

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО
ОБРАЗОВАНИЯ РЕСПУБЛИКИ УЗБЕКИСТАН

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ТАШКЕНТСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ НИЗАМИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«АСТРАХАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (АГУ)

ОКЛАДНИКОВА С.В., НАБИУЛИНА Л.М.,
ЗАКИРОВА М.Р.

Учебно-методическое пособие

«ВВЕДЕНИЕ В БАЗЫ ДАННЫХ: MS ACCESS»

5111000 – Профессиональное образование (5330400 Компьютерная
графика и дизайн)

090301 Информатика и вычислительная техника

090302 Информационные системы и технологии

Астрахань – 2019

ББК 32.972.134

УДК 004.65

Окладникова С.В. Учебно-методическое пособие «Базы данных»/Л.М. Набиулина, М.Р. Закирова/ – Астрахань: Издательство Сорокин Р.В., 2019. – 64 с.

Рецензенты:

Ханова Анна Алексеевна – д.т.н., профессор кафедры Прикладная информатика в экономике Астраханского государственного технического университета, Российская Федерация.

Турсунов Самар – к.пед.н., заведующий кафедры Методика преподавания информатики Ташкентского государственного педагогического университета им. Низами, Республика Узбекистан.

Учебно-методическое пособие предназначено для обучения студентов высших образовательных учреждений Республики Узбекистан и Российской Федерации по специальностям: 5111000 – Профессиональное образование (5330400 Компьютерная графика и дизайн), 090301 Информатика и вычислительная техника, 090302 Информационные системы и технологии, изучающих дисциплину «Базы данных».

В пособии изложены понятийный аппарат теории баз данных, основы проектирования в СУБД MS Access.

Для формирования профессиональных навыков студентов в области разработки баз данных в пособии представлены практические задания. Для закрепления приобретенных навыков по созданию БД в MS Access приведены задания для самостоятельной работы и вопросы для самоконтроля.

Учебно-методическое пособие рекомендовано студентам, аспирантам, преподавателям, слушателям институтов повышения и факультетов квалификации, изучающих основы создания баз данных.

© Издательство Сорокина Р.В., 2019

© Окладникова С.В., Набиулина Л.М., Закирова М.Р.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ЛАБОРАТОРНАЯ РАБОТА 1. ЗНАКОМСТВО С MS ACCESS. СОЗДАНИЕ ПРОСТЫХ ТАБЛИЦ.....	6
Теоретическая часть	6
Основная часть.....	8
ЛАБОРАТОРНАЯ РАБОТА 2. СОЗДАНИЕ БД, СОСТОЯЩЕЙ ИЗ НЕСКОЛЬКИХ ТАБЛИЦ.....	10
Теоретическая часть	10
Основная часть.....	13
Задания для самостоятельного выполнения	17
ЛАБОРАТОРНАЯ РАБОТА 3. СОЗДАНИЕ ЗАПРОСОВ	23
Теоретическая часть	23
Основная часть.....	24
Задания для самостоятельного выполнения	30
ЛАБОРАТОРНАЯ РАБОТА 4. СОЗДАНИЕ ФОРМ.....	37
Теоретическая часть	37
Основная часть.....	38
Задания для самостоятельного выполнения	42
ЛАБОРАТОРНАЯ РАБОТА 5. СОЗДАНИЕ ОТЧЕТОВ	43
Теоретическая часть	43
Основная часть.....	43
Задания для самостоятельного выполнения	46
ЛАБОРАТОРНАЯ РАБОТА 6. СОЗДАНИЕ МАКРОСОВ В ACCESS	49
Теоретическая часть	49
Основная часть.....	49
ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ	61
СПИСОК ЛИТЕРАТУРЫ.....	63
ИНФОРМАЦИЯ ОБ АВТОРАХ.....	64

ВВЕДЕНИЕ

В современном мире объемы информации, требующей хранения, увеличиваются ежесекундно. Поэтому использование баз данных является неотъемлемой частью профессиональной деятельности человека.

Простейшие базы данных можно создавать, не прибегая к специальным программным средствам: например, в электронных таблицах Excel. Но существует несколько отличий между ними. Во-первых, системы управления базами данных (СУБД) разрабатываются с целью обеспечения эффективной обработки больших объемов информации, намного больших, чем те, с которыми справляются электронные таблицы. Во-вторых, СУБД может легко связывать две таблицы так, что для пользователя они будут представляться одной таблицей. Реализовать такую возможность в электронных таблицах практически невозможно.

В-третьих, СУБД минимизирует общий объем базы данных. Для этой же цели таблицы, содержащие повторяющиеся данные, разбиваются на несколько связанных таблиц.

СУБД Microsoft Access, входит в пакет Microsoft Office. Основное назначение Access – дать пользователю простое и доступное средство для создания таких баз данных, которые ему нужны. Несмотря на простоту этой СУБД, она позволяет создавать приложения довольно сложной структуры. При желании систему можно развивать и настраивать собственными силами, используя язык программирования VBA.

Еще одним достоинством Access является ее интегрированность с Word, Excel и другими программами пакета Microsoft Office. Данные легко импортируются и экспортируются из одного приложения в другое.

Методические указания включают в себя шесть лабораторных работ, индивидуальные задания и перечень контрольных вопросов.

Индивидуальные задания выполняются студентами самостоятельно и оформляются в виде отчета о лабораторных работах.

Содержание отчета:

1. Титульный лист.
2. Постановка задачи (формулировка индивидуального задания).

3. Ход разработки базы данных (подробное описание действий, выполненных при разработке базы данных).
4. Заключение (основные итоги и выводы, предложения и рекомендации по использованию полученных результатов в практической деятельности).
5. Библиографический список.
6. Приложения (таблицы, схема данных, формы, запросы и отчеты, созданные при разработке базы данных).
7. Оглавление (автоматическое).

ЛАБОРАТОРНАЯ РАБОТА 1. ЗНАКОМСТВО С MS ACCESS. СОЗДАНИЕ ПРОСТЫХ ТАБЛИЦ

Цель: сформировать представление об особенностях создания БД в СУБД Access.

Теоретическая часть

База данных (БД) – упорядоченная совокупность данных, предназначенных для хранения, накопления и обработки с помощью ЭВМ. Для создания и ведения баз данных (их обновления, обеспечения доступа по запросам и выдачи данных по ним пользователю) используется набор языковых и программных средств, называемых *системой управления базами данных (СУБД)*.

К **объектам** базы данных Access относятся:

Таблицы – предназначены для упорядоченного хранения данных.

Запросы – предназначены для поиска, извлечения данных и выполнения вычислений.

Формы – предназначены для удобного просмотра, изменения и добавления данных в таблицах.

Отчеты – используются для анализа и печати данных.

Страницы доступа к данным – предназначены для просмотра, ввода, обновления и анализа данных через сеть или из любого места компьютера.

Макросы – используются для выполнения часто встречающегося набора макрокоманд, осуществляющих обработку данных.

Модули – предназначены для описания инструкций и процедур на языке VBA.

Основным объектом базы данных является таблица, которая состоит из записей (строк) и полей (столбцов). На пересечении записи и поля образуется ячейка, в которой содержатся данные.

Каждому полю таблицы присваивается уникальное имя, которое не может содержать более 64 символов. В каждом поле содержатся данные одного типа (таблица 1.1).

Таблица 1.1 – Типы данных

Тип	Описание
Текстовый	Используется для хранения символьных или числовых данных, не требующих вычислений. В свойстве <i>Размер поля</i> задается максимальное количество символов, которые могут быть введены в данное поле. По умолчанию размер устанавливается в 50 знаков. Максимальное количество символов, которые могут содержаться в текстовом поле, – 255
Поле МЕМО	Предназначено для ввода текстовой информации, по объему превышающей 255 символов; может содержать до 65 536 символов
Числовой	Предназначен для хранения числовых данных, используемых в математических расчетах. На вкладках <i>Общие</i> и <i>Подстановка</i> можно установить свойства числового поля, среди которых <i>Размер поля</i> , <i>Формат поля</i> , <i>Число десятичных знаков</i>
Дата/Время	Используется для представления даты и времени. Выбор конкретного формата даты или времени устанавливается в свойстве <i>Формат даты</i>
Денежный	Предназначен для хранения данных, точность представления которых колеблется от 1 до 4 знаков после запятой. Целая часть может содержать до 15 десятичных знаков
Счетчик	Предназначен для автоматической вставки уникальных последовательных (увеличивающихся на 1) или случайных чисел в качестве номера новой записи. Номер, присвоенный записи, не может быть удален или изменен. Поля с этим типом данных используются в качестве ключевых полей таблицы
Логический	Предназначен для хранения одного из двух значений, интерпретируемых как «Да / Нет», «Истина / Ложь», «Вкл. / Выкл.»
Поле объекта OLE	Содержит данные, созданные в других программах, которые используют протокол OLE. Это могут быть, например, документы Word, электронные таблицы Excel, рисунки, звуковые и видеозаписи и др. Объекты OLE связываются с базой данных

Тип	Описание
	Access или внедряются в нее. Сортировать, группировать и индексировать поля объектов OLE нельзя
Гиперссылка	Специальный тип, предназначенный для хранения гиперссылок
Мастер подстановок	Предназначен для автоматического определения поля. С его помощью будет создано поле со списком, из которого можно выбирать данные, содержащиеся в другой таблице или в наборе постоянных значений

Основная часть

Задание 1.1. Создайте базу данных – «Записная книжка», состоящую из одной таблицы.

Ход выполнения задания

1. Запустите MS Access.
2. Создайте новую БД с сохраните ее с именем *Записная книжка.accdb*.
3. На вкладке **Создание** нажмите на кнопку **Конструктор таблиц**.
4. В окне **Конструктора** введите имена полей и укажите типы данных, к которым они относятся. Данные для описания структуры таблицы приведены на рисунке 1.1.

Имя поля	Тип данных
№ п/п	Счетчик
Фамилия	Текстовый
Имя	Текстовый
Адрес	Текстовый
Индекс	Числовой
Телефон	Текстовый
Хобби	Текстовый
Эл. почта	Гиперссылка

Рисунок 1.1 – Структура данных таблицы

5. Завершите работу с **Конструктором**. При выходе сохраните таблицу с именем *Друзья*. **КЛЮЧЕВЫЕ ПОЛЯ НЕ ЗАДАВАЙТЕ!**

6. В области навигации **Все объекты Access** откройте таблицу **Друзья** двойным щелчком мыши и заполните в ней 10 строк тестовыми данными. Тестовые данные необходимы для проверки работы БД.

7. Внесите изменения в структуру таблицы и добавьте в нее новые поля «Отчество» и «Дата рождения», для этого:

а) установите курсор на поле, перед которым нужно вставить новый столбец;

б) перейдите на вкладку ленты Поля → Режим → Режим таблицы → Вставить поле;

с) переименуйте новые поля в «Отчество» и «Дата рождения».

8. Перейдите в режим **Конструктора** и установите типы полей:

а) «Дата рождения» – Дата/время; в свойствах поля выберите Краткий формат даты.

б) «Отчество» – текстовый.

9. Переименуйте поле «Хобби» в «Увлечения».

10. В режиме **Таблица**

а) удалите запись под номером 8.

б) измените размер ячеек так, чтобы были видны все данные.

с) расположите поля в следующем порядке: «№», «Фамилия», «Имя», «Отчество», «Телефон», «Дата рождения», «Увлечения», «Адрес», «Индекс», «Фото», «Эл_почта».

д) заполните пустые ячейки таблицы.

11. В режиме **Конструктор** добавьте поле «Семейное положение», в котором будет содержаться фиксированный набор значений – *замужем/не замужем, женат/не женат*. Для создания раскрывающегося списка необходимо использовать **Мастер подстановок**:

а) установите тип данных **Мастер подстановок**;

б) в появившемся диалоговом окне выберите строку «Будет введен фиксированный набор значений» и нажмите кнопку **Далее**;

с) число столбцов – 1;

д) введите данные списка – *замужем, не замужем, женат, не женат*;

е) нажмите кнопку **Готово**.

12. С помощью раскрывающегося списка заполните столбец «Семейное положение». Поскольку таблица получилась широкая, то при заполнении данного столбца возникают некоторые неудобства: не видно фамилии человека, для которого заполняется поле «Семейное

положение». Чтобы фамилия была постоянно видна при заполнении таблицы, необходимо воспользоваться командой **Закрепить столбцы** из контекстного меню поля «Фамилия».

13. Отформатируйте таблицу следующим образом:

- а) цвет сетки – темно-красный;
- б) цвет фона – голубой;
- с) цвет текста – темно-красный, размер – 12 пт, начертание – курсив.

ЛАБОРАТОРНАЯ РАБОТА 2. СОЗДАНИЕ БД, СОСТОЯЩЕЙ ИЗ НЕСКОЛЬКИХ ТАБЛИЦ.

Цель: сформировать навыки создания связей между таблицами БД в СУБД Access.

Теоретическая часть

СУБД Access относится к типу реляционных. Реляционная база данных представляет собой совокупность двумерных таблиц. Любая таблица реляционной базы данных состоит из строк, называемых записями, и столбцов, называемых полями. Строки таблицы содержат сведения об объектах. Каждый столбец в таблице должен содержать только определенный тип информации. Каждая строка таблицы содержит разнообразную (разного типа) информацию. Все данные, помещенные в одной строке, называют записью, каждый элемент записи – это поле. Таким образом, каждое поле содержит часть информации, находящейся на пересечении соответствующей строки и столбца. В таблице всевозможные значения одного типа в одном столбце называют доменом.

Поле является элементом записи и представляет собой ячейку таблицы. У каждого столбца есть свое неповторимое имя, описывающее тот вид информации, который содержится в нем. Это имя называют «именем поля базы данных»

Каждое поле имеет фиксированную длину, следовательно, и любая запись в таблице имеет фиксированную длину. Каждая запись характеризуется своим уникальным порядковым номером.

Данные в реляционной таблице должны удовлетворять следующим принципам:

1. Каждое значение поля должно быть атомарным, т.е. не

расчленяемым на несколько значений;

2. Значения данных домена (в одном и том же столбце) должны принадлежать к одному и тому же типу данных, доступному для использования в данной СУБД;
3. Каждая запись в таблице уникальна, т.е. в таблице не существует двух записей с полностью совпадающим набором значений ее полей;
4. Каждое поле имеет уникальное имя;
5. Последовательность полей в таблице несущественна;
6. Последовательность записей в таблице несущественна.

Существенное отличие реляционной модели от обыкновенного последовательного файла заключается в том, что все столбцы в таблице с точки зрения входа предполагаются эквивалентными. Именно это свойство делает эту модель весьма мощной и делает невозможным отображение ее на память в виде последовательного массива данных.

Поскольку записи в таблице не упорядочены, то необходимо указать поле (или набор нескольких полей) для уникальной идентификации каждой записи.

Первичный ключ – это поле или набор полей, которые однозначно идентифицируют (определяют) запись таблицы.

Обычно ключом является поле или совокупность полей фиксированной длины. Каждому значению первичного ключа соответствует одна и только одна запись. Первичный ключ любой таблицы обязан содержать уникальные непустые значения для каждой записи. Если первичный ключ состоит из нескольких полей, он называется составным первичным ключом (primary key).

Поле, указывающее на запись в другой таблице, связанную с данной записью, называется внешним ключом (foreign key).

Подобное взаимоотношение между таблицами называется связью (relationship). Связь между двумя таблицами устанавливается путем присвоения значений внешнего ключа одной таблице значениям первичного ключа другой. Таблица, содержащая внешний ключ, называется второстепенной, а таблица, содержащая первичный ключ, определяющий возможные значения внешнего ключа второстепенной таблицы, называется главной.

Типичная реляционная база данных состоит из нескольких связанных таблиц.

Все информационные объекты предметной области связаны

между собой. Соответствия, отношения, возникающие между объектами предметной области, называются **связями**. Связанные отношениями таблицы взаимодействуют по принципу главная, подчиненная. Возможны следующие отношения между таблицами:

1. Отношение «один – ко – многим» (обозначают 1:M): одной записи из главной таблицы может соответствовать ноль, одна или несколько записей подчинённой таблицы.

2. Отношение «один – к - одному» (обозначают 1:1): одной записи из главной таблицы соответствует только одна запись из подчинённой таблицы.

3. Отношение «многие – ко – многим» (обозначают 1:1): одной записи из главной таблицы может соответствовать ноль, одна или несколько записей подчинённой таблицы и наоборот.

Одним из правил ссылочной целостности (referential integrity) является то, что первичный ключ любой таблицы должен содержать уникальные непустые значения для данной таблицы. Некоторые СУБД могут контролировать уникальность первичных ключей. Если СУБД контролирует уникальность первичных ключей, то при попытке присвоить первичному ключу значение, уже имеющееся в другой записи, СУБД сгенерирует диагностическое сообщение, обычно содержащее словосочетания primary key violation. Это сообщение в дальнейшем может быть передано в приложение, с помощью которого конечный пользователь манипулирует данными.

Если две таблицы связаны соотношением главная-подчиненная, внешний ключ подчинённой таблицы должен содержать только те значения, которые имеются среди значений первичного ключа главной таблицы. Если корректность значений внешних ключей не контролируется СУБД, можно говорить о нарушении ссылочной целостности. Если же СУБД контролирует корректность значений внешних ключей, то при попытке присвоить внешнему ключу значение, отсутствующее среди значений первичных ключей главной таблицы, либо при удалении или модификации записей главной таблицы, приводящих к нарушению ссылочной целостности, СУБД сгенерирует сообщение, о котором говорилось выше.

Основная часть

Задание 2.1. Создайте базу данных Фирма, состоящую из трех таблиц *Сотрудники*, *Клиенты* и *Заказы*, и установите связи между таблицами.

Ход выполнения задания:

1. Запустите Microsoft Access.
2. Создайте базу данных **Фирма**, состоящую из 3 таблиц: *Сотрудники*, *Клиенты* и *Заказы*. Структура таблиц описана на рисунках 2.1-2.3.

Имя поля	Тип данных
Код сотрудника	Счетчик
Фамилия	Текстовый
Имя	Текстовый
Отчество	Текстовый
Должность	Текстовый
Телефон	Текстовый
Адрес	Текстовый
Дата рождения	Дата/Время
Заработная плата	Денежный
Фото	Объект OLE
Эл_почта	Гиперссылка

Рисунок 2.1 – Структура таблицы «Сотрудники»

Имя поля	Тип данных
Код клиента	Счетчик
Название компании	Текстовый
Адрес	Текстовый
Номер телефона	Текстовый
Факс	Числовой
Адрес электронной почты	Гиперссылка
Заметки	Поле MEMO

Рисунок 2.2 – Структура таблицы «Клиенты»

Имя поля	Тип данных
Код заказа	Счетчик
Код клиента	Числовой
Код сотрудника	Числовой
Дата размещения	Дата/Время
Дата исполнения	Дата/Время
Сумма	Денежный
Отметка о выполнении	Логический

Рисунок 2.3 – Структура таблицы «Заказы»

!	Отдельные таблицы, содержащие информацию по определенной теме, необходимо связать в единую структуру базы данных. Для связывания таблиц следует задать <i>ключевые поля</i> . Ключ состоит из одного или нескольких полей, значения которых <i>однозначно</i> определяют каждую запись в таблице.
---	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3. Используя **Конструктор** задайте каждой таблице ключевое поле. Для таблицы *Сотрудники* установите ключевое поле *Код сотрудника*, для таблицы *Клиенты* – ключевое поле *Код клиента*, для таблицы *Заказы* – *Код заказа*.

!	Наиболее подходящим в качестве ключевого поля является «Счетчик», так как значения в данном поле являются уникальными (т. е. исключают повторы). Если в таблице необходимо установить несколько ключевых полей, то выделить их можно, удерживая клавишу <i>Ctrl</i> .
---	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4. В таблице *Заказы* создайте раскрывающиеся списки для отображения сотрудников и клиентов. Для создания данного элемента управления необходимо воспользоваться **Мастером подстановки**. Для этого:

- а) откройте таблицу *Заказы* в режиме **Конструктора**.
- б) для поля *Код сотрудника* выберите тип данных **Мастер подстановок**.
- с) в появившемся окне выберите команду **Объект «столбец подстановки»** будет использовать значения из таблицы или запроса» и щелкните на кнопке **Далее**.

d) в списке таблиц выберите таблицу *Сотрудники* и щелкните на кнопке **Далее**.

e) в списке **Доступные поля** выберите поле *Код сотрудника* и щелкните на кнопке со стрелкой, чтобы ввести поле в список *Выбранные поля*.

f) аналогично добавьте поля *Фамилия* и *Имя* и щелкните на кнопке **Далее**.

g) выберите порядок сортировки списка по полю *Фамилия*.

h) в следующем диалоговом окне задайте необходимую ширину столбцов раскрывающегося списка.


i) установите флажок **Скрыть ключевой столбец** и нажмите кнопку **Далее**.

j) на последнем шаге **Мастера подстановок** замените при необходимости надпись для поля подстановок и щелкните на кнопке **Готово**.

k) Аналогичным образом создайте раскрывающийся список для поля *Код клиента*.

5. Для создания связей между таблицами их предварительно необходимо **ЗАКРЫТЬ**. Создавать или изменять связи между открытыми таблицами нельзя.

6. Используя вкладку ленты **Работа с базами данных** вызовите окно **Схема данных**. Если ранее никаких связей между таблицами базы не было, то при открытии окна **Схема данных** одновременно открывается окно **Добавление таблицы**, в котором выберите таблицы *Сотрудники*, *Клиенты* и *Заказы*.

	Если связи между таблицами уже были заданы, то для добавления в схему данных новой таблицы щелкните правой кнопкой мыши на схеме данных и в контекстном меню выберите пункт <i>Добавить таблицу</i> .
-------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

7. Установите связь между таблицами *Сотрудники* и *Заказы*. Для этого:

a) в таблице *Сотрудники* выберите поле *Код сотрудника* и перенесите его на соответствующее поле в таблице *Заказы*.

b) после перетаскивания откроется диалоговое окно **Изменение связей** (рисунок 2.4), в котором включите флажок **Обеспечение условия целостности**. Это позволит предотвратить случаи удаления записей из одной таблицы, при которых связанные с ними данные других таблиц останутся без связи.

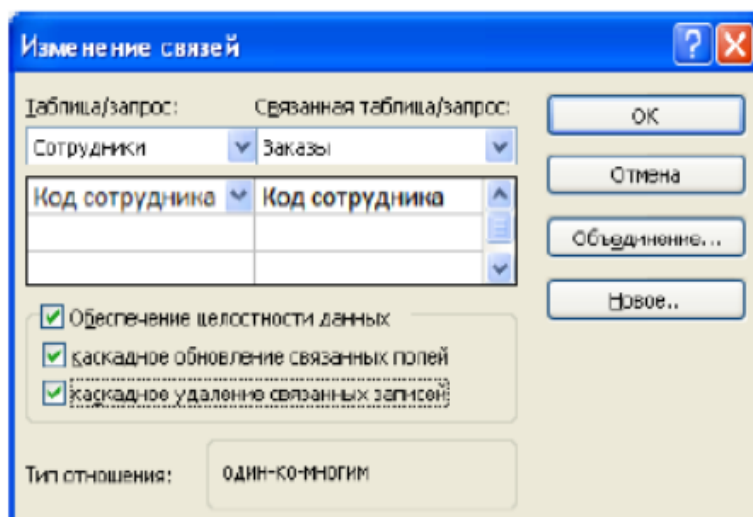


Рисунок 2.4 – Создание связи между таблицами

с) флажки **Каскадное обновление связанных полей** и **Каскадное удаление связанных записей** обеспечивают одновременное обновление или удаление данных во всех подчиненных таблицах при их изменении в главной таблице.

!	Параметры связи можно изменить, нажав на кнопку Объединение
---	--------------------------------------------------------------------

d) После установления всех необходимых параметров нажмите кнопку **ОК**.

8. Аналогично установите связь между таблицами *Клиенты* и *Заказы*.

9. В результате должна получиться схема данных, представленная на рисунке 2.5.

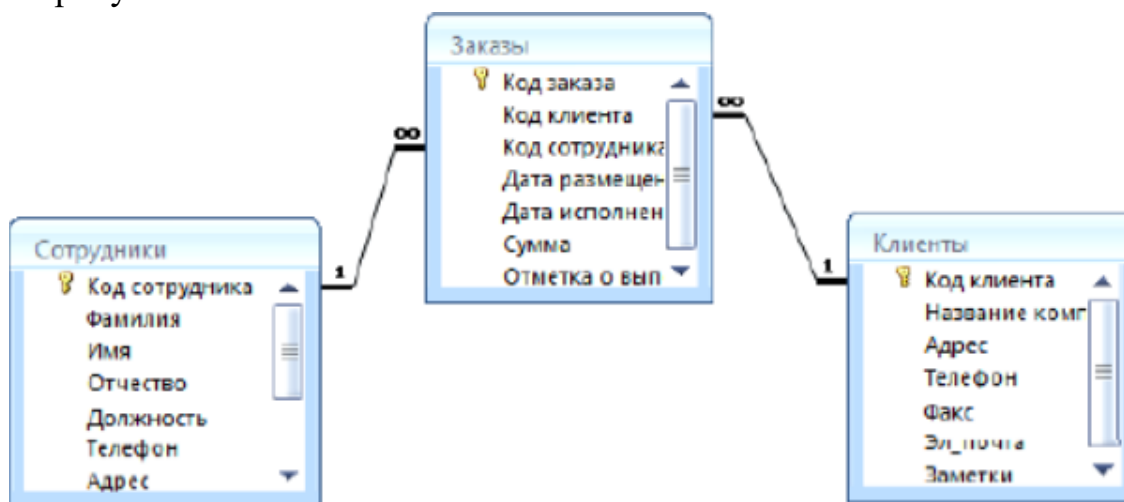


Рисунок 2.5 – Схема данных

!	<p>В приведенном примере используются связи «один-ко-многим». На схеме данных они отображаются в виде соединительных линий со специальными значками около таблиц. Связь «один-ко-многим» помечается «1» вблизи главной таблицы (имеющей первичный ключ) и «∞» вблизи подчиненной таблицы (имеющей внешний ключ). Связь «один-к-одному» помечается двумя «1» (оба поля таблиц имеют первичные ключи). Неопределенная связь не имеет никаких знаков. Если установлено объединение, то его направление отмечается стрелкой на конце соединительной линии (ни одно из объединенных полей не является ключевым и не имеет уникального индекса).</p>
---	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

10. В таблицу *Сотрудники* внесите тестовые данные о семи работниках.

11. В таблицу *Клиенты* внесите данные о десяти предприятиях, с которыми работает каждая фирма.

12. В таблице *Заказы* оформите несколько заявок, поступивших на фирму.

Задания для самостоятельного выполнения

Задание 2.2. Разработайте в MS Access структуру базы данных для хранения необходимой информации согласно варианту и установите связи между таблицами.

Вариант 1. База данных "Гостиница" должна хранить следующую информацию:

1. Фамилия, Имя, Отчество клиента.
2. Серия и номер паспорта клиента.
3. Домашний адрес клиента.
4. Телефон клиента.
5. Номер, в который заселен клиент.
6. Этаж, количество мест номера.
7. Класс номера, телефон номера.
8. Стоимость номера за сутки.
9. Дата и время заезда.
10. Дата и время выезда.

База данных должна содержать информацию о 10 клиентах, 15 номерах.

Предусмотреть, чтобы 5 клиентов останавливались в гостинице не менее 2 раз.

Вариант 2. База данных "Подписные издания" должна хранить следующую информацию:

1. Фамилия, Имя, Отчество подписчика.
2. Серия и номер паспорта подписчика.
3. Домашний адрес подписчика.
4. Телефон подписчика.
5. Вид (газета, журнал) и название периодического издания, на которое подписался подписчик.
6. Срок подписки на данное издание (3 месяца, 6 месяцев, 12 месяцев).
7. Стоимость подписки.
8. Дата подписки.

База данных должна содержать информацию о 10 подписчиках, 10 периодических изданий.

Предусмотреть, чтобы 5 подписчиков оформили подписку на 2 и более издания.

Вариант 3. База данных "Телефонные переговоры" должна хранить следующую информацию:

1. Фамилия, Имя, Отчество абонента.
2. Номер телефона абонента.
3. Домашний адрес абонента.
4. Паспортные данные абонента (серия, номер паспорта).
5. Телефонный код и название города, куда звонил абонент.
6. Тариф за 1 минуту разговора с указанным городом.
7. Дата разговора, время разговора.
8. Продолжительность разговора.

База данных должна содержать информацию о 10 абонентах, 5 городах.

Предусмотреть, чтобы 5 абонентов сделали не менее 2 телефонных звонков в различные города.

Вариант 4. База данных «Путешествие» должна хранить следующую информацию:

1. Фамилия, Имя, Отчество клиента.
2. Адрес и номер телефона клиента.

<p>3. Дата рождения клиента.</p> <p>4. Паспортные данные клиента (серия, номер паспорта).</p> <p>5. Наименование страны, в которую поедет клиент.</p> <p>6. Класс отеля, в котором остановится клиент.</p> <p>7. Длительность туристической поездки.</p> <p>8. Стоимость путевки.</p> <p>9. Дата поездки.</p> <p>База данных должна содержать информацию о 10 клиентах, 5 маршрутах.</p> <p>Предусмотреть, чтобы 5 клиентов заказали не менее 2 путевок по различным маршрутам.</p>
<p>Вариант 5. База данных «Студент» должна хранить следующую информацию:</p> <ol style="list-style-type: none"> 1. Фамилия, Имя, Отчество студента. 2. Номер студенческого билета. 3. Пол и дата рождения студента. 4. Адрес и номер телефона студента. 5. Специальность, на которой обучается студент. 6. Название дисциплины, по которой студент сдавал экзамен. 7. Количество аудиторных часов и количество часов самостоятельной работы по указанной дисциплине. 8. Дата экзамена по указанной дисциплине. 9. Оценка, полученная на экзамене. <p>База данных должна содержать информацию о 10 студентах, 5 дисциплинах.</p> <p>Предусмотреть, чтобы 5 студентов сдавали не менее 3 экзаменов.</p>
<p>Вариант 6. База данных «Оптовая торговля» должна хранить следующую информацию:</p> <ol style="list-style-type: none"> 1. Название, единицы измерения товара. 2. Цена товара и дата выпуска товара. 3. Название фирмы, которая купила указанный товар. 4. Фамилия, Имя, Отчество директора фирмы. 5. Адрес и номер телефона фирмы. 6. Количество купленного товара. 7. Дата, когда был куплен товар. <p>База данных должна содержать информацию о 10 фирмах-клиентах,</p>

<p>5 товарах.</p> <p>Предусмотреть, чтобы 5 фирм купили не менее 2 различных товаров.</p>
<p>Вариант 7. База данных «Ремонтные услуги» должна хранить следующую информацию:</p> <ol style="list-style-type: none"> 1. Фамилия, Имя, Отчество заказчика услуги. 2. Адрес, номер телефона, паспортные данные (серия и номер паспорта) заказчика. 3. Название и стоимость услуги, которую заказал заказчик. 4. Фамилия, Имя, Отчество мастера, выполнившего услугу. 5. Количество дней, требуемое на выполнение данной услуги. 6. Дата начала выполнения услуги. <p>База данных должна содержать информацию о 10 заказчиках, 5 услугах.</p> <p>Предусмотреть, чтобы 5 заказчиков сделали не менее 2 заказов различных услуг.</p>
<p>Вариант 8. База данных «Автобаза» должна хранить следующую информацию:</p> <ol style="list-style-type: none"> 1. Фамилия, Имя, Отчество водителя, отправляющегося в рейс. 2. Адрес, номер телефона и паспортные данные (серия и номер паспорта) водителя. 3. Марка и гос. номер транспортного средства водителя. 4. Норму расхода горючего на 1 километр (в литрах). 5. Цену горючего за 1 литр 6. Название пункта назначения, куда отправляется водитель. 7. Количество километров до пункта назначения. 8. Дата выезда и дата приезда водителя из рейса. <p>База данных должна содержать информацию о 10 водителях, 5 пунктах назначения.</p> <p>Предусмотреть, чтобы 5 водителей съездили в рейс не менее 2 раз в разные пункты назначения.</p>
<p>Вариант 9. База данных «Лекарственные средства» должна хранить следующую информацию:</p> <ol style="list-style-type: none"> 1. Название лекарственного средства. 2. Название фирмы-производителя. 3. Форма выпуска лекарственного средства. 4. Адрес и номер телефона аптеки, в которой есть указанное

лекарственное средство.

5. Фамилия, Имя, Отчество директора указанной аптеки.

6. Количество и цена лекарственного средства в указанной аптеке.

База данных должна содержать информацию о 10 наименованиях лекарственных средств, 5 аптеках.

Предусмотреть, чтобы в каждой аптеке было не менее 5 видов лекарств.

Вариант 10. База данных "Прокат товаров" должна хранить следующую информацию:

1. Фамилия, Имя, Отчество клиента.

2. Серия и номер паспорта клиента.

3. Домашний адрес клиента.

4. Телефон клиента.

5. Наименование товара.

6. Описание товара.

7. Стоимость товара.

8. Дата выдачи.

9. Дата возврата плановая и фактическая.

10. Стоимость проката за сутки.

База данных должна содержать информацию о 10 клиентах, 5 товарах.

Предусмотреть, чтобы 5 клиентов взяли на прокат не менее 2 различных товаров

Вариант 11. База данных "Коммунальные услуги" должна хранить следующую информацию:

1. Фамилия, Имя, Отчество квартиросъемщика.

2. Домашний адрес квартиросъемщика.

3. Номер лицевого счета.

4. Количество жильцов.

5. Площадь квартиры, кв.м.

6. Вид услуги (название платежа).

7. Стоимость услуги на квадратный метр площади.

8. Стоимость услуги на 1 жильца.

9. Дата оплаты.

База данных должна содержать информацию о 10 квартиросъемщиках, о 5 видах услуг. Стоимость одних услуг должна

определяться площадью квартиры, других — количеством жильцов. Предусмотреть, чтобы каждый квартиросъемщик оплачивал не менее 3-х коммунальных услуг.

Вариант 12. База данных "Салон видеопроката" должна хранить следующую информацию:

1. Фамилия, Имя, Отчество клиента.
2. Серия и номер паспорта клиента.
3. Домашний адрес клиента.
4. Телефон клиента.
5. Наименование видеодиска.
6. Жанр и описание видеодиска.
7. Залоговая стоимость видеодиска.
8. Стоимость проката за сутки.
9. Дата выдачи.
10. Дата возврата плановая и фактическая.

База данных должна содержать информацию о 10 клиентах, 10 видеодисках.

Предусмотреть, чтобы 5 клиентов пользовались услугами проката не менее 2 раз.

Вариант 13. База данных "Библиотека" должна хранить следующую информацию:

1. Фамилия, Имя, Отчество читателя.
2. Домашний адрес читателя.
3. Телефон читателя.
4. Дата рождения читателя.
5. Номер читательского билета.
6. Автор книги, которую взял читатель.
7. Название книги, жанр, год издания и издательство.
8. Цена книги.
9. Дата выдачи книги.
10. Дата возврата книги плановая и фактическая.

База данных должна содержать информацию о 5 читателях, 10 книгах.

Предусмотреть, чтобы каждый читатель брал не менее 2 книг.

Вариант 14. База данных "Преподаватели вуза" должна хранить следующую информацию:

1. Фамилия, Имя, Отчество преподавателя.

2. Телефон преподавателя.

3. Место работы преподавателя.

4. Домашний адрес преподавателя.

5. Название должности преподавателя.

6. Должностная почасовая ставка преподавателя.

7. Предмет, который ведет преподаватель.

8. Общее число часов занятий по предмету.

9. Рабочая программа предмета.

10. Дата проведения занятий.

11. Количество проведенных часов.

База данных должна содержать информацию о 10 преподавателях, 3 должностях, 4 предметах.

Предусмотреть, чтобы 5 преподавателей проводили занятия по 2 предметам одновременно.

Вариант 15. База данных "Услуги интернет-провайдера" должна хранить следующую информацию:

1. Фамилия, Имя, Отчество клиента.

2. Серия и номер паспорта клиента.

3. Домашний адрес и номер телефона клиента.

4. Название тарифа.

5. Скорость доступа к ресурсам сети.

6. Объемы внутреннего и внешнего входящих трафиков.

7. Абонентская плата в месяц.

8. Дата подключения.

База данных должна содержать информацию о 10 клиентах, 8 тарифах.

ЛАБОРАТОРНАЯ РАБОТА 3. СОЗДАНИЕ ЗАПРОСОВ

Цель: сформировать навыки создания запросов в БД MS Access.

Теоретическая часть

Запросы являются основой для алгоритмической обработки данных БД, используемой для формирования подсхемы данных, обеспечивающих создание многотабличных форм и отчетов.

Запросы используются и для задания условий фильтрации записей таблиц, формирования страниц доступа. Исходными данными для запросов являются таблицы или другие запросы. Имена запросов не должны совпадать с именами таблиц БД.

Запросы классифицируются различным образом:

1. По числу обрабатываемых таблиц:
 - однотабличные;
 - многотабличные (все таблицы должны быть связаны).
2. По типу алгоритмов обработки:
 - выборки – результат запроса отражается только на экране;
 - перекрестный - создание сводной таблицы, содержащей групповые итоги;
 - на создание таблицы – автоматическое формирование структуры записей новой таблицы и загрузка;
 - на добавление – ввод новой записи, являющейся результатом выполнения запроса;
 - на удаление – удаление группы записей из таблицы;
 - на обновление – запрос пересчитывает (обновляет) значения расчетных полей.
3. По типу языка запросов:
 - QBE – запрос по примеру, построенный на основе реляционного языка запросов графического типа;
 - SQL – реляционно-полный язык запросов.
4. По стабильности условий фильтрации записей:
 - статистические запросы с неизменными условиями;
 - динамические запросы с изменяемыми условиями.

Построение запроса выполняется в **Конструкторе запросов**. Для запроса выбираются источники информации – таблицы или другие запросы, устанавливаются связи между ними. Для каждого запроса в области **Бланк запроса** определяют тип, уточняются условия выполнения запроса, состав полей результирующей таблицы.

Основная часть

!	Для выполнения заданий используется БД Фирма , созданная в лабораторной работе 2.
---	------------------------------------------------------------------------------------------

Задание 3.1.

1. Запустите Microsoft Access. Откройте БД Фирма, созданную в ЛР 2 «Создание связей между таблицами»
2. Создайте запрос на выборку *Телефоны сотрудников*. Для этого:
 - а) На вкладке **Создание** вызовите **Мастер запросов**. В появившемся диалоговом окне выберите *Простой запрос*.
 - б) В появившемся диалоговом окне (рисунок 3.1) выберите таблицу **Сотрудники** и поля *Фамилия, Имя, Телефон*. Нажмите кнопку **Далее**.
 - с) Введите имя запроса *Телефоны сотрудников*, нажмите кнопку **Готово**. Появится запрос для просмотра телефонов сотрудников.
 - д) Проверьте работу запроса.
3. Создайте запрос *Адреса клиентов* с помощью **Конструктора запросов**. Для этого:
 - а) На вкладке **Создание** вызовите **Конструктор запросов**. В появившемся диалоговом окне **Добавление таблиц** выберите таблицу **Клиенты**, кликните по кнопке **Добавить** и затем **Заккрыть**.
 - б) Перенесите поля *Название компании, Адрес, Телефон* в бланк запроса, щелкнув по ним левой клавишей мыши в окне таблицы **Клиенты** (рисунок 3.2.)
 - с) Выполните сортировку записей по полю *Название компаний* по возрастанию.
 - д) Сохраните запрос с именем *Адреса клиентов*.
 - е) Проверьте работу запроса.

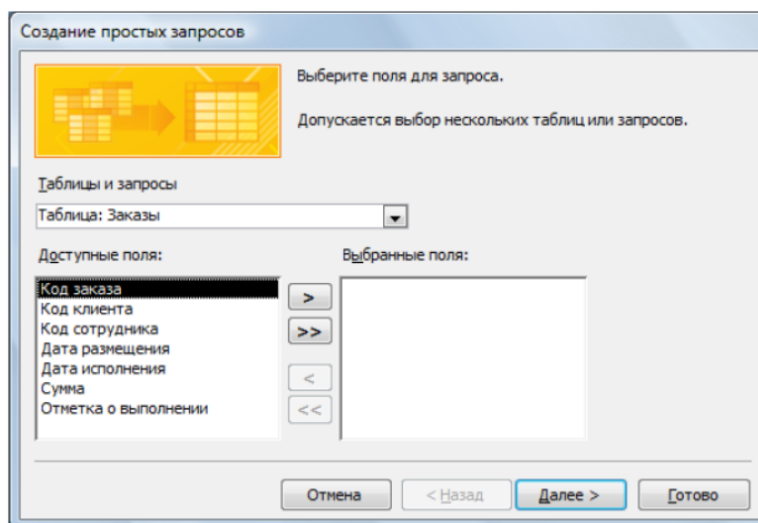


Рисунок 3.1 – Диалоговое окно «Создание простых запросов»

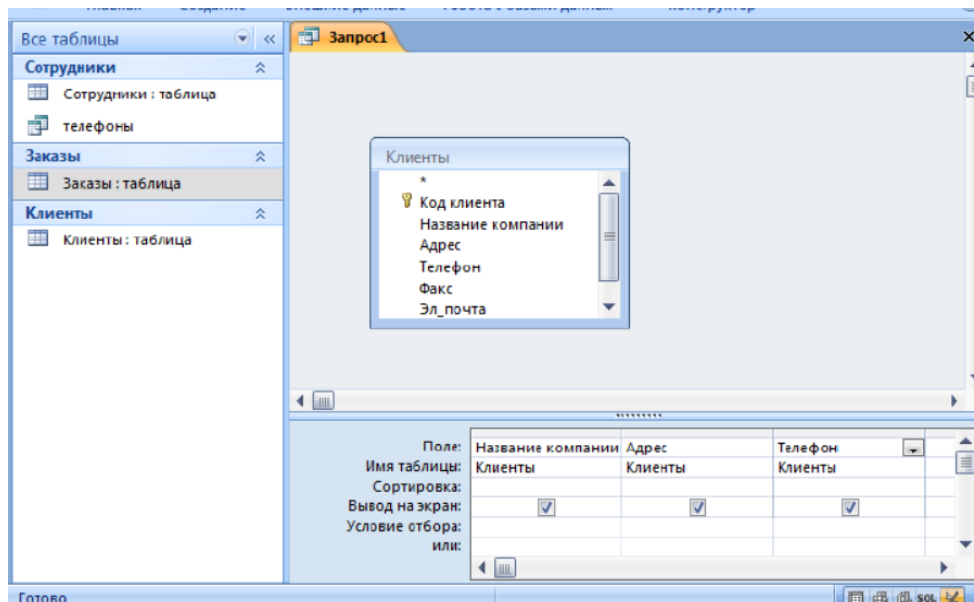


Рисунок 3.2 – Создание запроса в режиме Конструктора

4. Создайте запрос *Дни рождения сотрудников (апрель)*, в котором можно просмотреть дни рождения сотрудников, родившихся в апреле. Для этого:

- Вызовите окно **Конструктора запросов**.
- Перенесите поля таблицы **Сотрудники** (*Фамилия*, *Имя*, *Дата рождения*) в бланк запроса.
- В строке **Условие отбора** для поля «*Дата рождения*» введите значение **.04.**. В данной записи * означают, что дата и год рождения могут быть любыми, а месяц 4-м (т. е. апрель). После этого окно запроса должно выглядеть так, как оно представлено на рисунке 3.3

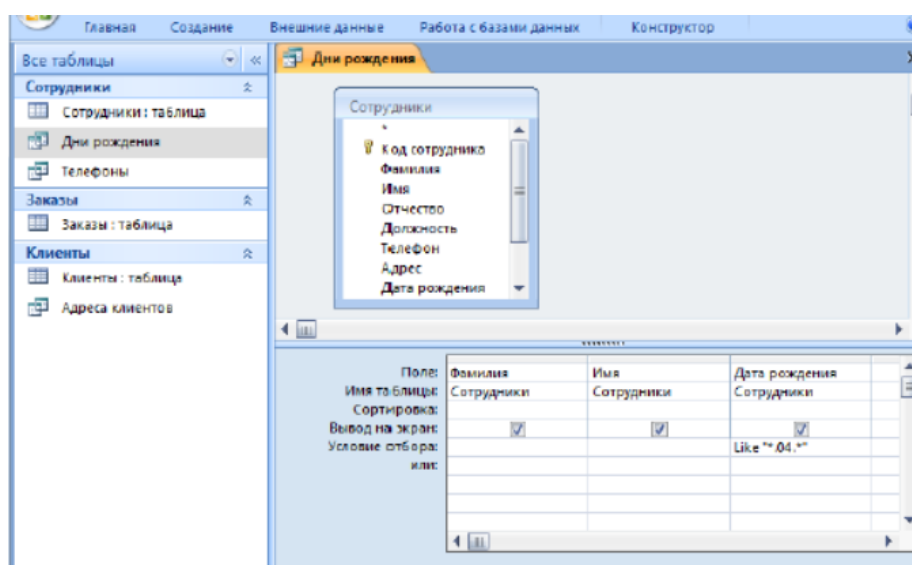


Рисунок 3.3 – Создание запроса с условием отбора

д) Проверьте работу запроса. Если в запросе *Дни рождения* нет ни одной записи, значит, в таблице **Сотрудники** нет ни одного человека, родившегося в апреле. Добавьте в таблицу **Сотрудники** несколько человек, родившихся в апреле, и посмотрите, как изменится запрос. Запросы автоматически обновляются при каждом открытии.

е) Аналогично создайте запрос *Дни рождения сотрудников (май)*, родившихся в мае.

5. Создайте запрос *Дни рождения сотрудников* с параметром. Для этого:

а) Создайте копию запроса *Дни рождения сотрудников (апрель)* и переименуйте запрос в *Дни рождения сотрудников*.

б) Откройте запрос *Дни рождения сотрудников* через **Конструктор запросов**.

с) В строке **Условие отбора** для столбца *Дата рождения* введите: **Like[Введите дату]** (рисунок 3.4).

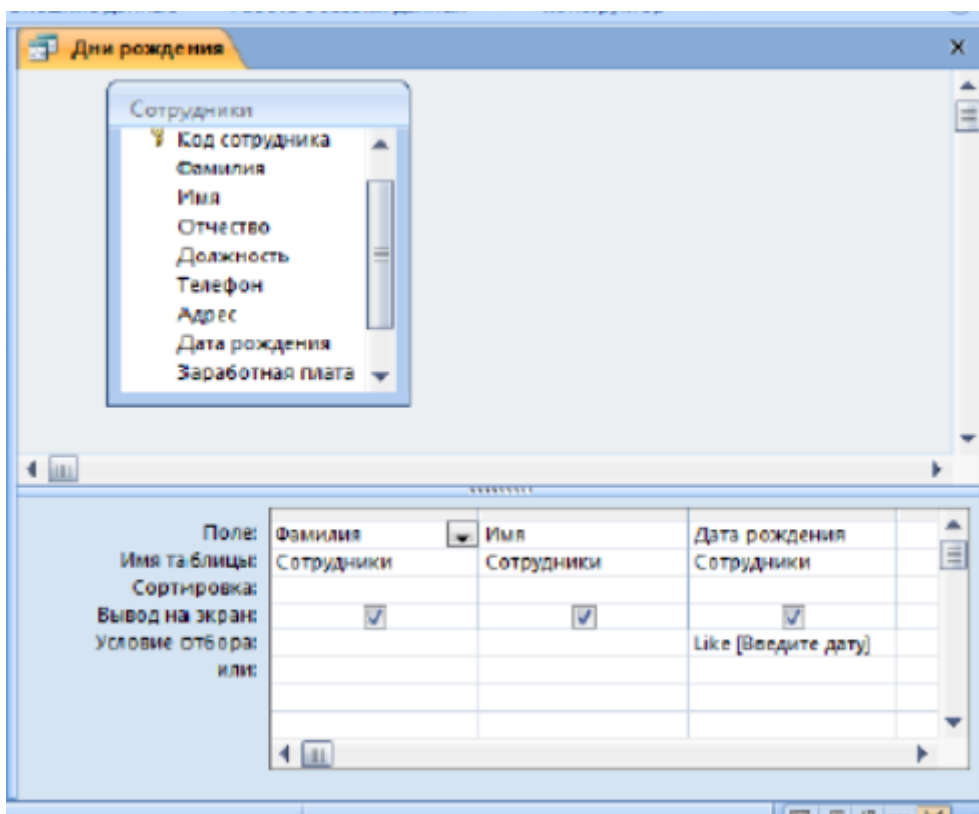


Рисунок 3.4 – Создание запроса с параметром

!	<p>Запись Like[Введите дату] означает, что при открытии запроса появится диалоговое окно (рисунок 3.5) с текстом «Введите дату» и полем для ввода условия отбора. Если ввести условие *.04.*, то в запросе появится список сотрудников, родившихся в апреле.</p>
---	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

d) Проверьте работу запроса для отбора сотрудников, родившихся в апреле, а затем в мае.

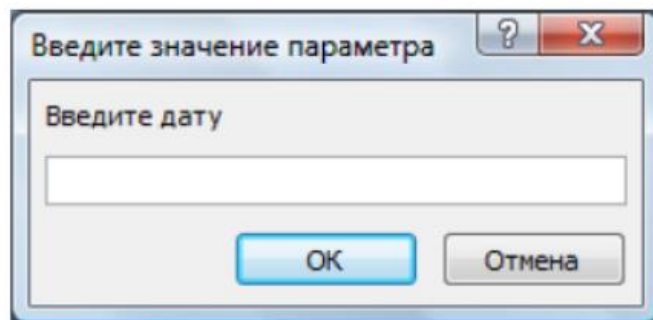


Рисунок 3.5 – Окно ввода параметра

6. Создайте запрос с параметром, чтобы при его запуске выводилось диалоговое окно с сообщением «Введите фамилию сотрудника». (Поскольку в запросе нужно вывести конкретную фамилию, в условии отбора слово Like писать не надо.)

7. Создайте запрос с параметром, чтобы при его запуске запрашивались фамилия и имя сотрудника.

8. Создайте запрос «Выполненные заказы», содержащий следующие сведения: фамилия и имя сотрудника, название компании, с которой он работает, отметка о выполнении и сумма заказа. Данные запроса возьмите из нескольких таблиц. В условии отбора для логического поля Отметка о выполнении введите Да, чтобы в запросе отображались только выполненные заказы. Сделайте так, чтобы столбец Отметка о выполнении не выводился на экран.

9. Создайте запрос *Сумма заказа (50 000)*, в котором будут отображаться заказы на сумму более 50 000 руб. Проверьте работу запроса. В случае отсутствия записей с данным условием, добавьте их в таблицу.

10. Создайте запрос, который будет выводить заказы суммой от 20 000 до 50 000 руб. В условии отбора можно использовать операторы сравнения >, <, =, >=, <=, < > и логические операторы And, Or, Not и др. Проверьте работу запроса. В случае отсутствия записей с данным условием, добавьте их в таблицу.

11. Создайте запрос, вычисляющий налог с прибыли в размере 13%. Для этого:

а) Запрос *Сумма заказа (50000)* пересохраните с именем *Сумма с налогом*. Откройте новый запрос в режиме **Конструктора**.

б) В пустом столбце бланка запроса щелкните правой кнопкой мыши на ячейке **Поле** и в появившемся контекстном меню выберите команду **Построить**. Перед вами появится окно **Построитель выражений** (рисунок 3.6), который состоит из трех областей: поля выражения, кнопок операторов и элементов выражения. Сверху располагается поле выражения, в котором оно и создается. Вводимые в это поле элементы выбираются в двух других областях окна **Построителя**.

с) В левом списке откройте папку *Запросы* и выделите запрос *Сумма заказа*. В среднем списке выделите поле *Сумма* и нажмите кнопку **Вставить**. Идентификатор этого поля появится в поле выражения **Построителя**.

д) Щелкните на кнопке ***** и введите 0,13. Таким образом, мы посчитаем налог 13 %.

е) Нажмите кнопку **ОК**, после чего в ячейке свойства **Поле** появится значение «Выражение1:[Сумма]*0,13».

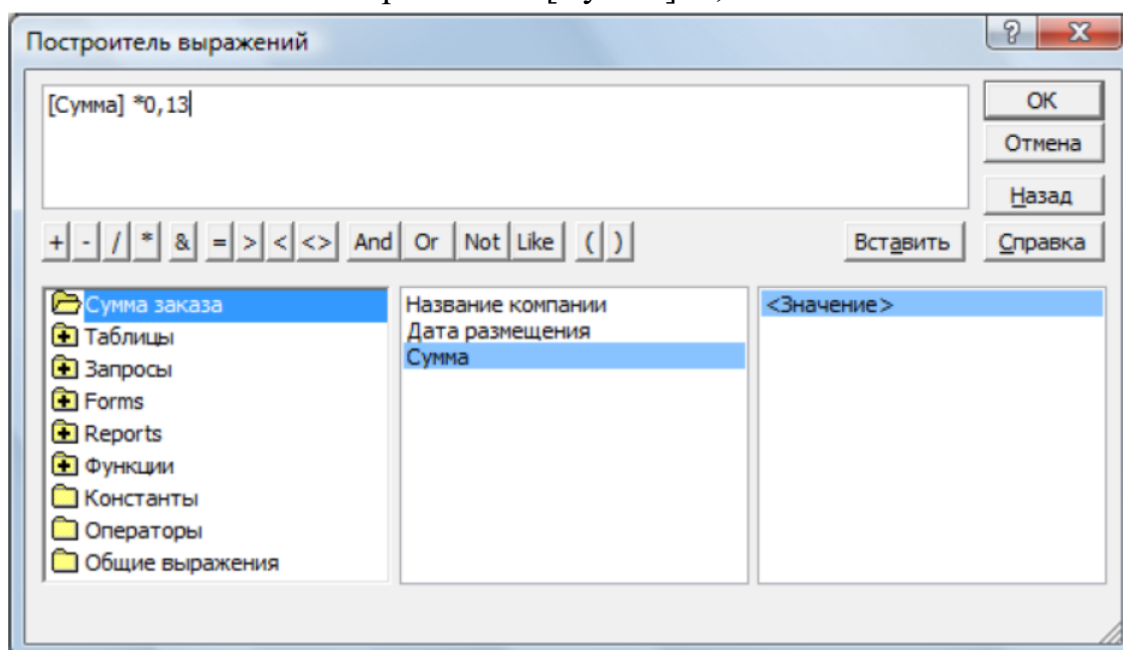


Рисунок 3.6 – Построитель выражений

ф) Замените Выражение1 на Налог и закройте **Конструктор**.

г) Проверьте работу запроса.

12. Запрос *Сумма заказа (50000)* пересохраните с именем *Сумма с доходом*. Откройте новый запрос в режиме **Конструктора**. Используя **Построитель выражений**, добавьте в запрос поле *Прибыль*, в котором будет вычисляться доход от заказа (т. е. сумма минус налог).

13. Создайте запрос *Менеджеры*, с помощью которого в таблице *Сотрудники* найдите всех менеджеров фирмы.

Задания для самостоятельного выполнения

Задание 3.2. В режиме **Мастера запросов**, сформировать простой запрос на выборку, позволяющий отобразить всю информацию из базовых таблиц (коды включать в запрос не нужно).

Задание 3.3. В режиме **Конструктора запросов** создать запрос, который выдает на экран информацию (согласно варианту).

Вариант 1. Список номеров в гостинице, стоимость которых за сутки не превышает определенного значения. Значением стоимости номера задаться самостоятельно.

Вариант 2. Список периодических названий, стоимость подписки на которые не менее определенного значения. Значением стоимости подписки задаться самостоятельно.

Вариант 3. Список городов, тариф за минуту разговора, с которым не более определенного значения. Значением тарифа за 1 минуту разговора задаться самостоятельно.

Вариант 4. Список маршрутов, цена путевки на которые не превышает определенного значения. Величиной цены путевки задаться самостоятельно.

Вариант 5. Список дисциплин, по которым количество аудиторных часов не менее определенного значения. Значением количества аудиторных часов задаться самостоятельно.

Вариант 6. Список товаров, цена которых не превышает определенного значения. Значением цены товара задаться самостоятельно.

Вариант 7. Список услуг, стоимость которых не менее определенного значения. Величиной стоимости услуги задаться самостоятельно.

Вариант 8. Список пунктов назначения, расстояние до которых не превышает определенного значения. Величину расстояния задаться самостоятельно.

Вариант 9. Список лекарственных средств, цена которых не превышает определенного значения. Значением цены лекарственного средства задаться самостоятельно.

Вариант 10. Список товаров, залоговая стоимость которых не превышает определенного значения. Значением стоимости товара задаться самостоятельно.

Вариант 11. Список коммунальных услуг, стоимость которых не

менее определенного значения. Величиной стоимости услуги задаться самостоятельно.

Вариант 12. Список видеодисков, залоговая стоимость которых не превышает определенного значения. Значением стоимости видеодиска задаться самостоятельно.

Вариант 13. Список книг, цена на которые не менее определенного значения. Значением цены книги задаться самостоятельно.

Вариант 14. Список преподавателей, должностная почасовая ставка которых не превышает определенного значения. Величиной ставки задаться самостоятельно.

Вариант 15. Список тарифов, у которых абонентская плата в месяц не превышает определенного значения. Значением величины абонентской платы задаться самостоятельно.

Задание 3.4. Создайте запрос, который выдает на экран информацию (согласно варианту).

Вариант 1. Список клиентов (Фамилия И.О.) с указанием паспортных данных (фамилии отсортировать по алфавиту).

Вариант 2. Список подписчиков (Фамилия И.О.) с указанием домашнего адреса (фамилии отсортировать по алфавиту).

Вариант 3. Список абонентов (Фамилия И.О.) с указанием номеров телефона (фамилии отсортировать по алфавиту).

Вариант 4. Список клиентов (Фамилия И.О.) с указанием паспортных данных (фамилии отсортировать по алфавиту).

Вариант 5. Список студентов (Фамилия И.О.) с указанием номера студенческого билета (фамилии отсортировать по алфавиту).

Вариант 6. Список фирм с указанием Фамилии И.О. директоров и номеров телефонов (название отсортировать по алфавиту).

Вариант 7. Список заказчиков (Фамилии И.О.) с указанием адреса и номера телефона (фамилии отсортировать по алфавиту)

Вариант 8. Список водителей (Фамилии И.О.) с указанием марки и гос.номера транспортного средства (фамилии отсортировать по алфавиту).

Вариант 9. Список аптек с указанием Фамилии И.О. директора, адреса и номера телефона.

Вариант 10. Список клиентов (Фамилии И.О.) с указанием

паспортных данных (фамилии отсортировать по алфавиту).
Вариант 11. Список квартиросъемщиков (Фамилия И.О.) с указанием адреса и лицевого счета (фамилии отсортировать по алфавиту)
Вариант 12. Список клиентов (Фамилия И.О.) с указанием паспортных данных (фамилии отсортировать по алфавиту).
Вариант 13. Список читателей (Фамилия И.О.) с указанием номера читательского билета, домашнего адреса и номера телефона (фамилии отсортировать по алфавиту).
Вариант 14. Список предметов, с указанием общего числа часов и Фамилии И.О. преподавателя.
Вариант 15. Список клиентов (Фамилия И.О.) с указанием паспортных данных и домашнего адреса (фамилии отсортировать по алфавиту).

Задание 3.5. Создайте запросы с вычисляемыми полями (согласно варианту).

Вариант	Включаемые в запрос поля			Сортиро- вать по указанно- му полю
	Из базовых таблиц	Вычисляемое поле		
			Имя	Формат
1	Фамилия Имя Отчество клиента, класс номера, количество мест номера, дата заезда, дата выезда, стоимость номера за сутки	Сумма к оплате (за проживание в номере)	Денеж- ный	Фамилия
2	Вид периодического издания, название периодического издания, срок подписки, стоимость подписки	1. Полное название издания (вид + название, н-р, журнал «Мурзилка», газета «Жизнь») 2. Новая стоимость подписки на издание (увеличение	Тексто-- вый Денеж- ный	Полное название

Вариант	Включаемые в запрос поля			Сортиро- вать по указанно- му полю
	Из базовых таблиц	Вычисляемое поле		
		Имя	Формат	
		стоимости на 25%)		
3	Фамилия, Имя, Отчество абонента, телефонный код и название города, дата разговора, время разговора, продолжительность разговора	Сумма к оплате	Денеж- ный	Фамилия
4	Фамилия, Имя, Отчество клиента, наименование страны, дата поездки	Дата приезда	Дата/вре- мя	Фамилия
5	Фамилия, Имя, Отчество студента, пол, дата рождения	Возраст	Числовой	Возраст
6	Название фирмы, название, цена и количество товара, дата покупки	Общая стоимость (на которую был куплен товар)	Денеж- ный	Название фирмы
7	Фамилия, Имя, Отчество заказчика услуги, название услуги, дата начала выполнения услуги.	Дата окончания выполнения услуги	Дата/вре- мя	Фамилия
8	Фамилия, Имя, Отчество водителя, название пункта назначения, дата выезда, дата приезда	Затраты на горючее	Денеж- ный	Дата выезда
9	Адрес и номер телефона аптеки, название лекарственного средства, количество и цена лекарственного	Новая цена (скидка 10%)	Денеж- ный	Название лекарствен- ного средства

Вариант	Включаемые в запрос поля			Сортиро- вать по указанно- му полю
	Из базовых таблиц	Вычисляемое поле		
		Имя	Формат	
	средства			
10	Фамилия, Имя, Отчество клиента, наименование товара, дата выдачи	Стоимость проката	Денеж- ный	Фамилия
11	Фамилия, Имя, Отчество квартиросъемщика, номер лицевого счета, количество жильцов, площадь квартиры	Стоимость услуг	Денеж- ный	Фамилия
12	Фамилия, Имя, Отчество клиента, наименование видеодиска, дата выдачи	Стоимость проката	Денеж- ный	Дата выдачи
13	Фамилия, Имя, Отчество читателя, автор книги, название книги, дата выдачи, дата возврата плановая и фактическая.	Штраф (1% от цены книги за каждый день просрочки, предполагается, что дата возврата плановая меньше или равна дате возврата фактической)	Денеж- ный	Фамилия
14	Фамилия, Имя, Отчество преподавателя, предмет, дата проведения занятий, количество проведенных часов	Сумма к выплате	Денеж- ный	Дата проведения занятий
15	Название тарифа, скорость доступа к ресурсам сети, абонентская плата	Новая аб. плата (увеличение на 35%)	Денеж- ный	Название

Задание 3.6. Создайте запросы с параметрами (согласно варианту).

Вариант 1. Список клиентов, проживающих в указанном номере, и стоимость номера в EUR
Вариант 2. Список подписчиков, подписавшихся на указанное издание, и стоимость издания в EUR.
Вариант 3. Список абонентов, звонивших в указанный город, и тариф за 1 минуту разговора в EUR.
Вариант 4. Список клиентов, отдыхающих в указанной стране, и стоимость путевки в EUR.
Вариант 5. Список студентов, не сдавших экзамен по указанной дисциплине, и дату пересдачи экзамена (через n дней).
Вариант 6. Список фирм, которые купили указанный товар, и новую цену товара с учетом скидки (размер скидки указывается в %).
Вариант 7. Список заказчиков указанной услуги и стоимость данной услуги с учетом скидки (размер скидки указывается в %).
Вариант 8. Список водителей, имеющих транспортное средство указанной марки, и цену горючего с учетом повышения цен (надбавка в %).
Вариант 9. Список аптек, в которых есть указанное лекарственное средство, и стоимость данного средства в EUR.
Вариант 10. Список клиентов, бравших напрокат указанный товар, и стоимость товара с учетом скидки (размер скидки указывается в %).
Вариант 11. Список квартиросъемщиков, пользующихся указанной коммунальной услугой, и стоимость этой услуги в EUR.
Вариант 12. Список клиентов, бравших для просмотра указанный видеодиск, и стоимость видеодиска с учетом надбавки (размер надбавки указывается в %).
Вариант 13. Список читателей, бравших в библиотеке указанную книгу, и цену книги в EUR.
Вариант 14. Список преподавателей, проводивших занятия по указанному предмету, и сумму к оплате в EUR.
Вариант 15. Список клиентов Интернет-провайдера с указанным тарифным планом и абонентскую плату по данному тарифу в EUR.

Задание 3.7. Создайте итоговый запрос (согласно варианту).

Вариант 1. Для каждого номера определить количество постояльцев. Определить общую выручку и количество постояльцев гостиницы.
Вариант 2. Определить среднюю стоимость подписки на каждый вид периодического издания. Определить общее число подписчиков и общую выручку от подписки.
Вариант 3. Определить общую продолжительность разговоров и общую сумму к оплате для каждого абонента. Определить итоговую сумму по всем переговорам и количество сеансов связи.
Вариант 4. Определить минимальную, максимальную и среднюю стоимости путевки. Определить общее количество туристов для каждой страны. Определить сумму к оплате для каждого клиента.
Вариант 5. Определить средний балл каждого студента. Для каждой дисциплины определить количество студентов, сдававших по ней экзамен и средний балл.
Вариант 6. Для каждой фирмы определить количество видов закупленного товара и сумму, затраченную на покупку. Определить общую выручку от продажи товара.
Вариант 7. Определить среднюю стоимость выполнения услуги. Для каждого мастера определить количество заказов услуг, сумму от выполнения заказов.
Вариант 8. Для каждого водителя определить количество рейсов и общее количество километров. Определить среднюю, минимальную и максимальную цену горючего за 1 литр.
Вариант 9. Для каждого лекарственного средства определить среднюю, минимальную и максимальную цену. Для каждой аптеки определить количество видов и общую стоимость лекарственных средств.
Вариант 10. Для каждого клиента определить, сколько товаров он брал напрокат и общую стоимость проката. Определить общую выручку от проката товаров.
Вариант 11. Для каждого квартиросъемщика определить количество оплаченных услуг и общую сумму. Для каждой услуги определить общую сумму.
Вариант 12. Для каждого клиента определить, сколько видеодисков он брал напрокат и общую стоимость проката. Определить среднюю, максимальную и минимальную стоимость проката видеодисков.

Вариант 13. Для каждого читателя определить количество книг, которые он брал в библиотеке. Определить сколько читателей посетило библиотеку и сколько книг взяли почитать.

Вариант 14. Определить заработную плату преподавателя. Определить общее количество проведенных часов по каждому предмету.

Вариант 15. Для каждого тарифа определить количество абонентов. Определить максимальную, минимальную и среднюю абонентскую плату.

ЛАБОРАТОРНАЯ РАБОТА 4. СОЗДАНИЕ ФОРМ

Цель: сформировать навыки создания форм в БД MS Access.

Теоретическая часть

Формы – это объекты базы данных, предназначенные для просмотра данных из таблиц и запросов, для ввода данных в базу, корректирования существующих данных и выполнения заданных действий. Форма может содержать графики, рисунки и другие внедренные объекты.

Новые формы могут создаваться на вкладке **Формы** с помощью командных кнопок вкладки **Создание** группы **Формы** (рисунок 4.1):

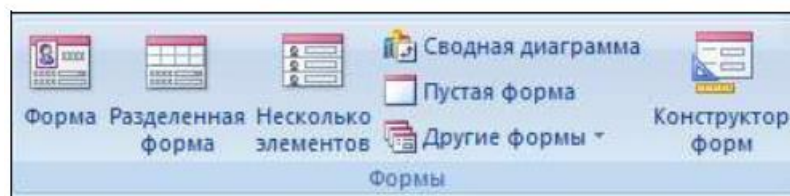


Рисунок 4.1 – Инструменты Access для создания форм.

- **Пустая форма** – позволяет быстро построить форму, особенно если в ней будет лишь несколько полей;
- **Конструктор форм** – наиболее трудоемкий способ, обеспечивает создание и редактирование форм различной сложности;
- **Мастер форм** – специальная программа для автоматизации создания форм;
- **Форма** – быстрое создание простейших однотабличных форм;
- **Сводная диаграмма** – построение диаграмм для числовых данных таблиц;

- **Разделенная форма** — позволяет одновременно отображать данные в двух представлениях — в режиме формы и в режиме таблицы;
- **Сводная таблица** – построение итоговой экранной формы, обеспечивающей формирование и просмотр итогов в электронной таблице Excel;
- **Несколько элементов** - форма предоставляет больше возможностей настройки, чем таблица. Например, к ней можно добавлять графические элементы, кнопки и другие элементы управления.

Основная часть

!	Для выполнения заданий используется БД Фирма , созданная в лабораторных работах 2 и 3.
---	-----------------------------------------------------------------------------------------------

I. Использование Мастера форм

Задание 4.1. С помощью **Мастера** создайте формы *Сотрудники*, *Клиенты*, *Заказы*, *Менеджеры*.

При создании форм с помощью **Мастера форм** используйте следующий алгоритм:

1. Выполните команду: вкладка ленты Создание → Мастер форм.
2. В диалоговом окне **Создание форм** выберите необходимые объекты: таблицы (запросы) и поля, которые будут помещены в форму. Щелкните по кнопке **Далее**.
3. В следующих диалоговых окнах мастера выберите внешний вид формы, стиль, задайте имя формы. Щелкните по кнопке **Готово**.

II. Использование Конструктора форм

!	Режим Конструктор форм предназначен, как правило, для создания и редактирования форм.
---	----------------------------------------------------------------------------------------------

Задание 4.2. С помощью **Конструктора форм** проведите редактирование формы *Сотрудники*.

1. Откройте, созданную в задании 1 форму *Сотрудники*, используя **Конструктор**.
2. Разместите элементы в удобном для вас порядке, измените размер и цвет текста.

3. В заголовок формы добавьте текст **Сотрудники фирмы**.

Задание 4.3. С помощью **Конструктора кнопок** добавьте в форму *Сотрудники* кнопки управления.

1. В примечание формы добавьте объект **Кнопка** (вкладка ленты Конструктор → Элементы управления).

2. После того как вы «нарисуете» кнопку указателем, на экране появится диалоговое окно **Создание кнопок** (рисунок 4.2)

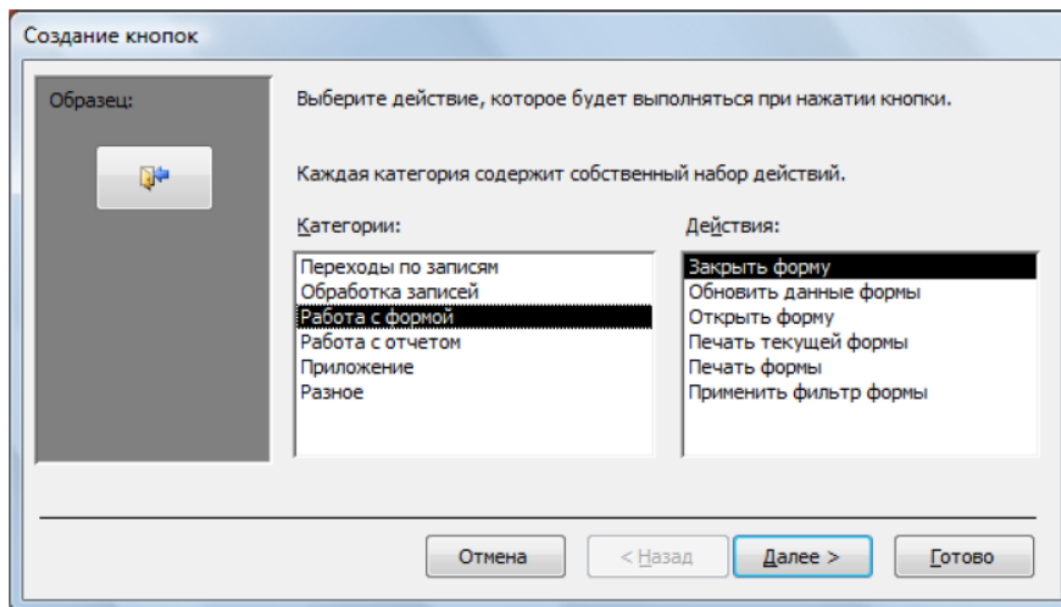


Рисунок 4.2 – Диалоговое окно «Создание кнопок»

3. В категории *Работа с формой* выберите действие *Заккрытие формы* и нажмите кнопку **Далее**.

4. Выберите рисунок или текст, который будет размещаться на кнопке.

5. В последнем диалоговом окне **Мастера кнопок** задайте имя кнопки и нажмите **Готово**.

!	Мастер кнопок написал для данной кнопки процедуру на языке Microsoft Visual Basic. Просмотреть процедуру обработки события можно с помощью команды Обработка событий контекстного меню кнопки.
---	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6. Самостоятельно создайте кнопки **Выход из приложения**, **Поиск записи**, **Удаление записи**.

!	Иногда на форме требуется разместить несколько страниц, содержащих данные из различных источников, справочную или вспомогательную информацию. Для этой цели можно использовать набор вкладок.
---	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Задание 4.4. Создайте форму, содержащую набор вкладок.

1. Создайте пустую форму (Создание → Форма).
2. Для добавления к форме набора вкладок щелкните по кнопке **Вкладка** на панели инструментов **Элементы управления**. Сначала добавятся только две вкладки с формальными именами *Вкладка 1* и *Вкладка 2*.
3. Добавьте еще одну вкладку.
4. Переименуйте ярлычки вкладок так, чтобы на них отображались названия данных, которые будут в них располагаться: *Сотрудники*, *Менеджеры*, *Помощь*.
5. Перейдите на вкладку *Сотрудники* и перетащите на нее мышкой из базы данных форму *Сотрудники*.
6. Аналогичным образом поместите форму *Менеджеры* на вкладку *Менеджеры*.
7. На вкладку *Помощь* поместите некоторые советы по работе с базой данных.
8. Данную форму сохраните с именем *Сотрудники фирмы*.



В Microsoft Access можно создавать кнопочные формы. Они содержат только кнопки и предназначены для выбора основных действий в базе данных.

Задание 4.5. Создайте кнопочную форму.



Для создания кнопочной формы необходимо на вкладке ленты **Работа с базами данных** выбрать команду **Диспетчер кнопочных форм**. Если на ленте **Диспетчер кнопочной формы** не отображается, то необходимо дополнительно настроить ленту. Для этого необходимо вызвать диалоговое окно **Настройка ленты** (Рисунок 4.3). Из списка команд выбрать **Диспетчер кнопочных форм** и добавить элемент на любую вкладку ленты, например, *Работа с базами данных*.

1. Откройте **Диспетчер кнопочных форм**. Будет выведен запрос на подтверждение ее создания. Нажмите **Да** в диалоговом окне подтверждения.
2. В появившемся окне **Диспетчера кнопочных форм** щелкните по кнопке **Создать**.
3. В диалоговом окне **Создание** (рисунок 4.4) введите имя новой кнопочной формы и нажмите **ОК**.

4. Имя новой кнопочной формы добавится в список **Страницы кнопочной формы** окна **Диспетчер кнопочных форм** (рисунок 4.5). Выделите имя новой кнопочной формы и щелкните по кнопке **Изменить**.

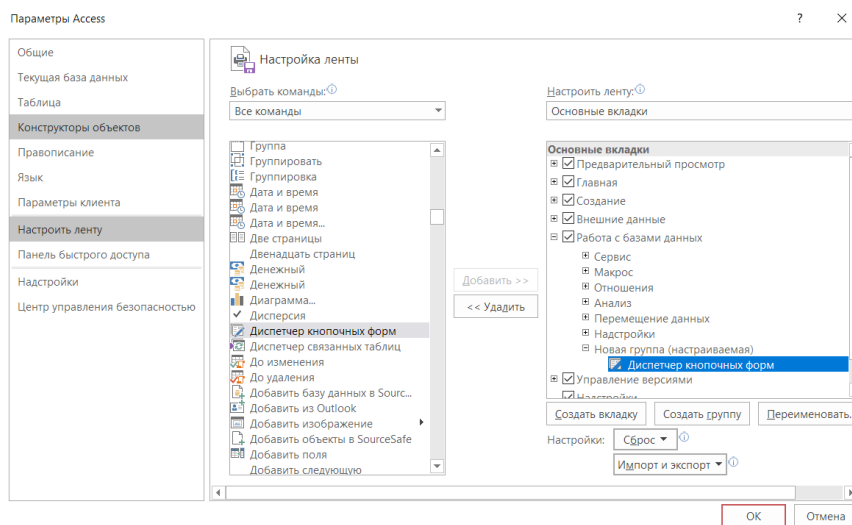


Рисунок 4.3 – Размещение на ленту Диспетчера кнопочных форм

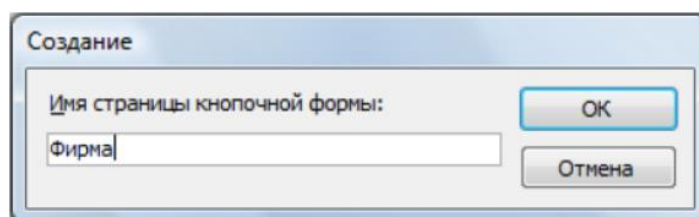


Рисунок 4.4 – Задание имени кнопочной форме.

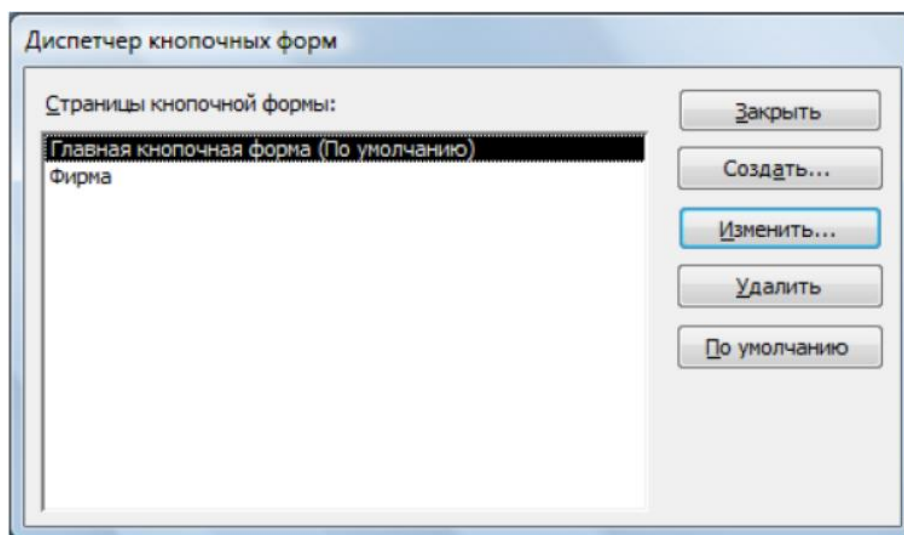


Рисунок 4.5 – Диспетчер кнопочных форм.

5. Выделите форму «Фирма» и нажмите **Изменить**. В диалоговом окне **Изменение элемента кнопочной формы** (рисунок 4.6) в поле **Текст** введите текст подписи для первой кнопки кнопочной формы, а

затем выберите команду из раскрывающегося списка в поле **Команда**. В поле **Форма** выберите форму, для которой будет выполняться данная команда.

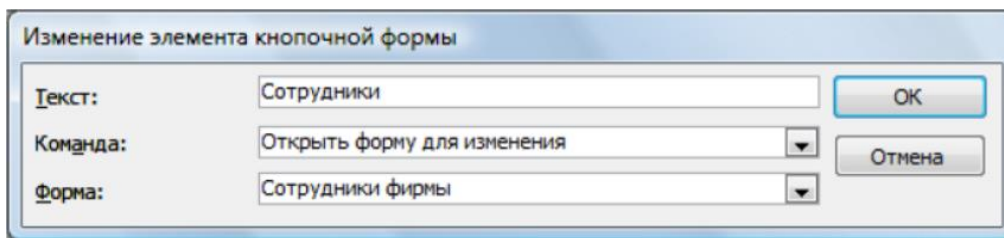


Рисунок 4.6 – Создание кнопок на форме «Фирма»

6. Аналогичным образом добавьте кнопки Клиенты, Заказы, Выход.
7. В диалоговом окне Диспетчер кнопочных форм выберите имя вашей кнопочной формы и щелкните по кнопке По умолчанию. Рядом с названием кнопочной формы появится надпись «(по умолчанию)».
8. Чтобы закончить создание кнопочной формы, щелкните по кнопке Закр \ddot{y} ть. В результате должна получиться форма, представленная на рисунке 4.7.

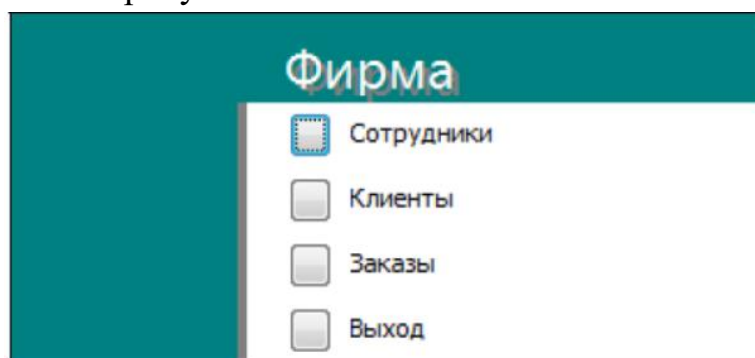


Рисунок 4.7 – Главная кнопочная форма

9. Для того чтобы главная кнопочная форма появлялась на экране при запуске приложения, необходимо в меню Файл выбрать Параметры Access → Текущая база данных → Форма просмотра → Выбрать кнопочную форму «Фирма» → нажать **ОК**.
10. Для проверки работы кнопочной формы необходимо закрыть БД Фирмы и открыть ее заново.

Задания для самостоятельного выполнения

Задание 4.6. Согласно варианту индивидуального задания выполнить:

1. Создать простую, многотабличную и кнопочную формы.
2. С кнопочной формы должны открываться все запросы, формы и отчеты.
3. Установить параметры запуска для базы данных.

ЛАБОРАТОРНАЯ РАБОТА 5. СОЗДАНИЕ ОТЧЕТОВ

Цель: сформировать навыки создания отчетов в БД MS Access.

Теоретическая часть

В БД объект «Отчеты» предназначены для вывода информации на печать. Часто данные в них располагаются в табличной форме. В отличие от распечаток таблиц или запросов отчет дает более широкие возможности сортировки и группировки данных, он предоставляет возможность добавлять итоговые значения, а также поясняющие надписи, колонтитулы, номера страниц, стили и различные графические элементы.

Создавать отчеты в СУБД Access можно несколькими способами:

- с помощью Мастера отчетов;
- на основе таблиц или запросов;
- в режиме Конструктора.

Основная часть

!	Для выполнения заданий используется БД Фирма , созданная в лабораторных работах 2 - 4.
----------	-----------------------------------------------------------------------------------------------

I. Использование Мастера отчетов.

Задание 5.1. С помощью **Мастера отчетов** на основе запроса *Адреса клиентов* создайте отчет *Адреса клиентов*.

1. Выполните команду: вкладка ленты **Создание** → **Мастер отчетов**.
2. В диалоговом окне **Создание отчетов** выберите запрос *Адреса клиентов*, переместите все доступные поля в область «выбранные поля» и щелкните по кнопке **Далее**.
3. Последовательно задайте порядок сортировки, структуру макета, ориентацию печатного документа, имя отчета и нажмите кнопку **Готово**.

Задание 5.2. С помощью **Мастера** создайте отчет *Дни рождения сотрудников*. В качестве источника данных используйте таблицу *Сотрудники*.

II. Использование Мастера наклеек.

Задание 5.3. Создайте почтовые наклейки для рассылки писем клиентам.

1. Выделите таблицу Клиенты. И выполните команду: вкладка ленты Создание → Отчеты → Наклейки.
2. В диалоговом окне **Создание наклеек** (рисунок 5.1) укажите размер наклейки, систему единиц, тип наклейки и нажмите кнопку Далее.

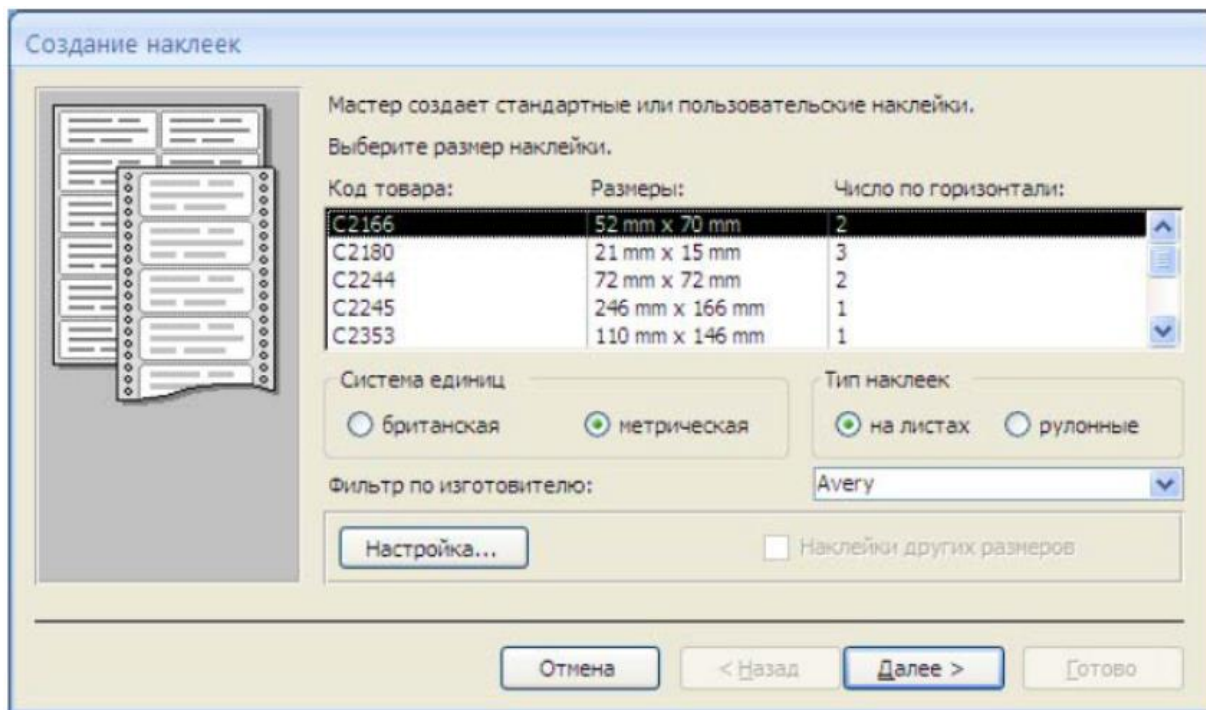


Рисунок 5.1 – Диалоговое окно Создание наклеек.

3. На следующем шаге создания отчета установите шрифт, размер, цвет текста и начертание. Нажмите кнопку **Далее**.
4. Выберите поля, которые будут размещаться на наклейке. Например, *Название компании, Адрес, Телефон и Факс*. Если на каждой наклейке требуется вывести определенный текст, то введите его в прототип наклейки.
5. При необходимости измените название отчета с наклейками и нажмите кнопку **Готово**.

III. Создание отчетов с вычисляемыми значениями

Задание 5.4. Создайте отчет с выводом итоговых значений, среднего, минимального и/или максимального значения, процентов по заказам клиентов. В качестве источника используйте запрос *Сумма заказов*.

1. Запустите **Мастер отчетов**, в качестве источника данных укажите запрос *Сумма заказа*.
2. В диалоговом окне **Мастера**, в котором задается порядок сортировки записей, нажмите кнопку **Итоги** (рисунок 5.2).

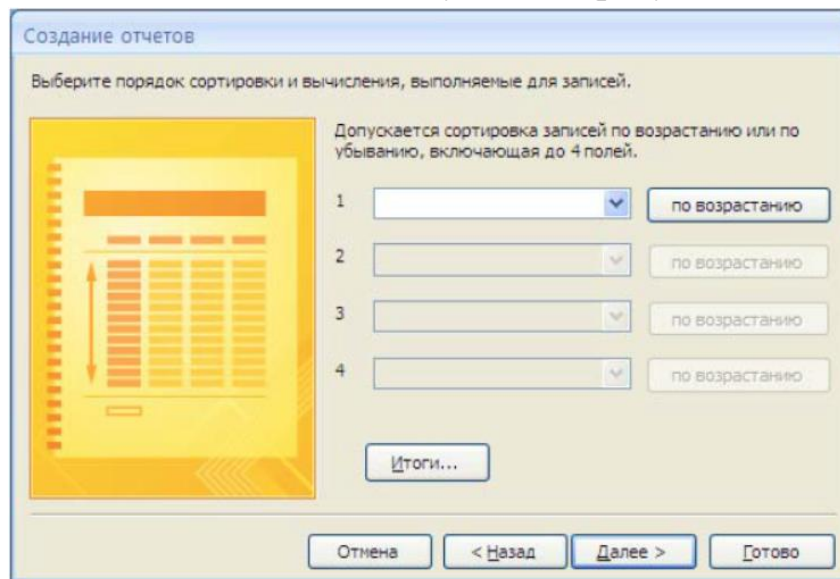


Рисунок 5.2 – Диалоговое окно **Создание отчетов**

3. В диалоговом окне **Итоги** (рисунок 5.3) для полей *Сумма* и *Налог* установите флажки в столбце *Sum*, чтобы посчитать итоговую сумму.

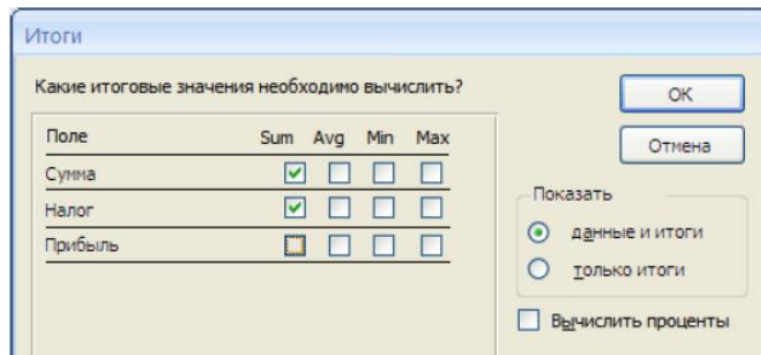


Рисунок 5.3 – Вычисление итоговых значений в отчетах

4. Далее выполните все шаги Мастера и нажмите кнопку **Готово**.
5. Выведите отчет на предварительный просмотр. При необходимости скорректируйте внешний вид отчета, используя **Конструктор отчетов**.

Задание 5.5. Создайте отчет *Дни рождения*, используя в качестве источника данных запрос *Дни рождения*.

Задание 5.6. Составьте отчет *Выполненные заказы*, в котором будут данные о компании и сумме заказа. Вычислите итоговую сумму, среднее значение (Avg) и максимальную сумму для каждой фирмы.

Задания для самостоятельного выполнения

Задание 5.7. Создать отчеты на основе запросов с вычисляемыми полями (согласно варианту).

Вариант 1. Список клиентов (Фамилия И.О.) с указанием паспортных данных (фамилии отсортировать по алфавиту), добавить № п/п1.
Вариант 2. Список подписчиков (Фамилия И.О.) с указанием домашнего адреса (фамилии отсортировать по алфавиту), добавить № п/п.
Вариант 3. Список абонентов (Фамилия И.О.) с указанием номеров телефона (фамилии отсортировать по алфавиту), добавить № п/п.
Вариант 4. Список клиентов (Фамилия И.О.) с указанием паспортных данных (фамилии отсортировать по алфавиту), добавить № п/п.
Вариант 5. Список студентов (Фамилия И.О.) с указанием номера студенческого билета (фамилии отсортировать по алфавиту), добавить № п/п.
Вариант 6. Список фирм с указанием Фамилии И.О. директоров и номеров телефонов (название отсортировать по алфавиту), добавить № п/п.
Вариант 7. Список заказчиков (Фамилии И.О.) с указанием адреса и номера телефона (фамилии отсортировать по алфавиту), добавить № п/п.
Вариант 8. Список водителей (Фамилии И.О.) с указанием марки и гос.номера транспортного средства (фамилии отсортировать по алфавиту), добавить № п/п
Вариант 9. Список аптек с указанием Фамилии И.О. директора, адреса и номера телефона, добавить № п/п.
Вариант 10. Список клиентов (Фамилии И.О.) с указанием паспортных данных (фамилии отсортировать по алфавиту), добавить № п/п.
Вариант 11. Список квартиросъемщиков (Фамилия И.О.) с указанием адреса и лицевого счета (фамилии отсортировать по алфавиту), добавить № п/п

Вариант 12. Список клиентов (Фамилия И.О.) с указанием паспортных данных (фамилии отсортировать по алфавиту), добавить № п/п.
Вариант 13. Список читателей (Фамилия И.О.) с указанием номера читательского билета, домашнего адреса и номера телефона (фамилии отсортировать по алфавиту), добавить № п/п.
Вариант 14. Список предметов, с указанием общего числа часов и Фамилии И.О. преподавателя, добавить № п/п.
Вариант 15. Список клиентов (Фамилия И.О.) с указанием паспортных данных и домашнего адреса (фамилии отсортировать по алфавиту), добавить № п/п.

Задание 5.8. Создать отчеты с добавлением уровней группировки и вычислением итогов (согласно варианту).

Вариант 1. № п/п, Фамилия Имя Отчество клиента, номер, дата заезда, дата выезда, стоимость номера за сутки, сумма к оплате (за проживание в номере). Добавить группировку в отчете по ФИО клиента, внизу отчета рассчитать итоговую сумму к оплате.
Вариант 2. № п/п, Фамилия Имя Отчество подписчика, название периодического издания, срок подписки, стоимость подписки, дата подписки. Добавить группировку в отчете по ФИО подписчика, внизу отчета рассчитать итоговую стоимость подписки.
Вариант 3. № п/п, Фамилия, Имя, Отчество абонента, номер телефона, название города, дата разговора, время разговора, продолжительность разговора, сумма к оплате. Добавить группировку в отчете по ФИО абонента, внизу отчета рассчитать итоговую сумму.
Вариант 4. № п/п, Фамилия, Имя, Отчество клиента, наименование страны, стоимость путевки, дата поездки. Добавить группировку в отчете по наименованию страны, внизу отчета рассчитать итоговую выручку турфирмы.
Вариант 5. № п/п, Фамилия, Имя, Отчество студента, специальность, пол, возраст. Добавить группировку в отчете по названию специальности. Внизу отчета рассчитать минимальный, средний и максимальный возраст студентов.
Вариант 6. № п/п, название фирмы, название, цена и количество товара, дата покупки, сумма покупки. Добавить группировку в отчете по названию фирмы, внизу отчета рассчитать итоговую сумму покупки.

<p>Вариант 7. № п/п, Фамилия, Имя, Отчество заказчика услуги, название услуги, дата начала выполнения услуги, стоимость услуги. Добавить группировку в отчете по названию услуги, внизу отчета рассчитать итоговую стоимость услуг.</p>
<p>Вариант 8. № п/п, Фамилия, Имя, Отчество водителя, название пункта назначения, дата выезда, дата приезда, затраты на горючее. Добавить группировку в отчете по названию пункта назначения, внизу отчета рассчитать суммарные затраты на горючее.</p>
<p>Вариант 9. № п/п, адрес аптеки, название лекарственного средства, количество и цена лекарственного средства, общая стоимость лекарственного средства. Добавить группировку в отчете по адресу аптеки, внизу отчета рассчитать итоговую стоимость лекарственных средств.</p>
<p>Вариант 10. № п/п, Фамилия, Имя, Отчество клиента, наименование товара, дата выдачи, дата возврата, стоимость проката. Добавить группировку в отчете по ФИО клиента, внизу отчета рассчитать итоговую стоимость проката.</p>
<p>Вариант 11. № п/п, Фамилия, Имя, Отчество квартиросъемщика, номер лицевого счета, количество жильцов, площадь квартиры, стоимость услуг. Добавить группировку в отчете по количеству жильцов. Внизу отчета рассчитать итоговую стоимость услуг.</p>
<p>Вариант 12. № п/п, Фамилия, Имя, Отчество клиента, наименование видеодиска, дата выдачи, дата возврата, стоимость проката. Добавить группировку в отчете по наименованию видеодиска, внизу отчета рассчитать итоговую стоимость проката.</p>
<p>Вариант 13. № п/п, Фамилия, Имя, Отчество читателя, автор книги, название книги, дата выдачи, дата возврата фактическая, штраф (1% от цены книги за каждый день просрочки, предполагается, что дата возврата плановая меньше или равна дате возврата фактической). Добавить группировку в отчете по ФИО читателя, внизу отчета рассчитать итоговую сумму штрафов.</p>
<p>Вариант 14. № п/п, Фамилия, Имя, Отчество преподавателя, предмет, дата проведения занятий, количество проведенных часов, сумма к оплате. Добавить группировку в отчете по предмету, внизу отчета рассчитать итоговую сумму к оплате.</p>
<p>Вариант 15. № п/п, фамилию, имя, отчество клиента, название тарифа, абонентская плата, дата подключения. Добавить группировку в отчете по ФИО клиента, внизу отчета рассчитать доход Интернет-провайдера.</p>

ЛАБОРАТОРНАЯ РАБОТА 6. СОЗДАНИЕ МАКРОСОВ В ACCESS

Цель: сформировать навыки создания макросов в БД MS Access.

Теоретическая часть

MS Access имеет широкий набор инструментов для работы с базами данных, таблицами, запросами, формами и отчетам. Но пользователю могут понадобиться сложные приложения. В этом случае прибегают к макросам и модулям.

Любая команда, выполняемая в Access, называется макрокомандой. Набор из одной или нескольких макрокоманд, каждая из которых выполняет определенное действие, называется макросом. Назначение макросов, автоматизация часто выполняемых операций без написания программ.

Макрос, в котором все макрокоманды выполняются последовательно, называется линейным. Макрос, в котором отдельные макрокоманды или их наборы выполняются в зависимости от выполнения некоторого условия, задаваемого логическим выражением, называется макросом с условием. Если в одном макросе объединены несколько логически связанных макросов, то такой макрос называется групповым.

Основная часть

Промышленное предприятие ежемесячно ведет учет выпуска своей продукции, отражая их в документе Выпуск продукции в январе месяце (таблица 6.1).

Таблица 6.1 – Выпуск продукции в январе месяце

Код изделия	Наименование изделия	Сорт	Выпуск изделий по плану	Выпуск изделий
T(4)	T(15)	T(6)	Ч(4)	Ч(4)
1238	стул	третий	2000	2005
...

Задание 6.1. Создайте БД *Продукция*. В БД *Продукция* создайте таблицу «*Выпуск*» со структурой по образцу структуры таблицы 6.1, НЕ ЗАДАВАЯ КЛЮЧЕВОГО ПОЛЯ.



Для правильного выполнения заданий, присвойте полям таблицы следующие имена (с учетом регистров): *Код*, *Наименование*, *Сорт*, *План*, *Факт*.

1. Заполните таблицу не менее 8 записями, повторяя значения в столбце *Сорт* (первый, второй, третий, высший).
2. Используя Мастер форм создайте форму **Форма_1** для просмотра и ввода данных всех записей таблицы **Выпуск**.
3. Используя Конструктор форм добавьте в форму **Форма_1** новое вычисляемое поле *Отклонение*. Для вычисления данных этого поля в **Окне свойств/Данные** в поле *Данные* (рисунок 6.1) введите формулу:

$$= [\text{Факт}] - [\text{План}] \quad (6.1)$$

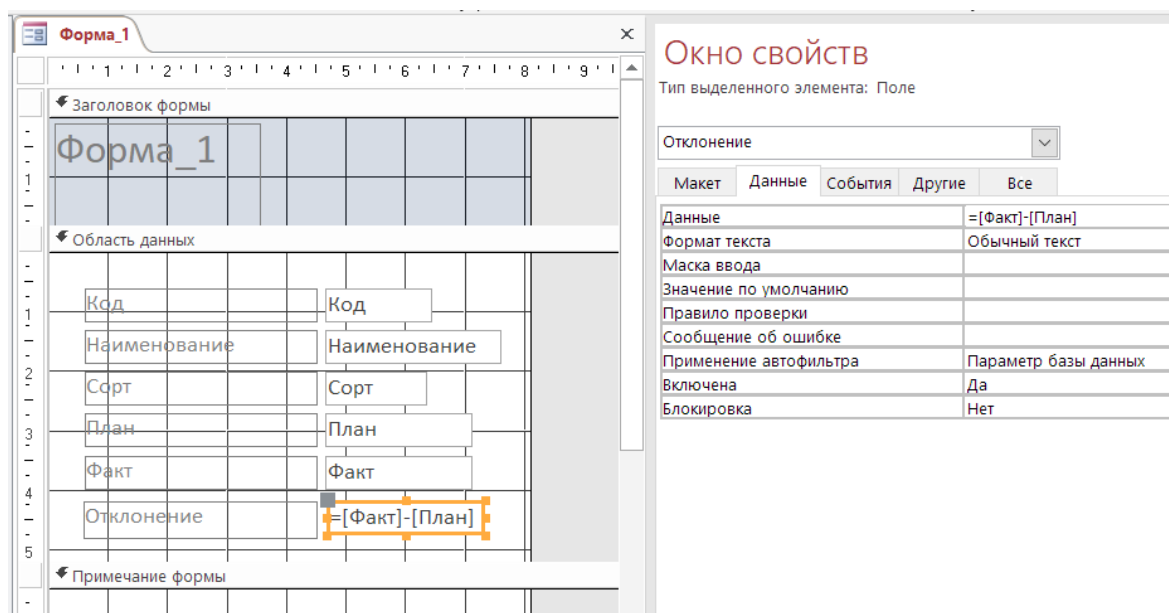


Рисунок 6.1 – Создание вычисляемого поля «Отклонение»

4. Сохраните форму.
5. На основе данных таблицы «Выпуск» сконструируйте отчет Отчет_1 в режиме Мастера отчетов.
6. В отчете **Отклонение**:
 - а) проведите группировку данных по полю Сорт;
 - б) сортировку - по полю Наименование;
 - в) подведите итоги по каждому сорту и общие итоги по полям План, Факт.
7. Используя **Конструктор отчетов**:
 - а) внесите необходимые изменения в дизайн и оформление отчета;

б) введите новое поле Отклонение, вычисляемое по формуле (6.1) (рисунок 6.2).

8. Сохраните отчет и выведите его на предварительный просмотр. Внешний вид отчета должен быть похож на пример на рисунке 6.3.

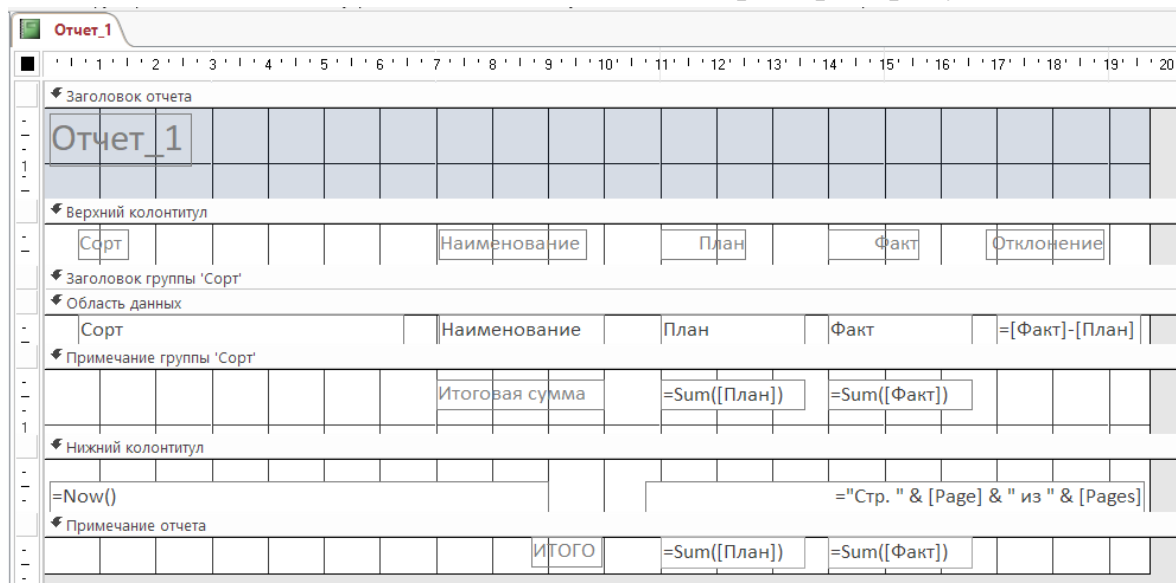


Рисунок 6.2 – Добавление вычисляемого поля в Отчет_1 с помощью Конструктора.

Отчет_1					
Сорт	Наименование	План	Факт	Отклонение	
второй	Товар 2	2050	2000	-50	
	Товар 5	1200	1000	-200	
	Товар 5	500	600	100	
	Итоговая сумма	3750	3600		
высший	Товар 1	1000	1000	0	
	Товар 3	1000	200	-800	
	Итоговая сумма	2000	1200		
первый	Товар 1	2000	1800	-200	
	Товар 10	800	1000	200	

Рисунок 6.3 – Фрагмент готового отчета.

Задание 6.2. Создайте макрос, который выбирает из таблицы **Выпуск** все записи с изделиями третьего сорта.

1. Для создания макроса в окне базы данных перейдите на вкладку Макросы. Откроется окно нового макроса
2. Для выбора первой макрокоманды щелкните левой кнопкой мыши в раскрывающемся списке «Добавить новую макрокоманду» и из раскрывшегося списка выберите Открыть таблицу.

3. После выбора станет активным окно Аргументы макрокоманды. В этом окне выберите таблицу, которую надо открыть, т.е. таблицу Выпуск. Остальные параметры оставьте без изменений (рисунок 6.4).

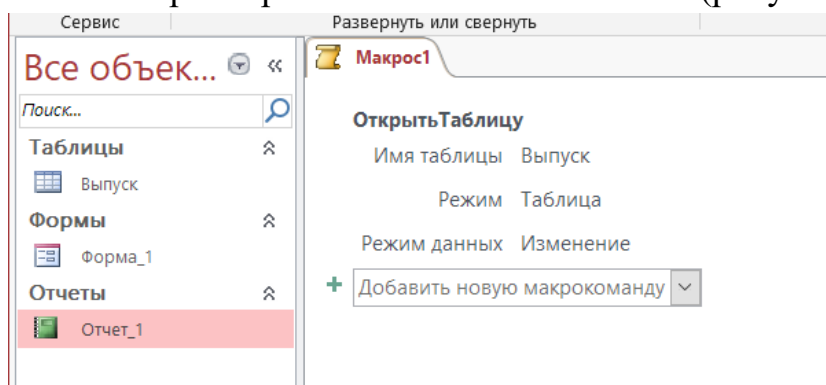


Рисунок 6.4 – Создание макрокоманды «Открыть таблицу»

4. Выберите следующую макрокоманду **ПрименитьФильтр**, щелкнув кнопкой мыши в следующей строке столбца *Макрокоманда*. В окне *Аргументы макрокоманды* укажите условия фильтрации. Для этого в строке *Условие отбора* напишите [Выпуск]![Сорт]='третий' (рисунок 6.5).

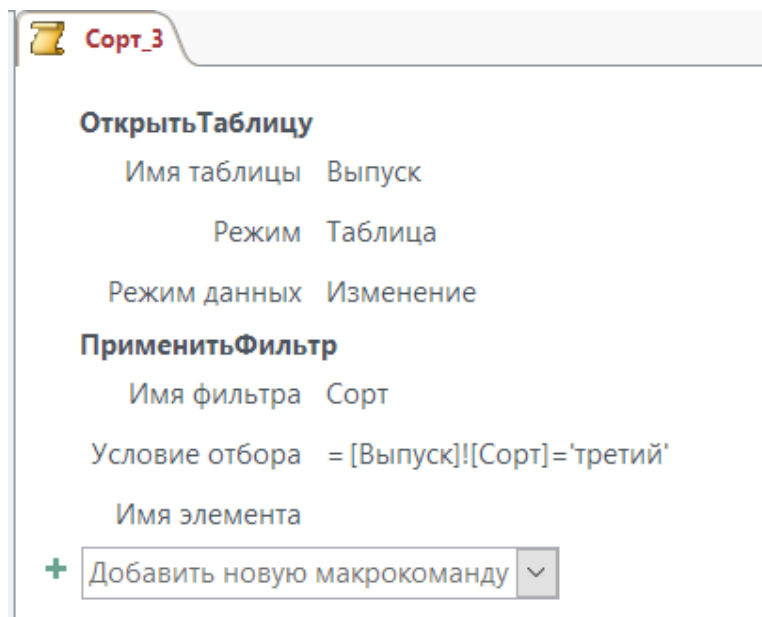


Рисунок 6.5 – Создание макрокоманды «Применить фильтр»

5. Сохраните макрос под именем **Сорт 3**.

6. Проверьте работу макроса. На вкладке Макросы дважды щелкните по имени макроса **Сорт_3**.

Задание 6.3. Создайте макрос, который выбирает из таблицы **Выпуск** все записи с изделиями первого и второго сорта.

1. Для создания макроса необходимо использовать следующий перечень макрокоманд:

Макрокоманда	Аргументы макрокоманды	
ОткрытьТаблицу	Имя таблицы	Выпуск
	Режим	Таблица
	Режим данных	Изменение
ПрименитьФильтр	Имя фильтра	Сорт_1_2
	Условие отбора	[Выпуск]![Сорт]="первый" OR [Выпуск]! [Сорт]="второй"

- Сохраните макрос под именем Сорт_1_2.
- Проверьте работу макроса.

!	Данную задачу можно решить иначе: для выбора всех записей с изделиями первого и второго сорта, используется метод исключения. Для этого в строке Условие отбора наберите: [Выпуск]![Сорт]<>"третий" AND [Выпуск]![Сорт]<>"высший"
---	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- Создайте новый макрос, описанным выше способом и проверьте его работу.

Задание 6.4. Создайте макрос, который при наличии в форме **Форма_1** отрицательного значения **Отклонения** выводит сообщение "*План не выполнен*". В противном случае выдавать сообщение "*План выполнен!*".

!	Для задания цикличности в работе макроса, т.е. для того, чтобы макрос повторял заданную операцию над каждой записью, необходимо создать два макроса: главный и подчиненный. Подчиненный макрос будет запускаться главным макросом определенное количество раз.
---	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

I. Создание подчиненного макроса

- Откройте окно создания нового макроса.
- В списке макрокоманд выберите Если. Заполните поле условием: [Формы]![Форма_1]![Отклонение]>=0.
- Внутри блока Если добавьте из списка новую макрокоманду Окно сообщения. Заполните поля данной макрокоманды по примеру на рисунке 6.6.

```

▢ Если [Формы]![Форма_1]![Отклонение]>=0 то
    Окно сообщения
        Сообщение План выполнен
        Сигнал Да
        Тип Информационное
        Заголовок Администрация

```

Рисунок 6.6 – Описание макрокоманды «Окно сообщение» при условии «Если»

4. Внутри блока Если добавьте следующую макрокоманду Иначе если и заполните поле условием: [Формы]![Форма_1]![Отклонение]<0.

5. Внутри блока Если добавьте из списка новую макрокоманду Окно сообщение. Заполните поля данной макрокоманды по примеру на рисунке 6.7.

```

▢ Иначе если [Формы]![Форма_1]![Отклонение]<0 то
    Окно сообщения
        Сообщение План не выполнен
        Сигнал Да
        Тип Предупреждающее!
        Заголовок Администрация
Конец блока "Если"

```

Рисунок 6.7 – Описание макрокоманды «Окно сообщение» при условии «Иначе если»

6. Выйдите из блока Если и создайте новую макрокоманду НаЗапись. Заполните поля макрокоманды по примеру на рисунке 6.8.

```

НаЗапись
    Тип объекта Форма
    Имя объекта Форма_1
    Запись Следующая
    Смещение 1

```

Рисунок 6.8 – Описание макрокоманды «НаЗапись»

7. Сверьте последовательность записи макрокоманд и правильность заполнения полей подчиненного макроса с рисунком 6.9.
8. Сохраните макрос с именем **Дополнительный**. Закройте макрос.

II. Создание главного макроса

1. Откройте окно создания нового макроса.
2. В списке макрокоманд выберите команду «ОткрытьФорму», затем «ЗапускМакроса». Заполните последовательно поля макрокоманд данными по примеру на рисунке 6.10.
3. Сохраните главный макрос под именем Сообщение и проверьте его работу.

```

❑ Если [Формы]![Форма_1]![Отклонение]>=0 то
    Окно сообщения
        Сообщение План выполнен
        Сигнал Да
        Тип Информационное
        Заголовок Администрация
❑ Иначе если [Формы]![Форма_1]![Отклонение]<0 то
    Окно сообщения
        Сообщение План не выполнен
        Сигнал Да
        Тип Предупреждающее!
        Заголовок Администрация
Конец блока "Если"
НаЗапись
    Тип объекта Форма
    Имя объекта Форма_1
    Запись Следующая
    Смещение 1
  
```

Рисунок 6.9 – Структура макрокоманд подчиненного макроса

```

ОткрытьФорму
    Имя формы Форма_1
    Режим Таблица
    Имя фильтра
    Условие отбора
    Режим данных
    Режим окна Обычное
ЗапускМакроса
    Имя макроса Дополнительный
    Число повторов 11
    Условие повтора
+ [Добавить новую макрокоманду]
  
```

Рисунок 6.10 – Структура макрокоманд главного макроса

Задание 6.5. Создайте макрос позволяющий изменить значение поля План следующим образом:

- если Отклонение = 0, то значение поля План увеличить на 5%;
- если Отклонение < 0, то значение поля План уменьшить на 15%;
- если Отклонение > 0, то значение поля План увеличить на 10%;

1. Откройте объект **Форма_1** в Режиме конструктора. Проверьте задано ли имя вычисляемого поля «Отклонение». Если имя не задано, то откройте **Окно свойств** данного поля, на вкладке **Все** (или **Другие**) в поле **Имя** введите *Отклонение* (рисунок 6.11).

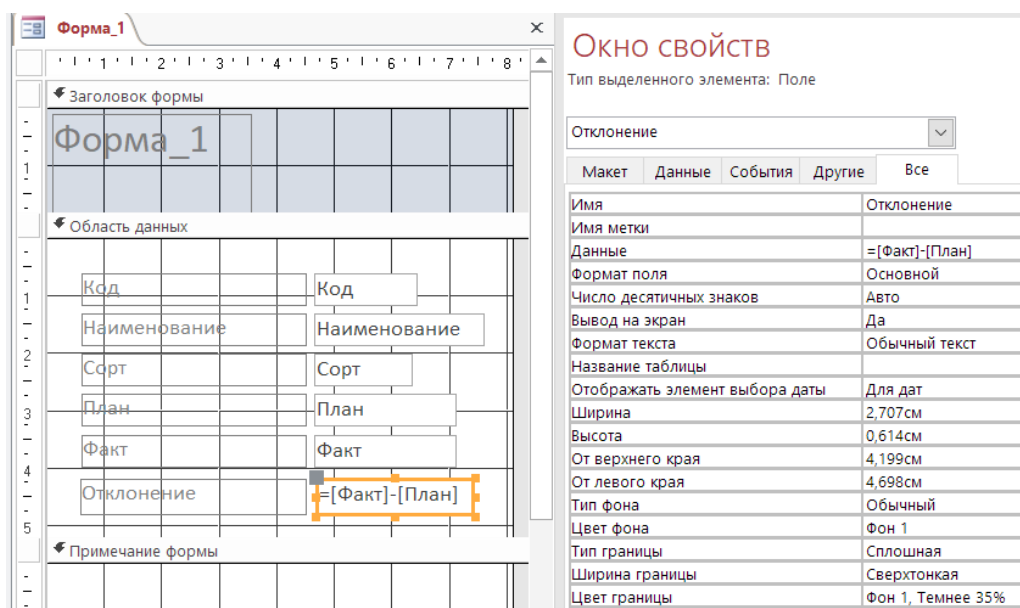


Рисунок 6.11 – Изменение свойств вычисляемого поля «Отклонение»

2. Сохраните Форма_1 с внесенными изменениями и закройте ее.
3. Создайте подчиненный макрос (с именем Изменение_подчиненная) с набором макрокоманд на рисунке 6.12.

!	В случае, если в списке макрокоманд отсутствует макрокоманда ЗадатьЗначение, необходимо на ленте нажать кнопку Показать все действия.
---	---------------------------------------------------------------------------------------------------------------------------------------

4. Создайте главный макрос с именем Изменение) с набором макрокоманд на рисунке 6.13 и проверьте его работу.

Задание 6.6. Создайте на основе таблицы **Выпуск** ленточную форму. Добавьте в нее шесть кнопок и сохраните под именем **Форма_2**.

1. Используя **Мастер форм** создайте на основе таблицы **Выпуск** ленточную форму и сохраните ее под именем **Форма_2**.

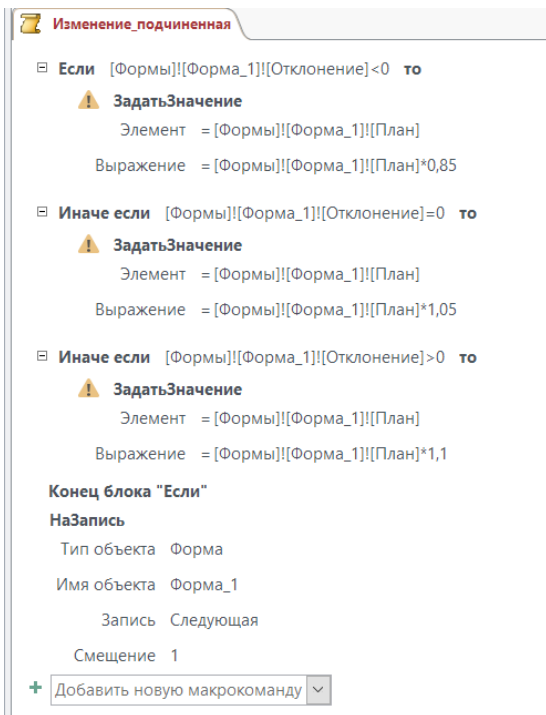


Рисунок 6.12 – Структура макрокоманд подчиненного макроса «Изменение_подчиненная»

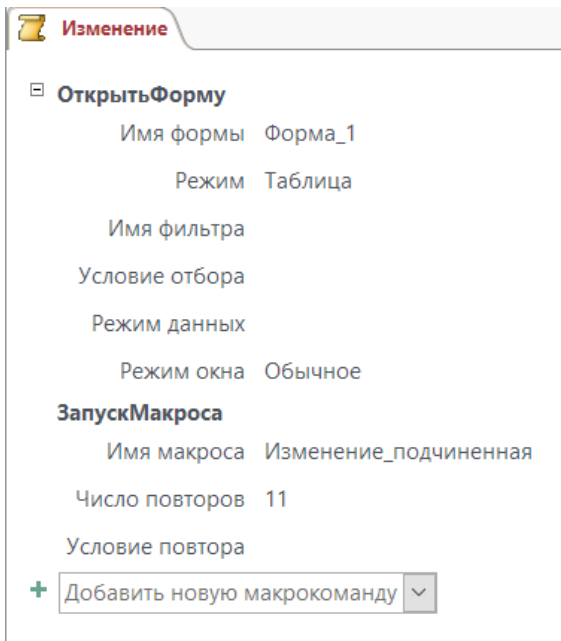


Рисунок 6.13 – Структура макрокоманд главного макроса «Изменение»

2. откройте форму *Форма_2* в **Режиме конструктора**.
3. На **Панели элементов** отключите кнопку **Мастера**, если она включена.
4. Добавьте в область **Примечание формы** *Форма_2* шесть кнопок, расположив их в строку одна за другой.
5. Закройте форму *Форма_2*, сохранив изменения.

Задание 6.7. Создать группу макросов, позволяющих закрепить за кнопками формы *Форма_2* следующие действия:

- просмотр сведений о товарах первого сорта;
- просмотр сведений о товарах второго сорта;
- просмотр сведений о товарах третьего сорта;
- просмотр сведений о товарах высшего сорта;
- просмотр сведений обо всех товарах;
- вывод выбранных сведений на предварительный просмотр.

1. Откройте окно создания нового макроса.

2. Выберите макрокоманду «Применить фильтр». Заполните поля макрокоманды по образцу на рисунке 6.14. Сохраните макрос с именем *C_1*.

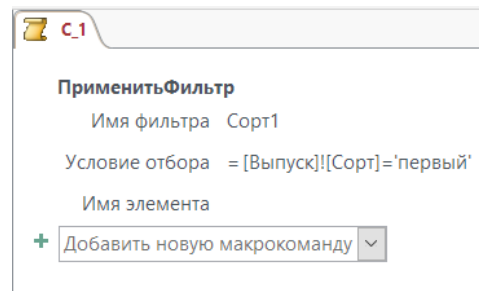


Рисунок 6.14 – Создание макроса *C_1*

3. Аналогично создайте новые макросы *C_2*, *C_3*, *C_В* для отбора товаров 2 сорта, 3 сорта и высшего сорта.

4. Создайте новый макрос (с именем *Все*) с макрокомандой **ПоказатьВсеЗаписи**.

5. Создайте новый макрос (с именем *Просмотр*) с макрокомандой **ПредварительныйПросмотр**.

6. Сохраните и закройте все созданные макросы в п.п. 2-5.

7. Откройте в режиме Конструктора *Форма_2*.

8. Выделите первую кнопку (см. Задание 6.4), откройте Окно свойств. На вкладке *Все* в поле *Подпись* введите имя первой кнопки «Сорт 1».

9. Аналогично задайте имена всем кнопкам: Сорт 2, Сорт 3, Высший, Все, Просмотр, поочередно их выделяя.

10. Выделите первую кнопку «Сорт 1», в Окне свойств перейдите на вкладку *События*.

11. В поле *Нажатие кнопки* из раскрывающегося списка выберите макрос *C_1*, созданный в п. 2 (рисунок 6.15).

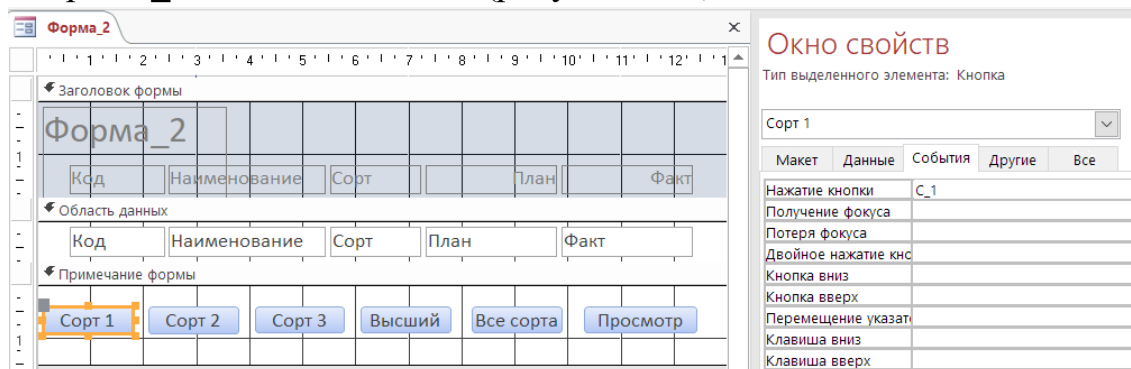


Рисунок 6.15 – Назначение макроса кнопке Сорт 1

12. Повторите выполнение п.11 для каждой кнопки, назначая им соответствующие макросы.

13. Сохраните форму *Форма_2* с внесенными изменениями.

14. Просмотрите результат работы макросов. Для этого откройте форму *Форма_2* в режиме просмотра записей (режим формы) и нажмите кнопки в следующей последовательности: Сорт 1, Просмотр,

Сорт 2, Просмотр, Сорт 3, Просмотр, Высший, Просмотр, Все, Просмотр.

Задание 6.8. Создать макрос (с именем **Кнопочный**), позволяющий в кнопочной форме открывать созданные ранее таблицу Выпуск, формы Форма_1 и Форма_2, отчет Отчет1 и осуществлять выход из Access.

1. Создайте кнопочную форму, используя режим Конструктора.
2. В рабочее поле формы добавьте Область заголовка (Заголовок/примечание формы), в которую введите название формы «Сведения о выпуске товаров» (рисунок 6.16).

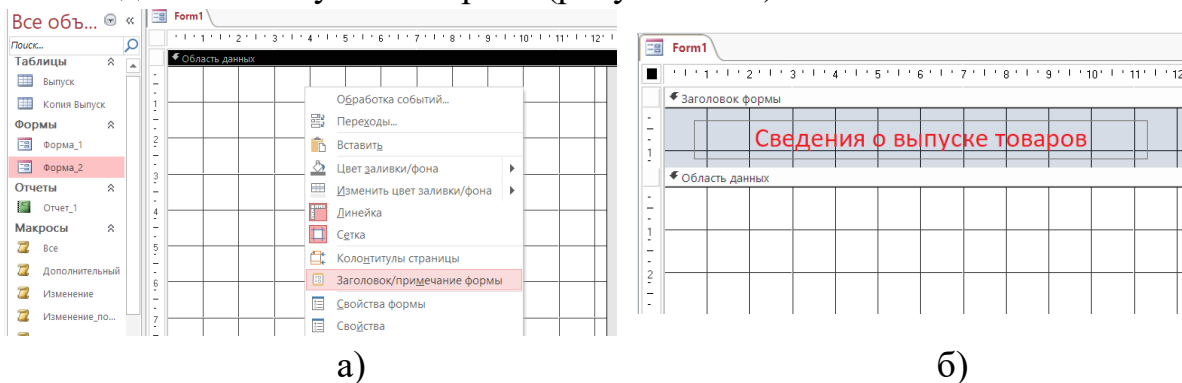


Рисунок 6.16 – Создание заголовка формы

3. В Область данных добавьте пять кнопок, расположив их в столбец одна под одной (предварительно отключив кнопку Мастер, если она включена).
4. Задайте имена кнопкам **Таблица Выпуск**, **Форма Форма_1**, **Форма Форма_2**, **Отчет Отчет_1**, **Выход из Access** (Рисунок 6.17).

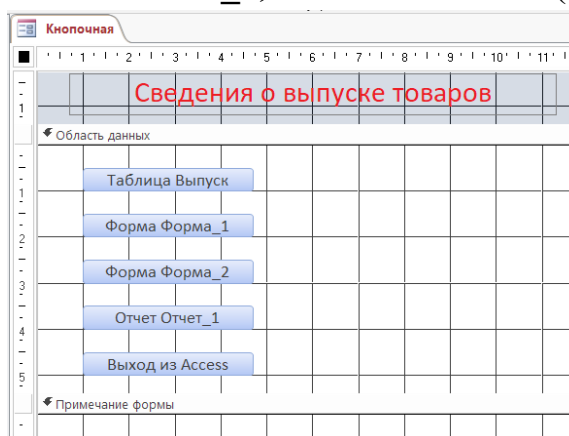


Рисунок 6.17 – Создание кнопок в Области данных формы

5. Закройте форму, сохранив ее под именем Кнопочная.
6. Создайте 5 макросов, каждый из которых будет выполнять соответствующую макрокоманду: Открыть_таблицу, Открыть форму 1, Открыть форму 2, Открыть отчет, Выход. Задайте макросам имена,

соответствующие действиям, которые они будут выполнять. Примеры структур макрокоманд приведены на рисунке 6.18.

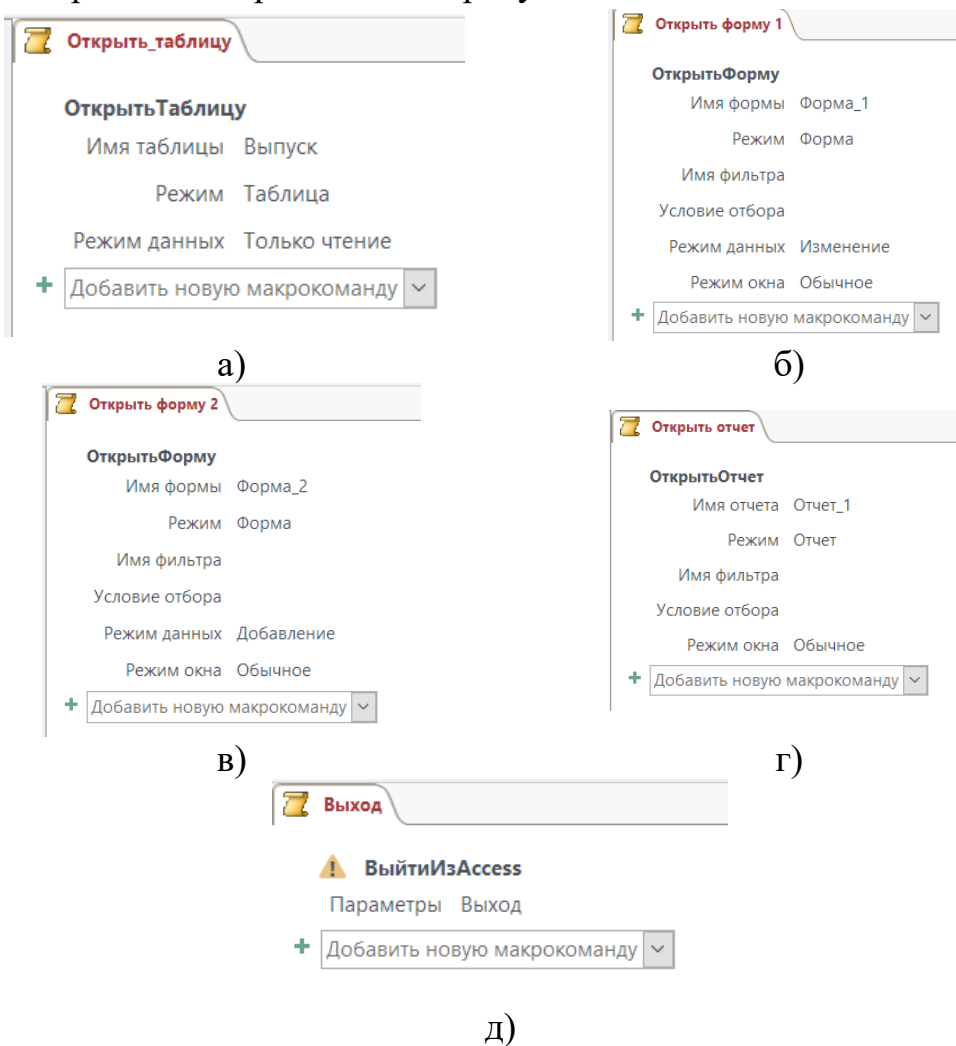


Рисунок 6.18 – Структуры макросов

7. Откройте кнопочную форму и назначьте каждой кнопке выполнение соответствующего макроса.
8. Сохраните изменения в кнопочной форме, закройте ее и проверьте работу макросов.

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Что называется базой данных (БД)?
2. Что такое система управления базами данных (СУБД)?
3. Чем отличается Microsoft Excel от Microsoft Access?
4. Какие объекты базы данных Microsoft Access вы знаете?
5. Какой объект в базе данных является основным?
6. Что называется полями и записями в БД?
7. Какие типы данных вы знаете?
8. Как можно переименовать поле?
9. Как можно создать поле с раскрывающимся списком?
10. С каким расширением сохраняется файл БД Access?
11. С помощью чего можно создавать таблицы?
12. Что такое ключевое поле?
13. Как установить несколько ключевых полей?
14. Как установить связи между таблицами?
15. Какие существуют отношения между таблицами?
16. Что означают на схеме данных «1» и «∞»?
17. Зачем нужен Мастер подстановок?
18. Что такое модель данных?
19. Для чего строится модель данных?
20. Укажите достоинства и недостатки иерархической модели данных.
21. Как организуется физическое размещение данных в БД иерархического типа?
22. Охарактеризуйте сетевую модель данных.
23. Охарактеризуйте реляционную модель данных.
24. Чем отличается реляционная модель данных от предшествующих ей моделей?
25. Что такое простой ключ и составной ключ?
26. Перечислите виды связей между объектами. Охарактеризуйте их.
27. Как проявляется иерархическая подчиненность в связи «один ко многим»?
28. Для чего предназначены запросы?
29. Какие виды запросов вы знаете?
30. С помощью чего можно создавать запросы?
31. Для чего используют запрос с параметром?

32. Как можно сделать вычисления в запросах?
33. Что означает запись в условии запроса « =50»?
34. Можно ли создавать запросы на основе нескольких таблиц?
35. Для чего предназначены формы?
36. Почему форма является незаменимым средством в БД?
37. С помощью чего можно создавать формы?
38. На основе чего можно создавать формы?
39. Как создать кнопку на форме?
40. Как можно разместить несколько таблиц и запросов на одной форме?
41. Как создать главную кнопочную форму?
42. Для чего предназначены отчеты?
43. Какие способы создания отчетов вы знаете?
44. Как в отчетах можно посчитать итоговые значения?
45. Какие итоговые значения можно посчитать в отчетах?
46. Как в Access напечатать почтовые наклейки?

СПИСОК ЛИТЕРАТУРЫ

1. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А.Д.Хомоненко. – 5-е изд., доп. – М.: Бином-Пресс; СПб.: КОРОНА принт, 2006. – 736 с.
2. Бекаревич Ю.Б., Пушкина Н.В. Самоучитель MS Access 2002. СПб.: БХВ-Петербург, 2004. – 720 с.
3. Карчевский Е.М., Филиппов И.Е. Access 2010 в примерах. Учебно-методическое пособие. КФУ, 2011.

ИНФОРМАЦИЯ ОБ АВТОРАХ

ОКЛАДНИКОВА Светлана Владимировна, к.т.н., доцент,
заведующая кафедрой информационных
технологий АГУ

НАБИУЛИНА Луиза Махмудовна, к.пед.н., доцент, доцент кафедры
информационных технологий ТашГПУ

ЗАКИРОВА Мадина Ринатовна, преподаватель кафедры
информационных технологий ТашГПУ