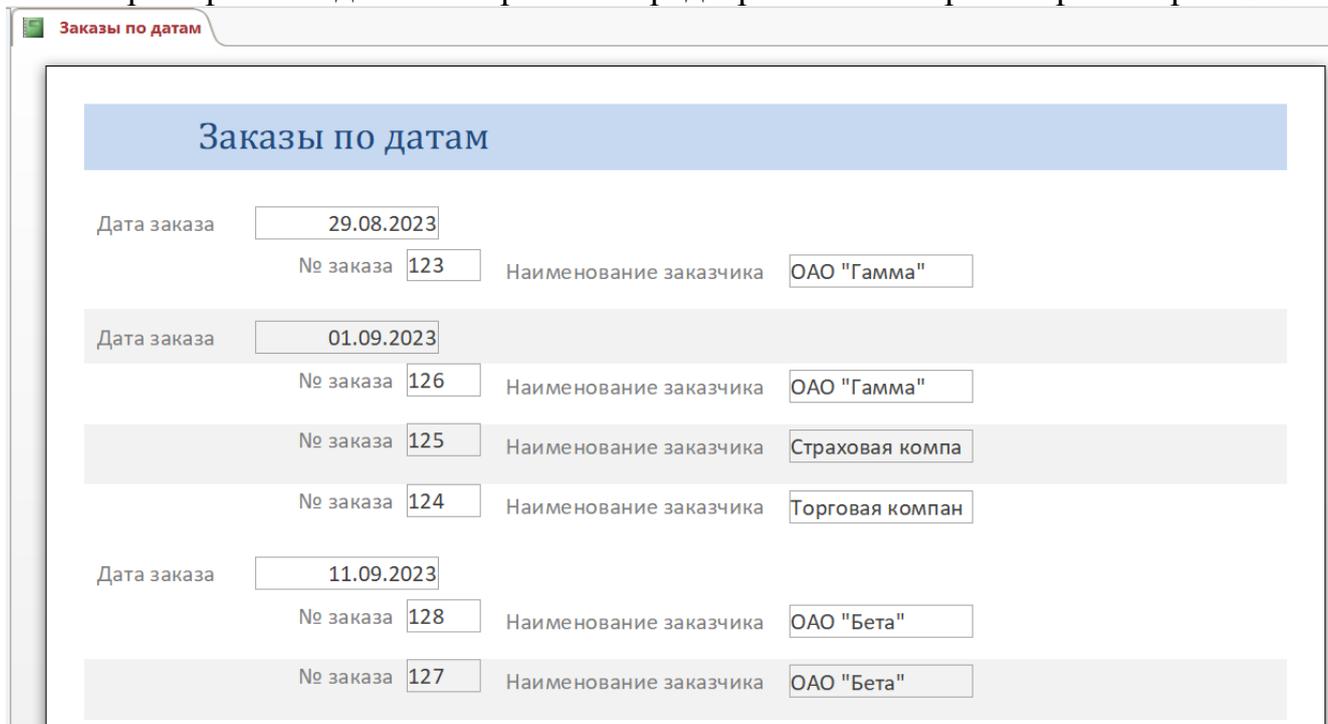


Рисунок 31 – Вид отчета в режиме предварительного просмотра

Рекомендации по выполнению задания 4.3:

Откройте Конструктор отчетов (без указания конкретной таблицы) и отобразите на экране **Список полей**.

В область данных отчета перетащите поля **№ заказа** и **Дата заказа** из таблицы **Заказ** и поле **Наименование заказчика** из таблицы **Заказчик**.

Отобразите на экране область **Группировка, сортировка и итоги** (Группировка). Добавьте в отчет группировку по полю **Дата заказа**.

В появившийся раздел заголовка группы перетащите поле **Дата заказа** (выделив и подпись, и само поле), а остальные поля оставьте в **Области данных** (см. рис. 32).

Поля в области данных расположите наиболее наглядно.

Для даты заказа настройте группировку по дням (рис. 33).

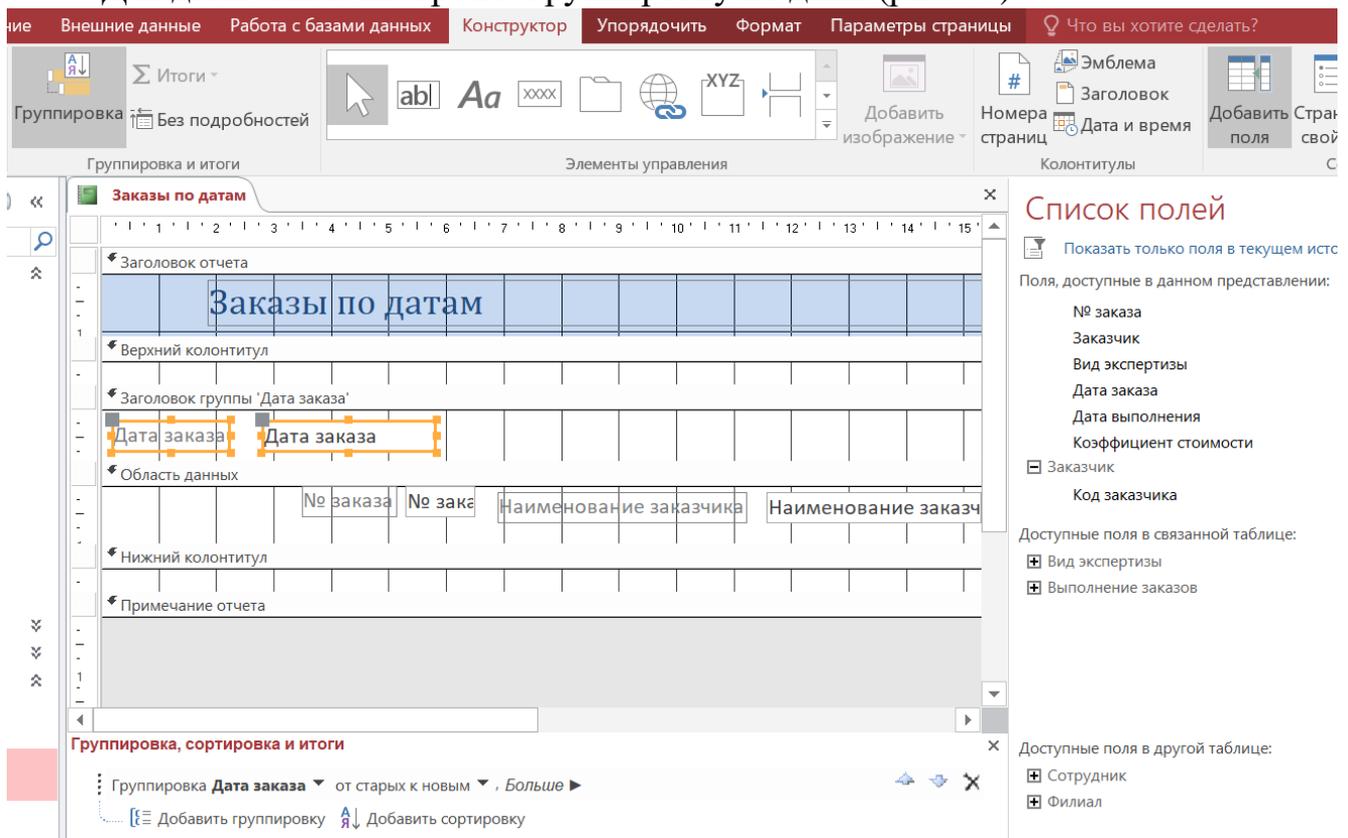


Рисунок 32 – Создание отчета с группировкой в режиме Конструктора отчета

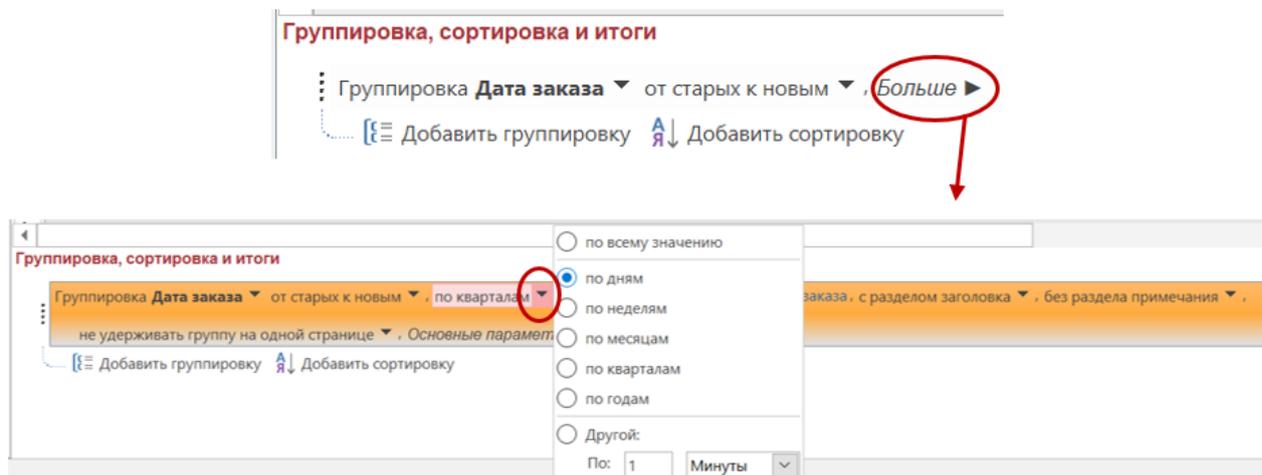


Рисунок 33 – Настройка группировки по дням

Вычисляемые поля в отчетах

Вычисляемые поля в отчетах используются в двух общих случаях:

- для вычисления значений нового вычисляемого поля. Например, для объединения фамилии и инициалов в одно поле, для вывода в новом поле возраста, вычисленного по дате рождения и т. д. В качестве *частного случая* нового вычисляемого поля можно привести пример вставки поля для нумерации записей;

- для получения итоговых значений по группе. Например, подсчет количества сотрудников в отделах, общей суммы выручки от продажи всех видов товаров.

Для создания вычисляемого поля необходимо в режиме Конструктора добавить элемент управления **Поле**  (лента Конструктор, группа **Элементы управления**). В это поле вручную или через свойства поля при помощи **Построителя выражений** внести нужное выражение, начинающееся со знака «=». Выражения в вычисляемом поле могут содержать функции, имена полей и конкретные значения.

Задание 4.4. В созданный ранее отчет **Заказы по датам** добавьте вычисляемое поле, в котором будет выводиться количество дней, прошедших с даты заказа.

Пояснения по выполнению задания 4.4:

- Откройте отчет **Заказы по датам** в режиме Конструктора.
- Добавьте элемент управления **Поле** в область заголовка группы.
- В него введите формулу, вычисляющую разность текущей даты и даты заказа (см. рис. 34). Функция вывода текущей даты может быть Date() либо Дата() в зависимости от версии Access.

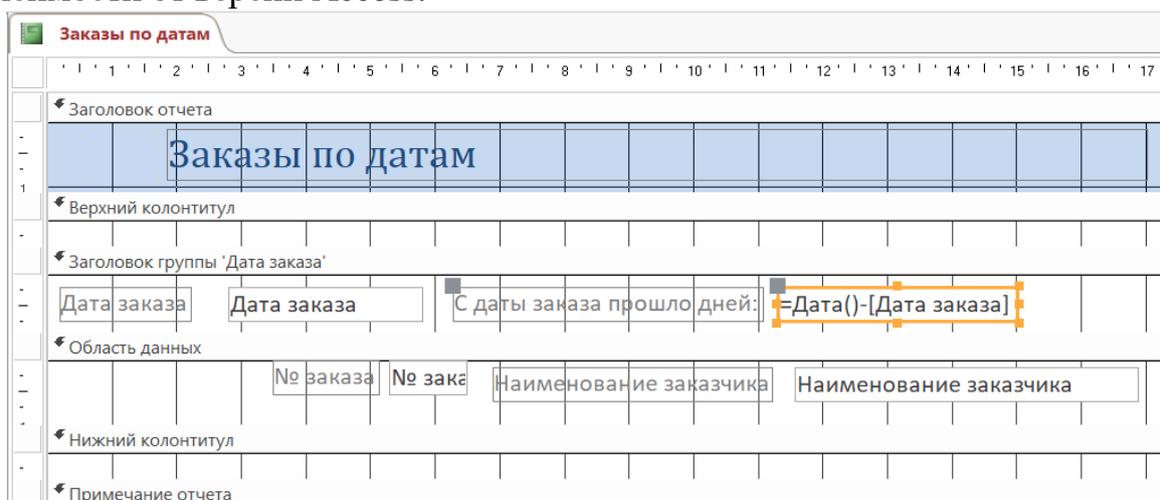


Рисунок 34 – Вычисляемое поле

Для **подведения итогов** есть функции, аналогичные функциям, используемым для создания аналитических запросов.

В Access начиная с версии 2010 для автоматического подведения итогов по группам есть специальная команда **Итоги** , расположенная на вкладке **Конструктор** в разделе **Группировка и итоги**. Для подведения итогов нужно выделить поле, по которому подводим итоги, нажать эту команду **Итоги** и выбрать нужную операцию: число записей, сумму, среднее значение или другие. Например, для того чтобы подсчитать количество сотрудников, сгруппированных по какому-либо признаку, нужно выделить поле, характеризующее сотрудника (например, **Табельный номер**) и в списке итогов выбрать команду **Число записей**.

Задание 4.5. В отчет **Заказы по датам** добавьте информацию о количестве заказов в каждую дату. Сделайте так, чтобы каждая дата начиналась с новой страницы. Добавьте нумерацию страниц.

Подсказка: для того, чтобы каждый день начинался с новой страницы, добавьте в раздел **Примечания группы** элемент управления **Вставить разрыв страницы**. Команда **Номера страниц** находится на вкладке **Конструктор**. Эффект разрыва страницы будет виден только в режиме предварительного просмотра.

Создание отчета при помощи инструмента Пустой отчет

Для создания отчета можно воспользоваться инструментом **Пустой отчет** (вкладка **Создание**, группа **Отчеты**). При запуске этого инструмента не надо выделять никакого источника данных, поскольку в любом случае будет открыт совершенно пустой новый отчет в режиме Макета. Также на экране откроется список полей, откуда можно перетаскивать нужные поля в пустой отчет.

Задание 4.6. Используя инструмент **Пустой отчет**, создайте отчет **Заказчики**, показывающий перечень всех данных обо всех заказчиках.

В любом подходящем режиме доработайте отчет так, чтобы при печати все сведения выводились понятно и корректно.

Задание 4.7. Любым способом создайте отчет **Сотрудники** так, чтобы выводилась информация о каждом сотруднике. Сделайте группировку по филиалам с подсчетом количества сотрудников в каждом филиале. Сделайте так, чтобы список сотрудников каждого филиала начинался на новой странице.

Лабораторная работа № 5

Главная кнопочная форма

В Access существует надстройка, которая позволяет создать своеобразный путеводитель по объектам базы данных – кнопочную форму. Когда форм и отчетов в базе данных становится много, полезно иметь систему указателей, которая позволит ориентироваться среди множества объектов. **Главная кнопочная форма** (рис. 35) может иметь иерархическую структуру и содержит кнопки, позволяющие выбирать основные объекты или функции.

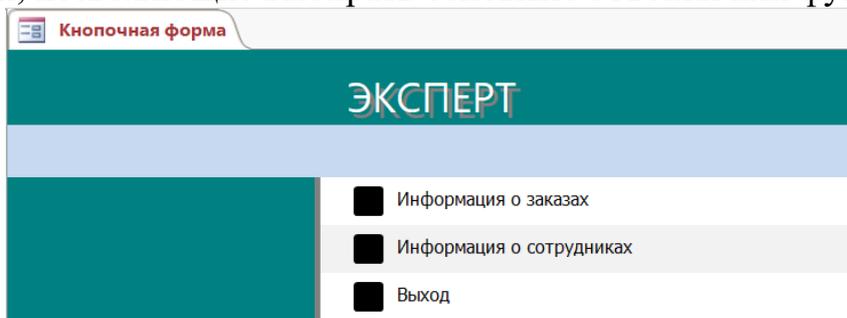


Рисунок 35 – Главная кнопочная форма

Эту форму можно создать с помощью специального мастера – Диспетчер кнопочных форм (вкладка **Работа с базами данных**).

Примечание: если команда **Диспетчер кнопочных форм** отсутствует на вкладке **Работа с базами данных**, то ее нужно вручную добавить. Для этого нужно:

- выполнить команду **Файл → Параметры**;
- в окне **Параметры Access** в разделе **Настройка ленты** выбрать команду **Диспетчер кнопочных форм** из списка всех команд.
- В разделе **Настройка ленты**, расположенной в правой части окна, на вкладке **Работа с базами данных** создать группу с именем **Кнопочная форма** и добавить в нее команду **Диспетчер кнопочных форм**.

Создание кнопочной формы и страниц кнопочной формы

При запуске **Диспетчера кнопочных форм**, если таких форм еще нет, выдается диалоговое окно, запрашивающее, нужно ли создать кнопочную форму. В случае утвердительного ответа мастером создается таблица **Swichboard Items** (Элементы кнопочной формы) и сама **Кнопочная форма**, а далее открывается диалоговое окно **Диспетчер кнопочных форм**.

В этом окне перечисляются все страницы кнопочной формы. Пока создана только одна страница, которая по умолчанию называется **Главная кнопочная форма**.

Для того, чтобы переименовать страницу, нажмите кнопку **Изменить**. В поле **Название кнопочной формы** можно ввести новое имя страницы (рис. 36).

Важно: при работе с диспетчером кнопочных форм обращайтесь внимание на заголовки открываемых диалоговых окон!

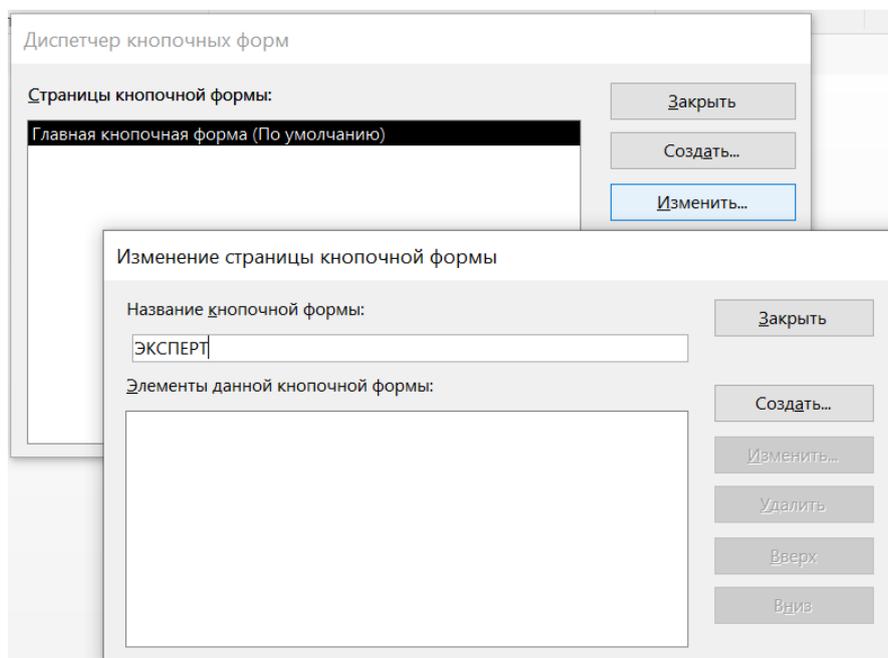


Рисунок 36 – Изменение названия страницы Главной кнопочной формы

Для создания новых страниц используйте кнопку **Создать**. При этом открывается окно **Создание**, в котором нужно ввести название новой страницы кнопочной формы.

Задание 5.1. Создайте Главную кнопочную форму вашей БД «ЭКСПЕРТ», состоящую из трех страниц: главную страницу назовите ЭКСПЕРТ, две другие –Заказы и Сотрудники.

***Примечание:** страницы Главной кнопочной формы пока будут пустыми, наполнение их будет выполнено в результате следующих заданий.*

Создание элементов (кнопок) на странице кнопочной формы

Страницы пока не содержат никаких элементов. Для создания кнопок на странице выделите ее и нажмите кнопку **Изменить**. Откроется окно **Изменение страницы кнопочной формы**.

Чтобы создать элемент, используйте кнопку **Создать**. Откроется диалоговое окно **Изменение элемента кнопочной формы** (рис. 37).

Здесь в поле **Текст** вводится название элемента (то, что мы увидим на форме), в поле **Команда** выбирается из списка действие, которое будет выполняться при нажатии этой кнопки, в поле **Кнопочная форма** вводится (или выбирается из списка) имя объекта, с которым будет произведено действие. Количество кнопок на одной странице ограничено восьмью.

Кроме того, что элементы можно добавлять, изменять и удалять, можно также менять их последовательность кнопками **Вниз** и **Вверх**.

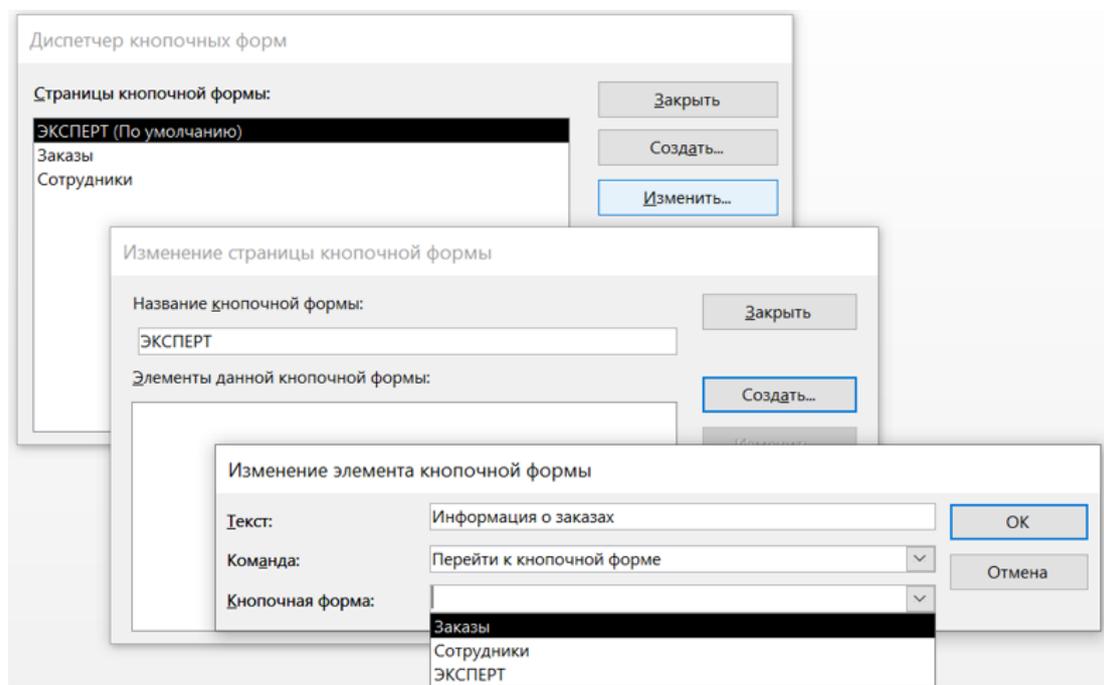


Рисунок 37 – Создание элемента кнопочной формы

Задание 5.2. На главной странице кнопочной формы сделайте кнопки перехода к страницам **Заказы** и **Сотрудники** и кнопку выхода из приложения MS Access (см. рис. 35). На странице **Заказы** создайте кнопки, открывающие формы и отчеты, связанные с заказами и заказчиками, а также кнопку возврата на Главную страницу. Аналогично поступите со страницей **Сотрудники**.

Оформление кнопочной формы

Внешний вид Главной кнопочной формы оформлен по умолчанию. Но если требуется изменить его, это можно сделать в режиме Конструктора. В окне базы данных на странице **Формы** присутствует созданная вами главная кнопочная форма под именем **Кнопочная форма**. Имя можно оставить таким или переименовать. В режиме Конструктора эта форма выглядит пустой, но если посмотреть свойства формы, то можно обнаружить команды. В режиме конструктора можно изменить цвет, вставить картинки и т. п.

!!!Важно: не удаляйте кнопки этой формы в режиме Конструктора!

Задание 5.3. Оформите Главную кнопочную форму по вашему усмотрению, добавьте надписи, вставьте картинку.

Для того чтобы при открытии базы данных сразу открывалась Главная кнопочная форма, в меню **Файл** выберите команду **Параметры**. В окне **Параметры Access** на вкладке **Текущая база данных** для параметра **Форма просмотра** выберите главную кнопочную форму.

Задание 5.4. Сделайте стартовой Главную кнопочную форму.

Задание по зачетной базе данных

Предметная область, по которой будет построена база данных – любая на ваш выбор.

Заранее необходимо сообщить преподавателю **тематику** базы данных.

Перед созданием базы данных в MS Access необходимо построить примерную **схему** базы данных и показать преподавателю.

База данных сдается в электронном виде **с пояснениями**.

Для того, чтобы БД была зачтена, нужно набрать **не менее 20 баллов**.

Задания	Баллы
1. База данных должна содержать не менее трех базовых таблиц , связанных между собой	8 <i>Из них:</i>
• корректность типов и свойств полей, наличие ограничений;	4
• правильность создания связей и их параметров, настройка ввода значений внешних ключей через поле со списком	4
2. Формы должны содержать элементы управления:	5 <i>Из них:</i>
• поле со списком, ссылающееся на данные из другой таблицы;	2
• кнопку, открывающую другую форму, связанную с текущими данными.	2
• рисунок или другие элементы оформления	1
3. Запросы следующих видов (или их комбинацией):	8 <i>Из них:</i>
• с условием;	2
• с параметром;	2
• с группировкой и использованием групповых операций (но не перекрестный);	2
• перекрестный.	2
4. В отчётах (отчёте) должны быть применены:	5 <i>Из них:</i>
• сортировка;	1
• группировка;	2
• вычисление групповых показателей (сумма, количество или др.).	2
5. Многостраничная главная кнопочная форма должна содержать ссылки на формы, отчеты.	3
6. Оформление (единый стиль, структурность).	1
ИТОГО	30

Список литературы

1. Дейт К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Диалектика, 2019 – 1328 с.
2. Ивашова О. Н. Проблемы обучения информационным технологиям студентов непрофильных направлений подготовки / О. Н. Ивашова, М. В. Петухова, Е. В. Щедрина // Международный журнал гуманитарных и естественных наук. – 2022. – № 3-1(66). – С. 31-34.
3. Ивашова О. Н. Система управления базами данных MS Access: практикум / О. Н. Ивашова, Ю. Р. Стратонович, Е. А. Яшкова; Российский государственный аграрный университет - МСХА имени К. А. Тимирязева — Москва: РГАУ-МСХА им. К. А. Тимирязева, 2020 — 58 с. —URL: <http://elib.timacad.ru/dl/local/umo488.pdf>
4. Исупова Т. Н. Возможности провоцирующих задач при реализации проблемного обучения на примере темы «Базы данных» / Т. Н. Исупова, М. В. Петухова // Вестник гуманитарного образования. – 2017. – № 2. – С. 20-26
5. Карпузова В. И. Информационные технологии в менеджменте. Проектирование информационной системы с использованием СУБД Access, аналитической платформы SAS Enterprise Guide 5/1: учебное пособие / В. И. Карпузова, К. В. Чернышева, Н. В. Карпузова; Российский государственный аграрный университет - МСХА имени К. А. Тимирязева — Москва: Росинформагротех, 2017 — 132 с. URL:<http://elib.timacad.ru/dl/local/t781.pdf>.
6. Лемешко Т. Б. Современные информационные технологии: учебное пособие / Т. Б. Лемешко, В. Н. Шурыгин; Российский государственный аграрный университет - МСХА имени К. А. Тимирязева — Москва: Росинформагротех, 2017 — 136 с. Режим доступа: <http://elib.timacad.ru/dl/local/t495.pdf>
7. Перевозчикова М. С. Практикум по программному обеспечению ЭВМ. Реляционные базы данных. Система управления базами данных Access / М. С. Перевозчикова, М. В. Петухова – Киров: Изд-во ВГГУ, 2005 – 112 с.

Петухова Мария Владиславовна
Гавриловская Надежда Владимировна
Ивашова Ольга Николаевна
Щедрина Елена Владимировна

ПРАКТИКУМ ПО ДИСЦИПЛИНЕ «БАЗЫ ДАННЫХ»

Учебно-методическое пособие

ISBN 978-5-00207-393-1



Подписано в печать 26.10.2023.

Формат 60×84 1/16. Гарнитура Times New Roman. Бумага офсетная.

Усл. печ. л. 2,91. Тираж 100 экз. Заказ № 4755-23.

Отпечатано в соответствии с предоставленными материалами
в ООО «Амирит», 410004, г. Саратов, ул. Чернышевского, 88.

Тел.: 8-800-700-86-33 | (845-2) 24-86-33

E-mail: zakaz@amirit.ru

Сайт: amirit.ru