

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ПРАВОСУДИЯ**

**В.Р. Ахметгалиева,
Л.Р. Галяутдинова**

БАЗЫ ДАННЫХ

MicrosoftAccess 2013

Учебно-методическое пособие

Москва
2017

УДК 681.518

ББК 32

Б 17

Авторы-составители:

Ахметгалиева В.Р., ст. преподаватель кафедры правовой информатики, информационного права и естественнонаучных дисциплин Казанского филиала Российского государственного университета правосудия;

Галютдинова Л.Р., доцент кафедры правовой информатики, информационного права и естественнонаучных дисциплин Казанского филиала Российского государственного университета правосудия, канд. физ.-мат. наук.

Рецензенты:

Аксанова А.В., профессор кафедры БСМЭ КНИТУ, д-р экон. наук;

Вахитов Р.Д., зав. кафедрой экономики Казанского филиала Российского государственного университета правосудия, д-р экон. наук, профессор.

Ахметгалиева В.Р., Галютдинова Л.Р.

Базы данных: Microsoft Access 2013: Учебно-методическое пособие. — М.: РГУП, 2017. — с. 94.

ISBN 978-5-93916-629-4

В пособии представлен комплекс задач и примеров, реализованных в среде Microsoft Access 2013, предназначенных для изучения и освоения принципов работы с базами данных, рассмотрены вопросы создания, редактирования и обработки таблиц, форм, запросов и отчетов.

Каждая работа содержит подробные инструкции и рекомендации по их выполнению, что делает возможным использование данных методических указаний для организации самостоятельной работы студентов.

Предназначено для студентов юридических вузов, обучающихся по программам среднего профессионального образования (специальности: 40.02.01 «Право и организация социального обеспечения», 40.02.03 «Право и судебное администрирование»).

© Ахметгалиева В.Р., 2017

© Галютдинова Л.Р., 2017

© Российский государственный университет правосудия, 2017

ОГЛАВЛЕНИЕ

Введение.	4
Лабораторная работа № 1. Знакомство с microsoftaccess (4ч)	9
Лабораторная работа № 2. Разработка инфологической и даталогической моделей данных (6ч).	21
Лабораторная работа № 3. Создание формы в режиме автоформы и с помощью мастера (4ч).	37
Лабораторная работа № 4. Создание и редактирование формы в режиме конструктора форм (4ч).	42
Лабораторная работа № 5. Создание запросов на выборку (6ч).	45
Лабораторная работа № 6. Создание отчетов (4ч).	62
Лабораторная работа № 7. Повторение пройденного материала (4ч)	66
Темы докладов и рефератов.	75
Контрольные вопросы к зачету	77
Приложение а. Самостоятельная работа по вариантам	79
Приложение1. Ег-модедь бд «суд»	93
Приложение2.Реляционная модель бд «суд».	93
Приложение 3. Схема данных «пенсионный фонд».	94
Приложение 4. Схема данных «офис».	94

ВВЕДЕНИЕ

Эффективность функционирования любой информационной системы в организации во многом определяется уровнем разработки информационного обеспечения, ведущим направлением которого является технология создания и эксплуатации баз данных. База данных является интегрированной системой обработки информации, способствующей сокращению избыточности в хранении данных, совместному использованию информации различными пользователями, удобству доступа к данным и безопасности хранения данных. С точки зрения взаимодействия с организацией и информационной системой управления база данных является основным средством создания и хранения информационной модели предприятия.

В связи с этим актуальным является освоение принципов создания и эксплуатации баз данных. Изучение технологии баз данных дает необходимые знания в области создания и проектирования организованных массивов информации.

Основные определения.

Под базой данных (БД) обычно понимается именованная совокупность данных, отображающая состояние объектов и их отношений в рассматриваемой предметной области.

База данных (БД)— совокупность взаимосвязанных данных, организованных в соответствии со схемой базы данных таким образом, чтобы с ними мог работать пользователь [из п. 2.1 ГОСТ 34.321-96].

Система управления базами данных(СУБД)—совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами [из п. 46 ГОСТ 20886-85].

Целостность базы данных — свойство базы данных, определяемое способностью системы управления базой данных защищать компоненты и связи базы данных от искажения в результате некорректных операций и сбоев технических средств [из п. 2 Прил. ГОСТ 20886-85].

Специализированные программные средства, при помощи которых пользователи управляют процессом обработки информацией,

называются **приложениями**. С одной БД могут работать несколько приложений. Приложения могут работать параллельно и независимо друг от друга, а СУБД призвана обеспечить их работу таким образом, чтобы каждое из них выполнялось корректно, и учитывало все изменения в БД, вносимые другими приложениями.

Архитектура БД. В настоящий момент используется предложенная американским комитетом по стандартизации ANSI (AmericanNationalS standardsInstitute) трехуровневая система БД.

1. Уровень внешних моделей. Этот уровень определяет точку зрения на БД отдельных приложений.

2. Концептуальный уровень — центральное управляющее звено, БД представлена в наиболее общем виде, который объединяет данные используемые всеми приложениями. Этот уровень отражает обобщенную модель предметной области (объектов реального мира), для которой создавалась БД.

3. Физический уровень — собственно данные, расположенные в файлах.

Эта архитектура позволяет обеспечить *логическую* (между 1 и 2) и *физическую* (между 2 и 3) независимость при работе с данными. Логическая независимость предполагает возможность изменения одного приложения без корректировки других. Физическая — возможность переноса информации с одних носителей на другие при сохранении работоспособности. Выделение концептуального уровня позволило разработать аппарат централизованного управления БД.



Рис. 2. Трехуровневая модель СУБД

Модели данных. Сами по себе данные не обладают определенной структурой, данные становятся информацией тогда, когда пользователь задает им определенную структуру, то есть осознает их смысловое содержание. Поэтому центральным понятием в области БД является понятие *модели*.

Модель данных – это некоторая абстракция, которая, будучи приложена к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию, то есть сведения, содержащие не только данные, но и взаимосвязь между ними.

На рис.3 представлена классификация моделей данных

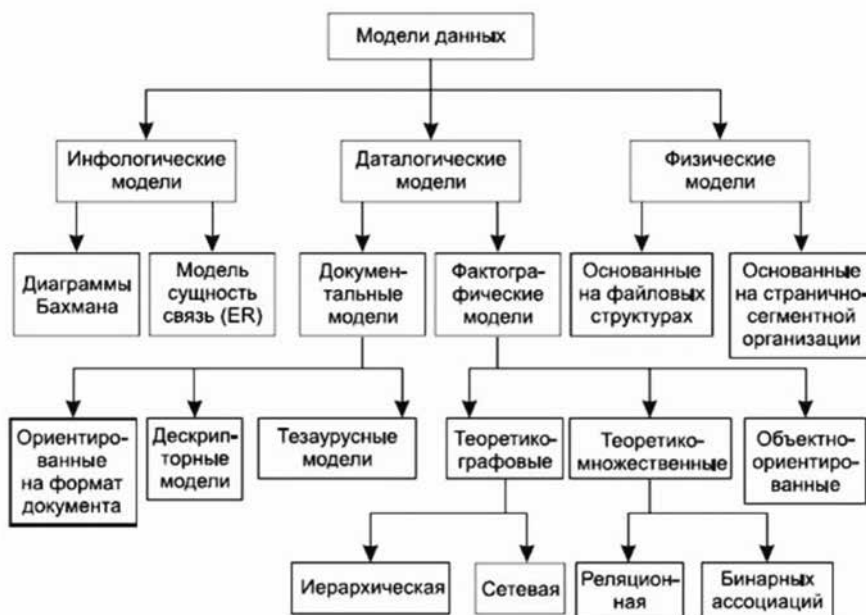


Рис. 3. Классификация моделей данных

Для каждого из трех уровней архитектуры БД можно сопоставить понятие модели.

Реляционная модель данных – основная модель в классе теоретико-множественных моделей. Простота и наглядность, и серьезное теоретическое

обоснование определили большую популярность этой модели. Теоретической основой этой модели стала теория отношений и реляционная алгебра. Американский математик Э.Ф.Кодд в 1970 году сформулировал основные понятия реляционной модели. Основной структурой данных в модели является отношение, именно поэтому модель получила название реляционной.

Инфологические модели. Модели внешнего уровня выражают информацию в виде, не зависящем от используемой СУБД, и называются *инфологическими* или *семантическими*. Они отражают в естественной форме информационно-логический уровень абстрагирования, связанный с описанием объектов, их свойств и взаимосвязей.

Инфологические модели используются на ранних стадиях проектирования для описания структур данных в процессе разработки приложения, а **дatalogические** модели уже поддерживаются конкретной СУБД.

В семидесятых годах было предложено несколько моделей данных, названных *семантическими (инфологическими)*. В настоящий момент наиболее широкое распространение получила **модель Чена** (Chen), предложенная в 1976 году.

В данном учебно-методическом пособии проектирование базы данных рассматривается в лабораторной работе №2, и начинается оно с построения инфологической модели предметной области, а именно **модели Чена**, также известной как **ER – модель** и **модель «сущность – связь»**. Эта модель в наибольшей степени согласуется с концепцией объектно-ориентированного программирования, которая в настоящий момент, несомненно, является базовой для разработки сложных программных систем. Именно на этом этапе выделяются объекты, их свойства и определяются связи между ними.

Следующим этапом является даталогическое проектирование, в ходе которого ER – модель необходимо преобразовать в реляционную модель базы данных. Выделение этого этапа обусловлено тем, что все современные системы управления базами данных являются реляционными, то есть поддерживают эту модель. Физическая реализация полученной реляционной модели будет осуществляться в среде MSAccess 2013 (лабораторные работы №1, №3-№7).

На сегодняшний день MS Access является наиболее популярной системой управления базами данных используемой для разработки настольных баз данных (БД). Богатый набор мастеров и строителей позволяет полностью автоматизировать создание пользовательского

интерфейса, а множество удобных средств разработки помогает сократить время создания законченного приложения. Полная интеграция с другими приложениями MS Office – также большой плюс MS Access.

Все лабораторные работы в пособии, кроме первой и последней, связаны друг с другом, поэтому рекомендуется выполнять их последовательно.

Первая лабораторная работа предусматривает знакомство со средой MSAccess 2013. В ней рассматриваются способы построения таблиц. Вторая работа содержит задания, помогающие освоить основные принципы разработки ER –моделей с различными типами связей и преобразования их в реляционные модели данных с последующей реализацией в среде Access: построением таблиц и заданием связей между таблицами.

Следующие две работы №3 и №4 посвящены способам создания и редактирования форм.

В работе №5 рассматриваются вопросы создания запросов на выборку.

Работа №6 посвящена созданию отчетов.

Работа №7 предназначена для повторения и систематизации полученных знаний и навыков работы с базами данных и рассчитана на уверенное владение ранее изложенным материалом. В этой работе рассматривается создание БД «Пенсионный фонд» и БД «Офис» в MSAccess 2013.

Практически все лабораторные работы заканчиваются заданиями для самостоятельной работы, контрольными вопросами и тестами для лучшего усвоения изученного материала студентами. Контрольные вопросы можно использовать для защиты лабораторной работы, а контрольные тесты рекомендуется решить перед выполнением заданий для самостоятельной работы.

На каждую лабораторную работу отводится 4 ч.

В конце пособия приведены темы рефератов.

ЛАБОРАТОРНАЯ РАБОТА № 1

Знакомство с microsoftaccess (4 ч)

***Цель работы:** научиться создавать новую базу данных и освоить способы создания и редактирования таблиц данных.*

СУБД MS ACCESS

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

В Access база данных включает в себя все объекты, связанные с хранимыми данными (таблицы, формы, отчеты, запросы, макросы, модули). Все объекты Access хранятся в одном файле с расширением **.accdb**. В таблицах хранятся данные, которые можно просматривать, редактировать, добавлять. Используя формы, можно выводить данные на экран в удобном виде, просматривать и изменять их. Запросы позволяют быстро выбирать необходимую информацию из таблиц. С помощью отчетов можно создавать различные виды документов для вывода на печать. Макросы и модули позволяют автоматизировать работу с базой данных.

Формирование Базы данных (БД) в Access состоит из ряда последовательных этапов. Первый этап этого процесса – **создание таблиц**. Таблицы в Access являются теми первичными, исходными файлами, на основе которых в дальнейшем строится все здание базы данных.

Таблицы предназначены для хранения структуры и данных БД, ввода и редактирования данных.

Данные в таблице организованы в столбцы (называемые **полями**) и строки (**записи**).

Рассмотрим вкратце типы данных в MS Access 2013, виды, назначение и допустимый размер данных, которые могут назначаться полям таблицы в Access (Таблица 1).

Таблица 1

Типы данных

Название типа	Назначение
Короткий Текст	Текст длиной до 255 символов
Длинный Текст	Текст длиной до 65 000 символов
Числовой	Числа различных форматов
Дата/время	Дата и/или время
Денежный	Денежные значения различных форматов
Счетчик	Счетчик, который автоматически увеличивается на единицу с добавлением каждой новой записи
Логический	Величины, способные принимать только два значения: да/нет или 1/0
Поле объекта OLE	Поля, позволяющие вставлять рисунки, звуки и данные других типов
Гиперссылка	Ссылки, дающие возможность открывать объект Access (таблицу, форму, запрос и т. п.), файл другого приложения или web-страницу

Наиболее детальным и основательным методом формирования таблиц является **режим конструктора**. В режиме конструктора задаются *имена полей* и *типы данных*. В зависимости от характера данных необходимо задать *свойства полей*.

Окно **Свойства поля**(рис.4) расположено в нижней половине окна конструктора таблиц. Его вид зависит от выбранного типа данных (текстовый, числовой, денежный и т.д.).

Свойства поля:

Размер поля — это свойство определяет максимальное число символов, которое может быть размещено в данном поле.

Формат поля — определяется для числовых полей и определяет форму записи числа, для полей типа дата-время и задает формат представления даты, для логических полей и задает один из двух вариантов представления значений «Истина» - «Ложно».

Маска ввода — это свойство задает шаблон для более удобного ввода данных в форме.

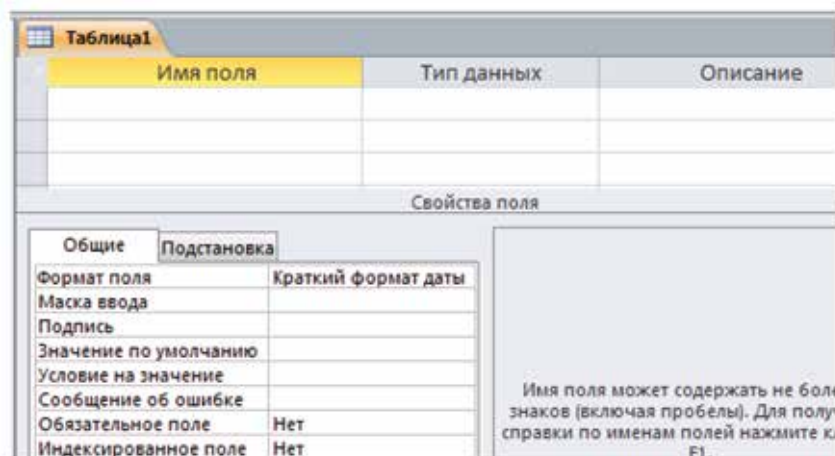


Рис. 4. Окно таблицы в режиме Конструктора

Подпись — свойство определяет заголовок поля, если подпись не задать, то в качестве заголовка будет выступать имя поля.

Значение по умолчанию — при вводе данных в поле автоматически будет введено значение по умолчанию, которое при необходимости может быть изменено.

Условие на значение — выражение, определяющее допустимые для данного поля данные.

Сообщение об ошибке — текст, выводимый в системном окне при нарушении заданного условия на вводимые данные.

Обязательное поле — устанавливается значение «Да» в случае недопустимости отсутствия данных в поле, значения всегда должны быть известны и введены в поле. Значение «Нет» означает возможность присутствия в этом поле значения Null — т.е. значение неизвестно.

Пустые строки — устанавливается значение «Да» в случае допустимости отсутствия данных в поле, т.е. значение просто отсутствует. Например у абитуриента отсутствует номер телефона.

Индексированное поле — устанавливается «Да» в случае необходимости выполнять упорядочивание (сортировку) по данному полю. Здесь же определяется допустимы ли совпадения значений в этом поле.

Каждая таблица должна содержать одно или несколько полей, однозначно идентифицирующих каждую запись в таблице. Такое поле

называется **ключевым полем** таблицы. Если поле содержит уникальные значения, такие как *Коды, Инвентарные номера, Номер зачетной книжки* или *Табельный номер*, то это поле можно определить как ключевое.

1. Создание новой базы данных. Запустить программу MSAccess и создать новую базу данных «Офис».

- Для этого воспользоваться пиктограммой на рабочем столе, либо выполнить **Пуск – Программы – Microsoft Access**.

- В открывшемся окне выбрать **Пустая база данных рабочего стола**, и, перед тем как выполнить команду **Создать**, дайте имя файлу «Офис» и укажите папку для сохранения файла (рис. 5).

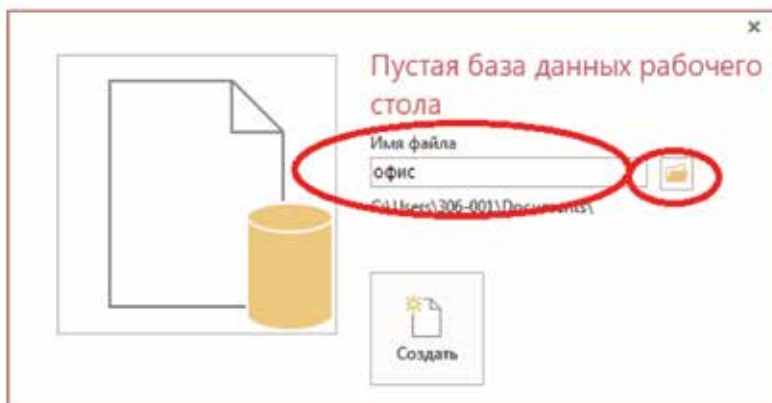


Рис. 5. Окно сохранения базы данных

- В результате будет создан файл новой БД. На экране будет отображено **Окно базы данных**, где автоматически будет создана таблица, готовая к вводу информации. В верхней части таблицы находятся заголовки полей (**Код, Добавить поле**).

- Закройте таблицу.
- Выбрать и рассмотреть по очереди различные вкладки верхней части **Окна базы данных**: **Главная, Создание, Внешние данные, Работа с базами данных**.

Для создания объектов БД: **Таблиц, Запросов, Форм, Отчетов, Макросов, Модулей** используют команды на вкладке **Создание**.

2. Создание новой таблицы.

Для создания **Таблиц** используют команды группы **Таблицы** на вкладке **Создание**. Таблицы можно создавать, например, в **режиме таблицы**, либо в **режиме конструктора**.

2.1. Создание новой таблицы в режиме конструктора.

При конструировании таблицы необходимо задать имена полей и тип данных, а также можно задать свойства полей (см. рис 3 и табл. 1). **Имя поля** — это заголовки столбцов таблицы. В разделе **Тип данных** можно задать, какие данные и в каком формате будут вводиться (числовой, текстовый, денежный и т.п.). **Описание** поля является необязательным параметром при конструировании таблицы.

- Создать новую таблицу в режиме конструктора. Для этого выполнить **Создание/Таблицы/Конструктор таблиц** (вкладка/ группа/ команда).

- В первой строке в разделе **Имя поля** набрать № п/п.

- Мышкой переключиться в раздел **Тип данных**. Справа от мышки появится стрелка раскрывающегося списка. Развернуть список и выбрать из него тип **Счетчик**. Переключиться на следующую строку.

Таблица 2.

Имя поля	Тип данных	Свойства
№ п/п	Счетчик	-
Фамилия	Короткий Текст	Размер: 25, Обязательное поле: Да, Пустые строки: нет.
Дата рождения	Дата/время	Формат поля: Краткий формат даты
Адрес	Короткий Текст	
Паспорт	Короткий Текст	
Средняя з/пл	Денежный	Число десятичных знаков: 2
Дети	Логический	Формат поля: Да/нет
Пособие	Денежный	Число десятичных знаков: 2

- В разделе **Имя поля** набрать *Фамилия*, в разделе **Тип данных** выбрать *Короткий Текст* (см таблицу 2).

- Заполнить свойства для поля *Фамилия*. В разделе *Размер поля* указать 25 (символов), в разделе *Обязательное поле* указать Да, *Пустые строки* — Нет.

- В следующей строке раздела Имя поля ввести *Дата рождения*. Тип данных задать *Дата/время*. Заполнить Свойства поля: Формат поля— *Краткий формат даты*(см. рис.3).

- Для полей *Адрес* и *Паспорт* установить тип *Короткий Текст*.
- Для *Средней з/п/и Пособия — Денежный*с двумя знаками после запятой.
- Для поля *Дети* установить тип *Логический*, форматполя выбрать *Да/нет*.

2.2. Задание ключевого поля.

Ключ служит для установления связей между таблицами и для предотвращения ввода повторяющихся данных. В данной таблице в качестве ключевого следует использовать поле *№ п/п*.

- Выделить поле *№ п/п*. и нажать кнопку  на вкладке **Рабо-**

та с таблицами/Конструкторили выбрать команду *Ключевое поле* в контекстном меню.

2.3. Сохранение таблицы.

Все объекты в MSAccess, в том числе таблицы сохраняются стандартным для Windows способом.

- Нажмите на кнопку **Закрыть «Таблица1»**(рис. 6).

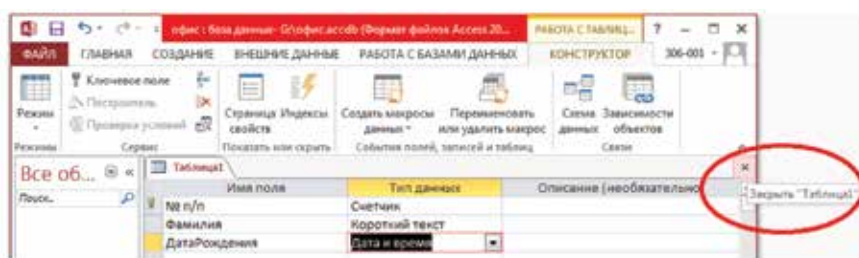


Рис.6. Закрытие таблицы

- В появившемся окне *Сохранение* (рис. 7) введите имя таблицы *Сотрудники*. Нажмите Ок.

3. Ввод данных в таблицу.

- Открыть таблицу *Сотрудники*. Для этого в окне переходов(слева) выделить таблицу и дважды щелкнуть левой кнопкой мыши.Таблица откроется в Режиме таблица.



Рис. 7. Сохранение таблицы

- Ввести 10 записей. Поле *Паспорт* не заполнять.
- Обратите внимание, что поле № п/п типом данных *счетчик* заполняется автоматически при добавлении новой записи.

4. Редактирование таблицы.

- Ознакомиться с командами на вкладке *Работа с Таблицами/Поля* (рис. 8).

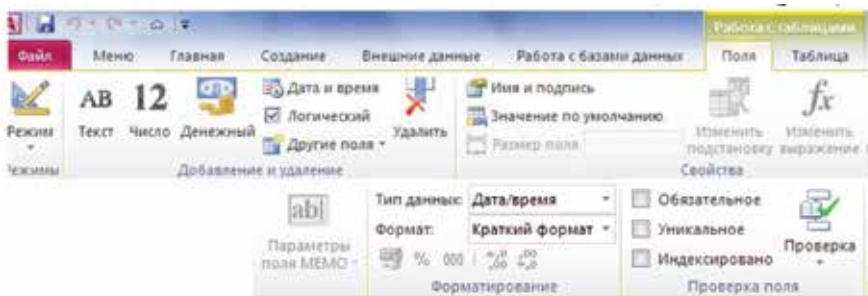


Рис. 8. Вкладка Работа с таблицами/Поля

- Выделяя каждое поле (№ п/п, *Фамилия*, *Дата рождения*, *Адрес*, *Паспорт*, *Средняя з/пл*, *Дети*, *Пособие*), проверьте, правильно ли указали тип данных и свойства.
- Передвигая границы таблицы, изменить высоту и ширину ячеек для наиболее выгодного представления таблицы на экране.
- Переименовать поле № п/п в *Табельный номер*. Для этого дважды щелкнуть мышью в области заголовка поля и изменить его.

5. Создание маски ввода.

Маски ввода используются для форматирования данных и управления вводимыми значениями. Маска ввода определяет типы значений,

которые могут быть введены в данную позицию. В основном маски ввода используются в *Текстовых* полях и полях *Дата/Время*. В приведенной ниже таблице представлены специальные символы, которые могут использоваться для задания маски ввода.

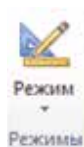
Таблица 3

Специальные символы для задания маски ввода

<i>Символ</i>	<i>Описание</i>
0	Цифра (от 0 до 9, ввод обязателен)
9	Цифра или пробел (ввод не обязателен)
L	Буква (от A до Z или от A до Я, ввод обязателен)
?	Буква (от A до Z или от A до Я, ввод не обязателен)
A	Буква или цифра (ввод обязателен)
a	Буква или цифра (ввод не обязателен)
&	Любой символ или пробел (ввод обязателен)
C	Любой символ или пробел (ввод не обязателен)
< , >	Указывает перевод следующих символов на нижний регистр, верхний регистр
\	Указывает ввод любого следующего символа в качестве текстовой константы. Используется для отображения всех перечисленных в данной таблице специальных символов как обычных текстовых констант (например, \A выводится как символ «A»).

- Переключиться в режим конструктора. Для этого выбрать в левом

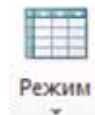
верхнем углу на ленте команду *Режимы*



и выбрать в выпадающем

списке команду *Конструктор*.

- Задать следующую маску ввода для поля *Паспорт*: **00\ -00\ -000000**.
- Сохранить изменения.



• Переключиться в режим таблицы
Паспорт.

и ввести данные в поле

- Переключиться в режим конструктора и для поля *Фамилия* в окне Свойства поля в графе *Формат* поставить знак **>** (знак «больше» переводит все символы в ПРОПИСНЫЕ).

- Переключиться в режим таблицы и посмотреть, как изменились данные.

6. Заголовок поля.

- Переключиться в режим конструктора.
- В поле *Паспорт* в окне Свойства поля в графе *Подпись* ввести **Паспортные данные**.
- Переключиться в режим таблицы. Обратите внимание, что заголовок поля изменился.

Таким образом, в окне Свойства в графе **Подпись** можно задавать *заголовок поля*, если он не совпадает с *именем поля*.

- Закрыть таблицу.

7. Копирование и удаление таблиц.

- Скопировать таблицу. (Для этого таблицу предварительно закрыть!) Копирование таблицы происходит в окне базы данных аналогично копированию файлов в Windows. Отметить таблицу. Скопировать ее в буфер обмена.

- Вставить таблицу из буфера обмена. При этом второй таблице автоматически присваивается имя Копия таблицы Сотрудники.

- Показать работу преподавателю.

- Удалить скопированную таблицу. Для этого выделить таблицу и нажать **Delete**.

Выводы:

- Формирование таблиц является первым этапом создания БД. Таблицы предназначены как для хранения структуры БД, так и данных БД.

- Наиболее детальным и основательным методом формирования таблиц является *режим конструктора*. В режиме конструктора задаются *имена полей и типы данных, ключевое поле (или ключевые поля)*. В зависимости от характера данных необходимо задать *свойства полей*.

- Опции **Формат** и **Маска ввода** позволяют задавать формат, в котором данные запоминаются при вводе в таблицу. Свойство **Маска ввода** целесообразно использовать при вводе данных в пустые графы бланков, таблиц и т.п.

Контрольные тесты:

1. Все данные в реляционной базе данных представлены в виде:

- a. Полей
 - b. Таблиц
 - c. Индексов
 - d. Записей
2. Записью в базе данных называется:
- a. объект базы данных
 - b. строка таблицы
 - c. столбец таблицы
 - d. каждое сохранение изменения информации в базе данных
3. Выражение 2009 относится к типу данных:
- a. Числовой
 - b. Короткий Текст
 - c. Логический
 - d. Счётчик
4. Выражение 25/05-09 относится к типу данных:
- a. Числовой
 - b. Короткий Текст
 - c. Дата/время
 - d. Гиперссылка
5. Выражение КодСотрудника относится к типу данных:
- a. Длинный Текст
 - b. Короткий Текст
 - c. Логический
 - d. Поле объекта OLE
6. Какое поле можно считать уникальным?
- a. поле, значения в котором не могут повторяться;
 - b. поле, которое носит уникальное имя;
 - c. поле, значение которого имеют свойство наращивания;
 - d. поле, значения в котором повторяются;
 - e. всё выше верно.
7. Ключевое поле таблицы – это
- a. номер первой по порядку записи;
 - b. любое поле числового типа;
 - c. одно или несколько полей, значения которых однозначно определяют любую запись в таблице;
 - d. первое поле числового типа
8. Почему при закрытии таблицы программа MSAccess не предлагает выполнить сохранение внесенных данных:

- а. недоработка программы;
- б. потому что данные сохраняются сразу после ввода в таблицу;
- с. потому что данные сохраняются только после закрытия всей базы данных.

9. Содержит ли какую-либо информацию таблица, в которой нет ни одной записи?

- а. пустая таблица, не содержит ни какой информации;
- б. пустая таблица содержит информацию о структуре базы данных;
- с. пустая таблица содержит информацию о будущих записях;
- д. таблица без записей существовать не может;
- е. всё выше верно.

10. В чем состоит особенность поля «счетчик»?

- а. служит для ввода числовых данных;
- б. служит для ввода действительных чисел;
- с. данные хранятся не в поле, а в другом месте, а в поле хранится только указатель на то, где расположен текст;
- д. имеет ограниченный размер;
- е. имеет свойство автоматического присваивания.

11. Без каких объектов не может существовать база данных:

- а) без модулей;
- б) без отчетов;
- с) без таблиц;
- д) без форм;
- е) без запросов.

12. Таблицы в базах данных предназначены:

- а) для хранения и обработки данных базы;
- б) для отбора и обработки данных базы;
- с) для ввода данных базы и их просмотра;
- д) для выполнения сложных программных действий.
- е) для автоматического выполнения группы команд;

Задания для самостоятельной работы.

1) Создать БД «Соцзащита».

В режиме Конструктора создать 2 таблицы:

Патронажные сёстры с полями: КодСестры, ФИО;

Пенсионеры с полями: КодПенсионера, ФИО, АдресПроживания, ПаспортныеДанные, СуммаПенсии, Телефон.

Типы данных и свойства для каждого поля задайте самостоятельно. Определите ключевое поле в каждой таблице.

2) Создать БД «Факультет».

В режиме Конструктора создать 4 таблицы:

Направления с полями: *Код направления, Направление, СрокОбучения*;

Группы с полями: *Номер группы, КоличествоЧеловек, КодНаправления.*

Студенты с полями: *Код Студента, Фамилия, Имя, Отчество, Дата Рождения, Бюджетная или внебюджетная форма обучения, Номер группы.*

Доп Сведения с полями: *КодСтудента, АдресПрописки, №СтраховогоСвидетельства, ПаспортныеДанные*

Типы данных и свойства для каждого поля задайте самостоятельно. Определите ключевое поле в каждой таблице.

3) Создать БД по вашему варианту.

Контрольные вопросы:

1. В каких режимах можно создать структуру таблицы в MSAccess?
2. Какая информация указывается в структуре таблицы?
3. Что такое ключевое поле, какие требования предъявляются к нему?
4. Как в СУБД MSAccess можно определить ключевое поле?
5. Какие типы данных используются в MSAccess?
6. Как называется атрибут, однозначно идентифицирующий каждую запись в таблице?
7. Какие виды числовых данных, используемые в СУБД MSAccess?
8. Поля, для которых недопустимо редактирование?
9. Тип СУБД, к которому принадлежит MSAccess?
10. Какой тип данных соответствует полю Оклад?
11. Назовите режимы просмотра таблиц.
12. Тип данных, используемый в СУБД Access для искусственного поля Код?
13. Тип данных, который необходимо использовать для телефонного номера (095)3568290?
14. Перечислите свойства для полей с текстовыми и числовыми типами данных.
15. Как при определении структуры таблицы использовать свойства полей?

ЛАБОРАТОРНАЯ РАБОТА № 2

Разработка инфологической и даталогической моделей данных (6 ч)

Цель работы: *научиться разрабатывать модели по заданному описанию предметной области, а также создавать и редактировать связи между таблицами БД*

Процесс проектирования БД представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели. Можно выделить следующие *этапы проектирования*.

1. Системный анализ и словесное описание информационных объектов предметной области.
2. Проектирование инфологической модели предметной области — частично формализованное описание объектов предметной области в терминах некой инфологической, например, ER — модели.
3. Даталогическое (или логическое) проектирование БД, то есть описание БД в терминах принятой даталогической модели.
4. Физическое проектирование БД, то есть выбор способа размещения БД на внешних носителях.

Системный анализ должен заканчиваться подробным описанием информации об объектах, формулировкой задач с кратким описанием алгоритмов их решения, описанием входных и выходных документов. (Описание предметной области в нашем курсе нам будет уже известно.)

Инфологическая модель должна выражать информацию в виде, не зависящем от используемой системы управления базами данных (далее — СУБД). Обычно эта модель отражает описание объектов, их свойства и взаимосвязи в виде схем. В настоящий момент наиболее широкое распространение получила модель Чена (Chen), которая называется «сущность — связь» или ER — модель (EntityRelationship). В ее основе лежат следующие базовые понятия.

Сущность — это класс однотипных объектов.

Сущность имеет уникальное имя. Предполагается, что в системе существует множество экземпляров данной сущности. (Пр.: сущность - Студент, экземпляры — студенты вуза; Преподаватель — преподаватели вуза).

Объект имеет свой набор *атрибутов* — свойств объекта. (Пр.: у сущности Студент атрибуты номер зачетной книжки, фамилия, имя, отчество, группа; у сущности Преподаватель — табельный номер, ФИО, кафедра).

Атрибут, однозначно идентифицирующий конкретный экземпляр сущности, называется **ключевым** (например, Номер зачетной книжки; Табельный номер).

Между сущностями могут быть установлены *связи*. По множественности связи делятся на три типа:

- ♦ **один-к-одному** (один экземпляр одной сущности связан только с одним экземпляром другой сущности),
- ♦ **один-ко-многим** (один экземпляр одной сущности связан с несколькими экземплярами другой сущности),
- ♦ **многие-ко-многим** (один экземпляр одной сущности связан с несколькими экземплярами другой сущности и наоборот).

Связь любого типа может быть обязательной, если в данной связи должен участвовать каждый экземпляр, и необязательной. Связь может быть обязательной с одной стороны и необязательной с другой.

При разработке модели «сущность — связь» каждая сущность изображается в виде прямоугольника, в верхней части которого отражается имя сущности, а в нижней — атрибуты данной сущности, ключевой атрибут подчеркнут.

Общепринятое графическое изображение сущностей и связей между ними представлено на рисунках 9-11.

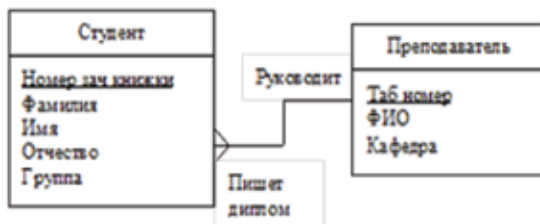


Рис. 9. Пример отношения «один-ко-многим»

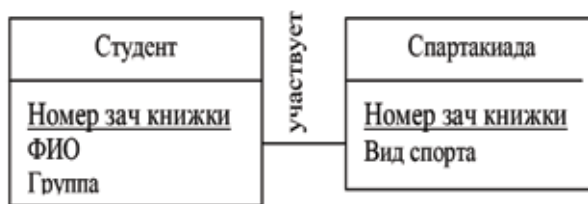


Рис. 10. Пример отношения «один-к-одному»



Рис. 11. Пример отношения «многие-ко-многим»

Сущность, которой соответствует конец связи «один», называется *главной* или *родительской*. На рис. 9 это сущность *Преподаватели*, на рис. 10 — *Студент*.

Сущность, которой соответствует конец связи «много», называется *подчинённой* или *дочерней*. На рис. 9 это сущность *Студент*.

Сущность *Спартакиада*, представленная на рис. 10, тоже буде подчинённой, так как в связи «один-к-одному», как и в типе связи «один-ко-многим» одна сущность главная, а другая — подчинённая.

Для реализации проекта в конкретной СУБД следует разработать даталогическую модель. На настоящий момент для этой цели используется реляционная модель данных. Основной структурой данных в модели является *отношение*, именно поэтому модель получила название реляционной (от англ. «relation» - «отношение»).

Существует **алгоритм** преобразования ER — модели в реляционную модель данных:

1. Каждой сущности ставится в соответствие отношение реляционной модели данных.

2. Каждый атрибут сущности становится атрибутом соответствующего отношения, задается тип данных и обязательность или необязательность данного атрибута.

3. Первичный ключ сущности становится ключевым полем соответствующего отношения.

4. В каждое отношение, соответствующее подчиненной сущности, добавляется атрибут основной сущности, и этот атрибут становится *внешним ключом*.

5. Для определения необязательного типа связи у атрибута, соответствующего внешнему ключу, устанавливается необязательность данного атрибута. При обязательном типе связи устанавливается его обязательность.

6. Если в ER – модели присутствуют связи «многие-ко-многим», то для перехода к реляционной модели данных (где такие связи не поддерживаются) вводится дополнительное связующее отношение. Оно связано с каждым исходным связью «один-ко-многим», а его атрибутами служат первичные ключи связываемых отношений.

В результате выполнения даталогического проектирования должна быть разработана схема БД, то есть совокупность отношений, которые моделируют объекты БД и связи между ними.

Рассмотрим проектирование инфологической (ER-модели) и даталогической (реляционной) модели на примере БД «Офис».

Описание предметной области.

В офисе есть несколько отделов. В отделах работают сотрудники (по 3-5 чел). У каждого сотрудника фиксируется его должность, оклад, стаж. У каждого сотрудника свои служебные обязанности. В одной должности могут работать несколько сотрудников.

База данных «Офис» предназначена для ведения учета сотрудников и должна обеспечивать получение сведений о сотрудниках, получаемых ими заработных платах, количестве работников в данной должности, имеющих вакансии.

Проектирование ER-модели БД «Офис»

1) Выделим следующие сущности: *отделы, сотрудники, должности и должностные обязанности*.

2) Зададим атрибуты и определим ключевой атрибут для каждой сущности.

Для сущности «Отделы»: **КодОтдела**, *Отдел(название отдела)*;

Для сущности «Сотрудники»: *Таб№*, *Фамилия*, *Имя*, *Отчество*, *Место работы*, *Должность*, *Стаж*;

Для сущности «Должности»: *КодДолжности*, *Должность*, *Оклад*, *Штат*.

Для сущности «СлужебныеОбязанности»: *Таб№*, *СлужебОбязанности*, *телефон*.

3) Установим связи.

Нам известно, что в одном отделе работают несколько сотрудников. А сотрудник может работать только в одном отделе. Таким образом, между сущностями *Отделы* и *Сотрудники* устанавливается связь *один-ко-многим*.

В одной должности могут работать несколько сотрудников. Поэтому выделили отдельно сущность *Должности*. А сотрудник может работать только в одной должности. Таким образом, между сущностями *Должности* и *Сотрудники* также устанавливается связь *один-ко-многим*.

У каждого сотрудника свои служебные обязанности. Значит между сущностями *Сотрудники* и *СлужебныеОбязанности* устанавливается связь *один-к-одному*. И, поэтому, у этих сущностей одинаковые ключевые атрибуты: *Таб№*.

Схема *ER-модели БД «ОФИС»* представлена на рис. 12.

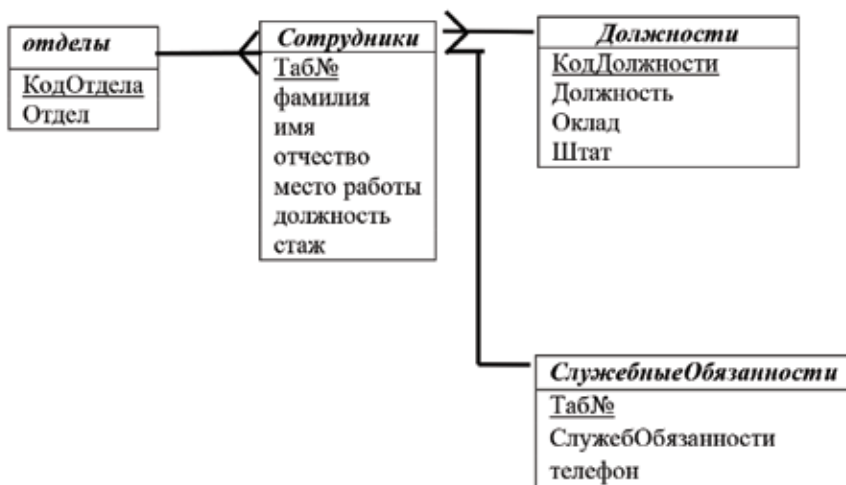


Рис. 12. Схема *ER-модели БД «Офис»*

Проектирование реляционной модели БД «ОФИС»

Согласно алгоритму преобразования ER – модели в реляционную, разработаем реляционную (дatalogическую) модель БД.

Названия отношений согласно алгоритму в реляционной модели совпадают с названиями сущностей ER-модели. Атрибуты, в том числе, ключевые, останутся теми же, за исключением атрибутов, участвующих в связях.

Так как каждый отдел, т.е. *Место работы* каждого сотрудника, определяется однозначно по ключевому атрибуту **КодОтдела**, то согласно п. 4 алгоритма преобразования ER – модели в реляционную модель, в подчиненное отношение *Сотрудник* для связи добавляется ключевой атрибут *КодОтдела*, который станет для него *внешним ключом*.

Аналогично, каждая *Должность* определяется однозначно по ключевому атрибуту **КодДолжности**. Поэтому, в подчиненное отношение *Сотрудник* для связи вместо атрибута *Должность* добавляется *КодДолжности*, который станет для него *внешним ключом*.

Для каждого атрибута укажем соответствующий ему *Тип данных*.

При этом необходимо учесть, что для *внешних ключей* устанавливается тип данных *Числовой*, если соответствующий ему в главном отношении ключевой атрибут имеет тип данных *Счетчик*.

Схема реляционной модели БД «ОФИС» будет выглядеть следующим образом (рис. 13).

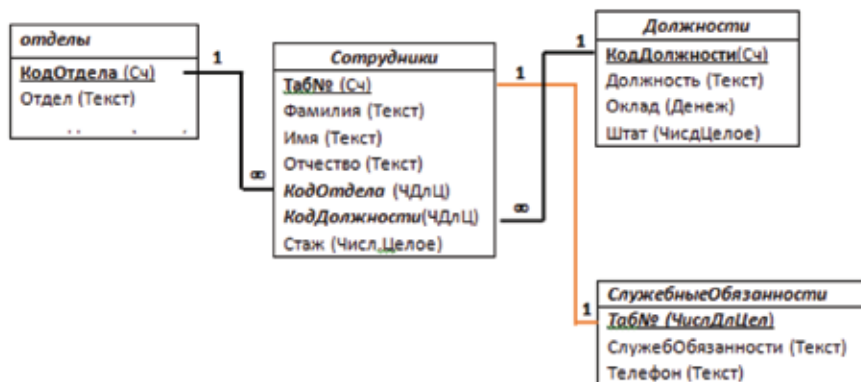
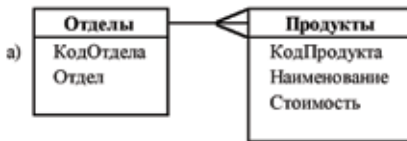


Рис. 13. Схема реляционная модели БД «Офис»

Задание 1: Преобразовать приведенные ER-модели в реляционные.

Указать ключевые атрибуты и тип связи. В реляционную модель в одно из отношений добавить внешний ключ для их связывания (см. п. 4 алгоритма).



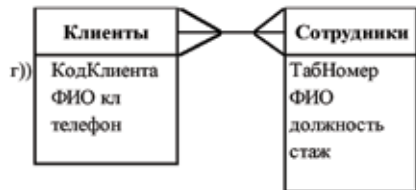
В каждом отделе торгуют продуктами определённого вида



Каждый сотрудник предлагает клиентам несколько услуг



Каждый клиент может заказать несколько услуг



Каждый сотрудник обслуживает многих клиентов. Каждый клиент может заказать различные услуги у нескольких сотрудников.

Задание 2: Разработать БД «СУД» согласно следующему описанию предметной области.

1. Описание предметной области. В каждом районе города имеется свой *районный суд*. В каждом суде имеются *судьи*, которые занимаются рассмотрением гражданских, административных и уголовных дел.

При разработке БД требуется отразить следующую информацию:

для дел — номер, название, дату открытия, дату закрытия, количество томов, кто рассматривает дело;

у судей — код, фамилия, имя, отчество, место работы (районный суд);

для районных судов — код, район, адрес.

2. Проектирование инфологической модели предметной области.

1) Выделить следующие сущности: *Районные суды*, *Судьи*, *Дела*.

2) Задать их атрибуты. У каждой сущности определить ключевой атрибут.

3) Установить связи. Обосновать типы связей и их обязательность. Какие сущности будут основными? Какие будут подчиненными?

3. Согласно алгоритму преобразования ER – модели в реляционную, разработать реляционную (даталогическую) модель БД.

Для установления связей ответьте на следующие вопросы:

Какие отношения друг с другом связаны? Какие их них в это связи главные, какие подчинённые?

Какие поля будут полями внешнего ключа? Зачем они нужны?

Атрибуты, между которыми устанавливается связь, должны иметь одинаковый тип и свойства (ключевому полю с типом данных *Счетчик* соответствует в другом отношении поле с типом данных *Числовой*, размер поля *Длинное целое*).

Проверьте, соответствуют ли спроектированные вами модели предметной области, приложению 1 и приложению 2.

4. Реализовать разработанную информационную модель в системе управления базами данных (СУБД) Access.

1) В MSAccess создать новую БД «СУД».

Создать структуру новой базы данных согласно реляционной модели (см. приложение 1), т.е. создать 3 таблицы: *Дела*, *Судьбы*, *Районные суды* (см. приложение 2.).

Характеристики задать самостоятельно, исходя из ожидаемых данных (например, для поля *Дата открытия* задать тип данных *Дата/Время*, для поля *Адрес суда* – *Короткий текст*, для поля *Код Судьбы* – *Счетчик*).

2) Создать связи между таблицами.

При создании связей между таблицами следует обратить внимание на следующее:

- одной связью можно связать только 2 таблицы;
- как правило, в связи участвует одно ключевое поле, реже – два;
- связь устанавливается между двумя полями, имеющими одинаковый смысл (при этом их названия могут отличаться).

Каждый атрибут таблицы должен иметь тип, соответствующий тем данным, которые будут храниться в этом атрибуте. Когда между двумя таблицами устанавливается *связь*, это означает, что значения из одной таблицы ставятся в соответствие значениям другой таблицы. Обычно связывают *ключевое поле родительской таблицы* с соответствующим ему полем в *дочерней таблице* (внешним ключом).

Установить связи между таблицами согласно разработанной в п.3 реляционной модели.

- **Перед созданием связей закрыть все таблицы!**
- На вкладке **Работа с базами данных** нажать на кнопку **Схема данных**.



- В диалоговом окне *Добавление таблицы* выделить все таблицы и нажать кнопку *Добавить*. Закрыть окно *Добавление таблицы*.
- В окне *Схема данных* установить связь между соответствующими полями таблиц **Районные суды** и **Судьи**.

Для этого щелкнуть мышью по полю *Код суда* таблицы *Районные суды*, удерживая ее, перетащить указатель мыши на поле *Код суда* таблицы *Судьи*.

На экране появится диалоговое окно **Изменение связей** (рис. 14.).

В левой части этого окна указано связываемое поле родительской таблицы, в правой — дочерней.

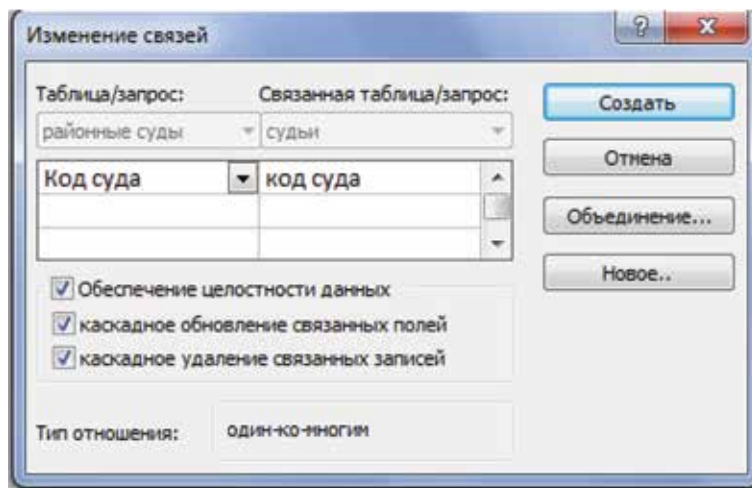


Рис. 14. Диалоговое окно Изменение связей

Поставить флажки на пунктах **Обеспечение целостности данных**, **Каскадное обновление связанных полей**, **Каскадное удаление связанных полей**, нажать кнопку **Создать**.

В окне **Схемы данных** появится линия связи между таблицами *Районные суды* и *Судьи*. Следует обратить внимание, что на одном конце линии связи стоит знак 1, а на другом ∞. Этот тип связи «*один-ко-многим*». Связь с отношением «*один-ко-многим*» является одним из наиболее часто используемых типов связей между таблицами. В такой связи *каждой записи* в таблице *Районные суды* могут соответствовать *несколько записей* в таблице *Судьи*, а запись в таблице *Судьи* не может иметь *более одной* соответствующей ей записи в таблице *Районные суды*.

- Аналогичным образом установить связи между остальными таблицами.

Должно получиться как на рис. 15.

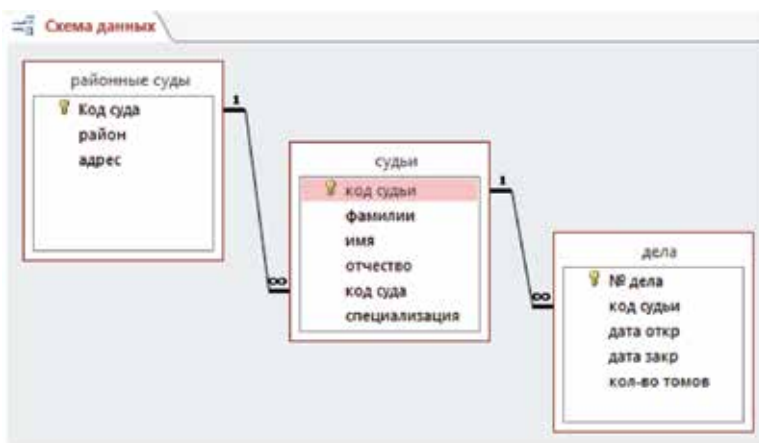


Рис. 15. Диалоговое окно **Схема данных**

- Сохранить БД «СУД».
- Показать результат работы преподавателю. Закрыть базу данных.

Обеспечение целостности данных, Каскадное обновление связанных полей, Каскадное удаление связанных полей.

Включение флажка *Обеспечение целостности данных* позволяет защититься от случаев удаления записей из одной таблицы, при которых связанные с ними данные других таблиц останутся без связи.

Чтобы условие целостности могло существовать, поле главной таблицы должно обязательно быть ключевым и оба поля должны иметь одинаковый тип и свойства.

Если для выбранной связи обеспечивается поддержание целостности, то можно задать режим **каскадного удаления связанных записей** и режим **каскадного обновления связанных полей**. Такие параметры делают возможным в главной таблице, соответственно, удаление записей и изменение значения в ключевом поле, т. к. при этих параметрах система автоматически выполнит необходимые изменения в подчиненных таблицах, обеспечив сохранение свойств целостности базы данных.

В режиме **каскадного удаления связанных записей** при удалении записи из главной таблицы будут автоматически удаляться все связанные записи в подчиненных таблицах.

В режиме **каскадного обновления связанных полей** при изменении значения ключевого поля в записи главной таблицы Access автоматически изменит значения в соответствующем поле в подчиненных записях.

Выводы.

- Для связывания данных из разных таблиц каждая таблица должна содержать одно или несколько полей, однозначно идентифицирующих каждую запись в таблице-ключевых полей.

- После создания таблиц необходимо определить связи между таблицами. Как правило, связывают **ключевое поле родительской таблицы** с соответствующим ему полем в **дочерней таблице** (внешним ключом). При этом образуется связь «**один-ко-многим**».

- При связи один-к-одному связываются ключевые поля. Если у одного из ключевых полей тип данных **Счетчик**, то у ключевого поля дочерней таблицы тип данных должен быть **Числовой** (свойство Формат: **Длинное целое**).

Задания для самостоятельной работы

Разработать ER – модель и реляционную модель для нижеприведенных примеров.

Пример 1:

Описание предметной области. На кафедре «Социальная работа и» несколько групп (94-21, 94-22, 93-21, 93-22, 92-21). В каждой группе несколько студентов (2-3), у которых фиксируется Фамилия, имя, отчество, дата рождения, бюджетная или внебюджетная форма обучения. Кроме этого, необходимо создать таблицу дополнительных сведений

о студенте, куда включить адрес прописки, номер страхового свидетельства, паспортные данные.

Пример 2:

Описаниепредметной области. Пенсионный фонд ежемесячно собирает со всех предприятий данные о сотрудниках. Данные включают в себя ИНН, Фамилию И.О., домашний адрес, телефон, место работы (предприятие), стаж работы, оклад. В пенсионном фонде ежемесячно фиксируются зарплата каждого работника за отчетный период и отчисления: на накопительную и страховую части пенсии.

Пример 3:

Описаниепредметной области. В отделении социальной защиты работают 5 патронажных сестер, которые обслуживают лежащих пенсионеров. У каждой патронажной сестры по 2-3 пенсионера, для которых указывается адрес проживания, паспортные данные, сумма пенсии, телефон. Кроме этого имеются дополнительные сведения об основном диагнозе болезни, справочные данные о родственниках, проживающих по другим адресам.

Пример 4:

Описание предметной области каждый берет по своему номеру варианта (см. Приложение А).

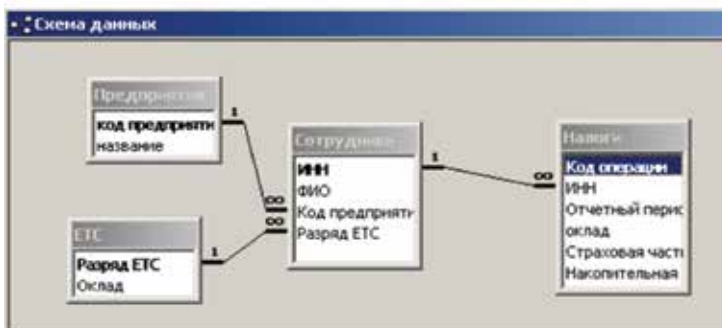
Контрольные тесты:

1. К классу инфологических моделей относятся:

- а) документальные;
- б) модель сущность-связь (ER-модель);
- в) основанные на файловых структурах.

2. СУБД Access поддерживает связи типа:

- а) один к одному;
- б) один ко многим;



в) многие ко многим.

3. На рисунке представлена схема данных базы данных.

4. В таблице «Предприятие» поле «Код предприятия» имеет тип Счетчик. Каким может быть тип поля «Код предприятия» в таблице «Сотрудники»?

- а) числовым;
- б) текстовым;
- в) логическим;
- г) любым.

5. В каком диалоговом окне создают связи между полями таблиц базы данных:

- а) таблица связей;
- б) схема связей;
- с) схема данных;
- д) таблица данных;

6. отчёт данных Структура файла реляционной базы данных (БД) меняется:

- а) при добавлении одной или нескольких записей;
- б) при удалении диапазона записей;
- с) при изменении любой записи;
- д) при уничтожении всех записей;
- е) при удалении любого поля.

7. Содержит ли какую-либо информацию таблица, в которой нет полей?

- а) содержит информацию о структуре базы данных;
- б) не содержит ни какой информации;
- с) таблица без полей существовать не может;
- д) содержит информацию о будущих записях;
- е) всё выше верно.

8. Какое поле можно считать уникальным?

- а) поле, значения в котором не могут повторяться;
- б) поле, которое носит уникальное имя;
- с) поле, значение которого имеют свойство наращивания;
- д) поле, значения в котором повторяются;
- е) всё выше верно.

9. На рисунке представлена таблица базы данных, реализованной в СУБД Access. Какие из полей данной таблицы могут быть ключевыми?

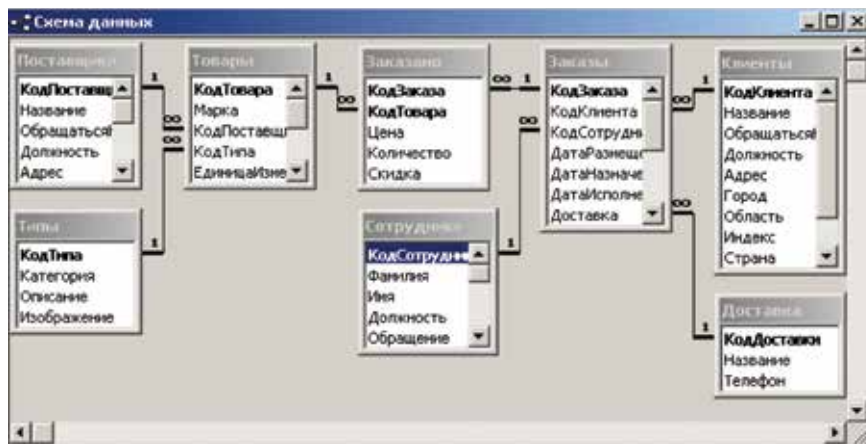
Таблица1 : таблица				
	Поле1	Поле2	Поле3	Поле4
	55	14	Иванов	100
	58	15	Петров	100
►	55	27	Сидоров	180

- а) Поле1;
б) Поле2;
в) Поле3;
г) Поле4.

10. Укажите соответствие понятий модели «сущность-связь» и понятий СУБД Access:

- | | |
|------------------------|---------------------------------------|
| 1. сущность; | а) запись в таблице (строка таблицы); |
| 2. экземпляр сущности; | б) поле таблицы (столбец таблицы); |
| 3. атрибут сущности. | в) таблица. |

11. На рисунке представлена схема данных базы данных. Между таблицами «Товары» и «Заказы» установлена связь типа:

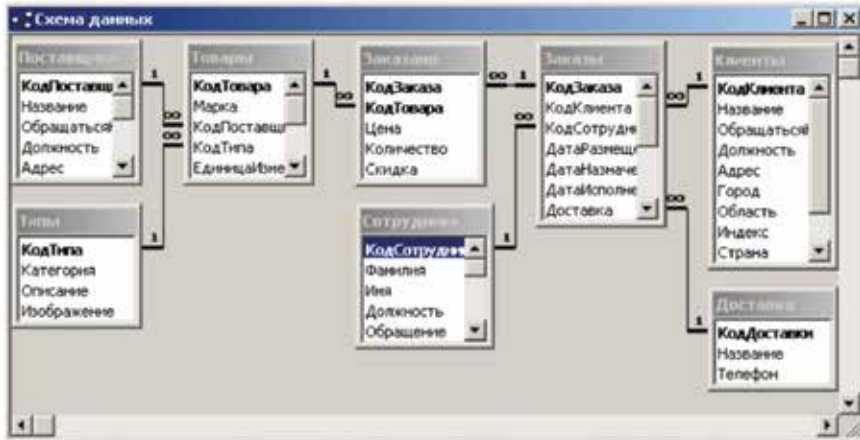


- а) один к одному;
б) один ко многим;
в) многие ко многим;
г) нет связи.

12. На рисунке представлена схема данных базы данных. Связи какого типа установлены между сущностями в модели, реализованной в данной базе?

- а) один к одному;

- б) один ко многим;
- в) многие ко многим.



13. Укажите соответствие между сущностями «преподаватель» и «студент», «студент» и «учетная карточка» в ER-модели и типом связи:

- 1. преподаватель проводит занятия — студент посещает занятия;
 - 2. преподаватель руководит дипломными проектами — студент пишет дипломный проект;
 - 3. на студента заводится учетная карточка.
- а) один к одному;
 - б) один ко многим;
 - в) многие ко многим.

14. В реляционных базах данных используется:

- а. несвязанные между собой таблицы;
- б. одна таблица, содержащая все данные;
- с. таблицы, между которыми устанавливаются связи;
- д. списки однородных данных.

Контрольные вопросы:

- 1. Назовите этапы проектирования.
- 2. Какие модели называются инфологическими, реляционными?
- 3. Чем характеризуется каждая сущность?
- 4. В модели «сущность-связь» что понимают под сущностью, атрибутом?

5. Типы связей «один-к-одному» и «один-ко-многим»: как они реализуются в реляционных базах данных?
6. Если один экземпляр одной сущности может быть связан только с одним экземпляром другой сущности, то это связь какого типа?
7. Что в реляционной модели является основной структурой данных?
8. Модели, получаемые из модели «сущность — связь»?
9. Какие модели создаются после инфологического проектирования?
10. Первичным ключом (ключевым атрибутом) отношения Продажи (Код, Код_товара, Количество, Дата_продажи, Цена) можно выбрать какой атрибут?
11. Первичными ключами отношения Студент (№группы, ФИО, № паспорта, Адрес, № зачетки) можно выбрать какие атрибуты?
12. Тип СУБД, к которому принадлежит Access?
13. Какие виды связей, поддерживают СУБД Access?
14. Сколько связей можно создать между двумя таблицами?
15. Могут ли разные таблицы содержать одинаковые поля? Если да, то в каких случаях?
16. С какой целью подключают условие «Обеспечение целостности данных» при связывании таблиц в Access?
17. С какой целью подключают условие «Каскадное удаление связанных данных» при связывании таблиц в Access?
18. С какой целью подключают условие «Каскадное обновление связанных данных» при связывании таблиц в Access?
19. В каком окне происходит связывание таблиц? Какую команду надо выполнить для его открытия?
20. По каким полям происходит связывание двух таблиц? Как они называются и какие требования к ним предъявляются?

ЛАБОРАТОРНАЯ РАБОТА № 3

Создание формы в режиме автоформы и с помощью мастера (4ч)

Цель работы: научиться создавать формы для работы с данными в режимах автоформы и с помощью мастера форм.

После построения таблиц в режиме конструктора необходимо ввести данные в таблицу (*записи*). Это можно делать как в самой таблице в режиме таблицы, так и используя объект БД *Формы*. *Форма* представляет собой *бланк для заполнения таблиц*. Она создает комфорт при работе с Access, защищает базу от действий неквалифицированных пользователей и скрывает конфиденциальную информацию. Кроме того, формы используются при получении ответов на запросы.

Формы являются основным средством организации интерфейса пользователя в MS Access. Хорошо разработанные формы позволяют работать с приложением даже неподготовленному пользователю.

Формы позволяют задавать два различных набора параметров: один — для вывода формы на экран, а другой — для вывода на печать.

Создать форму можно с помощью инструментов группы *Формы* вкладки *Создание* (рис. 16).



Рис. 16. Группа *Формы* вкладки *Создание*

1. Создание формы с помощью инструмента *Форма*.
2. Создание с помощью инструмента *Мастер форм*.
3. Создание с помощью инструмента *Конструктор форм*.

4. Создание разделенной формы при помощи инструмента *Разделенная форма*.

5. Создание формы, в которой отображается несколько записей при помощи инструмента *Несколько элементов*.

6. Создание формы при помощи инструмента *Пустая форма*.

7. Создание формы в виде *сводной таблицы*.

8. Создание формы в виде *сводной диаграммы*.

Для создания формы с помощью инструмента **Формы** в области *Все объекты Access* необходимо выбрать таблицу с данными, которые должны отображаться в форме, и выполнить щелчок мышью по пиктограмме этого инструмента в группе **Формы**. MS Access 2013 автоматически создаст форму и отобразит ее в *режиме макета*. В *режиме макета* можно внести изменения в структуру формы при одновременном отображении данных. При необходимости можно настроить размер полей в соответствии с данными.

С помощью **Мастера форм** можно создавать формы как на основе одной таблицы или запроса, так и на основе нескольких связанных таблиц.

Мастер форм разбивает процесс создания формы на несколько этапов. На каждом из них выбираются определенные параметры в предложенном диалоговом окне.

1. На первом этапе выбирается объект (таблица или запрос), на основании которого будет создаваться форма. Здесь же определяются поля, которые будут отображены в создаваемой форме.

2. На втором шаге определяется внешний вид формы. Существует шесть видов форм в зависимости от представления на них данных: в один столбец, ленточный, табличный, выровненный, сводная таблица и сводная диаграмма.

3. На третьем шаге выбирается стиль формы.

4. На четвертом шаге требуется указать название формы и выбрать опцию переключателя *Открыть форму* для просмотра и ввода данных. Если требуется внести изменения в форму, созданную мастером, то можно выбрать *Изменить макет формы*.

Если на одном из этапов сделана ошибка и необходимо изменение уже выбранных параметров, то Мастер всегда позволяет вернуться к предыдущему шагу.

Задания:

1. Автоматическое создание формы.

• Открыть созданную на практическом занятии БД «СУД». В **Окне базы данных** выбрать вкладку **Создание группа Формы**.

- Выделить таблицу *Районные суды* и выполнить команду **Создание – Формы – Форма**. В результате появится Макет автоматически созданной формы в виде столбца и с полями из подчиненной таблицы *Судьи*.

- Переключитесь в Режим формы. Для этого на вкладке **Работа с макетами форм – Конструктор** выполните команду **Режим – Режим формы**.

- Ввести 5 записей для судов, и для каждого суда по 2-3 фамилии судей.

Для этого заполните только поля **район** и **адрес**, а **Код суда** заполнится автоматически порядковыми номерами.

Затем перейдите ниже в таблицу и введите информацию о 2-х или 3-х судьях, работающих в данном суде.

После этого для перехода к новой записи внизу формы выполните команду *Следующая запись*.

- Закрыть форму.

2. Создание формы с помощью мастера.

- Для этого выполните команду **Создание – Формы – Другие формы – Мастер форм**.

- В качестве **источника данных** выбрать таблицу *Дела*. В отличие от *Формы*, *Мастерформ* работает более гибко: есть возможность включить в проектируемую форму *произвольный набор полей* из различных выбранных таблиц.

- Используя кнопку >>, перенести в форму все поля таблицы *Дела*.

Обратить внимание, что в таблице присутствует поле *Код судьи*, но нет фамилии судьи, который ведет это дело. Чтобы добавить его в форму:

- выбрать таблицу *Судьи*.

- Выделить поле **Фамилия** в левой половине окна и **Код судьи** в правой половине, нажать >. Тогда поле *Фамилия* судьи будет помещена после поля *Код судьи* в создаваемой форме. Нажать кнопку **Далее**.

- В следующем окне **Вид представления данных** выбрать *Судьи* и далее следовать инструкциям мастера до завершения создания формы.

В результате получится вложенная форма, где основная форма – *Судьи*, а подчиненная – *Дела*. Для того чтобы заносить данные о делах, необходимо сначала открыть запись одного из судей в основной форме, а потом внести записи о делах этого судьи в подчиненную форму.

- Для каждого из судей занести по 2-3 записи о делах (не дублировать *№ дела*, так как ключевые поля не повторяются). Для перехода в другое поле используйте клавишу **Tab**. Сохранить форму и закрыть.

Обратить внимание, что в предыдущем случае (п.1) была создана одиночная форма, хотя в ней также использовались поля из двух таблиц. Но в предыдущем случае в качестве основной использовалась таблица *Районные суды*. Связь между таблицами *Судьи* и *Районные суды* имеет тип отношения «многие-к-одному», то есть каждой записи таблицы *Судьи* соответствует одна запись таблицы *Районные суды*. А в последнем случае (п.2) связь *Судьи* – *Дела* имеет тип отношения «один-ко-многим», то есть одной записи таблицы *Судьи* может соответствовать несколько записей таблицы *Дела* (один судья ведет несколько дел).

3. С помощью мастера форм создать форму **Дела1**, как в п.2, но вид представления не выбирать (рис. 17).

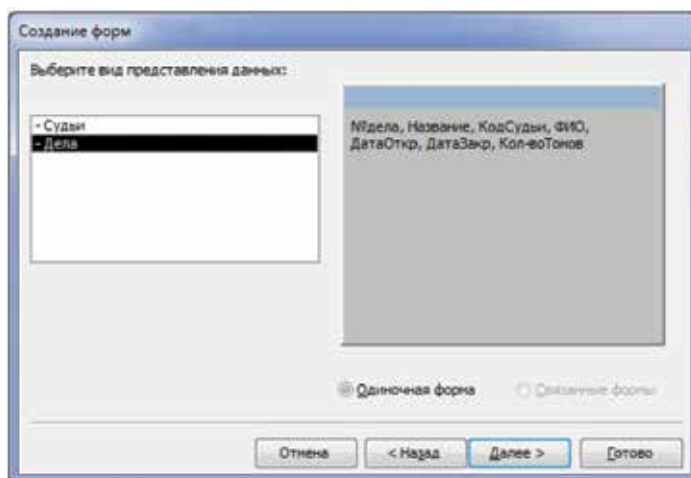


Рис. 17. Диалоговое окно *Создание форм*

4. С помощью мастера форм создать форму *Помощники судьи* (аналогично п.2).

5. Открыть таблицы *Районные суды*, *Судьи*, *Помощники судьи* и *Дела* посмотреть внесенные данные. Показать результат работы преподавателю. Закрыть таблицы. Закрыть базу данных.

6. Создать формы для просмотра выборочных данных:

- а) *ФамилияСудьи, Специализация, Район;*
- б) *ФамилияСудьи, № дела, ДатаОткр.*
- с) *Район, №Дела, специализация.*

Выводы:

- Для ввода данных в БД можно пользоваться не только таблицами, но и формами. Это наиболее приемлемый способ для конечного пользователя.

- При создании автоматической формы с/п инструмента *Форма* в нее включаются все поля выбранной таблицы. Если она является главной, то в нее включаются и все поля из подчиненной таблицы. При этом будут созданы две формы *основная* и *подчиненные* формы.

- Создание формы с помощью *Мастера форм* является более целесообразным, так как существует возможность включить в проектируемую форму *произвольный набор полей* из *различных* выбранных таблиц.

- Формы могут быть *одинокими* и *вложенными* (составными). У вложенных форм существуют *основная* и *подчиненные* формы. Подчиненная форма сохраняется как самостоятельная форма.

Задания для самостоятельной работы

Создать формы, используя все вышеописанные способы и заполнить их данными для:

- 1) БД «Соцзащита», созданной в лаб. работе №1
- 2) БД «Факультет», созданной в лаб. работе №1
- 3) БД по номеру варианта (см. Приложение А).

ЛАБОРАТОРНАЯ РАБОТА № 4

Создание и редактирование формы в режиме конструктора форм (4 ч)

Цель работы: *освоить основные приемы редактирования в режиме конструктора ранее созданной формы; работа с окном свойств.*

1. Открыть БД «СУД».
2. Открыть форму *Дела 1* в режиме конструктора.

Перемещение *Полей* и их имен (*Надписей*) (рис.18) по форме производится следующим образом:



Рис. 18. Вид *Надписи* с *Поле*м в режиме *Конструктор форм*

- Выделить *Поле* с *Именем (Надписью)* щелчком мыши (щёлкать между полем и именем до появления стрелки). Вокруг него появятся маркеры перемещения  и изменения размеров . Перемещать поле можно вместе с привязанным к нему именем или отдельно от него.
- Для перемещения поместить указатель мыши на квадратик, находящийся в левом верхнем углу элемента (рис. 19). Указатель мыши в виде четырех направленной стрелки  позволяет перемещать объект.
- Нажать кнопку мыши и, удерживая ее, буксировать поле или его имя в нужное место в форме. Затем отпустить кнопку мыши.
- Для изменения надписи, связанной с полем необходимо выполнить на ней двойной щелчок мышью и выполнить необходимые изменения. Затем закрыть окно.
- Для изменения размеров поместить курсор на размерные маркеры, при этом курсор примет вид двунаправленной стрелки . Нажать



Рис. 19. Вид квадратика у Поля (Надписи) в режиме Конструктор форм, используемый для перемещения только Поля (Надписи)

кнопку мыши, буксировать в нужном направлении, затем отпустить кнопку мыши.

- Для удаления поля выделить его, нажать клавишу Delete или другим способом.

Окно свойств

Каждая форма и элемент управления имеют набор свойств, с помощью которых можно изменить вид или характеристики данного объекта. Эти свойства доступны для просмотра и изменения в окне свойств. Для того чтобы открыть окно свойств, следует в контекстном меню выбрать команду **Свойства** (или **Конструктор – Сервис – Страница свойств**). При выборе элемента **Страница свойств** на экране появится окно свойств формы. Существуют различные категории свойств формы, которые представлены на следующих пяти вкладках.

- 1) **Макет** – свойства, которые принадлежат способу отображения объекта.
- 2) **Данные** – свойства, которые принадлежат данным объекта, независимо от того, каким способом они получены.
- 3) **События** – свойства, которые принадлежат событиям и связанным с ними процедурам.
- 4) **Другие** – свойства, которые принадлежат характеристикам объекта или его признакам.
- 5) **Все** – все категории и свойства объекта.

Следует обратить внимание, что в окне *Свойства* отображаются свойства выделенного в данный момент объекта. В строке заголовка окна свойств показано, какой элемент выбран.

3. Выбрать по очереди различные управляющие элементы формы и **Область данных** (центральную часть формы).

4. Рассмотреть **Окно свойств**. Окно имеет пять вкладок: **Макет**, **Данные**, **События**, **Другие** и **Все**.

На первой вкладке собраны свойства, относящиеся к внешнему виду объекта (ширина, высота, оформление и т.п.). На второй вкладке, свойство **Данные** позволяет задать *источник данных* для элемента управления (то есть с каким полем таблицы связан элемент управления).

5. В режиме конструктора отредактировать внешний вид формы **Дела1**: поменять место расположения надписей согласно образцу, представленному на рис. 20.

Не редактировать текст *Полей!* Только *Надписи*. (Примечание. в случае ошибочного редактирования после переключения в режим формы в полях ввода будут стоять символы **#Имя?**. В этом случае закрыть форму, не сохраняя внесенных изменений и повторить редактирование).

Рис. 20. Образец *Формы*

6. Для полей *№ дела* и *Название дела* установить высоту 0,8 см, цвет фона — желтый, используя на вкладке Конструктор команду **Свойства/Макет**.

7. Для надписей *№дела* и *Название дела* задать тип границы — точечная, цвет текста - фиолетовый.

8. Посмотреть свойство **Данные** для поля *Дата открытия* и для надписи *Датаоткрытия*. В чем отличие? Почему нельзя редактировать текст поля на форме?

9. Переключиться в режим формы и ввести 3 дополнительные записи по делам. Вернуться в режим конструктора.

10. Добавить в заголовок формы эмблему, задать пунктирную границу для заголовка.

При создании или открытии формы в режиме *Конструктора* на экране появятся три дополнительные вкладки *Конструктор*, *Макет* и *Упорядочить*.

Элементы управления на вкладке *Конструктор – Инструменты Конструктора Форм* используется для размещения объектов в форме. Наиболее используемые элементы формы и соответствующие им кнопки представлены на рис.21 и в таблице 4.



Рис. 21. Инструменты вкладки Конструктор

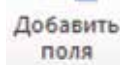
Таблице 4

	Надпись. Размещение в форме произвольного текста.
	Поле. Размещение в форме данных из соответствующего поля базовой таблицы/запроса, вывод результатов вычислений, а также прием данных, вводимых пользователем.
	Выключатель. Создание выключателя, кнопки с фиксацией.
	Переключатель. Создание селекторного переключателя.
	Флажок. Создание контрольного переключателя.

	Поле со списком. Размещение элемента управления, объединяющего поле и раскрывающийся список.
	Список. Создание списка, допускающего прокрутку. В режиме формы выбранное из списка значение можно ввести в новую запись или использовать для замены уже существующего значения.
	Кнопка. Создание командной кнопки, позволяющей осуществлять разнообразные действия в форме (поиск записей, печать отчёта, установка фильтров и т.п.).
	Гиперссылка. Создание ссылки на веб-страницу, рисунок, адрес электронной почты или программу.

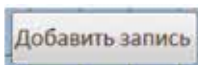
При размещении в форме полей из базовой таблицы/запроса удобно

пользоваться командой . Он представляет собой окно, содержащее



перечень всех полей базовой таблицы/запроса. Чтобы поместить поле в форму, достаточно просто перетащить его левой кнопкой мыши из списка полей в нужный раздел формы.

11. Добавить в форму элемент *Кнопка*



- Выбрать **Кнопка**  (**Конструктор — Элементы управления**).
- В диалоговом окне **Создание кнопок** выбрать категорию — *обработка записей, действие — добавить запись* (рис. 23). Нажмите **Далее**.
- Выберите **Текст** (рис. 24). Нажмите **Далее, Готово**.

12. Добавить кнопку



для закрытия формы (аналогично п.11,

только выбрать категория — *работа с формой, действие — закрыть форму*).

13. Отредактировать другие формы. Показать работу преподавателю.

Выводы:

- Форму, созданную с помощью *Мастера форм*, можно отредактировать в режиме *Конструктора* для наиболее выгодного отображения данных.

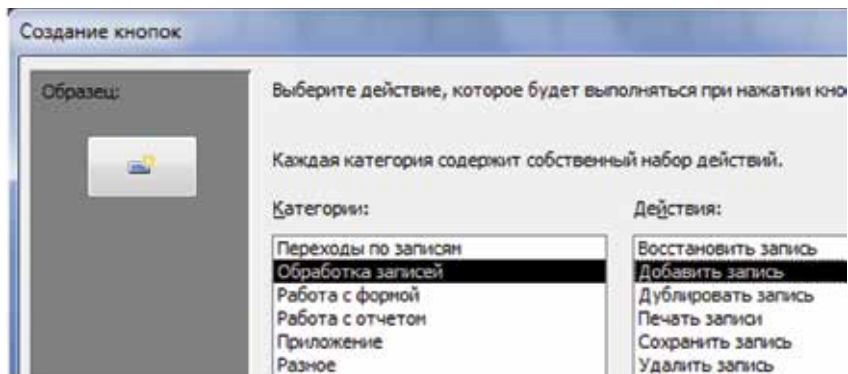


Рис. 23. Окно *Создание кнопок* для выбора *Категории* и *Действия* из выбранной категории

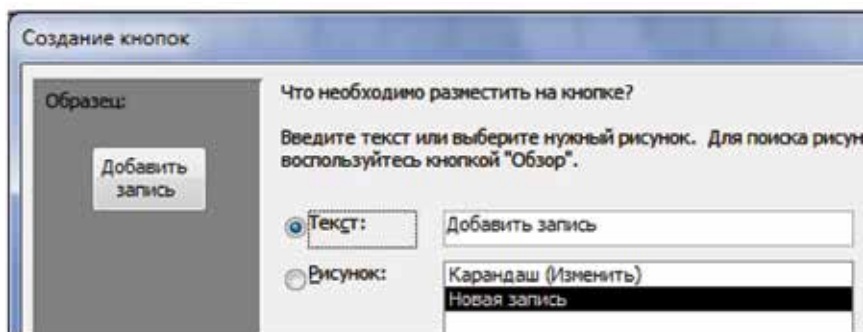


Рис. 24. Окно *Создание кнопок* для выбора подписи на кнопке *Текста* или *Рисунка*

• Каждая форма и элемент управления имеют набор свойств, с помощью которых можно изменить вид и характеристики данного объекта. Эти свойства доступны для просмотра и изменения в окне *Свойства*.

Задания для самостоятельной работы

Отредактировать все формы для:

- 1) БД «Соцзащита», созданной в лаб. работе №1
- 2) БД «Факультет», созданной в лаб. работе №1
- 3) БД по номеру варианта (см. Приложение А).

Контрольные вопросы

1. *С какой целью создаются формы? Назовите виды форм MS Access.*
2. *Какие инструменты создания форм существуют?*
3. *Для создания каких форм предназначена команда Мастер форм?*
4. *Чем отличаются изученные способы создания форм друг от друга?*
5. *Какие виды форм, созданные автоматически, существуют?*
6. *Перечислите основные разделы форм. Для чего каждый из них предназначен?*
7. *Для чего используется раздел Данные в форме?*
8. *Как разместить в форме элемент управления?*
9. *Какие элементы можно добавить в форму?*
10. *Как используется при создании форма Кнопка?*
11. *Как изменить порядок размещения элементов в форме?*
12. *Как построить форму по двум связанным таблицам?*
13. *Можно ли редактировать Поля в режиме Конструктор форм?*
14. *Какие свойства элементов форм можно настроить?*
15. *Как добавить в режиме Конструктор форм новые поля из таблиц?*

ЛАБОРАТОРНАЯ РАБОТА № 5

Создание запросов на выборку (6 ч)

Цель работы: *ознакомиться с основными типами запросов на выборку и научиться создавать запросы в режиме конструктора запросов.*

При работе с таблицами можно в любой момент выбрать из базы данных необходимую информацию с помощью запросов.

Запрос — это обращение к БД для поиска или изменения в базе данных информации, соответствующей заданным критериям.

В Access существует два основных типа запросов: ***запросы на выборку, запросы на изменение.***

Запрос на выборку выполняет отбор данных из одной или нескольких таблиц по заданным пользователем критериям, причём в самой базе данных никаких изменений не происходит.

К **запросам на выборку** относятся:

- групповые запросы;
- перекрёстные запросы;
- параметрические запросы.

Групповые запросы используются тогда, когда необходимо выполнение групповых операций, например, определение количества дел у конкретного судьи. При создании группового запроса используется строка ***Групповая операция*** в режиме конструктора запросов.

Для выполнения групповых операций имеются следующие четыре категории функций, которые позволяют производить вычисления на основании значений полей:

• **группировка** — используется для указания поля, которое будет применяться в качестве группирующего;

• **итоговые функции** — их 9:

Sum, Avg (среднее), Min, Max, Count (количество), StDev, Var, First (первая), Last (последняя). Каждая из этих функций выполняет вычисления над полями;

• **выражение** — используется для указания на создание поля, значения которого будут вычисляться;

• условие — для указания поля, записи которого выбираются на основании заданного условия отбора.

Перекры́стные запросы — специальный вид группового запроса, представляющий собой выборку данных, записанных в виде двухмерной матрицы, в качестве заголовков строк и столбцов которой используются сгруппированные значения заданных полей или выражения.

Параметрические запросы — вид запросов, при котором выбор информации из таблиц осуществляется на основании задания условий отбора записей (параметров). Вводимые параметры предварительно необходимо описать в окне, вызываемом при выполнении пункта меню *Запрос-Параметры*.

Запрос на изменение — это запрос, который позволяет выполнять более сложные действия по сравнению с выборкой определённых групп записей. Этот запрос можно представить как запрос на выборку, который должен выполнять определённые действия над выбранной группой записей.

Основные запросы на изменение:

- *запрос на создание таблицы* — позволяет создать новую таблицу с помощью условия отбора записей из других таблиц;
- *запрос на добавление* — позволяет добавить отобранные записи из таблицы или запроса в конец другой таблицы;
- *запрос на обновление* — позволяет изменить в таблице группу записей данных, удовлетворяющих определённым условиям;
- *запрос на удаление* — позволяет удалить группу записей данных, удовлетворяющих заданным условиям.

Запросы могут быть созданы двумя способами: с помощью мастера создания запросов и с помощью конструктора.

Задания:

1. Создание простых запросов. Простые запросы формируются с помощью мастера и не содержат каких-либо критериев, условий, фильтров.

- Открыть БД «СУД» .
- Перейти на вкладку **Запросы**. Выбрать **Мастер запросов - Простой запрос**. На экране появится окно **Создание простых запросов**.
- Из таблицы *Районные суды* внести в запрос поля *Код суда* и *Район*, из таблицы *Судьи* — *Фамилия судьи* и *Специализация*.
- Завершите создании запроса, следуя инструкции, сохранив его под именем *Запрос 1*.

2. Создание запроса в режиме конструктора. Создание запроса с помощью конструктора – это основной способ формирования запросов.

- Перейти в **окно БД.** Перейти на вкладку **Запросы.** Выбрать **Конструктор запросов**

- В окне **Добавление таблицы** добавить таблицы *Районные суды* и *Судьи*. Закрыть окно **Добавление таблицы.** В результате на экране останется окно конструктора запросов (см. рис.10).

Чтобы поместить поля в бланк запроса, надо дважды щелкнуть на соответствующем поле, либо поставить курсор в строку Поле в нижней части окна и из списка полей выбрать нужные (см. рис.25). Как и в предыдущем случае, включить в бланк запроса поля *Код* и *Район*, а также *Фамилия* судьи и *Специализация*. Нажать кнопку запуска запроса

 . В результате получится таблица, полностью совпадающая с

Выполнить

п.1. Сохранить запрос под именем **Запрос2** .

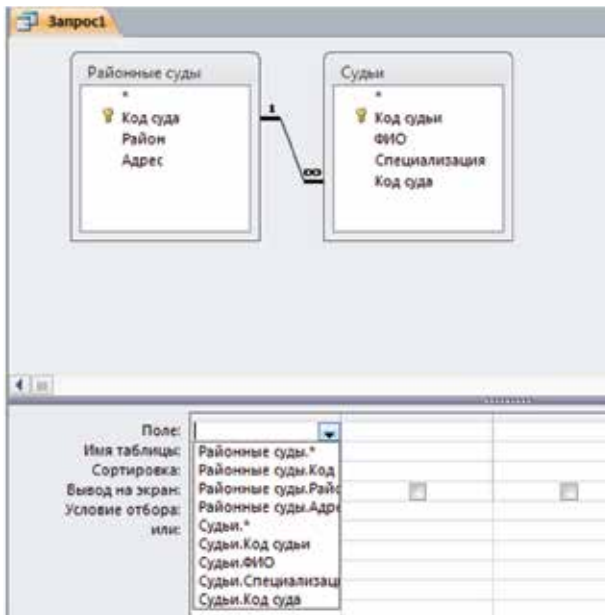


Рис. 25. Окно конструктора запросов

- Заккрыть БД «СУД».

3. Создать новую БД *Факультет*.

4. Создать таблицу *Абитуриент*. Её поля с типами данных и свойствами указаны в нижеприведённой таблице.

№п/п	Фамилия	Имя	Ср. балл аттестата	Сумма баллов вступит. Экзамены
Ключевое, счетчик	Текстовое	Текстовое	Числовой, размер поля — одинарное с плавающей точкой	Числовой, длинное целое

6. Создать форму для этой таблицы согласно образцу (см. рис. 26).

• Для этого в *окне* базы данных на вкладке **Формы** выбрать **Конструктор форм**.

• Далее щелкнуть мышью на кнопке *Добавить поля* и переместить все поля таблицы *Абитуриент* из **Списка полей** в **Область данных**.

• Отредактировать форму согласно образцу (рис. 26).

Рис. 26. Образец формы для таблицы *Абитуриент*

• Ввести 10 записей (несколько имен и фамилий должны начинаться на буквы «П» или «Р», средний балл должен лежать в интервале от 4 до 5, сумма баллов от 10 до 20).

• Сохранить и закрыть форму.

7. Создание запросов с использованием групповых операций. *Запросы с использованием групповых операций* предназначены для первичной обработки полученной информации — ее суммирования, осреднения и т.п.

5.1. Подсчитать количество записей в таблице абитуриент.

- В Окне базы данных выбрать Запросы, Создание в режиме конструктора добавить таблицу *Абитуриент*.

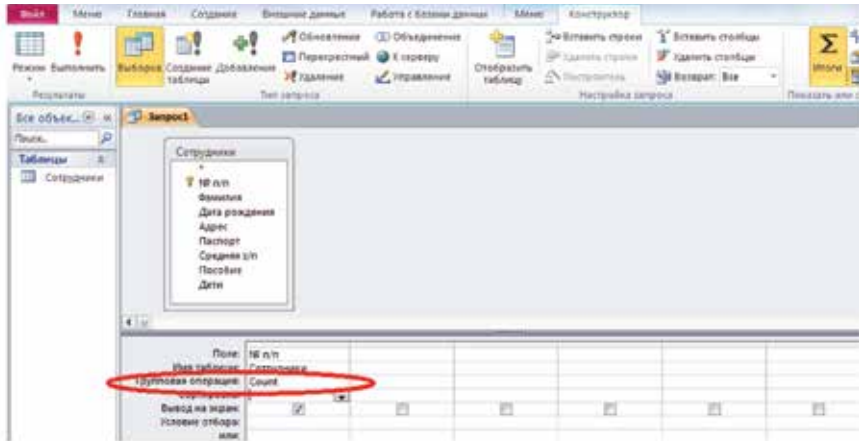


Рис. 27. Запрос в режиме Конструктора

- Для того, чтобы опция **Групповые операции** стала доступна, необходимо щелкнуть на кнопке Σ на вкладке Конструктор. Тогда в

Итого

бланке запроса появится дополнительная строка **Групповая операция**.

- Включить в бланк запроса поля: $N/n/n$,
- Щелкнуть в строке **Групповые операции**, развернуть стрелку прокрутки и выбрать функцию **Count**, которая *подсчитывает количество записей*, возвращаемых запросом (рис.27).

- Запустить запрос, закрыть его, сохранив под именем Запрос5.

5.2. Рассчитать среднее значение поля *Средний балл* и поля *Сумма баллов*. Для этого для полей *Средний балл* и *Сумма баллов* за экзамены выбрать функцию **Avg**, вычисляющую *среднее арифметическое* набора чисел поля. Сохранить под именами Запрос5_1.

5.3. Найти **минимальное значение (Min)** поля *Сумма баллов*. Сохранить под именами Запрос5_2.

5.4. Показать последнюю запись (**First**) в таблице *Абитуриент*. В запрос должны быть включены все поля таблицы. Сохранить под именем *Запрос5_3*.

8. Создание запросов с условиями отбора. *Условия отбора* - это ограничения, которые пользователь накладывает на запрос для определения записей, с которыми он будет работать.

Чтобы установить условие отбора для поля в бланке запроса, в ячейку **Условие отбора** для данного поля необходимо ввести выражение. При написании выражения необходимо учитывать несколько общих правил.

- ♦ При работе с текстовыми данными их заключают в кавычки (= "Иванов").

- ♦ При работе с датами используют символ # (<#12.03.98#).

- ♦ Числа набирают без каких-то дополнительных символов.

- ♦ Пустые позиции обозначаются ключевым словом Null.

6.1. Отбор записей, содержащих значения, которые больше, меньше, равны, не равны или находятся между указанными значениями.

а) Построить запрос к таблице *Абитуриент*. Включить все поля. Для поля *Средний балл* ввести условие **>4,5**. Запустить запрос.

б) Вернуться в режим конструктора. Удалить это условие. В поле *Фамилия* ввести условие **>=Иванов** (Все фамилии, начиная с «Иванов» и до конца алфавита). Запустить запрос.

в) Вернуться в режим конструктора. Удалить условие. В поле *№* задать условие **Between 4 And 8** (числа в диапазоне от 4 до 8). Запустить запрос. Сохранить последний запрос под именем *Запрос6_1*. Закрыть.

6.2. Отбор записей, начинающихся или кончающихся определенными наборами символов. Для этого используются подстановочные знаки:

|| * – заменяет *любое* количество цифр или символов,
 || ? – заменяет *один* символ.

Построить запрос к таблице *Абитуриент*. Включить все поля. Используя эти символы, вывести (по очереди):

а) все записи с фамилиями, начинающимися на букву «Р»;

б) все записи с именами из 4 букв;

в) все записи, в которых средний балл аттестата заканчивается на цифру «5». Сохранить последний запрос под именем *Запрос6_2*. Закрыть.

6.3. Отбор записей, кроме определенных. Для отбора записей кроме определенных используется ключевое слово **Not**.

Построить запрос к таблице *Абитуриент*. Включить все поля. Вывести:

а) фамилии, кроме начинающихся с буквы «П»: **Not П***;

б) непустые позиции в поле *Имя*: **Not Null**.

Сохранить последний запрос под именем *Запрос 6_3*. Заккрыть.

6.4. Запросы с дополнительными условиями отбора. Для одного и того же поля или для нескольких полей можно ввести *дополнительные условия отбора*.

Если выражения вводятся в несколько ячеек **Условие отбора**, то они автоматически комбинируются с помощью операторов **And** или **Or**.

Если выражения находятся в *разных* ячейках, но в *одной* строке, то автоматически используется оператор **And**, что означает, что будут возвращены только записи, отвечающие условиям отбора, указанным *во всех ячейках*.

Если же выражения находятся в *разных* строках бланка запроса, то автоматически используется оператор **Or**, что означает, что возвращены, будут записи, отвечающие *любому* из условий отбора, указанных в ячейках.

Выполнить следующие запросы, сохраняя каждый из них под именами *Запрос6_41*, *Запрос6_42*,....

1) Вывести записи, у которых *Средний балл* выше 4,6 и *Сумма баллов*>16.

2) Вывести записи, у которых *Средний балл* выше 4,6 или *Сумма баллов*>16.

3) Вывести записи, у которых *Фамилия* начинается на П или на Р.

4) Вывести записи, у которых *Фамилия* начинается на П или *Имя* на Р.

5) Вывести записи, у которых (*Фамилия* начинается на П и *Средний балл*>4.6) или (*Имя* начинается на Р и *Сумма баллов*>16).

6) Заккрыть БД Факультет.

9. Запросы с участием нескольких связанных таблиц.

- Открыть БД «СУД» .

- Создать запрос в режиме конструктора. В запрос добавить все таблицы (*Дела*, *Судьи*, *Районные суды*).

- В запрос включить следующие поля: *Название дела*, *Фамилия судьи*, *Название района*, *Количество Томов*.

- Запустить запрос. Сохранить запрос под именем *Запрос7*.

10. Запросы, создаваемые на базе других запросов.

- Создать запрос на базе *Запроса 7* (в качестве исходной таблицы выбрать *Запрос 7*).

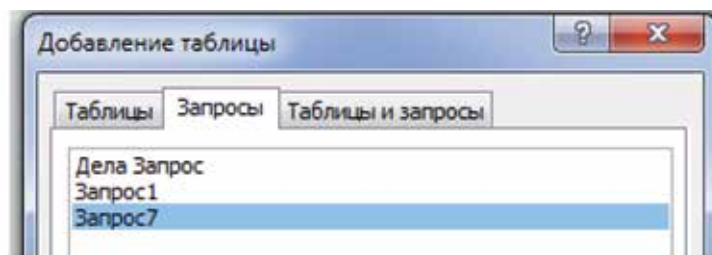


Рис. 28. Диалоговое окно Добавление таблицы (запросов)

- Сосчитать количество записей в запросе.
 - Запустить запрос. Сохранить под именем *Запрос8*. Заккрыть.
11. Открыть *Запрос7* в режиме конструктора. Выполнить следующие запросы, сохраняя каждый из них под именами *Запрос9_1*, *Запрос9_2*...
- 9.1. Показать записи, у которых поле *Количество Томов*. не пустое.
 - 9.2. Показать дела, рассматриваемые в Вахитовском районном суде.
 - 9.3. Показать дела, рассматриваемые в Вахитовском районном суде с указанием фамилии судьи, рассматривающего данное дело.
12. Подсчитать в каждом районном суде число работающих судей.
- Создать запрос в режиме Конструктора. В запрос добавить таблицы *Судьи* и *Районные суды*.
 - В запрос включить следующие поля *Район*, *Фамилия*.
 - Подключить опцию Групповая операция.
 - Для поля *Район* в строке Групповые операции, выбрать функцию Группировка, для поля *Фамилия* -- функцию Count.
- Функцию Группировка выбирается для тех полей, записи которых в таблице или запросе повторяются.*
- Запустить запрос, закрыть его, сохранив под именем Запрос 10.
13. Подсчитать сколько дел у каждого судьи (аналогично п.10). Сохранить под именем Запрос 11.
14. На основе предыдущего Запроса 11, создать запрос 12, вычисляющий среднее количество дел.
15. Подсчитать количество судей с *уголовной специализацией*.
- Сначала создать запрос 13, выводющий список судей с *уголовной специализацией*.
 - Затем на основе созданного запроса 13, создать запрос 13_1, подсчитывающий количество судей (записей в запросе 13).

Задания для самостоятельной работы

1. Создать итоговые запросы для БД «Суд»:

- а) Вывести Дела, которые были закрыты после 1 января 2015г.
- б) Закрыть базу данных.
- с) Рассчитать среднее количество томов в делах, используя функцию Avg, вычисляющую среднее арифметическое значений поля.
- д) Показать последнюю запись таблицы Дела (в запрос включить все поля таблицы), используя функцию Last.
- е) Определить минимальное количество томов в деле.
- ф) Определить максимальное количество томов в деле.
- г) Определить суммарное количество томов во всех делах.
- х) Определить дату о открытия дела, открытого последним.
- и) Определить дату закрытия дела, закрытого раньше других.

2. Создать запросы с условиями отбора для БД «Суд»:

- а) Показать фамилии судей, работающих в Советском районе. Для этого в бланк запроса включить поле Фамилия таблицы Судьи и поле Районтаблицы Районный суд. Для поля Район ввести Условие отбора = «Советский»
- б) Показать полную информацию о делах, которые ведет судья Иванов (фамилию судьи можно взять другую).
- с) Показать полную информацию о делах, открытых после 20.03.2005, суказанием судьи, ведущего дело и района.
- д) Показать информацию о делах, открытых между 18.05.2015 и 26.05.2016 (Between #18.05.2015# And #26.06.2016#).
- е) Показать информацию о делах, в которых количество томов меньше 2.
- ф) Показать информацию о делах, в которых количество томов от 2 до 4.
- г) Показать сведения о делах, открытых в 2015 году.
- х) Показать сведения о судьях, фамилии которых начинаются с букв А(А*).
- и) Показать сведения о делах, открытых в мае 2016 г
- и) Показать сведения о делах, закрытых в сентябре или марте.
- к) Показать сведения о судьях, работающих в Советском или Вахитовском районах.
- л) Показать информацию о делах, которые ведет судья Иванов, открытых после 20.04.2016.

- м) Показать полную информацию о деле, закрытом раньше других.
- н) Показать фамилию судьи и название района, который вел дело, открытое последним.

3. Создать запросы для БД по номеру варианта (см. Приложение А).

Выводы

- Запросы являются основным средством просмотра, изменения и анализа информации, которая содержится в одной или в нескольких таблицах БД.
- В основном, запросы формируются в режиме конструктора.
- Существует несколько типов запросов: простые, с использованием групповых операций и с условиями отбора (критериями).
- Запросы с использованием групповых операций предназначены для обработки полученной информации — ее суммирования, осреднения и т.п.
- Запросы с условиями отбора используются для отбора записей по определенному критерию.

Контрольные тесты:

1. Проанализируйте информацию на картинках. Информация о каких сотрудниках будет получена в результате выполнения Запроса1?

Запрос 1 : запрос на выборку

Сотрудники

*
ИНН
ФИО
Код предприятия
Разряд ЕТС

1

∞

Предприятия

*
код предприятия
название

Поле:	ФИО	Разряд ЕТС	название
Имя таблицы:	Сотрудники	Сотрудники	Предприятия
Сортировка:			
Вывод на экран:	☒	☒	☒
Условие отбора:	>10		"ККЗ"

- а) Петров А.Т.;
- б) Смирнова М.И.;

- в) Кириллов А.Н.;
- г) Рогов Т.Т.;
- д) Пименова А.Р.;
- е) Ван П.Л.

2. Проанализируйте информацию на картинках. Информация о каких сотрудниках будет получена в результате выполнения Запроса 1?

Сотрудники: таблица

	ИНН	ФИО	Код предпри	Разряд ЕТС
+	1231	Петров А.Т.	3	12
+	1232	Смирнова М.И.	1	10
+	1234	Кириллов А.Н.	3	7
+	1236	Рыкова Т.С.	5	8
+	1238	Рогов Т.Т.	3	11
+	1239	Мешкова Т.Л.	2	9
+	12345	Пименова А.Р.	1	12
+	123780	Аникина М.И.	4	8
+	12345670	Ван П.Л.	2	13

Запрос: 1 9 10 11 12 13

Предприятия: таблица

	код предпри	название
+	1	Тасма
+	2	Мега
+	3	ККЗ
+	4	Прогресс
+	5	Фортуна

Запрос: 1 2 3 4 5

Сотрудники

ИНН
ФИО
Код предпри
Разряд ЕТС

Предприятия

код предпри
название

Поле: ФИО Разряд ЕТС название
Имя таблицы: Сотрудники Сотрудники Предприятия
Сортировка:
Вывод на экран: ☒ ☒ ☒
Условие отбора: >10 "ККЗ"

- а) Петров А.Т.;
- б) Смирнова М.И.;
- в) Кириллов А.Н.;
- г) Рогов Т.Т.;
- д) Пименова А.Р.;
- е) Ван П.Л.

3. Предположим, что некоторая база данных содержит поля ФАМИЛИЯ, ГОД РОЖДЕНИЯ, ДОХОД. При поиске по условию ГОД РОЖДЕНИЯ>1958 AND ДОХОД<3500 будут найдены фамилии лиц:

- 1) имеющих доход менее 3500, или тех, кто родился в 1958 году и позже;
- 2) имеющих доход менее 3500, и родившихся в 1958 году и позже;
- 3) имеющих доход менее 3500, и тех, кто родился в 1958 году;
- 4) имеющих доход менее 3500, и родившихся в 1959 году и позже;
- 5) имеющих доход не менее 3500, и старше тех, кто родился в 1958 году.

4. Для чего предназначен объект «запрос»?

- a) для ввода данных базы и их просмотра,
- b) для выборки и обработки данных,
- c) для хранения данных,
- d) для удаления данных из базы.

5. Каких запросов не бывает?

- a) на выборку,
- b) на изменение,
- c) на уничтожение,
- d) перекрестных,

6. итоговых. Какой тип запроса предназначен для вывода информации в виде таблицы, где на пересечении строк и столбцов также выводится информация?

- a) на выборку,
- b) на изменение,
- c) на уничтожение,
- d) перекрестный,
- e) итоговый,
- f) параметрический.

7. Функция Avg выполняет операцию:

- a) Вычисления среднего арифметического всех значений поля в каждой группе
- b) Вычисления среднеквадратичного отклонения значений полей от нормы группы
- c) Вывода на экран среднего значения арифметической прогрессии группы записей
- d) Вывода на экран среднегеометрического изменения значений группы записей

8. Функция Count выполняет операцию:

- a) Вычисления количества непустых записей в каждом поле запроса
- b) Вычисления математического ожидания для данного поля каждой группы писей

с) Вычисления дисперсии значений данного поля в каждой группе записей

д) Вычисления числа связей между таблицами, участвующими в данном запросе

9. К запросам на выборку не относятся:

а) Запросы на обновление

б) Перекрёстные запросы

с) Параметрические запросы

д) Групповые запросы

Контрольные вопросы

1. Виды запросов, используемые в СУБД Access?

2. Назовите способы создания запросов.

3. Какие групповые операции используются для подсчета абитуриентов, подавших документы на каждый факультет?

4. Для чего предназначены запросы с условием?

5. Для чего предназначены итоговые запросы?

6. Для чего предназначены запросы на изменение?

7. Какой символ заменяет один символ? Любое количество символов?

8. Какая итоговая функция подсчитывает среднее арифметическое?

9. Какая функция выводит первую запись таблицы?

10. Какой оператор предназначен для вывода информации, удовлетворяющей заданному диапазону?

ЛАБОРАТОРНАЯ РАБОТА № 6

Создание отчетов (4 ч)

Цель работы: *научиться создавать отчеты и диаграммы; научиться сохранять статическую копию отчета.*

Технология создания отчета в Access аналогична созданию формы. Разница между этими двумя объектами заключается в их целевом назначении. Если форма необходима для работы с данными на экране, то отчет предназначен для печати и представляет ценность как материальная копия документа, полученного из базы данных.

Отчёты используются для представления данных в удобном виде на экране и для вывода на печать. В них можно группировать и сортировать данные в любом порядке, получать итоговые значения, средние значения и другие статистические величины, а также помещать в него графические объекты.

Основными типами отчётов являются:

- 1) *ленточные отчёты* — данные распечатываются в виде строк и столбцов с группировкой и итогами. Их разновидности включают отчёты с итоговым и групповым суммированием;
- 2) *отчёты в столбец* — данные распечатываются в виде формы с возможным включением итогов и графиков;
- 3) *отчёты для рассылки* — создаются как шаблоны писем;
- 4) *почтовые наклейки* — создаются наклейки для конвертов.

Различия между формами и отчётами:

- главное различие между формами и отчётами — в их назначении. Если формы задуманы для ввода данных, то отчёты — для просмотра данных;
- в формах можно использовать вычисляемые поля. В отчётах вычисляемые поля формируются на основе общей группы записей, страницы записей или всех записей отчёта (итоги);
- всё, что можно сделать с формой, можно сделать и с отчётом, за исключением ввода данных;

• форму можно сохранить в виде отчёта, а затем редактировать в окне конструктора отчётов; обратное — невозможно.

В отчётах выделяют следующие разделы:

1) *заголовок отчёта* — печатается один раз в начале каждого отчёта. Здесь же может быть размещена дата создания отчёта;

2) *верхний колонтитул* — печатается вверху каждой страницы, служат обычно в качестве заголовков столбцов для итоговых данных;

3) *заголовок группы* — печатается перед обработкой первой записи группы;

4) *область данных* — печатается каждая запись таблицы или запроса;

5) *примечание группы* — печатается после обработки последней записи группы, отображает итоговые данные в записях текущей группы;

6) *нижний колонтитул* — печатается внизу каждой страницы, содержит номер страницы и вычисляемые элементы управления. Здесь могут быть подведены итоги по страницам и по группам;

7) *примечание отчёта* — печатается один раз в конце отчёта после всех записей данных и примечаний групп, содержит общий итог и другие статистические данные.

Алгоритм составления отчёта:

1. В окне базы данных выбирается закладка Отчёты.
2. Выбирается Создание отчёта с помощью мастера.
3. В окне выбора таблицы или запроса выбирается нужное.
4. Выбираются интересующие поля.
5. Если необходимо, вводятся уровни группировки и задаётся порядок сортировки.
6. Выбирается макет, ориентация листа, стиль таблицы.
7. Отчёт готов.
8. Если необходимо внести изменения в элементы отчёта, то прибегают к помощи Конструктора.

Отчеты можно формировать с помощью: *автоотчета (отчет)*, *мастера отчетов* и *конструктора отчетов*.

Задания:

1. Создание отчета с помощью автоотчета. Отчет, созданный таким образом, будет отображать *все поля и записи* базовой таблицы или запроса.

1) Создать отчет по таблице *Районные суды* БД «СУД» .

• Для этого открыть БД «СУД» .

- В окне БД в разделе **Объекты** выбрать **Отчеты**. Создать новый отчет в режиме **автоотчета** (кнопка **Отчет**).

- В качестве **источника данных** выбрать таблицу *Районные суды*. Завершить создание отчета. Сохранить отчет под именем *Районные суды*.

- Закрыть созданный отчет.

2) Аналогично в режиме автоотчета создать отчет по таблице *Судьи*. Сохранить отчет под именем *Судьи*.

2. Создание отчета с помощью мастера. В отличие от форм, в отчетах можно *группировать* и *сортировать* данные.

- С помощью *мастера отчетов* создать отчет по таблице *Дела*, включив в нее все поля, кроме поля *Код судьи*.

- На следующем шаге добавить **уровень группировки** по дате открытия.

- Далее добавить **сортировку** данных по названию дела.

- Следуя инструкциям мастера, завершить создание отчета. Сохранить его под именем *Дела*.

3. Создание отчета с помощью конструктора отчетов аналогично конструированию форм.

1) Создать в режиме конструктора отчет *Районные суды*.

- Для этого в **окне базы данных** выбрать **Создать – Конструктор отчетов**, в качестве источника данных указать таблицу *Районные суды*. ОК.

- В области **заголовка** и **верхнего колонтитула** создать надписи *Код суда*, *Район*, *Адрес* (Arial, 10пт, курсив).

- В область **данных** перетащить из **окна списка полей** поле *Фамилия судьи* (при этом помимо *поля Фамилия* будет автоматически добавлена *надпись Фамилия*; *поле* оставить, а *надпись* удалить).

- В область **нижнего колонтитула** вставить номер страницы. Для этого в группе **Элементы управления** выбрать **Номера страниц**. В открывшемся диалоговом окне отметить соответствующие образцу параметры, **Ок**.

- В области **примечания** создать **надписи** «*Отчет создан*» и свои фамилии. Установите время создания отчета. Оформление фона задать по своему усмотрению. Сохранить отчет.

4. Создание отчета на основе запроса. Для того, чтобы выводить на печать данные, полученные с помощью запросов, используются отчеты, созданные на *базе запросов*. Технология создания таких отчетов полностью совпадает с созданием отчетов на базе таблиц, только в качестве источника необходимо выбрать не таблицу, а соответствующий запрос.

Создать **автоотчет** на базе *Запроса 11* БД «СУД». Сохранить его под именем *Отчет 11*.

Выводы:

♦ Технология создания отчета в Access аналогична созданию формы. Разница между этими двумя объектами заключается в их целевом назначении. Если форма необходима для работы с данными на экране, то отчет предназначен для печати и представляет ценность как материальная копия документа, полученного из БД.

♦ Отчет, созданный с помощью *автоотчета*, будет отображать все поля и записи базовой таблицы или запроса. Он наиболее прост в построении, но наименее гибок. Создание отчета с помощью *мастера* позволяет включать данные из нескольких таблиц (запросов), а также добавлять уровни группировки и сортировки данных. *Конструктор* отчетов позволяет получить наиболее гибкую структуру отчета, но требуются определенные навыки работы.

ЛАБОРАТОРНАЯ РАБОТА № 7

Повторение пройденного материала (4ч)

Цель работы: повторить пройденный материал и систематизировать полученные знания.

Описание предметной области. Пенсионный фонд ежемесячно собирает с предприятий данные о сотрудниках. Данные включают в себя ИНН, Фамилию И.О., домашний адрес, телефон, место работы (предприятие), стаж работы, оклад. В пенсионном фонде ежемесячно фиксируются зарплата каждого работника и отчисления: на накопительную и страховую части пенсии.

1. Разработать инфологическую и даталогическую модели.
2. Создать таблицы и установить связи между ними.
3. Создать формы для ввода данных. Родительская таблица должна содержать 5 записей, остальные — по 10. Формы отредактировать и добавить элементы формы (эмблемы, кнопки,...)
4. Для таблиц использовать данные со следующими ограничениями:
 - Среди фамилий сотрудников несколько должны начинаться на букву А или Д;
 - Зарплата сотрудников от 10000 до 20000 рублей.
 - Стаж работы от 1 до 20 лет.
5. Создать и сохранить следующие запросы:
 - Вывести список сотрудников, фамилии которых начинаются на букву А или Д;
 - Вывести список сотрудников, стаж работы которых больше 7 лет;
 - Вывести список сотрудников с указанием их домашнего адреса, телефона и заработной платы;
 - Вывести список сотрудников, у которых зарплата от 12000 до 18000 рублей;
 - Вывести список сотрудников, у которых нет телефона.
6. Создать отчеты по всем запросам с сортировкой и группировкой.

Описание предметной области.

В офисе есть несколько отделов. В отделах работают сотрудники (по 3-5 чел). У каждого сотрудника фиксируется его должность, оклад, стаж.

У каждого сотрудника свои служебные обязанности. В одной должности могут работать несколько сотрудников.

База данных **Офис** предназначена для ведения учета сотрудников и должна обеспечивать получение сведений о сотрудниках, получаемых ими заработных платах, количестве работников в данной должности, имеющихся вакансиях.

Задание 1.

Создать файл БД «Офис»

Порядок работы

1. Запустите Access.
2. В диалоговом окне Access установите мышью на пункте **Новая база данных** и дважды щелкните мышью. Появится диалоговое окно **Файл новой базы данных**.

3. В поле ввода Имя файла введите название новой базы данных **Офис_имя_**, где **имя** – уникальное имя вашей работы.

4. В окне **Папка** выберите имя папки, в которой должна быть база данных. Нажмите кнопку **Создать**.

При загрузке своей базы данных необходимо в Access выполнить команду **Файл-открыть** и в появившемся окне выбрать имя своей базы данных.

Задание 2

Используя режим конструктора таблицы, создать таблицу **Должностис** полями: **КодДолжности** (ключ. поле, счетчик); **Должность** (тестовый); **Зарплата**(денежный), **ЧистаяЗарплата** (денежный), **Штат** (числовой длинное целое).

Порядок работы.

1. В окне базы данных щелкните вкладку **Таблица**, а затем - **Создать**.
2. В появившемся окне **Новая таблица** выберите пункт меню **Конструктор** и нажмите кнопку **ОК**. При этом откроется окно конструктора таблицы.

3. Введите имя первого поля в первой пустой строке **Имя поля** и нажмите клавишу **Tab**, чтобы перейти к столбцу **Тип данных**.

Примечание. При написании имен полей желательно использовать короткие, легко запоминающиеся поля, которые **не** должны содержать символов «.», «!», «[», «]».

4. Откройте выпадающий список **Тип данных** и выберите в нем тип поля. Столбец **Описание** не является обязательным, поэтому его можно не заполнять.

5. При необходимости перейдите в секцию **Свойства поля**, отображающую свойства выбранного типа поля, и внесите необходимые изменения.

- Для числовых и текстовых полей установите минимально возможный размер.

- Если определенное значение в поле заведомо будет встречаться больше других, то его можно установить и записать в свойство **Значение по умолчанию**.

- Поля, по которым будет часто осуществляться поиск в базе данных следует сделать индексированными, например, поле «Фамилия» таблицы «Сотрудники». Однако таких полей не должно быть много.

- Поля, требующие обязательного заполнения, в строке **Обязательное поле** должны содержать значение **Да**.

6. Не забудьте для ключевого поля таблицы установить соответствующий признак. Для этого нужно выделить поле (или несколько полей), проехав мышкой по самой левой (серой) колонке окна конструктора таблиц и нажать кнопку *Ключевое поле*. Слева от имени поля появится символ ключа. Если ключевое поле не будет определено пользователем, Access попытается добавить ключевое поле самостоятельно, что приведет к ошибкам в структуре таблицы.

7. Для завершения работы с конструктором нажмите кнопку (x) **Закрыть**. В появившемся запросе на сохранение выберите кнопку **Да**. После этого появится диалоговое окно **Сохранение**.

8. В поле ввода **Имя таблицы** укажите название таблицы и нажмите **ОК**.

Задание 3

Создать в Access таблицу «Сотрудники», используя режим **Конструктор таблицы** со следующими полями: **КодСотрудника** (ключ. поле, счетчик); **Имя**, **Отчество**, **Фамилия**, **РабочийТелефон** (текстовый); **ДатаНайма**, **ДатаРождения** (дата/время), **Пол** (логический), **КодДолжности** (числовой длинное целое) (рис. 29).

Задание 4

Создать в Access таблицу **Адреса**, используя режим **Конструктор таблицы** со следующими полями: **КодСотрудника** (ключ. поле, числовое длинное целое); **ОбластьКрайРеспублика** (текст.), **Город**(текст.), **Адрес**(текст.), **Домашний телефон** (текст.); **ПочтовыйИндекс** (числовое длинное целое).

Служебные обязанности : таб...

Имя поля	Тип данных	Писани
КодСотрудника	Числовой	
Служб обязанности	Текстовый	

Свойства поля

Общие

Подстановка

Размер поля

Формат поля

Число десятичных знаков

Авто

Маска ввода

Подпись

Значение по умолчанию

Условие на значение

Сообщение об ошибке

Обязательное поле

Индексированное поле

Да (Совпадения не дои

Адреса : таблица

Имя поля	Тип данных	Писани
Адрес	Текстовый	
Город	Текстовый	
ОбластьКрайРеспублика	Текстовый	
ПочтовыйИндекс	Текстовый	
ДомашнийТелефон	Текстовый	
РабочийТелефон	Текстовый	

Свойства поля

Общие

Подстановка

Размер поля

Формат поля

30

Сотрудники : таблица

Имя поля	Тип данных	Иса
КодСотрудника	Счетчик	
НазваниеОтдела	Текстовый	
НомерПаспорта	Текстовый	
Имя	Текстовый	
Отчество	Текстовый	
Фамилия	Текстовый	
КодДолжности	Числовой	
ДатаРождения	Дата/время	
ДатаНайма	Дата/время	

Свойства поля

Общие

Подстановка

Формат поля

Краткий формат д

99.99.00;0

Маска ввода

Должности - оклады : таб...

Имя поля	Тип данных	Иса
КодДолжности	Счетчик	
Должность	Текстовый	
Зарплата	Денежный	
Штат	Числовой	
Чистая зарплата	Денежный	

Свойства поля

Общие

Подстановка

Формат поля

Число десятичных знаков

Авто

Маска ввода

Подпись

Рис. 29. Таблицы БД «Офис»

Задание 5

Используя режим конструктора таблицы, создать таблицу Служебные-Обязанности (с полями: КодСотрудника (ключ. поле, числовой длинное целое), СлужебОбязанности (текст.).

Задание 6

Создайте связи между таблицами типа *один-ко-одному*

- 1) Сотрудники и Адреса по полю КодСотрудника,
- 2) Сотрудники и СлужебОбязанности по полю КодСотрудника.

Создайте связи между таблицами типа *один-ко-многим*

- 3) Должности и Сотрудники по полю КодДолжности.

Примечание. Для правильного функционирования связей необходимо, чтобы поля, по которым устанавливаются связи, имели одинаковый тип данных и размер.

Порядок работы

1. Выполните команду **Работа с базами данных — Отношения — Схема данных** . Откроется окно **Схема данных**, которое позволит создать связи между таблицами.

2. Если окно базы данных открывается впервые, то появится диалоговое окно **Добавление таблицы**.

3. Выберите все таблицы, для которых будут устанавливаться связи (встаньте на название таблицы и щелкните на кнопке **Добавить**).

4. Щелкните на кнопке **Заккрыть**. При этом появится окно **Схема данных**.

5. Для установки первой связи щелкните левой кнопкой мыши на поле КодСотрудника таблицы Сотрудники.

6. Переместите его в поле КодСотрудника таблицы Адреса. При отпускании кнопки мыши автоматически откроется диалоговое окно **Изменение связей**.

7. Включите поддержку **Обеспечение целостности данных**, **Каскадное обновление связанных полей**, **Каскадное удаление связанных записей** и щелкните на кнопке **Создать**. Связь между таблицами будет отображена в виде линии с более толстыми окончаниями с символами «1» и «1».

8. Аналогичным образом установите связи между остальными таблицами (см. п. 2) — 3)).

Для того, чтобы узнать в какой последовательности заполнять таблицы (или автоформы), в окне **Схема данных** расположите таблицы таким образом, чтобы таблицы с концом «1» были выше таблицы с концом «∞».

Создание и форм

Задание 1

Создайте форму для таблицы **Должности**.

1. В окне базы данных щелкните мышью вкладку **Формы**.

2. Выполните команду **Создание – Формы – Мастер форм**.

3. Выберите в окне мастера таблицу **Должности**.

4. Выберите все поля таблицы с помощью кнопки . Нажмите кнопку **Далее**

5. *Внешний вид* формы выберите *в один столбец*. Нажмите кнопку **Далее**.

6. *Стиль* выберите *стандартный*. Нажмите кнопку **Далее**

7. Задайте имя формы: **Должности**.

8. Нажмите **готово**. Форма создана.

9. Заполните форму. Введите в форму 5-6 записей.

10. Для завершения работы с формой нажмите кнопку (x) **Закреть**.

Задание 2

Создайте автоформу для таблицы **Сотрудники**.

Порядок работы

1. Выделите название таблицы **Сотрудники**.

2. Выполните команду **Создание – Формы – Форма**. Появится макет формы со всеми полями таблицы **Сотрудники**.

3. Переключитесь на **режим формы**, щелкнув лев.кнопкой мыши на названии формы. Заполните форму. Введите в форму 6-10 записей.

4. Для завершения работы с формой нажмите кнопку (x) **Закреть**.

5. Аналогичным образом создайте формы для таблиц **Служебные Обязанности** и **Адреса**.

Задание 3

Создайте форму, в которой бы выводилась только часть данных, например, фамилия, должность, телефон и адрес.

Указание: Для создания используйте **Мастер форм**.

Задание 4

Измените внешний вид формы **Сотрудники**.

1. Щелкните на название формы лев.кнопкой мыши.

2. Выберите команду **Конструктор**.

3. Измените размеры областей для ввода данных, шрифт названия полей.

Указание: Используйте инструменты конструктора форм.

4. Дайте форме заголовок. Щелкните на кнопке «надпись»  и растяните появившийся символ до нужных размеров, введите название заголовка.

Задание 5

В форме **Должности-Оклады** создать кнопку **Выход** для выхода из формы.

1. Откройте форму **Должности-Оклады** в режиме конструктора. В группе **Элементы управления** выберите элемент **Кнопка**.

2. Растяните этот элемент в область данных на место расположения кнопки. При этом откроется окно мастера **Создание кнопок**.

3. Выберите категорию **Работа с формой**, а действие - **Закрытие формы**. Нажмите кнопку **Далее**.

4. В следующем окне можно выбрать обозначение кнопки (ввести текст или рисунок). Выбрав текст в окне, введите надпись **Выход**. Нажмите кнопку **Готово**.

Создание запросов

Задание 1. Простой запрос.

Сформируйте **Простой запрос** из таблицы «Список сотрудников» на поля «Фамилия» и «Должность» *с помощью мастера*. Для этого необходимо выбрать название таблицы «Список сотрудников» и выбрать интересующие нас поля («Фамилия», «Должность»). Дайте имя запросу **ФДолжность**. Нажмите **Далее** и **Готово**. Запрос сформирован.

Сам/раб: Создайте простой запрос с полями Фамилия, ДатаРождения, Пол

Задание 2.

Создайте простой запрос **На фамилию** сотрудника, который показывает фамилию сотрудника, его должность, «чистую» и без учета подоходного налога зарплату.

Указание: Для создания такого запроса удобно воспользоваться Мастером запросов. Мастер простых запросов позволяет выделить поля, которые необходимо отобразить из разных таблиц:

1. В окне мастера простых запросов выбираем таблицу «Список сотрудников», а в ней поля «Фамилия» и «Должность»;

2. Здесь же выбираем таблицу «Должности - оклады», а в ней поля «Зарплата», «Чистая зарплата». Нажимаем **Далее** и **Готово**. Запрос сформирован.

Сам/раб: Создайте простой запрос с полями Фамилия, Имя, Адрес, ДомашнийТелефон, Зарплата.

Задание 3. Запрос с параметром (**Оклад**). Создайте запрос, который выводит список должностей с окладом, превышающим 10000руб.

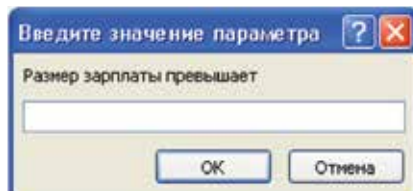
Порядок работы:

1. С помощью мастера сформируйте простой запрос на поля «Должность» и «Зарплата» таблицы «Должности - оклады».

2. Чтобы наложить условие отбора на зарплату, превышающую заданное значение, перейдите в режим конструктора и в строке *Условие отбора* под полем «Зарплата» наберите:

> [Размер зарплаты превышает].

Для запуска запроса на панели инструментов нажмите на кнопку . При выполнении запроса будет отображаться диалоговое окно предлагающее ввести размер зарплаты.



Сам/раб: Создайте запрос, который выводит список сотрудников, моложе 40 лет (т.е. родившихся до 01.01.1972г).

Задание 4. Групповой запрос (Должность).

Подсчитать количество специалистов работающих на одной и той же должности в «Офисе продаж и обслуживания абонентов».

1. Создать запрос в режиме конструктора.
2. Добавить таблицы «Должности-оклады» и «Список сотрудников».

3. На вкладке работа с запросами выполнить команду  .

4. Выбрать **Групповую операцию**.

- для поля «Должность» таблицы «Должности-оклады» — **Группировка**.
- для поля «Фамилия» таблицы «Список сотрудников» — **Count**.

Назвать поле как **ЕСТЬ: Фамилия**

- для поля «Штат» таблицы «Должности-оклады» — **Last**.

Назвать поле как **НУЖНО: Штат**

- для поля «Зарплата» таблицы «Должности-оклады» — **Last**.

5. Сохранить запрос под именем **Должность**.

Задание 6. Запрос, включающий функцию (Вакансия).

Создать запрос в режиме конструктора.

1. Добавить запрос **Должность**.
2. Из запроса **Должность** выбрать поле «Должность».
3. Поле Вакансия запроса сформировать как выражение **Вакансия: [НУЖНО] – [ЕСТЬ]**.

4. Поле Зарплата запроса соответствует полю Last-Зарплата Запроса **ДолжностьЗарплата: Last-Зарплата**

5. Сохранить запрос под именем **Вакансия**.

Задание 7. Запрос с условием. (**Вакансия с окладом выше заданного**).

Данный запрос показывает наличие вакансий с окладом выше заданного.

Создать запрос в режиме конструктора.

1. Выбрать все поля **запроса Вакансия**.

2. Наложить условие на поле «Зарплата» :> [**Размер оклада**].

3. Наложить условие на поле «Вакансия» : > **0**. (Отрезаем нулевые вакансии)

4. Сохранить запрос.

Перекрестные запросы используют для расчетов и представления данных в структуре, облегчающей их анализ. Перекрестный запрос подсчитывает сумму, среднее, число значений или выполняет другие статистические расчеты, после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк. (В режиме конструктора запросов: для заголовков строк и столбцов выбираем групповую операцию *Группировка*, а для поля значение — одну из групповых операций, соответствующую задаче.

Сам/раб: Создайте перекрестный запрос, который выводит список сотрудников с должностями и зарплатами.

Сам/раб: Создайте разные отчеты по таблице **Сотрудники** и по любым двум запросам с группировкой и сортировкой.

ТЕМЫ ДОКЛАДОВ И РЕФЕРАТОВ

1. Базы данных. Системы управления базами данных
2. Автоматизированное рабочее место юриста
3. Информационные технологии в деятельности суда
4. Использование пакета Microsoft Access для ведения базы данных в области юриспруденции
5. Приемы и методы работы со сжатыми данными
6. Информационные справочные системы в человеческом обществе
7. Базы данных и Интернет.
8. Использование информационных технологий для обработки данных эксперимента
9. Информационные технологии организационного управления (корпоративные информационные технологии)
10. Информационные технологии в промышленности и экономике
11. Информационные технологии автоматизированного проектирования
12. Программные средства информационных технологий
13. Технические средства информационных технологий
14. Этапы эволюции информационных технологий
15. Геоинформационные технологии. Основные понятия
16. Геоинформационные системы в ведении земельных кадастров
17. CASE — технологии
18. Основные стандарты мультимедиа — технологий
19. Аппаратные средства мультимедиа — технологий
20. Компьютерные сети. Основные понятия
21. Глобальные компьютерные сети
22. Локальные компьютерные сети
23. Инструментальные программные средства для создания экспертных систем.
24. Иерархические классификационные системы
25. Системы автоматизированного проектирования в машиностроении
26. Автоматизированные системы управления технологическими процессами

27. Информационно- справочные системы и информационно – поисковые технологии

28. Системы автоматизации документооборота и учета

29. Экспертные системы в отраслях народного хозяйства

30. Информационные сетевые технологии

31. Мультимедиа – технологии. Основные понятия

32. Информационно – справочные правовые системы (ИСПС).

33. Информационные технологии искусственного интеллекта

34. Экспертные системы. Основные понятия

35. Информационные технологии защиты информации

36. Информационные технологии в образовании

37. Информационные технологии в медицине

38. Телекоммуникационные технологии

39. 1 С: Бухгалтерия

40. Обзор современных систем автоматизированного бухгалтерского учета (САБУ)

41. Информационные технологии автоматизации офиса

42. Информационная справочно – правовая система (ИСПС) «Консультант – плюс»

КОНТРОЛЬНЫЕ ВОПРОСЫ К ЗАЧЕТУ

1. Назовите этапы проектирования БД.
2. Чем характеризуются инфологические модели?
3. Для чего предназначено даталогическое проектирование?
4. Что понимают под классом объектов? Свойствами объектов?
5. Типы связей «один-к-одному» и «один-ко-многим»: как они реализуются в реляционных базах данных?
6. Какие объекты можно создать в СУБД Access?
7. В каких режимах можно создать структуру таблицы в Access?
8. Какая информация указывается в структуре таблицы?
9. Что такое ключевое поле, и какие требования предъявляются к нему?
10. Как в СУБД Access можно определить ключевое поле?
11. Как и для чего в СУБД Access создается схема данных?
12. Какие типы данных используются в Access?
13. Как при определении структуры таблицы использовать свойства полей?
14. Как в Access определить связи между таблицами?
15. Какие возможности предоставляет Access по изменению структуры таблиц?
16. Как добавить\удалить поле в структуре таблицы?
17. Как изменить шрифт записей и цвет фона в таблице?
18. Как изменить ширину столбца и высоту строки в таблице?
19. Как отсортировать записи по одному полю?
20. Как отсортировать записи по нескольким полям?
21. С какой целью создаются формы? Назовите виды форм MS Access.
22. Какие инструменты создания форм существуют?
23. Какие виды автоформ существуют?
24. Перечислите основные разделы форм. Для чего каждый из них предназначен?
25. Для чего используется раздел Данные в форме?
26. Как разместить в форме элемент управления?

27. Как построить форму по двум связанным таблицам?
28. Как произвести поиск и замену данных в Access?
29. Что такое запрос?
30. Какие элементы окна конструктора запросов существуют?
31. Что такое бланк запроса?
32. Где записываются критерии условия выбора для запроса?
33. Как удалить таблицу из запроса?
34. Как удалить поле из запроса?
35. Какие основные операторы, используются в запросе?
36. В каких случаях используется оператор OR?
37. В каких случаях используется оператор AND?
38. В чем различие между операторами OR и AND?
39. Для чего используются итоговые запросы?
40. Как вычислить сумму значений заданного поля?
41. Как осуществить сортировку записей в запросе?
42. Какие виды вычислений можно произвести в итоговых полях?
43. Типы запросов в Access. Как выбрать тип запроса?
44. Как посмотреть результат запроса?
45. Что представляют собой отчёты в Access?
46. В чем их преимущества над другими методами вывода данных?
47. С чего начинается создание отчёта?
48. Какие существуют мастера для создания отчёта?
49. В чем отличие автоотчётов от других способов создания отчёта?
50. Чем характеризуется режим Конструктора?
51. Какие существуют разделы отчёта?
52. Какие инструменты окна отчёта используются?
53. В каком разделе отчёта выводятся итоги по группе?
54. В поле, какого типа, можно задать группировку по интервалу? По годам?
55. Какие изменения можно внести в отчёт в режиме предварительного просмотра?

ПРИЛОЖЕНИЕ А.

Самостоятельная работа по вариантам

Разработать базу данных согласно заданному описанию предметной области.

В работе должны быть представлены:

- 1) исследование предметной области;
- 2) модель «сущность-связь»;
- 3) реляционная модель;
- 4) отчёты по всем таблицам и запросам;
- 5) файл базы данных.

Реализовать разработанную модель в СУБД Access.

1. Создать таблицы и установить связи между ними.
2. Создать формы для ввода данных. Родительская таблица должна содержать не более 5 записей, остальные – не менее 10. Отформатировать формы и добавить элементы формы (кнопки...).
3. Создать запросы.
4. Создать отчёты по всем таблицам и запросам.

Вариант 1. Создать базу данных детского сада.

Описание предметной области.

В детском садике 4 группы (младшая, средняя, старшая, подготовительная) со своим воспитателем и нянечкой. В каждой группе несколько детей (2-3), у которых учитывается фамилия, имя, возраст и домашний адрес. Кроме этого необходимо создать таблицу дополнительных сведений о ребенке, где будет учитываться имя отчество одного родителя, его место работы, телефон рабочий и домашний.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий детей несколько должны начинаться на букву А и Г;
- Возраст детей от 3 до 7 лет;
- Среди воспитателей должна быть одна фамилия Сидорова;

Создать и сохранить следующие запросы:

- Вывести список детей, фамилии которых начинаются на буквы А или Г;
- Вывести список детей, возраст, которых равен 7 лет;
- Вывести список детей, у которых воспитатель Сидорова;
- Вывести список детей в возрасте до 4 лет, их родителей, домашний адрес, место работы и номер рабочего и домашнего телефонов;
- Подсчитать число детей: а) в детском саду; б) у воспитателя Сидоровой.

Вариант 2. Создать базу данных туристической базы

Описание предметной области.

На туристической базе 5 тренеров, которые занимаются с туристами различными видами спорта. У каждого тренера несколько туристов (2-3), у которых фиксируется фамилия, имя отчество, дата прибытия, стоимость путевки и каким видом спорта он занимается. Кроме этого необходимо создать для чрезвычайной ситуации таблицу дополнительных сведений о туристе, куда включить домашний адрес, место работы, группу крови.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий туристов несколько должны начинаться на букву В и Н;
- Дата прибытия должна находиться в интервале от 01.01.2004 до 31.03.2004;
- Виды спорта: горные лыжи, слалом, сноуборд;
- Стоимость путевки от 10 до 15 тысяч.

Создать и сохранить следующие запросы:

- Вывести список туристов, фамилии которых начинаются на буквы В или Н;
- Вывести список туристов, прибывших в январе;
- Вывести список туристов, которые занимаются горными лыжами;
- Вывести список туристов, которые занимаются слаломом их домашний адрес, место работы и группу крови;
- Подсчитать число туристов: а) занимающихся слаломом; б) со второй группой крови.

Вариант 3. Создать базу данных факультета

Описание предметной области.

На кафедре «Социальная работа и» несколько групп (94-21, 94-22, 93-21, 93-22, 92-21). В каждой группе несколько студентов (2-3), у которых фиксируется Фамилия, имя, отчество, дата рождения, бюджетная или внебюджетная форма обучения. Кроме этого, необходимо создать таблицу дополнительных сведений о студенте, куда включить адрес прописки, номер страхового свидетельства, паспортные данные

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий студентов несколько должны начинаться на букву А и К;
- Дата рождения от 01.01.87 до 31.03.87
- Адрес прописки должен быть Казань и другие города РТ

Создать и сохранить следующие запросы:

- Вывести список студентов, фамилии которых начинаются на букву К или А;
- Вывести список студентов, родившихся в феврале;
- Вывести список студентов, которые обучаются по вне бюджету;
- Вывести список студентов, прописанных в г. Казань с указанием их домашнего адреса, паспортных данных и номера страховки;
- Подсчитать число студентов, прописанных в Казани.
- Подсчитать число студентов, у которых внебюджетная форма обучения.

Вариант 4. Создать базу данных фирмы «Эдельвейс»

Описание предметной области.

Фирма «Эдельвейс-М» производит и реализует молочную продукцию. На предприятии существуют следующие отделы: Отдел продаж, технический отдел, отдел маркетинга и бухгалтерия. В каждом отделе несколько сотрудников (2-3), у которых своя должность, оклад и стаж работы. Кроме этого необходимо создать таблицу дополнительных сведений о сотруднике, где будут фиксироваться его домашний адрес, паспортные данные, дата рождения, телефон.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий сотрудников несколько должны начинаться на букву В и М;

- Стаж работы сотрудников от 3 до 10 лет
- Должности: инженер, менеджер, маркетолог, бухгалтер и т.д.
- Оклад от 5000 до 12000 рублей

Создать и сохранить следующие запросы:

- Вывести список сотрудников, фамилии которых начинаются на буквы В или М;

- Вывести список сотрудников, стаж работы которых больше 5 лет;

- Вывести список сотрудников, у которых оклад от 8000 до 10000 рублей;

- Вывести список менеджеров, с указанием их домашнего адреса, паспортных данных и телефона;

- Вывести наибольший оклад и должность, соответствующую этому окладу.

- Подсчитать сумму, необходимую для выплаты заработной платы всем сотрудникам.

Вариант 5. Создать базу данных отделения социальной защиты

Описание предметной области.

В отделении социальной защиты работают 5 патронажных сестер, которые обслуживают лежачих пенсионеров. У каждой патронажной сестры по 2-3 пенсионера, для которых указывается адрес проживания, паспортные данные, сумма пенсии, телефон. Кроме этого имеются дополнительные сведения об основном диагнозе болезни, справочные данные о родственниках, проживающих по другим адресам.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий пенсионеров несколько должны начинаться на букву В и М;

- Сумма пенсии составляет от 1500 до 2500 руб.

- Не у всех пенсионеров есть телефон.

Создать и сохранить следующие запросы:

- Вывести список пенсионеров, фамилии которых начинаются на буквы В или М;

- Вывести список пенсионеров, у которых пенсия ниже 2000 руб.;

- Вывести список пенсионеров, у которых нет телефона;
- Вывести список пенсионеров у первой патронажной сестры;
- Подсчитать число пенсионеров у каждой патронажной сестры;
- Подсчитать количество пенсионеров с диагнозом инсульт.

Вариант 6. Создать базу данных ателье мод

Описание предметной области.

Ателье мод представляет собой небольшой комплекс различных залов и служб для оказания услуг населению. В этот комплекс входят зал кроя, пошивочный зал, обметочный зал, зал приемки заказов у населения. В каждом из этих залов работают несколько сотрудников (2-3), у которых своя специальность оклад, телефон и стаж работы. У некоторых сотрудников есть постоянные клиенты (по 4 человека). Имеются сведения об этих клиентах: фамилия имя отчество, домашний адрес и телефон.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий сотрудников несколько должны начинаться на букву С и П;
- Стаж работы сотрудников от 3 до 10 лет
- Специальности: закройщик, портниха, приемщик
- Оклад от 3000 до 7000 рублей

Создать и сохранить следующие запросы:

- Вывести список сотрудников, фамилии которых начинаются на буквы С или П;
 - Вывести список сотрудников, стаж работы которых больше 5 лет;
 - Вывести список сотрудников, у которых оклад от 3000 до 5000 рублей;
 - Вывести список постоянных клиентов у одного из сотрудников;
 - Подсчитать число постоянных клиентов у каждого сотрудника;
- Вывести среднюю зарплату сотрудников. Описание предметной области.

Вариант 7. Создать базу данных фирмы «М-вид»

Описание предметной области.

Фирма «М-вид» реализует товары 5 видов: компьютеры, бытовую технику, оргтехника, торговое оборудование и инструменты для дома.

Каждого вида по 2-3 наименования, для которых указывается цена, дата изготовления и производитель. Для каждого вида товаров работает свой консультант-менеджер. У менеджера имеется свой табельный номер, стаж работы и оклад.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий менеджеров несколько должны начинаться на букву В и М;

- Стаж работы сотрудников от 3 до 10 лет

- Оклад больше 8000 рублей

Создать и сохранить следующие запросы:

- Вывести список сотрудников, фамилии которых начинаются на буквы В или М;

- Вывести список сотрудников, стаж работы которых меньше 7 лет;

- Вывести список сотрудников, у которых оклад от 8000 до 1000 рублей;

- Вывести список менеджеров, с указанием их домашнего адреса, паспортных данных и телефона;

- Вывести среднюю зарплату менеджеров;

- Подсчитать число сотрудников с окладом от 8000 до 1000 рублей.

Вариант 8. Создать базу данных фирмы «ЖЖЖ»

Описание предметной области.

Фирма «ЖЖЖ» реализует гастрономическую продукцию и работает с 5 поставщиками. Каждый поставщик поставяет продукцию 5 видов: компьютеры, бытовую технику, оргтехника, торговое оборудование и инструменты для дома. Каждого вида по 2-3 наименования, для которых указывается цена, дата изготовления и производитель. Для каждого вида товаров работает свой консультант-менеджер. У менеджера имеется свой табельный номер, стаж работы и оклад.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий менеджеров несколько должны начинаться на букву В и М;

- Стаж работы сотрудников от 3 до 10 лет

- Оклад от 5000 до 12000 рублей

Создать и сохранить следующие запросы:

- Вывести список сотрудников, фамилии которых начинаются на буквы В или М;
- Вывести список сотрудников, стаж работы которых больше 5 лет;
- Вывести список сотрудников, у которых оклад от 8000 до 1000 рублей;
- Вывести список менеджеров, с указанием их домашнего адреса, паспортных данных и телефона;
- Вывести средний стаж сотрудников;
- Вывести список консультантов-менеджеров, продающих компьютеры.

Вариант 9. Создать базу данных фирмы, реализующей сотовые телефоны

Описание предметной области.

Фирма реализует телефоны 5 видов «Nokia», «Samsung», «Siemens», «LG», «Motorola». Каждого вида по 2-3 модификации. Для каждой модификации указывается код, стоимость, год выпуска, вес. Кроме производителя телефонов есть страны, где их производят. Так, например телефон «Nokia» производят в Венгрии и Германии.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Стоимость телефонов от 2000 до 13000 руб;
- Год выпуска — последние 3 года;
- Вес от 100 до 300 г

Создать и сохранить следующие запросы:

- Вывести список телефонов, стоимость которых ниже 10000руб с указанием стран — производителей и других характеристик;
- Вывести список телефонов последнего года производства;
- Вывести список телефонов с весом более 200г;
- Подсчитать количество телефонов, произведенных в Германии;
- Вывести наибольшую цену телефона и его модификацию;
- Вывести список телефонов с их стоимостью и видом.

Вариант 10. Создать базу данных библиотеки

Описание предметной области.

В библиотеке книги сортируются по отделам: классическая литература, приключения и фантастика, детективы, женский роман и детская

литература. В каждом отделе 2-3 писателя, которые написали по несколько всем известных книг. Для писателей фиксируется страна проживания, годы жизни, для книг — год издания, стоимость, количество страниц.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий писателей несколько должны начинаться на букву А и К;

- Год издания книг — последние 5 лет;

- Стоимость книг от 50 до 400руб.

Создать и сохранить следующие запросы:

- Вывести список писателей, фамилии которых начинаются на буквы А или К;

- Вывести список книг, изданных в последний год;

- Вывести список книг, стоимость которых ниже 100 рублей»;

- Вывести список книг с названием и автором из отдела детективы;

- Подсчитать суммарную стоимость всех книг библиотеки;

- Вывести книгу, изданную раньше всех остальных.

Вариант 11. Создать базу данных библиотеки

Описание предметной области.

В библиотеке книги сортируются по отделам: классическая литература, приключения и фантастика, детективы, женский роман и детская литература. В каждом отделе по несколько книг. В библиотеку приходят 10 читателей. Для книг фиксируется автор, год издания, стоимость, для читателей — номер читательского билета, адрес, паспортные данные, возраст, сдал или не сдал книги.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий читателей несколько должны начинаться на букву С и Р;

- Год издания книг — последние 5 лет;

- Возраст читателей — от 16 до 50 лет;

- Стоимость книг от 50 до 1000 рублей.

Создать и сохранить следующие запросы:

- Вывести список книг, которые брали читатели, с фамилия на буквы С или Р;

- Вывести список читателей, которые брали книги из отдела приключения и фантастика;
- Вывести список книг, изданных за последние 5 лет;
- Подсчитать сколько книг на руках.
- Подсчитать общее количество книг в каждом отделе;
- Вывести возраст самого молодого читателя.

Вариант 12. Создать базу данных тренировочной базы

Описание предметной области.

На базе тренируются 5 команд: молодежная¹, молодежная², основная и дублирующая¹, дублирующая². В каждой команде по несколько игроков (2-4), у которых учитывается фамилия имя отчество, вес, рост, возраст. Кроме того, необходимо создать таблицу дополнительных сведений о спортсмене, его домашний адрес, паспортные данные, номер страховки и группу крови.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий игроков несколько должны начинаться на букву К и Р;

- Возраст игроков от 18 до 30 лет;

- Вес игроков от 60 до 90 кг;

Создать и сохранить следующие запросы:

- Вывести список игроков, фамилии которых начинаются на буквы К или Р;

- Вывести список игроков в возрасте до 25 лет;

- Вывести список игроков, которые имеют вес до 70 кг;

- Вывести список игроков молодежной команды², с указанием их домашнего адреса, паспортных данных, номера страховки и группы крови;

- Подсчитать число игроков в каждой команде;

- Вывести средний возраст игроков в каждой команде.

Вариант 13. Создать базу данных факультета

Описание предметной области.

На социально-экономическом факультете 3 кафедры: «Правовой информатики и ИТ», «Менеджмент» «Экономика». У каждой кафедры

номер и фамилия заведующего. На каждой кафедре работают несколько преподавателей (3-4 человека). У каждого преподавателя свой табельный номер, должность, стаж работы и предмет, который он ведет. Кроме того, у каждого преподавателя несколько групп, в которых он ведет свой предмет. У группы фиксируется ее номер и количество студентов.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий преподавателей несколько должны начинаться на букву П и Л;
- Стаж работы преподавателей от 5 до 20 лет
- Количество студентов в группах от 25 до 35.

Создать и сохранить следующие запросы:

- Вывести список преподавателей, фамилии которых начинаются на буквы П или Л;
- Вывести список преподавателей, стаж работы которых более 15 лет;
- Вывести список преподавателей кафедры «Экономика» с их стажем работы;
- Вывести список групп, в которых преподает один из преподавателей;
- Подсчитать количество групп у каждого преподавателя;
- Подсчитать число преподавателей

Вариант 14. Создать базу данных ателье мод

Описание предметной области.

Ателье мод представляет собой небольшой комплекс различных залов и служб для оказания услуг населению. В этот комплекс входят зал кроя, пошивочный зал, обметочный зал, зал приемки заказов у населения. В каждом из этих залов работают несколько сотрудников (2-3), у которых своя специальность, оклад, телефон и стаж работы. Кроме этого для экстренных случаев фиксируются дополнительные сведения о сотрудниках, его домашний адрес и телефон, паспортные данные.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий сотрудников несколько должны начинаться на букву С и П;
- Стаж работы сотрудников от 3 до 10 лет;
- Специальности: закройщик, портниха, приемщик;

- Оклад от 3000 до 7000 рублей.

Создать и сохранить следующие запросы:

- Вывести список сотрудников, фамилии которых начинаются на буквы С или П;
- Вывести список сотрудников, стаж работы которых больше 5 лет;
- Вывести список сотрудников, у которых оклад от 4000 до 5500 рублей;
- Вывести список закройщиков, с указанием их домашнего адреса, паспортных данных и телефона;
- Подсчитать число сотрудников каждой специальности;
- Подсчитать сумму, необходимую для выплаты заработной платы всем сотрудникам.

Вариант 15. Создать базу данных фирмы, реализующей кондитерскую продукцию

Описание предметной области.

Фирма «Дубль В» производит и реализует кондитерскую продукцию. На предприятии существуют следующие отделы: Отдел продаж, технический отдел, отдел маркетинга и бухгалтерия. В каждом отделе несколько сотрудников (2-3), у которых своя должность, оклад и стаж работы. Кроме этого необходимо создать таблицу дополнительных сведений о сотруднике, где будут фиксироваться его домашний адрес, паспортные данные, дата рождения, телефон.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий сотрудников несколько должны начинаться на букву В и М;
- Стаж работы сотрудников от 3 до 10 лет;
- Должности: инженер, менеджер, маркетолог, бухгалтер и т.д.;
- Оклад от 5000 до 12000 рублей.

Создать и сохранить следующие запросы:

- Вывести список сотрудников, фамилии которых начинаются на буквы В или М;
- Вывести список сотрудников, стаж работы которых меньше 6 лет;
- Вывести список сотрудников, у которых оклад от 8000 до 1000 рублей;
- Вывести список менеджеров, с указанием их домашнего адреса, паспортных данных и телефона;

- Вывести список сотрудников, работающих в должности маркетолог и стаж у которых больше 7 лет;
- Подсчитать число сотрудников, работающих в должности менеджера.

Вариант 16. Создать базу данных оптового склада

Описание предметной области.

Имеется оптовый склад, где есть 5 отделов: бытовая химия, канцелярские товары, видеокассеты, садовый инвентарь, оргтехника. В каждом отделе несколько сотрудников (2-3), у которых своя должность, оклад и стаж работы. Кроме этого у каждого отдела есть постоянные клиенты (2-3 человека у каждого отдела). Имеются сведения о месте работы клиентов, их рабочий телефон.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий сотрудников несколько должны начинаться на букву В и М;
- Стаж работы сотрудников от 3 до 10 лет
- Оклад от 5000 до 12000 рублей

Создать и сохранить следующие запросы:

- Вывести список сотрудников, фамилии которых начинаются на буквы В или М;
- Вывести список сотрудников, стаж работы которых от 5 до 8 лет;
- Вывести список сотрудников, у которых оклад больше 10000 рублей;
- Вывести список клиентов отдела бытовой химии;
- Подсчитать число сотрудников, стаж которых меньше 4 лет;
- Вывести среднюю зарплату сотрудников.

Вариант 17. Создать базу данных факультета

Описание предметной области.

На социально-экономическом факультете 4 кафедры: «Социальной работы», «Менеджмент» «Экономика» и «Государственное и муниципальное управление (ГМУ)». У каждой кафедры номер и фамилия заведующего. На каждой кафедре работают несколько преподавателей (3-4 человека). У каждого преподавателя свой табельный номер, должность, стаж работы и предмет, который он ведет. Кроме того, необходимо составить таблицу дополнительных сведений о пре-

подавателе. Его страховой номер, ИНН, домашний адрес, домашний телефон.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди фамилий преподавателей несколько должны начинаться на букву А и К;

- Стаж работы от 5 до 25 лет

- Домашний телефон может быть не у всех преподавателей

Создать и сохранить следующие запросы:

- Вывести список преподавателей, фамилии которых начинаются на буквы К или А;

- Вывести список преподавателей, стаж работы которых более 10 лет;

- Вывести список преподавателей кафедры «Социальная работа»;

- Вывести список преподавателей, у которых нет телефона;

- Вывести средний стаж преподавателей;

- Подсчитать число преподавателей на каждой кафедре.

Вариант 18. Создать базу данных ЖЭУ

Описание предметной области.

ЖЭУ обслуживает 5 улиц своего района. На каждой улице по 2-3 дома, для которых указывается номер, этажность, год постройки. В каждом доме по 3-4 квартиры, для которых фиксируется номер, фамилия владельца, кв. метры

Родительская таблица должна содержать 5 записей, остальные по 10 записей.

Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди улиц одна называется Восстание;

- Год постройки от 3 до 10 лет

- Этажность 2 до 14

- Метраж от 36 до 120 кв. м

Создать и сохранить следующие запросы:

- Вывести список домов по улице Восстания;

- Вывести список домов, которые были построены за последние 2 года;

- Вывести список владельцев квартир, метраж которых мене 40 кв. м.

- Вывести список владельцев, проживающих на улице Восстания и имеющих метраж квартиры больше 50 кв. м.

- Подсчитать число домов, этажей у которых меньше 10.
- Подсчитать число домов на каждой улице.

Вариант 19. Создать базу данных ЖЭУ

Описание предметной области.

ЖЭУ обслуживает 5 улиц своего района. Каждую улицу обслуживает один управляющий домами, один дворник и один сантехник. Для улиц указывается название и количество домов, для рабочих — фамилия, имя, отчество, стаж работы, адрес, паспорт, оклад, количество дней выхода на работу. Кроме того, необходимо составить таблицу дополнительных сведений о работниках. Его страховой номер, ИНН, домашний адрес, домашний телефон.

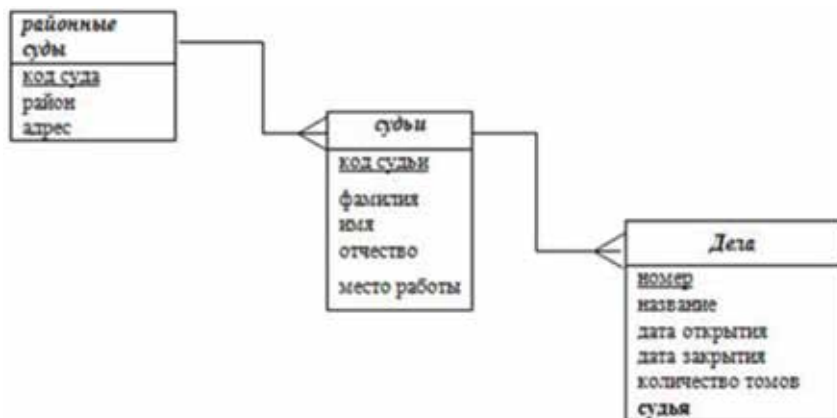
Для таблиц использовать произвольные данные со следующими ограничениями:

- Среди улиц одна называется Декабристов;
- Количество домов на улице от 5 до 10;
- Стаж работы сотрудников от 2 до 5 лет;
- Оклад сотрудников от 2000 до 5000 рублей.

Создать и сохранить следующие запросы:

- Вывести список сотрудников, обслуживающих улицу Декабристов;
- Вывести список улиц, на которых только 3 дома;
- Вывести дворников ЖЭУ;
- Вывести список работников, оклад у которых меньше 3000 и стаж работы больше 3 лет;
- Вывести средний оклад работников;
- Подсчитать число всех работников ЖЭУ.

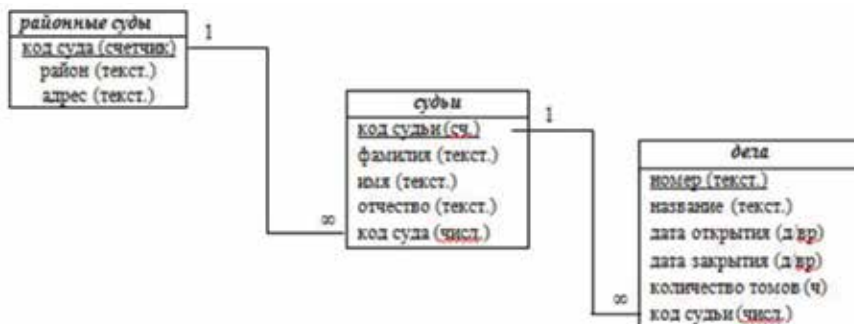
Приложение 1. ER-модель БД «Суд»



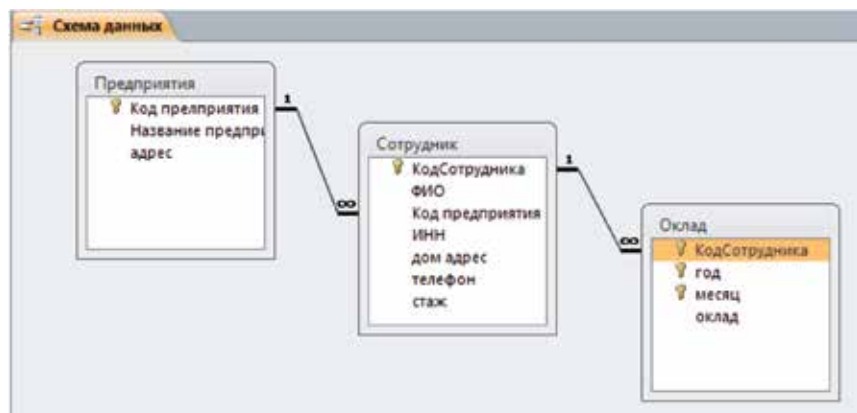
В каждом районном суде могут работать несколько судей, но каждый судья может работать только в одном районном суде. Таким образом, между сущностями районные суды и судьи устанавливается связь один ко многим.

Каждый судья может вести несколько дел, но каждое дело ведет только один судья. Таким образом, между сущностями судьи и дела устанавливается связь один ко многим.

Приложение 2. Реляционная модель БД «Суд»



Приложение 3. Схема данных «Пенсионный фонд»



Приложение 4. Схема данных «Офис».



Учебное издание

В.Р. Ахметгалиева

Л.Р. Галяутдинова

БАЗЫ ДАННЫХ

Microsoft Access 2013

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Издание осуществлено в авторской редакции

Корректурa Е.В. Ермичева

Оформление, верстка М.А. Иванова

Подписано в печать 13.10.2017. Формат 60х90 1/16.

Усл. п. л. 6. Тираж 100 экз.