

М. А. Ковалева

2019

Создание баз данных в Microsoft Access

Учебно-методическое пособие



УДК 004.65
ББК 32.973
К 56

Рецензент(ы): Болотаева И. И. – к.т.н., доцент кафедры «Информационные технологии и системы» ФГБОУ ВО «Северо-Кавказский горно-металлургический институт (государственный технологический университет)».

Ковалева, Мария Александровна

К 56 Создание баз данных в Microsoft Access. Учебно-методическое пособие – М.: Мир науки, 2019. – Сетевое издание. Режим доступа:
<https://izd-mn.com/PDF/35MNNPU19.pdf> – Загл. с экрана.

ISBN 978-5-6043306-5-4

Учебное-методическое пособие предназначено для студентов всех форм обучения всех специальностей.

Цель данного пособия – оказать помощь в самостоятельной подготовке студентов направления подготовки 38.03.05 «Бизнес-информатика» изучающих основы проектирования баз данных, и способствовать формированию системного представления относительно основных теоретико-методологических вопросов.

Данное учебно-методическое пособие адресовано студентам младших курсов технических и экономических специальностей для изучения основных понятий теории баз данных, а также получения практических навыков по работе в СУБД MS Access. В практических работах подробно рассмотрены способы создания таблиц, запросов, форм и отчетов. Индивидуальные задания и контрольные вопросы помогут обобщить и систематизировать полученные знания.

Учебно-методическое пособие будет полезно преподавателям, научным сотрудникам, студентам, изучающих применение вычислительных технологий при изучении основ проектирования баз данных, для широкого круга читателей.

ISBN 978-5-6043306-5-4

© Ковалева Мария Александровна
© ООО Издательство «Мир науки», 2019

Оглавление

Введение	4
1. Базы данных	6
2. Работа с базой данных	16
Практическая работа № 1	17
Практическая работа № 2	20
Практическая работа № 3	24
Практическая работа № 4	30
Практическая работа № 5	35
Задания для самостоятельного выполнения	38
Библиографический список.....	43

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

В настоящее время основными направлениями использования средств вычислительной техники и программных продуктов для информационного обеспечения товароведения и экспертизы товаров являются следующие:

- использование существующих автоматизированных информационных систем и их конфигурирования с учетом определенного типа предприятия и класса решаемых задач;

- разработка новых информационных систем для решения конкретных задач.

Информационной системой называется программно-аппаратный комплекс, обеспечивающий выполнение следующих функций:

- надежное хранение информации в памяти компьютера
- выполнение специфических операций преобразования информации
- предоставление пользователю удобного интерфейса.

Эти задачи могут быть решены с помощью универсальной системы автоматизации деятельности предприятия 1С:Предприятие. Эта система может быть использована для автоматизации самых различных аспектов деятельности предприятия, таких, например, как оперативный учет товарных и материальных средств, учет складских и торговых операций, обслуживания отдела кадров, расчетов заработной платы и др.

Отличительной особенностью системы 1С:Предприятие является ее конфигурируемость.

Конфигурация представляет собой модель предметной области и создается штатными средствами системы. Средства конфигурирования позволяют описать структуры данных, характеризующие объекты предметной области, и алгоритмы их обработки с целью отражения особенностей учета.

Для описания алгоритмов решения прикладной задачи используется специальный предметно-ориентированный язык программирования, встроенный в систему 1С:Предприятие. Описание языка программирования 1С:Предприятие занимает примерно 900 страниц. Этот язык предназначен для специалистов, выполняющих конфигурирование системы для решения конкретной задачи автоматизации учета.

Одним из базовых объектов системы являются документы, которые предназначены для хранения основной информации о событиях, происходящих на складе, отделе, предприятии - объектах предметной области. Типовыми примерами документов являются такие, например, как "Приходная накладная", "Расходная накладная", "Накладная на внутреннее перемещение", "Платежное поручение", "Счет", "Протокол испытаний", акты экспертиз и др.

Для ввода информации используются экранные формы. Каждый документ отражает определенный тип событий. Структура документа и свойства событий описываются в конфигурации.

Конфигурация может быть поставлена фирмой "1С". Система 1С:Предприятие вместе с конфигурацией является готовым программным продуктом, ориентированным на определенные классы решаемых задач для конкретной предметной области.

Очевидно, недостатком системы 1С:Предприятие является ее некоторая консервативность, обусловленная тем, что модули системы и алгоритмы должны настраиваться в процессе ее конфигурации специалистами фирмы "1С". Следует также иметь в виду, что изменение форм отчетности и условий функционирования предприятия каждый раз вызывает необходимость реконфигурирования системы 1С:Предприятие.

Другим более эффективным способом решения задач, которые сформулированы в определении информационной системы, являются базы данных (БД) и системы управления базами данных (СУБД).

База данных - это именованная совокупность структурированных данных, отражающая состояние объектов и их отношений в конкретной предметной области.

В узком смысле база данных представляет собой набор данных о конкретном объекте, организованных определенным образом.

В широком смысле база данных представляет собой совокупность сведений о конкретных объектах.

Целью базы данных является необходимость упорядочить фактические данные по различным признакам, так чтобы иметь возможность быстро извлекать требуемую информацию.

Как правило, с одной базой данных может работать несколько пользователей. Программы, с помощью которых пользователи работают с базой данных, называются приложениями.

Для работы с одной базой данных может использоваться несколько приложений. Например, если база данных моделирует предприятие, то одно приложение может обслуживать отдел складского учета товаров, другое приложение может учитывать движение материальных ценностей, третье приложение может обслуживать планирование технологических операций.

Создание базы данных, ее поддержка, управление, доступ пользователя к самим данным осуществляется с помощью СУБД.

СУБД представляет собой программные средства для создания, ведения и совместного использования базы данных многими пользователями.

СУБД обеспечивает корректную работу нескольких приложений с единой базой данных, так чтобы каждое приложение учитывало все изменения, вносимые в базу данных другими приложениями.

Среди наиболее доступных и широко известных СУБД следует отметить СУБД Microsoft Access, которая объединяет сведения из разных источников в одной реляционной базе данных.

Создаваемые с помощью Microsoft Access формы, запросы и отчеты позволяют быстро и эффективно обновлять данные, получать ответы на вопросы, осуществлять поиск нужных данных, анализировать данные, выводить на печать структурированные данные в удобном для пользователя формате утвержденных форм и документов (различных накладных, платежных поручений и др.), печатать отчеты и диаграммы.

1. Базы данных

Классификация баз данных

В зависимости от принципов обработки данных базы данных делятся на следующие:

- централизованные базы данных (ЦБД)
- распределенные базы данных (РБД).
- В централизованной БД база данных размещается на одном компьютере, который может иметь или не иметь поддержки сети.
- Если компьютер не имеет поддержки сети, то в этом случае может быть реализован только локальный доступ.
- Если же компьютер включен в локальную сеть, то в этом случае может быть реализован централизованный доступ.

В настоящее время наиболее широко применяются централизованные базы данных с централизованным доступом. При такой технологии возможны два способа обработки данных:

- Файл-сервер
- Клиент-сервер

Для реализации технологии **файл-сервер** необходимо, чтобы один из компьютеров сети функционировал в качестве сервера, на котором хранятся файлы централизованной базы данных.

В соответствии с запросами пользователей файлы с файл-сервера передаются на рабочие станции пользователей, где и осуществляется обработка данных. По завершении работы с базой данных пользователи копируют файлы с обработанными данными обратно на сервер.

Возможности пользователей вычислительной сети определяются их правами доступа. При регистрации пользователя на рабочей станции возможны два уровня доступа к данным: администратор и рядовой пользователь.

Администратор получает возможность работать не только со всеми командами, доступными рядовому пользователю, но и с командами, которые определяют его привилегированное положение в сети и доступны только ему.

Рядовой пользователь получает только те права доступа, которые определяет администратор при создании пользователя или группы пользователей.

Для того чтобы войти в сеть и работать с сетевыми ресурсами, пользователь должен знать свой логин (сетевое имя) и пароль. Логин и пароль для пользователя устанавливает администратор. Пароль пользователя должен быть уникальным.

Основными недостатками технологии файл-сервер являются следующие:

- Между рабочими станциями пользователей и сервером по сети передаются большие объемы информации.
- При одновременном обращении к одним и тем же данным пользователи вынуждены дожидаться освобождения этих данных.

Технология клиент-сервер реализуется следующим образом.

Центральный сервер обеспечивает не только хранение файлов базы данных, но и выполняет основную часть обработки данных. Пользователи обращаются к центральному серверу с запросами, которые формулируют с помощью специального языка структурированных запросов - SQL. Запрос описывает действия и список задач, которые должен выполнить сервер. Запросы принимаются сервером и инициируют процессы обработки данных. В соответствии с полученными инструкциями сервер выполняет необходимые действия. В ответ пользователь получает по сети только обработанный набор данных.

При такой технологии между пользователем и сервером передается не весь объем данных, а только данные, необходимые пользователю.

Архитектура централизованной базы данных с централизованным доступом по технологии клиент-сервер имеет ряд преимуществ по сравнению с технологией файл-сервер, а именно:

По сети передается не весь объем информации, а только данные, необходимые пользователю.

Позволяет избежать конфликтов изменений одних и тех же данных множеством пользователей.

Организует согласованное изменение данных множеством пользователей, поддерживая автоматически целостность данных.

Недостатком этой технологии являются высокие требования к характеристикам центрального сервера.

Распределенная база данных может состоять из нескольких частей, хранимых в различных ЭВМ вычислительной сети. Для работы с такими базами данных используют системы управления распределенными базами данных (СУРБД).

В зависимости от характера информационных ресурсов, содержащихся в базе данных, базы данных делятся на документальные и фактографические.

Документальные базы данных представляют собой информационно-поисковые системы документов, отсортированных по какому-либо признаку.

Фактографические базы данных предназначены для решения задач обработки данных - задач, связанных с вводом, хранением, сортировкой, отбором по заданным критериям и группировкой записей данных однородной структуры. Характерной особенностью фактографических баз данных является то, что они содержат фактические сведения, представленные в виде совокупностей форматированных записей данных.

Структура MS Access

Первая версия MS Access была разработана в 1993 г. фирмой Microsoft. MS Access - это функционально полная реляционная СУБД, работающая в среде Windows. Access позволяет создавать сложные базы данных, задавая структуру таблиц и определяя связи между ними. MS Access содержит возможности для создания запросов, отчетов и форм любой сложности. В Access можно использовать все возможности Windows обмена данными между приложениями (DDE и OLE), что позволяет включить в базу данных графическую и звуковую информацию.

База данных Access состоит из объектов, связанных с хранимыми данными (таблицы, запросы, формы, отчеты, страницы, макросы, модули). Все объекты Access хранятся в одном файле с расширением .mdb. В таблицах хранятся данные, которые можно добавлять, редактировать, просматривать. Запросы позволяют -- 8

быстро выбирать необходимую информацию из таблиц. Используя формы, можно вводить данные в таблицы, выводить на экран в удобном виде, просматривать и изменять их. С помощью отчетов можно создавать различные виды документов для вывода на печать. Опубликованная средствами Access веб-страница, имеющая подключение к базе данных, позволяет просматривать, добавлять, изменять и обрабатывать данные, хранящиеся в базе данных, из обозревателя. Макросы и модули используются для автоматизации работы с базой данных.

Реляционная модель данных

Центральным понятием в области баз данных является модель данных.

Модель представляет собой такое структурированное представление конкретных данных о свойствах объекта, которое позволяет разработчикам баз данных трактовать их как сведения, содержащие не только данные, но и взаимосвязь между ними.

В настоящее время среди различных моделей данных наиболее перспективной является реляционная модель данных, которая фактически является монополистом на рынке баз данных.

Реляционная модель данных была предложена американским математиком, сотрудником фирмы IBM Э. Коддом в 1970 году.

Теоретической основой реляционной модели данных стала теория отношений. Основной структурой данных в этой модели является отношение. Э. Кодд показал, что любое представление данных можно свести к совокупности двумерных таблиц особого вида, называемых отношением. В английской терминологии *relation* - это отношение. Поэтому модель данных получила название реляционной.

Каждый элемент отношения может быть представлен в формате

$$A[i,j],$$

где: A - элемент данных, i - строка отношения, j - номер атрибута отношения

Количество атрибутов в отношении определяет степень этого отношения.

Отношение степени 1 называется унарным, отношение степени 2 - бинарным, отношение степени 3 - тернарным, а отношение степени n - n-арным.

Множество всех значений A[i,j] при i = const и всех возможных j образует кортеж или строку таблицы.

Множество всех кортежей отношения образует тело отношения.

Реляционной считается такая база данных, в которой все данные представлены в виде двумерных таблиц и все операции над базой сводятся к манипуляциям с таблицами или отношениями (в терминологии баз данных отношение - это таблица или сущность).

Реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:

- каждый элемент таблицы является минимальным элементом данных;
- все элементы в пределах одного столбца имеют одинаковый тип данных (текстовый, числовой, логический и т. д.);
- все столбцы должны иметь уникальное имя;
- в таблице нет совпадающих строк;
- порядок следования столбцов и строк может быть произвольным (см., например, таблицу **Сессия**, в соответствии с рисунком 1.

дата	вид_контроля	предмет	группа
18.12.2012	зачет	Информатика	МЕН2-3
18.12.2012	зачет	Ин.язык	ПО1-1
18.12.2010	зачет	Математика	ПО1-1
19.12.2012	зачет	История	СОЦ1-3
19.12.2012	экзамен	Информатика	МЕН2-3
19.12.2012	зачет	Статистика	МЕН2-3
20.12.2012	экзамен	Информатика	ПО1-1
20.12.2012	зачет	Философия	СОЦ1-3
21.12.2012	экзамен	Информатика	СОЦ1-3
21.12.2012	зачет	Ин.язык	МЕН2-3

Рисунок 1 – Таблица Сессия

Каждый из атрибутов расположен в отдельном столбце. Каждая строка представляет собой набор атрибутов, описывающий конкретный экземпляр объекта - Сессия.

Каждое поле таблицы должно являться отдельным атомарным значением для конкретной предметной области. Это атомарное значение не должно разлагаться на более простые составляющие.

В некоторых случаях в качестве атомарного значения могут рассматриваться агрегированные объекты. Например, фамилия, имя и отчество или адрес и его атрибуты.

Между реляционными таблицами должны быть установлены связи. Связь между таблицами характеризует способ, которым информация одной таблицы связана с информацией в другой.

С этой целью необходимо установить идентификатор или ключ, который бы позволил уникально опознать любую строку таблицы.

Ключом называется множество атрибутов, задание значений которых позволяет однозначно определить значения остальных атрибутов таблицы.

В качестве ключа может выступать учебный шифр студента (номер зачетки), номер группы, код заказа, код клиента и др.

Один из возможных ключей отношения может быть выбран в качестве первичного ключа.

Проектирование реляционных баз данных

Реляционная база данных представляет собой совокупность двумерных реляционных таблиц, в которых хранится вся информация о конкретной предметной области и все операции над базой данных сводятся к манипуляциям. Кроме таблиц в базе данных могут храниться и другие объекты, а именно: экранные формы, отчеты, представления и др.

Таблицы реляционной базы данных должны быть взаимосвязаны между собой. В каждой связи одно отношение может выступать как главное, а другое - как подчиненное. Для поддержки этих отношений используются ключи.

Реляционная таблица состоит из строк (записей) и столбцов (полей) и имеет уникальное в пределах базы данных имя. Каждая таблица имеет один или несколько столбцов, значения в которых однозначно идентифицируют каждую ее строку. Такой столбец или их комбинация называется первичным ключом. В таблице **Сессия** первичным ключом является комбинация столбцов **Код преподавателя** и **Код дисциплины** (один клиент может иметь несколько заказов). Первичный ключ должен содержать уникальные непустые значения.

Таблица отражает сущность (класс объектов) реального мира, а каждая ее строка - конкретный экземпляр этой сущности. Так, например, таблица **Преподаватели**, рисунок 1, содержит всю необходимую информацию о регистрации и исполнении заказов.

Целью проектирования базы данных является сокращение избыточности хранимых данных. База данных должна обеспечить оптимальное использование оперативной и дисковой памяти, возможность изменения данных и обеспечить целостность данных.

К проекту реляционной базы данных предъявляются следующие требования.

- Каждая таблица должна иметь уникальное в пределах базы данных имя.
- Все строки таблицы должны быть однотипны.
- Каждая строка таблицы должна отличаться от любой другой строки хотя бы одним значением.
- Каждая строка таблицы не должна зависеть от значений, хранимых в другой строке.
- Каждому столбцу таблицы должно быть присвоено уникальное в пределах базы данных имя.
- Разрешается свободно обращаться к любой строке или столбцу.

Нормализация данных

Нормализацией называется процесс приведения структур данных в состояние, обеспечивающее лучшие условия выборки, включения, изменения и удаления данных. Это

достигается разбиением одной большой таблицы на две или более мелких. Конечной целью нормализации является получение такого проекта базы данных, в котором каждый факт появляется лишь в одном месте.

Таблица, в которую включены все интересующие атрибуты, называется универсальным отношением. При использовании универсального отношения база данных будет состоять лишь из одной таблицы, в которой будет храниться вся информация о рассматриваемом объекте.

Таблицу, содержащую в одном или нескольких полях большое количество повторяющихся данных, можно разделить на две или более связанных таблиц. Такой способ, позволяющий более эффективно хранить данные, называют нормализацией таблиц.

В некоторых СУБД, например Access, предусмотрен мастер анализа таблиц, который позволяет нормализовать таблицы базы данных. При использовании мастера пользователь имеет возможность самостоятельно определить создаваемые таблицы или позволить мастеру провести нормализацию таблиц.

Мастер анализа таблиц преобразует таблицу, содержащую повторяющиеся данные, в набор связанных таблиц, где уже нет повторений. Это повышает эффективность работы с базой данных и уменьшает ее размер. После создания набора таблиц данные по-прежнему можно просматривать и обрабатывать вместе, создав для этого запрос.

Универсальное отношение порождает ряд проблем.

- Избыточность
- Аномалии обновления (потенциальная противоречивость)
- Аномалии включения
- Аномалии удаления

Большая часть проблем исчезнет, если данные из универсальной таблицы разнести в несколько более мелких таблиц. Эту задачу можно решить путем нормализации.

Процесс нормализации состоит из нескольких этапов. На каждом из этапов изменяют структуру данных так, чтобы она удовлетворяла определенным критериям - требованиям нормальной формы.

Нормализация представляет собой последовательное изменение структуры данных в соответствии с требованиями нормальных форм.

Всего существует шесть нормальных форм. Теория нормализации опирается на довольно сложный математический аппарат реляционной алгебры, изложение которого вытекает за рамки данного пособия.

Дадим определение первых трех нормальных форм.

Таблица находится в первой нормальной форме, если ни одно поле строки не содержит более одного значения и любое ключевое поле не является пустым. Как следует из определения, любая таблица реляционной базы данных автоматически удовлетворяет требованиям первой нормальной формы.

Таблица находится во второй нормальной форме, если она удовлетворяет требованиям первой нормальной формы, и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом. Как следует из этого определения, необходимо чтобы только первичный ключ однозначно идентифицировал значения полей в любом столбце и в то же время значения полей в столбцах не зависели от любой части составного ключа. Если первичный ключ состоит из одного столбца, то это требование удовлетворяется автоматически. Если же первичный ключ является составным, то есть состоит из двух и более столбцов, то такая таблица не обязательно будет находиться во второй нормальной форме. Если таблица не удовлетворяет требованиям второй нормальной формы, то она должна быть разбита на две или более таблиц так, чтобы первичный ключ однозначно идентифицировал значение в любом столбце.

Таблица находится в третьей нормальной форме тогда и только тогда, когда она удовлетворяет требованиям второй нормальной формы и ни одно из ее не ключевых полей не

зависит функционально от любого не ключевого поля. Любое не ключевое поле должно зависеть только от первичного ключа.

Практически таблицы реляционной базы данных не обязательно должны удовлетворять всем требованиям нормальных форм.

Связывание таблиц

Пока данные хранятся в универсальном отношении можно сразу получить всю необходимую информацию. После нормализации доступ к данным усложняется, так как вместо одной таблицы имеется набор множества таблиц. Чтобы выбрать необходимую информацию необходимо рассмотреть несколько таблиц.

Для связывания данных в таблицах необходимо использовать ключи. Взаимосвязь таблиц поддерживается внешними ключами. Внешний ключ создается в таблице, поля которой ссылаются на строки главной таблицы. Каждому значению внешнего ключа должно быть сопоставлено значение первичного ключа. В отличие от первичного, внешний ключ не должен быть уникальным. В зависимой таблице могут быть строки, имеющие одинаковые значения внешнего ключа.

Группа связанных таблиц называется схемой данных. Пример схемы данных показан на рис.2.

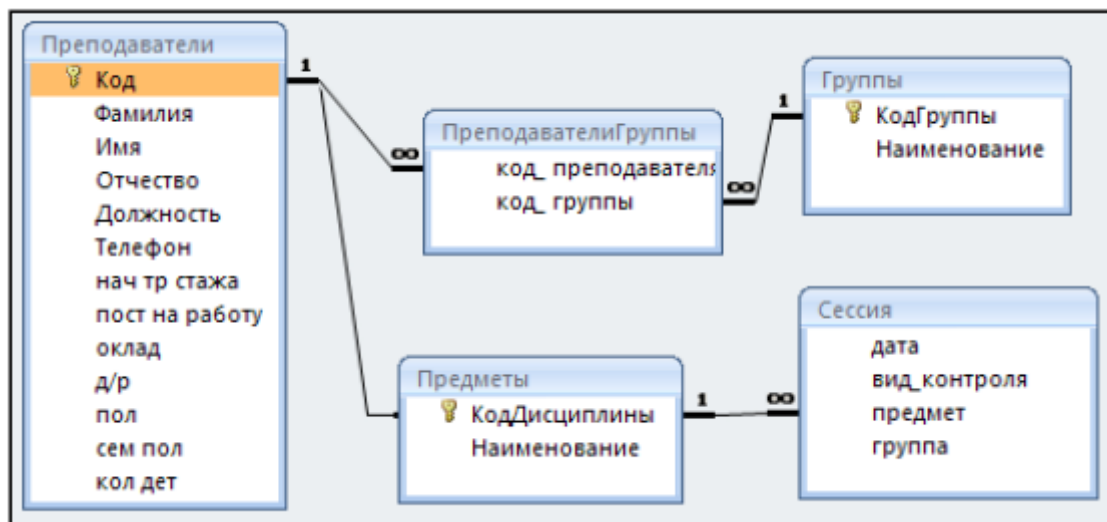


Рисунок 2 – Фрагмент схемы данных базы данных

Типы связей между таблицами

Тип связи определяет правила сопоставления строк между двумя таблицами. Существуют следующие виды связей или отношений.

Один - к - одному (1:1). Такое отношение означает, что каждой строке первой таблицы соответствует только одна строка во второй таблице и, наоборот, каждой строке второй таблицы соответствует только одна строка в первой таблице. Таблицы, связанные отношением один к одному можно объединить в одну общую таблицу, состоящую из полей обеих таблиц. Отношение один к одному может использоваться для разделения таблиц, состоящих из большого числа полей. Такое разбиение может потребоваться, если некоторые поля таблицы содержат конфиденциальную информацию или требуется создать условия для ускоренного просмотра данных.

Один - ко - многим. Такая связь определяет отношение между таблицами, когда одна из них является главной, а другая подчиненной. При этом каждой строке главной таблицы может соответствовать несколько строк в подчиненной таблице, а каждой строке в подчиненной таблице соответствует только одна в главной таблице. Примером такого отношения является связь между таблицами Преподаватели и Дисциплина, устанавливаемая между полями **Код клиента** и Код клиента (см., например, схему данных базы данных). В

отношении один – ко – многим главной таблицей является таблица, которая содержит первичный ключ, который составляет часть один в отношении один - ко - многим (Код клиента в таблице Клиенты). Каждый преподаватель может иметь или одну дисциплину, или несколько, или не иметь их совсем. Каждая дисциплина в подчиненной таблице **Преподаватели** должен принадлежать только одному преподавателю.

Многие - ко - многим. При такой связи каждой строке первой таблицы может соответствовать несколько строк во второй таблице и наоборот. Примером такой связи является связь между таблицами **Заказы** и **Каталог**. Один заказ может содержать много моделей мебели и каждая конкретная модель может быть включена во множество заказов. Такая связь может быть реализована только через третью таблицу, с которой исходные таблицы будут иметь связи один - ко - многим.

Для рассматриваемого примера такой таблицей является таблица **Состав заказа**, которая имеет внешние ключи, образованные из составных первичных ключей таблиц **Заказы** и **Каталог**.

Формирование запросов к базе данных

Доступ к данным осуществляется с помощью запросов к базе данных. Эти запросы формулируются на стандартном языке запросов. Для большинства СУБД таким языком является структурированный язык запросов -Structured Query Language (SQL).

Его появление и развитие связано с созданием теории реляционных баз данных. Первая версия языка была разработана в 1970 году фирмой IBM.

За прошедшие с момента появления этого языка годы SQL претерпел существенные изменения и стал более гибким. Вследствие конкуренции на рынке программных продуктов многие фирмы создавали различные модификации и диалекты языка. Эта ситуация негативно отражалась на совместимости программных продуктов, что и явилось объективной необходимостью стандартизации языков доступа к данным, применяемых в различных СУБД. В 1992 году американским национальным институтом стандартов (ANSI) был разработан стандарт языка, названный ANSI SQL-92.

Действующим в настоящее время стандартом языка SQL для большинства реляционных СУБД является принятая Американским национальным институтом стандартов (ANSI) версия SQL 3.

Однако производители СУБД продолжают модернизировать и улучшать возможности языка доступа к данным. Так, например, фирма Microsoft разработала свою версию языка доступа к данным, назвав его Transact-SQL. Этот язык, удовлетворяя требованиям стандарта ANSI SQL-92, предлагает ряд дополнительных возможностей для более эффективного доступа к данным. Transact-SQL и используется для доступа к данным в SQL Server 7.0.

Язык SQL имеет унифицированный набор инструкций. Он также имеет графическую версию - это Query-By Example (QBE). Используя специальный бланк запроса QBE, запросы формулируются посредством графического представления.

Язык SQL не является языком программирования в традиционном понимании. Он не содержит операторы, управляющие ходом выполнения программы. Он содержит только набор стандартных операторов доступа к данным, хранящимся в базе данных. На нем формулируются только запросы к базе данных. С помощью языка SQL можно сформулировать, что необходимо получить в базе данных. Кроме оператора выбора SELECT, язык SQL содержит операторы определения и администрирования данных, операторы манипулирования данными и другие средства.

Наиболее важные операторы, которые входят в стандарт ANSI SQL, приведены в таблице 1

Таблица 1 – Операторы доступа к данным

Оператор	Действие
Оператор выбора SELECT	
SELECT	Оператор, формирующий в соответствии с SQL-запросом результирующее отношение
Операторы определения данных	
CREATE TABLE	Создает новую
DROP TABLE	Удаляет таблицу из базы данных
ALTER TABLE	Изменяет структуру существующей таблицы
CREATE VIEW	Создает виртуальную таблицу, соответствующую SQL-запросу
DROP VIEW	Удаляет ранее созданное представление
ALTER VIEW	Изменяет ранее созданное представление
CREATE INDEX	Создает индекс для обеспечения быстрого доступа
DROP INDEX	Удаляет ранее созданный индекс
DELETE	Удаляет в соответствии с условиями фильтрации строки из таблицы
INSERT	Вставляет строку в базовую таблицу
UPDATE	Обновляет в соответствии с условиями значения одного или нескольких столбцов
Операторы администрирования данных	
ALTER PASSWORD	Изменить пароль для доступа к базе данных
CREATE DATABASE	Создать новую базу данных
DROP DATABASE	Удалить существующую базу данных
GRANT	Предоставить права доступа на ряд действий с объектом базы данных
REVOKE	Лишить прав доступа к некоторому объекту

Оператор выбора SELECT

Одним из основных инструментов обработки данных в СУБД является выборка данных с помощью запросов. Запрос строится на основе одной или нескольких таблиц. Запрос позволяет выбрать необходимые данные из одной или нескольких таблиц, произвести вычисления и получить результат в виде таблицы. Через запрос можно производить обновление данных в таблицах, создать новую таблицу, используя данные из существующих объектов базы данных. Результатом запроса на выборку является таблица.

Ключевое слово **SELECT** сообщает СУБД, что эта команда - запрос. Все запросы начинаются этим словом с последующим пробелом.

Синтаксис оператора (команды) выбора **SELECT** имеет вид:

SELECT[ALL / DISTINCT] (<список полей>/*)

FROM (<список таблиц>)

[WHERE <предикат - условие выборки или соединения>]

[**GROUP BY**< список полей группировки>]

[**HAVING** < предикат - условие для группы>]

[**ORDER BY** < список полей, определяющих порядок сортировки>]

Звездочка в операторе **SELECT** означает выбор всех столбцов из таблицы.

Ключевое слово **ALL** означает, что результат выборки включает все строки, возвращаемые запросом. Указание параметра **ALL** не является обязательным, так как он включается в конструкцию оператора выбора **SELECT** по умолчанию.

Ключевое слово **DISTINCT** позволяет исключить из результата выборки повторяющиеся строки и тем самым обеспечить уникальность каждой строки результата. Если параметр **DISTINCT** не указывается, то будет использоваться **ALL**.

Раздел **FROM** должен быть представлен в каждом запросе. В этом разделе **FROM** указываются таблицы и представления, из которых будет производиться выборка данных.

Раздел **WHERE** предназначен для ограничения количества строк, включаемых в результат выборки в соответствии с логическим условием.

В выражении условий ключевого слова **WHERE** могут использоваться следующие основные предикаты:

- Предикаты сравнения (=, <>, >, <, >=, <=), имеющие традиционный смысл.
- Предикат **Between A and B**. Предикат принимает значение "истина", когда сравниваемое значение попадает в диапазон A-B, включая его границы.
- Предикат **Not Between A and B**. Предикат истинен тогда, когда сравниваемое значение не попадает в диапазон A-B, включая его границы.
- Предикат вхождения в множество **IN** (множество). Предикат принимает значение "истина", когда сравниваемое значение входит в множество, которое может быть задано простым перечислением. Существует также предикат **NOT IN** (множество), который принимает значение "истина", когда сравниваемое значение не входит в заданное множество.
- Предикаты сравнения с образцом **LIKE** и **NOT LIKE**. Предикат сравнения **LIKE** принимает значение "истина", когда сравниваемое значение соответствует заданному шаблону, в противном случае этот предикат принимает значение "ложь".
- Предикат **NOT LIKE** имеет противоположный смысл.

Тестовые задания:

1.1. Что означает **SQL**?

- A. Стандартный язык запросов для работы с реляционными базами данных.
- B. Программа.
- C. Алгоритмический язык для работы с базой данных.

1.2. Что такое **QBE**?

- A. Диалоговое окно базы данных.
- B. Язык программирования.
- C. Графическая версия языка **SQL**, бланк запроса по образцу.

1.3. Что является результатом выполнения запроса

SELECT КодКлинта, КодЗаказа

FROM Клиенты;

- A. Набор данных
- B. Таблица с двумя столбцами.
- C. Таблица с двумя столбцами **КодКлинта** и **КодЗаказа**, выбранными из исходной таблицы **Клиенты**

1.4. Что является результатом выполнения запроса

SELECT *

FROM Заказы

WHERE Город = "Владикавказ"

А. Таблица, в которой выбраны все столбцы исходной таблицы **Заказы** и те строки в которых поле **Город** имеет значение **Владикавказ**.

В. Таблица из одного столбца

С. Таблица, в которой имеются только те строки, поле **Город** в которых имеет значение **Владикавказ**.

1.5 Что является результатом выполнения запроса

SELECT КодКлинта, КодЗаказа

FROM Заказы

WHERE Город = "Владикавказ"

AND Стоимость >= 1000

А. Таблица, в которой выбраны два столбца исходной таблицы **Заказы** и те строки в которых поле **Город** имеет значение **Владикавказ**.

В. Таблица из 2-х столбцов **КодКлинта** и **КодЗаказа**, в которой отражены заказы стоимостью не менее 1000, размещенные клиентами из Москвы.

С. Таблица, в которой имеются только те строки, поле **Город** в которых имеет значение **Владикавказ**.

Контрольные вопросы для самопроверки:

1. Что такое запрос?
2. Назовите элементы окна конструктора запросов?
3. Что такое бланк запроса?
4. Где записываются критерии условия выбора для запроса? Как удалить таблицу из запроса?
5. Как удалить поле из запроса?
6. Перечислите основные операторы, используемые в запросе.

2. Работа с базой данных

Несмотря на то, что в мире существует множество систем управления базами данных, все они имеют некоторые сходства. Поэтому для изучения мы выбрали СУБД Microsoft Access 2007, входящую 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 Visual Basic.

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 теоретический материал, пять практических работ, индивидуальные задания и перечень контрольных вопросов. Индивидуальные задания выполняются студентами самостоятельно и оформляются в виде отчета о практических работах.

Содержание отчета:

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

Практическая работа № 1

Знакомство с Access. Создание таблиц

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)*.

Таблица 2 – 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
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 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

Объекты базы данных Access


К объектам базы данных Access относятся:

1. *Таблицы* – предназначены для упорядоченного хранения данных.
2. *Запросы* – предназначены для поиска, извлечения данных и выполнения вычислений.
3. *Формы* – предназначены для удобного просмотра, изменения и добавления данных в таблицах.
4. *Отчеты* – используются для анализа и печати данных.
5. *Страницы доступа к данным* – предназначены для просмотра, ввода, обновления и анализа данных через сеть или из любого места компьютера.
6. *Макросы* – используются для выполнения часто встречающегося набора макрокоманд, осуществляющих обработку данных.
7. *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, в соответствии с таблицей 2.

Начало работы:**Создание базы данных**

1. Запустите Microsoft Access 2007.
2. Нажмите на кнопку  .
3. Задайте имя новой базы данных – «Записная книжка.accdb».
4. На вкладке ленты *Создание* в панели инструментов *Таблицы* нажмите на кнопку *Конструктор таблиц*.
5. Введите имена полей и укажите типы данных, к которым они относятся.

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

6. Выйдите из режима Конструктора, предварительно сохранив таблицу под именем «Друзья»; ключевые поля не задавайте.
7. Откройте таблицу «Друзья» двойным щелчком мыши и заполните в ней 10 строк.

8. Добавьте поля «Отчество» и «Дата рождения», для этого:
 - 1) установите курсор на поле, перед которым нужно вставить новый столбец;
 - 2) выполните команду: вкладка ленты *Режим таблицы* → панель инструментов *Поля и столбцы* → *Вставить*;
 - 3) щелкнув два раза на *Поле1*, переименуйте его в «Отчество», а *Поле2* – «Дата рождения».
9. Перейдите в режим Конструктора командой: вкладка ленты *Главная* → *Режимы* → *Конструктор*.
10. Для поля «Дата рождения» установите тип данных *Дата / время*; в свойствах поля выберите *Краткий формат даты*.
11. Отформатируйте таблицу следующим образом:
 - 1) цвет сетки – темно-красный;
 - 2) цвет фона – голубой;
 - 3) цвет текста – темно-красный, размер – 12 пт, начертание – курсив.
12. Переименуйте поле «Хобби» в «Увлечения».
13. Удалите запись под номером 8.
14. Измените размер ячеек так, чтобы были видны все данные. Для этого достаточно два раза щелкнуть левой кнопкой мыши на границе полей.
15. Расположите поля в следующем порядке: «№», «Фамилия», «Имя», «Отчество», «Телефон», «Дата рождения», «Увлечения», «Адрес», «Индекс», «Фото», «Эл_почта».
16. Заполните пустые ячейки таблицы.
17. В режиме Конструктора добавьте поле «Семейное положение», в котором будет содержаться фиксированный набор значений – замужем, не замужем, женат, не женат. Для создания раскрывающегося списка будем использовать *Мастер подстановок*:
 - 1) установите тип данных *Мастер подстановок*;
 - 2) в появившемся диалоговом окне выберите строку «Будет введен фиксированный набор значений» и нажмите кнопку *Далее*;
 - 3) число столбцов – 1;
 - 4) введите данные списка – замужем, не замужем, женат, не женат; 11) нажмите кнопку *Готово*.
18. С помощью раскрывающегося списка заполните новый столбец. 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».
19. Покажите работу преподавателю.

Контрольные вопросы :

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?

Практическая работа № 2

Создание связей между таблицами

1. Запустите Microsoft Access 2007.

2. Создадим базу данных «Фирма». Сотрудники данной организации работают с клиентами и выполняют их заказы.

Если все сведения поместить в одной таблице, то она станет очень неудобной для работы. В ней начнутся повторы данных. Всякий раз, когда сотрудник Иванов будет работать с какой-либо фирмой, придется прописывать данные о сотруднике и клиенте заново, в результате чего можно допустить множество ошибок. Чтобы уменьшить число ошибок, можно исходную таблицу разбить на несколько таблиц и установить связи между ними. Это будет более рационально, чем прежде.

Таким образом, необходимо создать 3 таблицы: *Сотрудники*, *Клиенты* и *Заказы*.

Сотрудники

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

Клиенты

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

Заказы

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

3. Отдельные таблицы, содержащие информацию по определенной теме, необходимо связать в единую структуру базы данных. Для связывания таблиц следует задать *ключевые поля*. Ключ состоит из одного или нескольких полей, значения которых *однозначно* определяют каждую запись в таблице. Наиболее подходящим в качестве ключевого поля является «Счетчик», так как значения в данном поле являются уникальными (т. е. исключают повторы).

4. Откройте таблицу *Сотрудники* в режиме Конструктора.

5. Нажмите правой кнопкой мыши на поле *Код сотрудника* и в появившемся контекстном меню выберите команду *Ключевое поле*. Если в таблице необходимо установить несколько ключевых полей, то выделить их можно, удерживая клавишу *Ctrl*.

6. Для таблицы *Клиенты* установите ключевое поле *Код клиента*, а для таблицы *Заказы* – *Код заказа*.

7. Таблица *Заказы* содержит поля *Код сотрудника* и *Код клиента*. При их заполнении могут возникнуть некоторые трудности, так как не всегда удастся запомнить все предприятия, с которыми работает фирма, и всех сотрудников с номером кода. Для удобства можно создать раскрывающиеся списки с помощью *Мастера подстановок*.

8. Откройте таблицу *Заказы* в режиме Конструктора.

9. Для поля *Код сотрудника* выберите тип данных *Мастер подстановок*.

10. В появившемся окне выберите команду «Объект «столбец подстановки» будет использовать значения из таблицы или запроса» и щелкните на кнопке *Далее*.

11. В списке таблиц выберите таблицу *Сотрудники* и щелкните на кнопке *Далее*.

12. В списке *Доступные поля* выберите поле *Код сотрудника* и щелкните на кнопке со стрелкой, чтобы ввести поле в список *Выбранные поля*. Таким же образом добавьте поля *Фамилия* и *Имя* и щелкните на кнопке *Далее*.

13. Выберите порядок сортировки списка по полю *Фамилия*.

14. В следующем диалоговом окне задайте необходимую ширину столбцов раскрывающегося списка.

15. Установите флажок *Скрыть ключевой столбец* и нажмите кнопку *Далее*.

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

17. Аналогичным образом создайте раскрывающийся список для поля *Код клиента*.


18. После создания ключевых полей можно приступить к созданию связей. Существует несколько типов отношений между таблицами:

1) при отношении «*один-к-одному*» каждой записи ключевого поля в первой таблице соответствует только одна запись в связанном поле другой таблицы, и наоборот. Отношения такого типа используются не очень часто. Иногда их можно использовать для разделения таблиц, содержащих много полей, для отделения части таблицы по соображениям безопасности;

2) при отношении «*один-ко-многим*» каждой записи в первой таблице соответствует несколько записей во второй, но запись во второй таблице не может иметь более одной связанной записи в первой таблице;

3) при отношении «*многие-ко-многим*» одной записи в первой таблице могут соответствовать несколько записей во второй таблице, а одной записи во второй таблице могут соответствовать несколько записей в первой.

19. Закройте все открытые таблицы, так как создавать или изменять связи между открытыми таблицами нельзя.

20. Выполните команду: вкладка ленты *Работа с базами данных* → кнопка  .

21. 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*.

22. 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 *Добавить таблицу*.

23. Установите связь между таблицами *Сотрудники* и *Заказы*, для этого выберите поле *Код сотрудника* в таблице *Сотрудники* и перенесите его на соответствующее поле в таблице *Заказы*.

24. После перетаскивания откроется диалоговое окно *Изменение связей* (рисунок 3), 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.

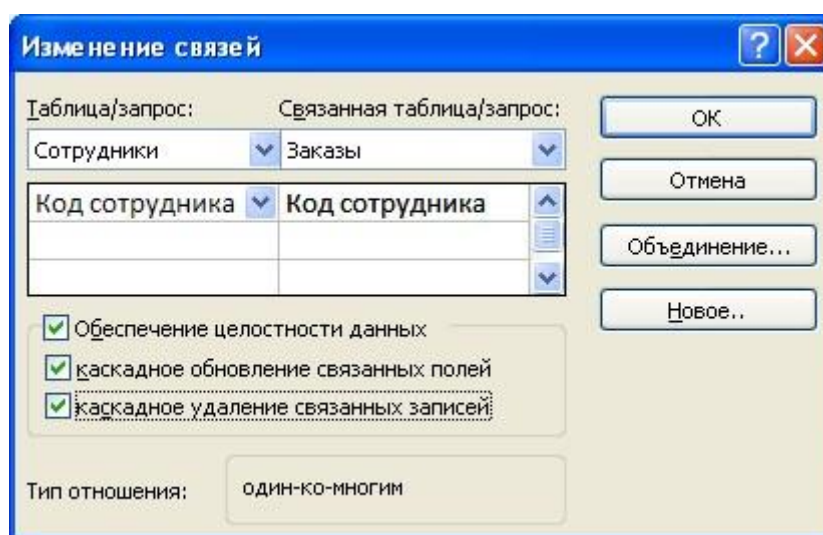


Рисунок 3 – Создание связи между таблицами

25. 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.

26. Параметры связи можно изменить, нажав на кнопку *Объединение*.

27. После установления всех необходимых параметров нажмите кнопку *OK*.

28. Связь между таблицами *Клиенты* и *Заказы* установите самостоятельно.
29. В результате должна получиться схема данных, представленная на рисунке 4.

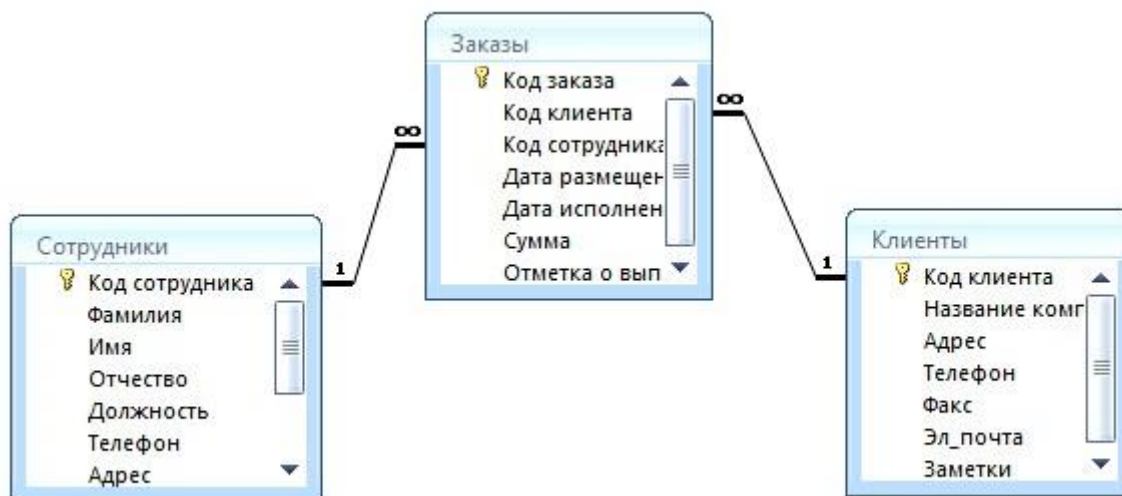


Рисунок 4 – Схема данных

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).

30. В таблицу *Сотрудники* внесите данные о семи работниках.
31. В таблицу *Клиенты* внесите данные о десяти предприятиях, с которыми работает данная фирма.
32. В таблице *Заказы* оформите несколько заявок, поступивших на фирму.
33. Покажите работу преподавателю.

Контрольные вопросы:

1. С помощью чего можно создавать таблицы?
2. Что такое ключевое поле?
3. Как установить несколько ключевых полей?
4. Как установить связи между таблицами?
5. Какие существуют отношения между таблицами?
6. Что означают на схеме данных «1» и «∞»?
7. Зачем нужен *Мастер подстановок*?

Практическая работа № 3

Отбор данных с помощью запросов

Запросы являются основным средством просмотра, отбора, изменения и анализа информации, которая содержится в одной или нескольких таблицах базы данных.

Существуют различные виды запросов, но наиболее распространенными являются запросы на выборку, с них и начнем наше знакомство.

1. Откройте базу данных «Фирма», созданную ранее.
2. Выполните команду: вкладка ленты *Создание* → *Мастер запросов* → *Простой запрос*.
3. В появившемся диалоговом окне (рисунок 5) укажите таблицу *Сотрудники* и выберите поля *Фамилия*, *Имя*, *Телефон*. Нажмите кнопку *Далее*.

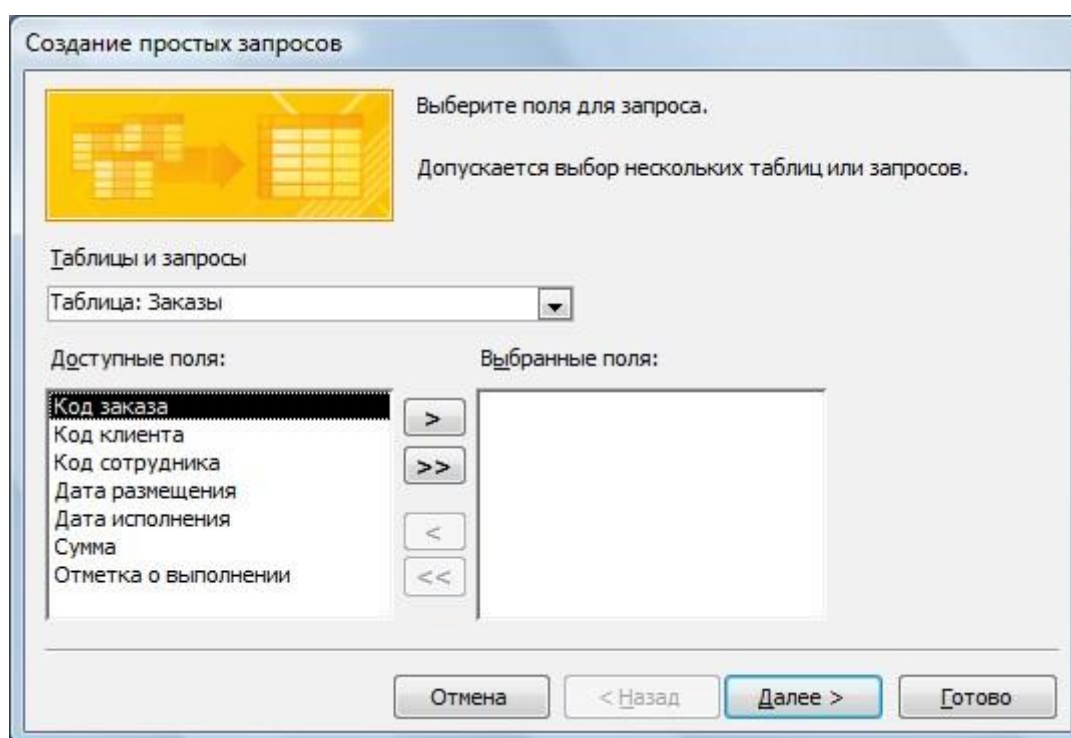


Рисунок 5 – Создание простого запроса

4. Введите имя запроса – *Телефоны* – и нажмите кнопку *Готово*. Перед вами появится запрос, в котором можно просмотреть телефоны сотрудников.
5. Следующий запрос попробуйте создать с помощью Конструктора, для этого выполните команду: вкладка ленты *Создание* → *Конструктор запросов*.
6. В диалоговом окне *Добавление таблиц* выберите таблицу *Клиенты* и щелкните на кнопке *Добавить*, а затем – на кнопке *Заккрыть*.
7. Чтобы перенести нужные поля в бланк запроса, необходимо по ним дважды щелкнуть левой кнопкой мыши (рисунок 6).

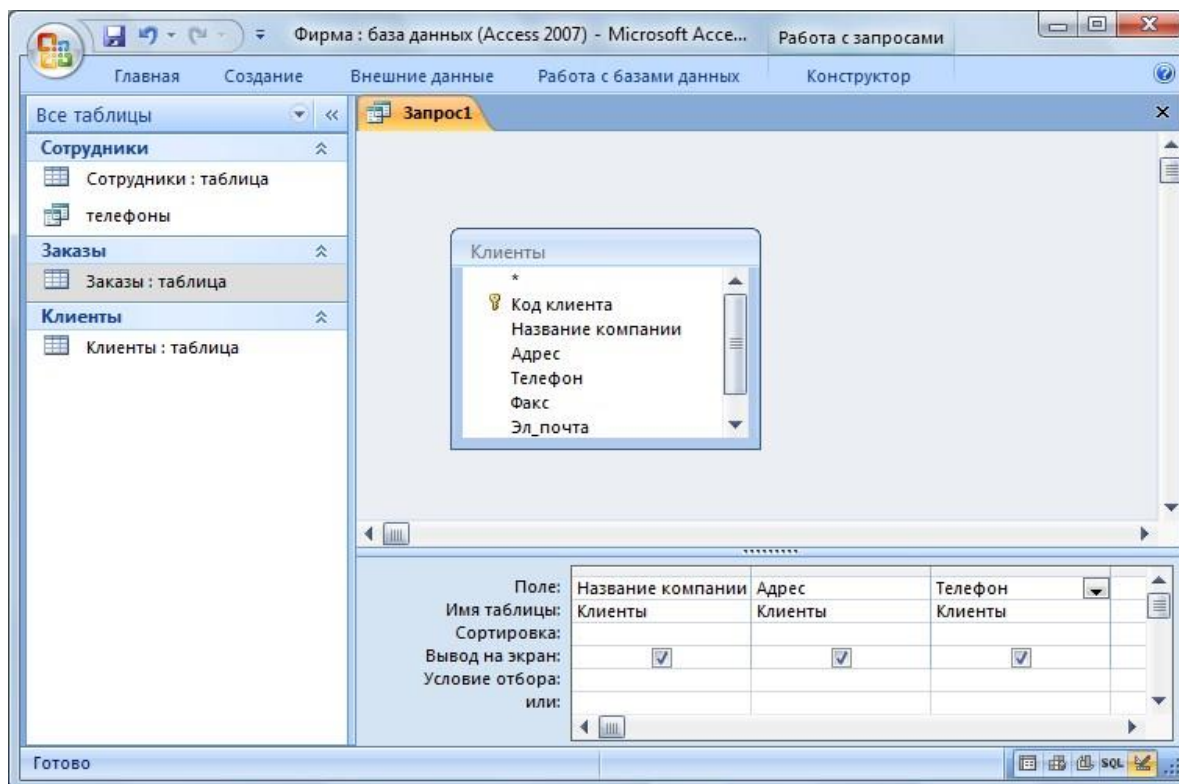


Рисунок 6 – Создание запроса в режиме Конструктора

8. Чтобы отсортировать записи в поле *Название компании* в алфавитном порядке, необходимо в раскрывающемся списке строки *Сортировка* выбрать пункт *по возрастанию*.
9. Сохраните запрос с именем «*Адреса клиентов*».
10. Самостоятельно создайте запрос «*Дни рождения*», в котором можно будет просмотреть дни рождения сотрудников.
11. Допустим, мы хотим узнать, у кого из сотрудников день рождения в текущем месяце, например в апреле. Для этого откройте запрос в режиме Конструктора.
12. 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 7.

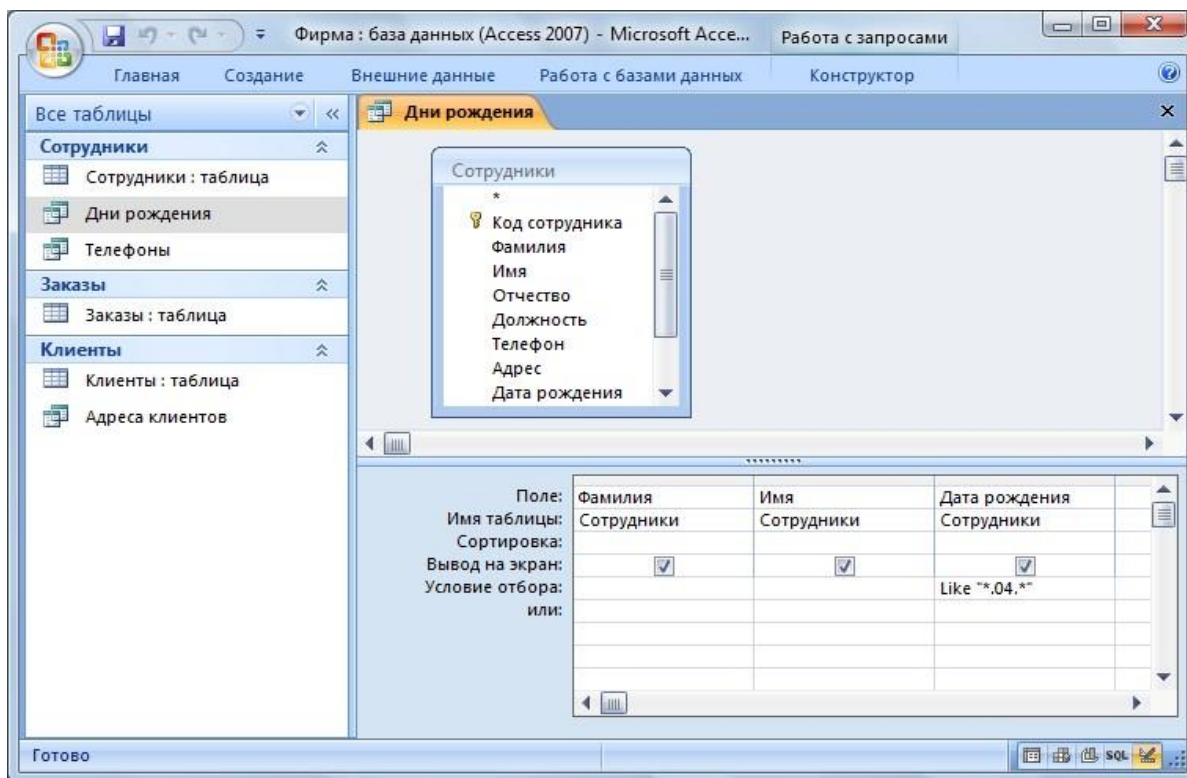


Рисунок 7 – Создание запроса

13. Закройте Конструктор и просмотрите полученный результат. 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.

14. Если нам нужно узнать, кто из сотрудников родился в мае, то придется создать новый запрос или изменить условие в существующем запросе *Дни рождения*. Данная процедура является неудобной и занимает много времени. Если приходится часто выполнять запрос, но каждый раз с новыми значениями условий используют *запрос с параметром*. При запуске такого запроса на экран выводится диалоговое окно для ввода значения в качестве условия отбора. Чтобы создать запрос с параметром, пользователю необходимо ввести текст сообщения в строке *Условие отбора* бланка запроса (рисунок 8).

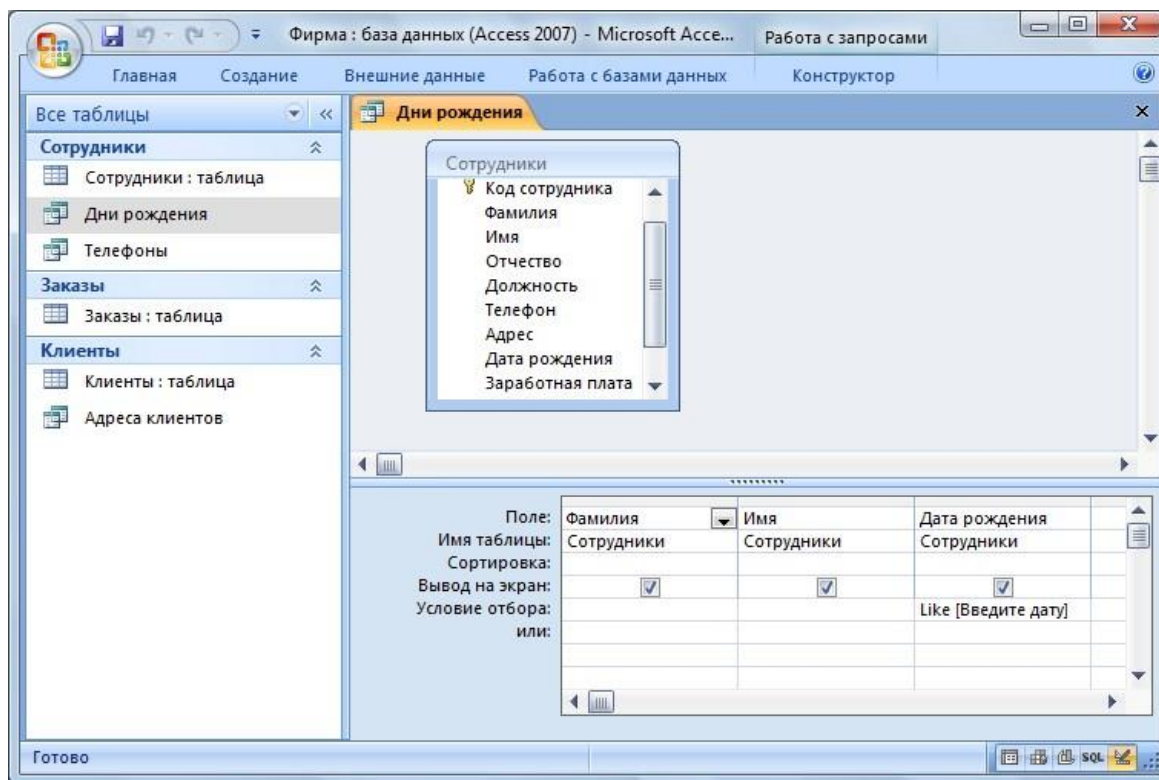


Рисунок 8 – Создание запроса с параметром

15. Запись *Like[Введите дату]* означает, что при открытии запроса появится диалоговое окно (рисунок 9) с текстом «Введите дату» и полем для ввода условия отбора. Если ввести условие **.04.**, то в запросе появится список сотрудников, родившихся в апреле. Запустите запрос еще раз и введите значение **.05.**, посмотрите, как изменился запрос.

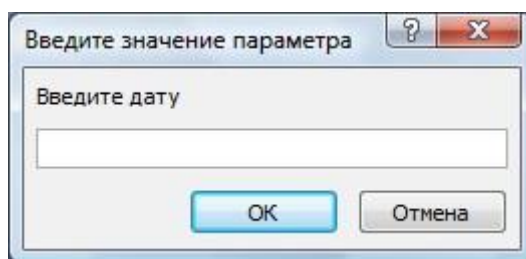


Рисунок 9 – Окно для ввода условия отбора

16. Измените запрос «Телефоны» так, чтобы при его запуске выводилось диалоговое окно с сообщением «Введите фамилию». Поскольку в запросе нужно вывести конкретную фамилию, в условии отбора слово *Like* писать не надо.

17. Измените запрос «Телефоны» так, чтобы при его запуске запрашивались не только фамилия, но и имя сотрудника.

18. Самостоятельно 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.

19. В условии отбора для логического поля *Отметка о выполнении* введите *Да*, чтобы в запросе отображались только выполненные заказы.

20. Сделайте так, чтобы столбец *Отметка о выполнении* не выводился на экран.

21. Создайте запрос *Сумма заказа*, в котором будут отображаться заказы на сумму более 50 000 руб.

22. Измените запрос, чтобы сумма заказа была от 20 000 до 50 000 руб. Для данных запросов в условии отбора можно использовать операторы сравнения >, <, =, >=, <=, <> и логические операторы *And*, *Or*, *Not* и др.

23. Иногда в запросах требуется произвести некоторые вычисления, например посчитать подоходный налог 13 % для каждой сделки. Для этого откройте запрос *Сумма заказа* в режиме Конструктора.

24. 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* (рис. 8), 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.

25. 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.

26. Щелкните на кнопке * и введите 0,13, в соответствии с рисунком 10.. Таким образом, мы посчитаем подоходный налог 13 %.

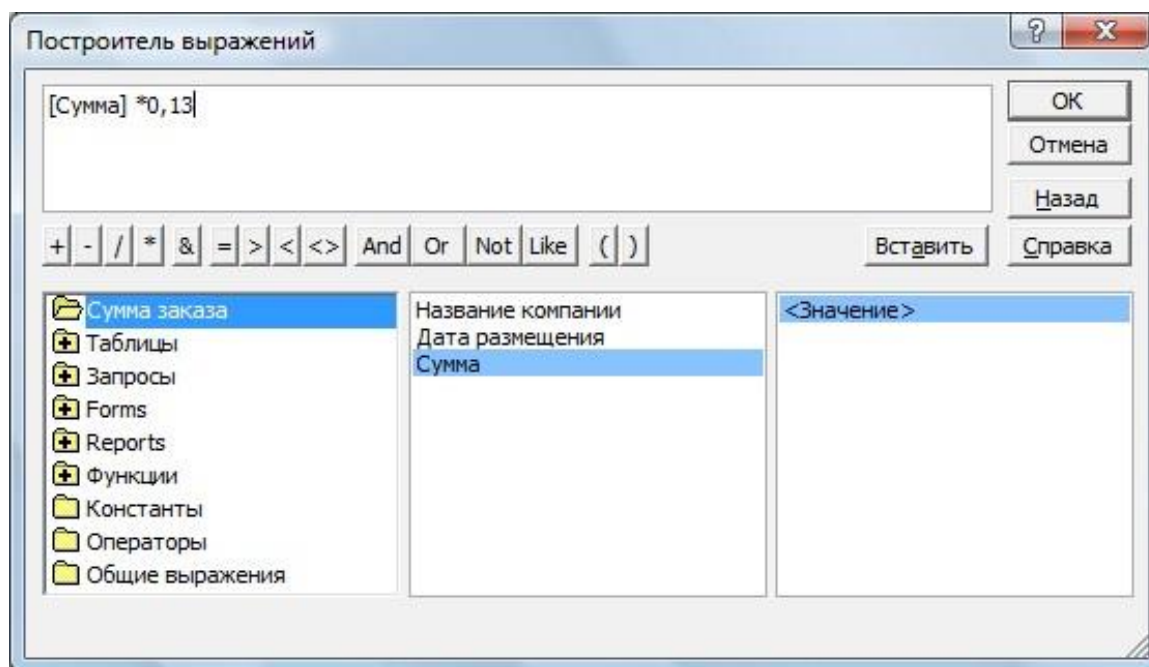


Рисунок 10 – Построитель выражений

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

28. Замените *Выражение1* на *Налог* и закройте Конструктор.

29. Откройте запрос и посмотрите, что у вас получилось.

30. Используя *Построитель выражений*, добавьте в запрос *Сумма заказа* поле *Прибыль*, в котором будет вычисляться доход от заказа (т. е. сумма минус налог).

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

32. Покажите работу преподавателю.

Контрольные вопросы:

1. Для чего предназначены запросы?
2. Какие виды запросов вы знаете?
3. С помощью чего можно создавать запросы?
4. Для чего используют запрос с параметром?
5. Как можно сделать вычисления в запросах?
6. Что означает запись в условии запроса «=50»?
7. Можно ли создавать запросы на основе нескольких таблиц?

Практическая работа № 4

Использование форм в базе данных

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.

Можно вносить данные в таблицы и без помощи каких-либо форм. Но существует несколько причин, которые делают формы незаменимым средством ввода данных в базу:

- 1) при работе с формами ограничен доступ к таблицам (самому ценному в базе данных);
- 2) разные люди могут иметь разные права доступа к информации, хранящейся в базе. Для ввода данных им предоставляются разные формы, хотя данные из форм могут поступать в одну таблицу;
- 3) вводить данные в форму легче, чем в таблицу, и удобнее, так как в окне формы отображается, как правило, одна запись таблицы;
- 4) в большинстве случаев информация для баз данных берется из бумажных бланков (анкет, счетов, накладных, справок и т. д.). Экранные формы можно сделать точной копией бумажных бланков, благодаря этому уменьшается количество ошибок при вводе и снижается утомляемость персонала.

Создавать формы можно на основе нескольких таблиц или запросов с помощью Мастера, используя средство автоформы, «вручную» в режиме Конструктора, сохраняя таблицу или запрос как форму. Созданную любым способом форму можно затем изменять в режиме Конструктора. Рассмотрим некоторые из перечисленных способов.

1. Выполните команду: вкладка ленты Создание → панель инструментов *Формы* → *Другие формы* → *Мастер форм*.
2. В диалоговом окне *Создание форм* выберите 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*.
4. С помощью Мастера создайте формы *Сотрудники*, *Клиенты*, *Заказы*, *Менеджеры*.
5. Откройте форму *Сотрудники* в режиме Конструктора. Этот режим предназначен для создания и редактирования форм.
6. Разместите элементы в удобном для вас порядке, измените размер и цвет текста.
7. В заголовок формы добавьте текст *Сотрудники фирмы*.
8. В примечание формы добавьте объект *Кнопка* (вкладка ленты *Конструктор* → панель инструментов *Элементы управления*).
9. После того как вы «нарисуете» кнопку указателем, на экране появится диалоговое окно *Создание кнопок*, в соответствии с рисунком 11.

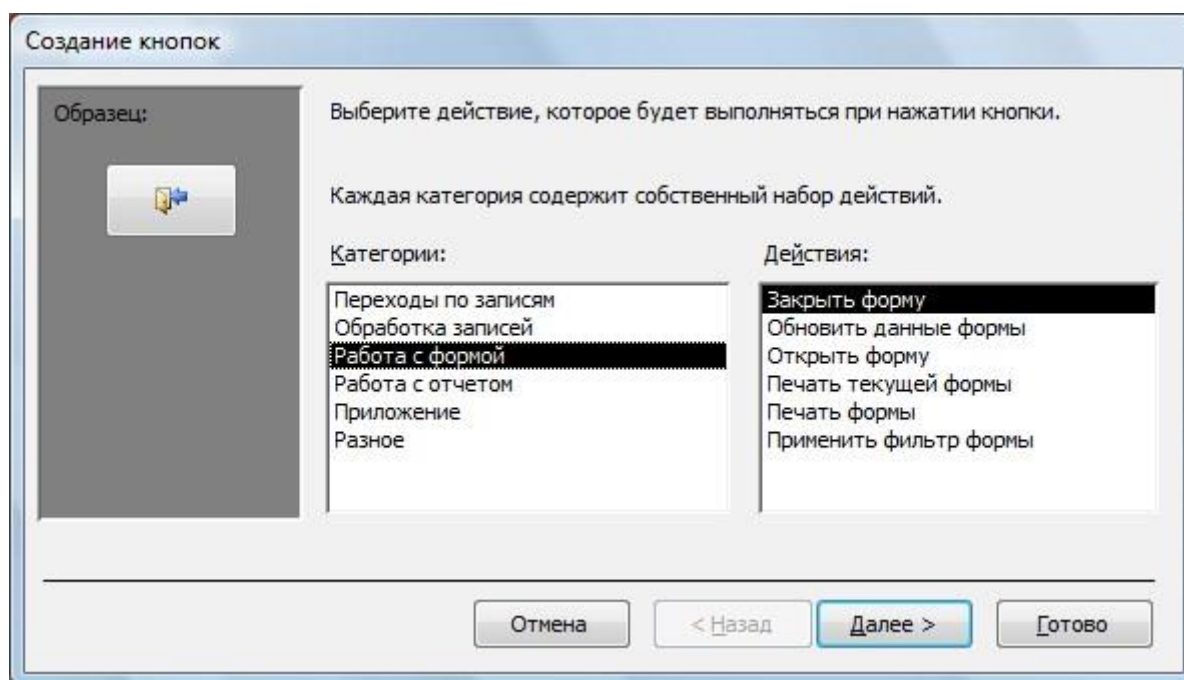


Рисунок 11 – Создание кнопок на форме

10. В категории *Работа с формой* выберите действие *Закрывать форму* и нажмите кнопку *Далее*.
11. Выберите рисунок или текст, который будет размещаться на кнопке.
12. В последнем диалоговом окне *Мастера кнопок* задайте имя кнопки и нажмите *Готово*.
13. *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.
14. Самостоятельно создайте кнопки *Выход из приложения*, *Поиск записи*, *Удаление записи*.
15. Иногда на форме требуется разместить несколько страниц, содержащих данные из различных источников, справочную или вспомогательную информацию. Для этой цели можно использовать набор вкладок.
16. Создайте пустую форму.
17. 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*.
18. Добавьте еще одну вкладку.
19. Переименуйте ярлычки вкладок так, чтобы на них отображались названия данных, которые будут в них располагаться: *Сотрудники*, *Менеджеры*, *Помощь*.
20. Перейдите на вкладку *Сотрудники* и перетащите на нее мышкой из базы данных форму *Сотрудники*.
21. Аналогичным образом поместите форму *Менеджеры* на вкладку *Менеджеры*.
22. На вкладку *Помощь* поместите некоторые советы по работе с базой данных.
23. Данную форму сохраните с именем *Сотрудники фирмы*.
24. В *Microsoft Access* можно создавать кнопочные формы. Они содержат только кнопки и предназначены для выбора основных действий в базе данных. Для создания кнопочной формы необходимо на вкладке ленты *Работа с базами данных* выбрать команду *Диспетчер кнопочных форм*.

25. Если кнопочной формы в базе данных нет, то будет выведен запрос на подтверждение ее создания. Нажмите *Да* в диалоговом окне подтверждения.

26. Перед вами появится *Диспетчер кнопочных форм*, в котором щелкните по кнопке *Создать*.

27. В диалоговом окне *Создание*, в соответствии с рисунком 12, введите имя новой кнопочной формы и нажмите *ОК*.

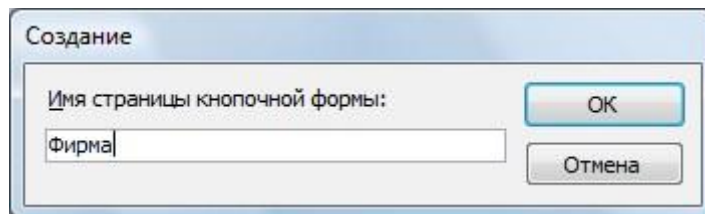


Рисунок 12 – Задание имени кнопочной формы

28. Имя новой кнопочной формы добавится в список *Страницы кнопочной формы* окна *Диспетчер кнопочных форм* в соответствии с рисунком 13. Выделите имя новой кнопочной формы и щелкните по кнопке *Изменить*.

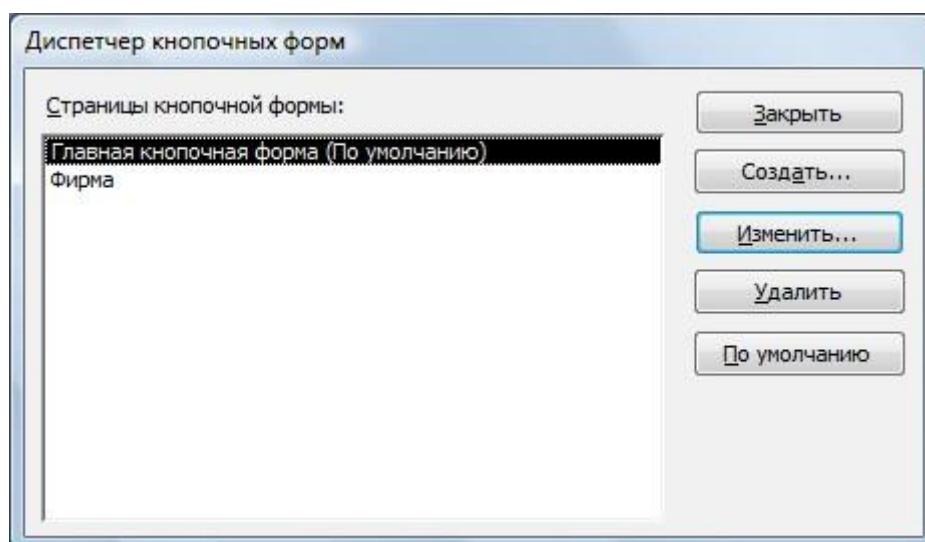


Рисунок 13 – Диспетчер кнопочных форм

29. В диалоговом окне *Изменение страницы кнопочной формы* щелкните по кнопке *Создать*. Появится диалоговое окно *Изменение элемента кнопочной формы*, в соответствии с рисунком 14.

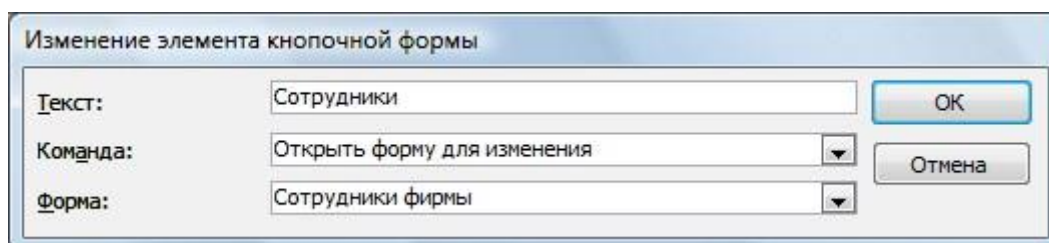


Рисунок 14 – Создание кнопок на форме

30. 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.

31. Аналогичным образом добавьте кнопки *Клиенты*, *Заказы*, *Выход*.

32. 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)».

33. Чтобы закончить создание кнопочной формы, щелкните по кнопке *Закреть*.

34. В результате должна получиться форма, представленная на рисунке 15.

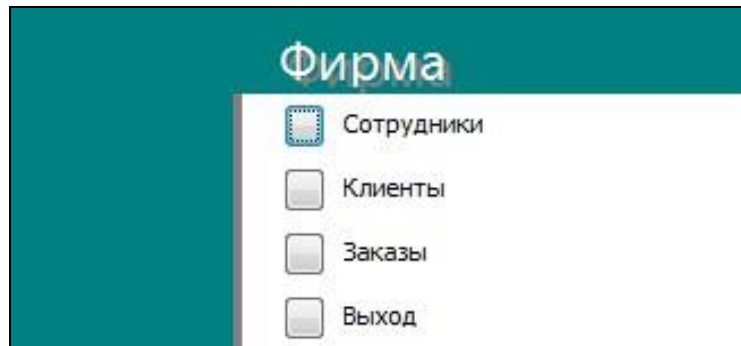


Рисунок 15 – Главная кнопочная форма

35. Добавьте в форму какой-нибудь рисунок.

36. Для того чтобы главная кнопочная форма появлялась на экране при запуске приложения, необходимо в главном меню нажать на кнопку *Параметры Access*, в соответствии с рисунком 16. Для текущей базы данных установите форму просмотра – «кнопочная форма».

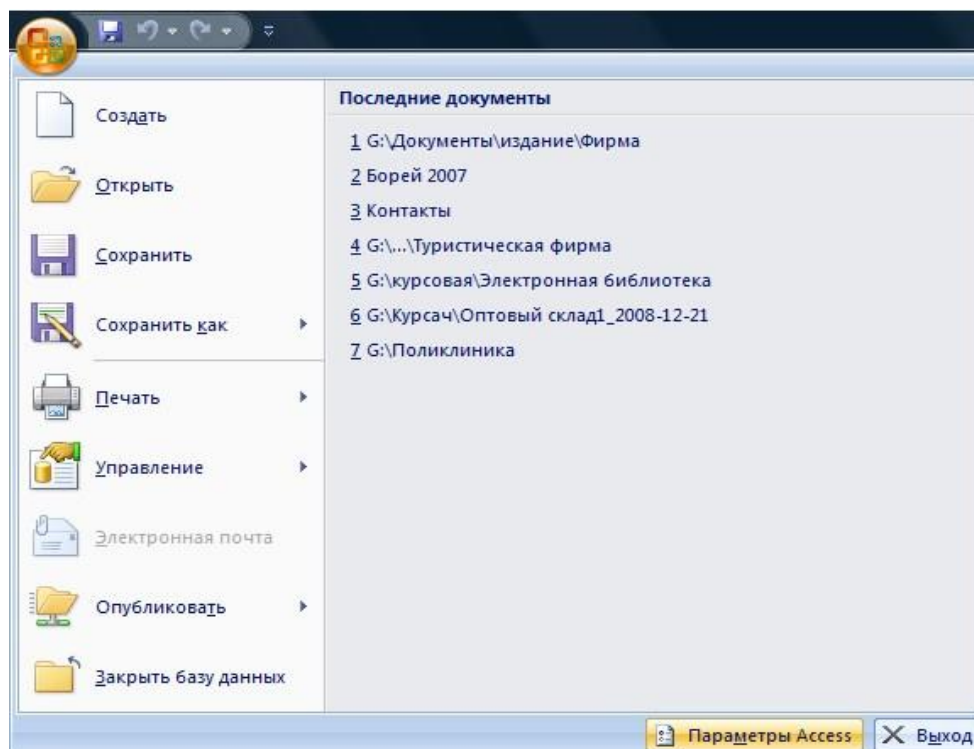


Рисунок 16 – Задание параметров Access

37. Покажите работу преподавателю.

Контрольные вопросы:

1. Для чего предназначены формы?
2. Почему форма является незаменимым средством в БД?
3. С помощью чего можно создавать формы?
4. На основе чего можно создавать формы?
5. Как создать кнопку на форме?
6. Как можно разместить несколько таблиц и запросов на одной форме?
7. Как создать главную кнопочную форму?

Практическая работа № 5

Создание отчетов

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 базе данных Access можно несколькими способами:

- 1) с помощью Мастера отчетов;
- 2) на основе таблиц или запросов;
- 3) в режиме Конструктора.

1. В окне базы данных выполните команду: вкладка ленты *Создание* → панель инструментов *Отчеты* → *Мастер отчетов*.

2. Выберите из списка таблиц (или запрос), которая будет использована как источник данных (например, запрос *Адреса клиентов*).

3. В появившемся диалоговом окне *Создание отчетов*, в соответствии с рисунком 17, переместите все доступные поля в область «выбранные поля».

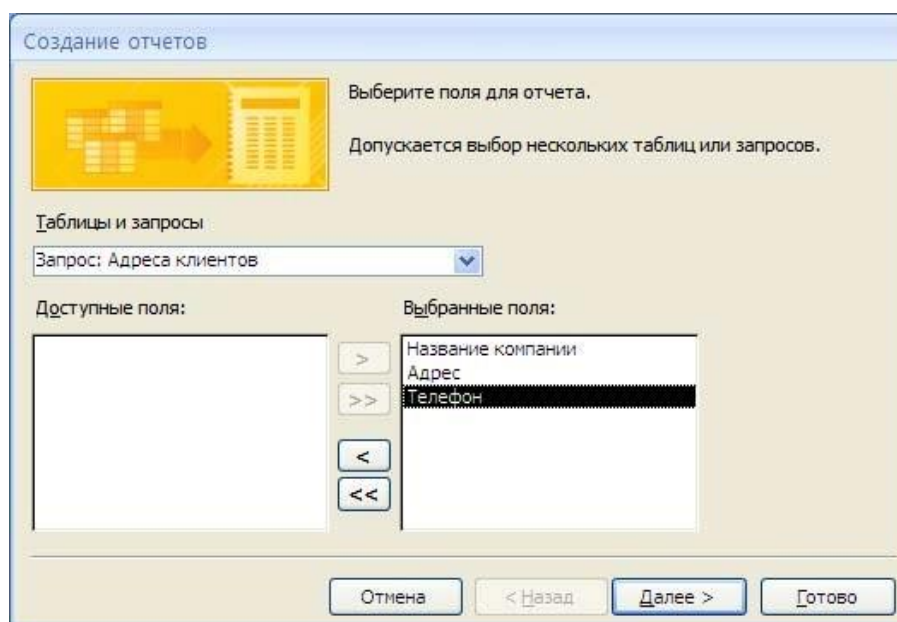


Рисунок 17 – Мастер отчетов

4. С помощью *Мастера отчетов* создайте отчет *Дни рождения*. В качестве источника данных используйте таблицу *Сотрудники*.

5. Если требуется напечатать почтовые наклейки, Access предоставляет такую возможность. Для этого выделите таблицу *Клиенты* и выполните команду: вкладка ленты *Создание* → панель инструментов *Отчеты* → *Наклейки*.

6. В появившемся диалоговом окне, в соответствии с рисунком 18, укажите размер наклейки, систему единиц, тип наклейки и нажмите кнопку *Далее*.

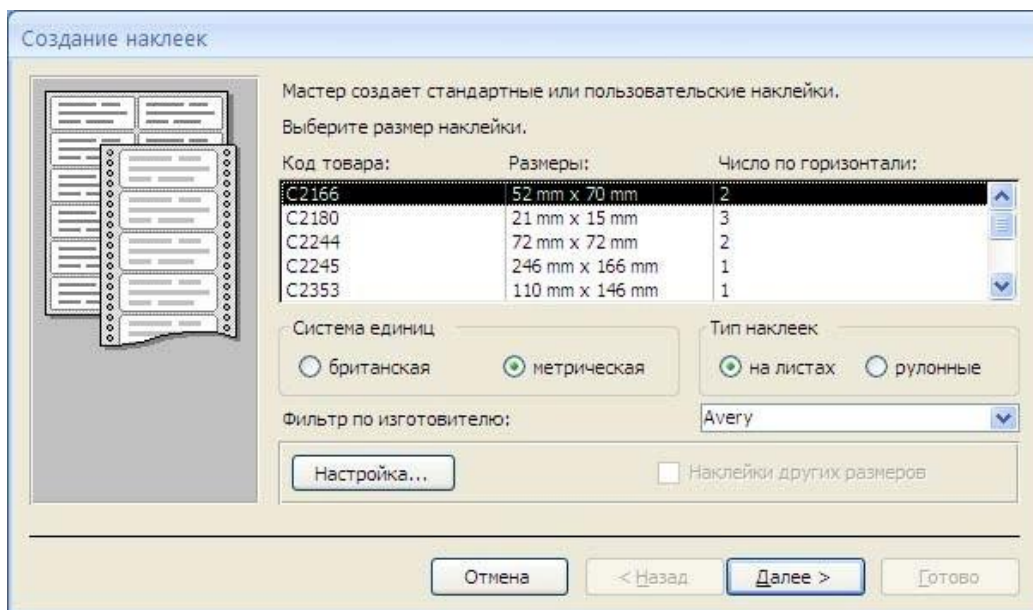


Рисунок 18 – Диалоговое окно Создание наклеек

7. На следующем шаге создания отчета установите шрифт, размер, цвет текста и начертание. Нажмите кнопку *Далее*.

8. 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.

9. При необходимости измените название отчета с наклейками и нажмите кнопку *Готово*.

10. Иногда в отчетах требуется вычислять итоговые значения, среднее, минимальное или максимальное значения, а также проценты. Для этого запустите *Мастер отчетов* и в качестве источника данных укажите запрос *Сумма заказа*.

11. В диалоговом окне *Мастера*, в котором задается порядок сортировки записей, нажмите кнопку *Итоги*, в соответствии с рисунком 19.

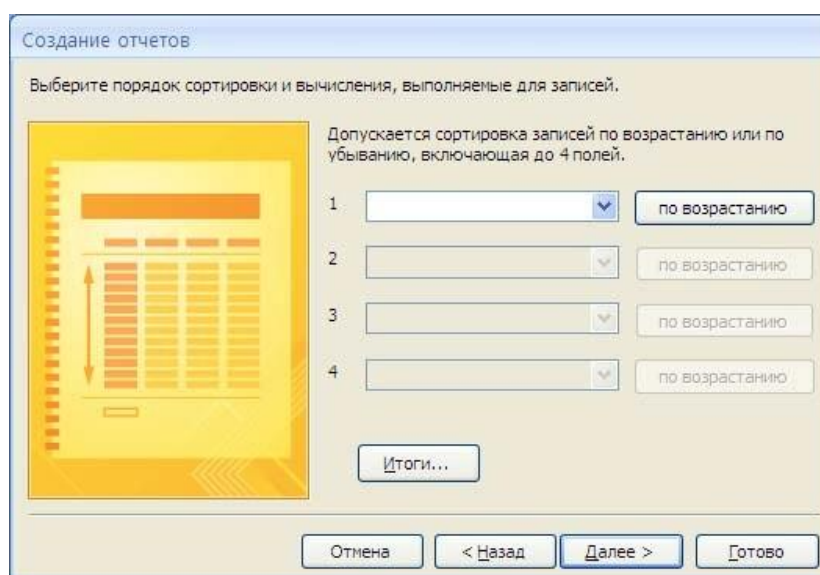


Рисунок 19 – Вычисление итоговых значений в отчетах

12. В диалоговом окне *Итоги* (рисунок 20 для полей *Сумма* и *Налог* установите флажки в столбце *sum*, чтобы посчитать итоговую сумму.

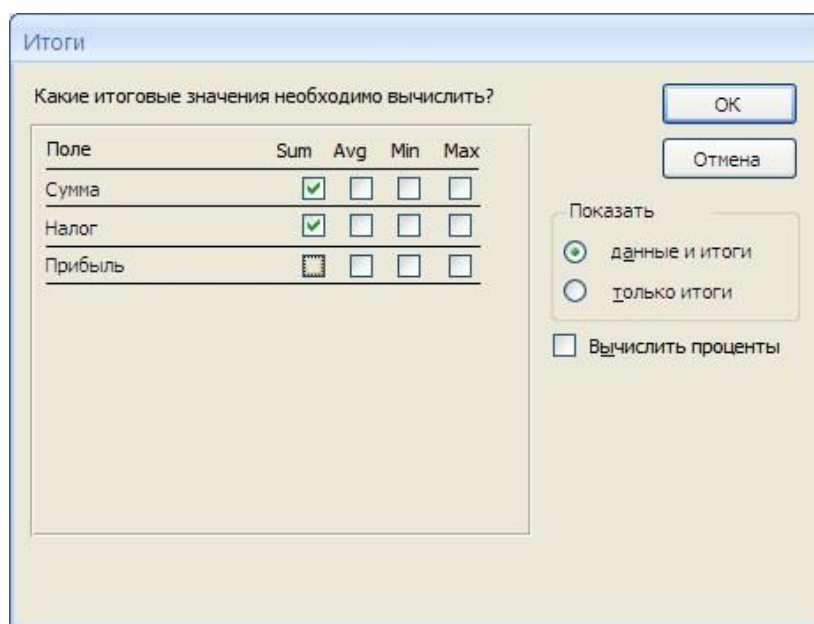


Рисунок 20 – Вычисление итоговых значений суммы

13. Далее выполните все шаги Мастера и нажмите кнопку *Готово*.
14. Создайте отчет *Дни рождения*, используя в качестве источника данных запрос *Дни рождения*.
15. 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.
16. Покажите работу преподавателю.

Контрольные вопросы :

1. Для чего предназначены отчеты?
2. Какие способы создания отчетов вы знаете?
3. Как в отчетах можно посчитать итоговые значения?
4. Какие итоговые значения можно посчитать в отчетах?
5. Как в Access напечатать почтовые наклейки?

Задания для самостоятельного выполнения

Вариант 1

1. Разработайте базу данных «*Электронная библиотека*», состоящую из трех таблиц со следующей структурой:

Книги – шифр книги (ключевое поле), автор, название, год издания, количество экземпляров.

Читатели – читательский билет (ключевое поле), фамилия, имя, отчество, адрес.

Выданные книги – шифр книги, читательский билет, дата выдачи, дата возвращения, дата фактического возвращения.

2. Установите связи между таблицами.

3. С помощью запроса отберите все книги, выпущенные с 1990 по 2007 годы.

4. Создайте запрос с параметром для отбора книг определенного автора.

5. Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 2

1. Разработайте базу данных «*Продуктовый магазин*», которая состоит из четырех таблиц со следующей структурой:

Товары – код товара (ключевое поле), наименование товара, количество товара.

Поступление товаров – код товара, дата поступления, цена приобретения товара за единицу, код поставщика.

Продажа товаров – код товара, месяц продажи, проданное количество за месяц, цена продажи товара.

Поставщики – код поставщика (ключевое поле), название поставщика, адрес поставщика, телефон поставщика.

2. Установите связи между таблицами.

3. С помощью запроса отберите товары, цены которых от 100 до 450 руб.

4. Создайте запрос с параметром для отбора товаров, проданных в определенном месяце.

5. Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 3

1. Разработайте базу данных «*Сессия*», состоящую из четырех таблиц со следующей структурой:

Студенты – шифр студента (ключевое поле), фамилия, имя, отчество, курс, группа.

Экзамены – шифр студента, дата, шифр дисциплины, оценка.

Зачеты – шифр студента, дата, шифр дисциплины, зачет.

Дисциплины – шифр дисциплины (ключевое поле), название дисциплины, количество часов.

2. Установите связи между таблицами.

3. С помощью запроса отберите студентов, сдавших экзамен на 4 или 5.

4. Создайте запрос с параметром для отбора студентов, получивших или не получивших зачет.

5. Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 4

1. Разработайте базу данных «*Оптовый склад*», состоящую из четырех таблиц со следующей структурой:

Склад – код товара, количество, дата поступления.

Товары – код товара (ключевое поле), название товара, срок хранения.

Заявки – код заявки (ключевое поле), название организации, код товара, требуемое количество.

Отпуск товаров – код заявки (ключевое поле), код товара, отпущенное количество, дата отпуска товара.

2. Установите связи между таблицами.

3. С помощью запроса отберите товары, количество которых от 50 до 200 штук.

4. Создайте запрос с параметром для отбора товаров, поступивших на склад какого-либо числа.

5. Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 5

1. Разработайте базу данных «*Абитуриенты*», состоящую из четырех таблиц со следующей структурой:

Анкета – номер абитуриента (ключевое поле), фамилия, имя, отчество, дата рождения, оконченное среднее учебное заведение (название, номер, населенный пункт), дата окончания учебного заведения, наличие красного диплома или золотой / серебряной медали, адрес, телефон, шифр специальности.

Специальности – шифр специальности (ключевое поле), название специальности.

Дисциплины – шифр дисциплины (ключевое поле), название дисциплины.

Вступительные экзамены – номер абитуриента, шифр дисциплины, экзаменационная оценка.

2. Установите связи между таблицами.

3. Составьте запрос для отбора студентов, сдавших экзамены без троек.

4. Создайте запрос с параметром для отбора студентов, поступающих на определенную специальность.

5. Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 6

1. Разработайте базу данных «*Транспортные перевозки*», состоящую из трех таблиц со следующей структурой:

Транспорт – марка автомобиля, государственный номер (ключевое поле), расход топлива.

Заявки – код заявки (ключевое поле), дата заявки, название груза, количество груза, пункт отправления, пункт назначения.

Доставка – № п/п, дата и время отправления, дата и время прибытия, код заявки, государственный номер автомобиля, пройденное расстояние.

2. Установите связи между таблицами.

3. С помощью запроса отберите заявки с количеством груза от 100 до 500 кг.

4. Создайте запрос с параметром для отбора транспорта по марке автомобиля.

5. Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант № 7

1. Разработайте базу данных «Прокат спортивного оборудования», состоящую из трех таблиц со следующей структурой:

Клиенты – код клиента (ключевое поле), фамилия, имя, отчество, телефон, адрес, паспортные данные, залог.

Склад – код оборудования (ключевое поле), название, количество, залоговая стоимость, остаток.

Прокат – № п/п, клиент, оборудование, дата выдачи, срок возврата, отметка о возврате, оплата проката.

2. Установите связи между таблицами.

3. Создайте запрос для отбора оборудования с залоговой стоимостью от 10000 до 50000 руб.

4. Создайте запрос с параметром для отбора клиентов, возвративших оборудование.

5. Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 8

1. Разработайте базу данных «Банк», состоящую из трех таблиц со следующей структурой:

Клиенты – код клиента (ключевое поле), фамилия, имя, отчество, паспорт, телефон, адрес, заработная плата.

Виды кредитов – код кредита (ключевое поле), название кредита, процентная ставка, условия предоставления.

Предоставленные кредиты – № п/п, клиент, кредит, дата предоставления, срок, дата возврата, сумма, отметка о возврате.

2. Установите связи между таблицами.

3. Создайте запрос для отбора клиентов, взявших кредит от 500 000 до 1 000 000 руб.

4. Создайте запрос с параметром для отбора кредитов по процентной ставке.

5. Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 9

1. Разработайте базу данных «Туристическая фирма», состоящую из четырех таблиц со следующей структурой:

Клиенты – код клиента (ключевое поле), фамилия, имя, отчество, телефон, адрес, паспорт.

Сотрудники – код сотрудника (ключевое поле), фамилия, имя, отчество, должность, телефон, адрес, паспортные данные.

Туристические маршруты – код маршрута (ключевое поле), название, описание маршрута, страна, стоимость путевки, количество дней, вид транспорта.

Заказы – код заказа (ключевое поле), клиент, маршрут, сотрудник (менеджер, оформивший заказ), дата, отметка об оплате.

2. Установите связи между таблицами.

3. Создайте запрос для отбора маршрутов со стоимостью от 10000 до 20000 руб.

4. Создайте запрос с параметром для отбора клиентов, выбравших определенный вид маршрута.

5. Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Вариант 10

1. Разработайте базу данных «Поликлиника», состоящую из четырех таблиц со следующей структурой:

Врачи – код врача (ключевое поле), ФИО, должность, специализация, стаж работы, адрес, телефон.

Болезни– № п/п (ключевое поле), название заболевания, рекомендации по лечению, меры профилактики.

Пациенты – код пациента (ключевое поле), ФИО, адрес, телефон, страховой полис, паспорт.

Диагноз – № п/п (ключевое поле), пациент, заболевание, лечащий врач, дата обращения, дата выздоровления.

2. Установите связи между таблицами.

3. С помощью запроса отберите врачей-стоматологов и ортопедов.

4. Создайте запрос с параметром для отбора пациентов с определенным видом заболевания.

5. Создайте формы для ввода данных, отчеты и главную кнопочную форму.

Контрольные вопросы для самопроверки

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. Для чего предназначены запросы?
19. Какие виды запросов вы знаете?
20. С помощью чего можно создавать запросы?
21. Для чего используют запрос с параметром?
22. Как можно сделать вычисления в запросах?
23. Можно ли создавать запросы на основе нескольких таблиц?
24. Для чего предназначены отчеты?
25. Какие способы создания отчетов вы знаете?
26. Как в отчетах можно посчитать итоговые значения?
27. Какие итоговые значения можно посчитать в отчетах?
28. Как в Access напечатать почтовые наклейки?

Библиографический список

1. Практикум по Microsoft Office 2007 (Word, Excel, Access), PhotoShop: Учебно-методическое пособие / **Кравченко Л.В.**, - 2-е изд., испр. и доп - М.:Форум, НИЦ ИНФРА-М, 2015. - 168 с.: 70x100 1/16. -(Профессиональное обр.) (Обложка) ISBN 978-5-91134-656-0 - Режим доступа: <http://znanium.com/catalog/product/478844> .
2. **Грошев, А.С.** Основы работы с базами данных : курс лекций / Грошев А.С. — Владикавказ : Интуит НОУ, 2016. — 256 с. — URL: <https://book.ru/book/917933> (дата обращения: 17.09.2019). — Текст : электронный/
3. **Мясникова, Н.А.** Алгоритмы и структуры данных : учебное пособие / Мясникова Н.А. — Владикавказ : КноРус, 2018. — 185 с. — (для бакалавров). — ISBN 978-5-406-04894-8. — URL: <https://book.ru/book/926560> (дата обращения: 17.09.2019). — Текст : электронный.
4. **Осипов, Д.Л.** Системы управления базами данных : практикум / Осипов Д.Л., Огур М.Г., сост. — Ставрополь : Северо-Кавказский федеральный университет, 2017. — 148 с. — URL: <https://book.ru/book/929919> (дата обращения: 17.09.2019). — Текст : электронный.
5. **Сирант, О.В.** Работа с базами данных : курс лекций / Сирант О.В., Коваленко Т.А. — Владикавказ : Интуит НОУ, 2016. — 150 с. — ISBN 978-5-9556-0136-6. — URL: <https://book.ru/book/918170> (дата обращения: 17.09.2019). — Текст : электронный.
6. **Дубнов, П. Ю.** Access 2000. Проектирование баз данных / П.Ю. Дубнов. – М. : ДМК, 2000. – 272 с.
7. **Каратыгин, С. А.** Access 2000. Руководство пользователя с примерами / С.А. Каратыгин. – М. : Лаборатория базовых знаний, 2000. – 376 с.
8. **Кошелев, В.Е.** Access 2003. Практическое руководство / В.Е. Кошелев. – М. : ООО «Бином-Пресс», 2005. – 464 с.
9. **Робинсон, С.** Microsoft Access 2000 : учебный курс / С. Робинсон. – СПб. : Питер, 2001. – 512 с.
10. **Советов, Б.Я.** Базы данных: теория и практика [Электронный ресурс]: учебник для бакалавров / Б.Я.Советов, В.В. Цехановский, В.Д.Чертовской. - 2-е изд. - М. : Юрайт, 2017. - 463с. – Режим доступа: <http://biblio-online.ru>.
11. **Кумскова, И.А.** Базы данных [Электронный ресурс]: учебник / И.А. Кумскова.- М.: КноРус, 2018.- 488с. – Режим доступа: <http://www.book.ru>.
12. **Агальцов, В.П.** Базы данных. В 2-х кн. Книга 2. Распределенные и удаленные базы данных [Электронный ресурс]: учебник / В.П. Агальцов. — М. : ИНФРА-М, 2017. — 271 с.: ил. – Режим доступа: <http://www.znanium.com>.

Ковалева Мария Александровна

Создание баз данных в Microsoft Access

Учебно-методическое пособие издано в авторской редакции

Сетевое издание

Главный редактор – Кирсанов К.А.

Вёрстка – Павлов А.А.

Ответственный за выпуск - Алимова Н.К.

Учебное издание

Системные требования:

операционная система Windows XP или новее, macOS 10.12 или новее, Linux.

Программное обеспечение для чтения файлов PDF.

Объем данных 1,45 Мб

Принято к публикации «23» сентября 2019 года

Режим доступа: <https://izd-mn.com/PDF/35MNNPU19.pdf> свободный. – Загл. с экрана. - Яз. рус., англ.

ООО «Издательство «Мир науки»

«Publishing company «World of science», LLC

Адрес:

Юридический адрес — 127055, г. Москва, пер. Порядковый, д. 21, офис 401.

Почтовый адрес — 127055, г. Москва, пер. Порядковый, д. 21, офис 401.

<https://izd-mn.com/>

**ДАННОЕ ИЗДАНИЕ ПРЕДНАЗНАЧЕНО ИСКЛЮЧИТЕЛЬНО ДЛЯ ПУБЛИКАЦИИ НА
ЭЛЕКТРОННЫХ НОСИТЕЛЯХ**