

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УДМУРТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Филиал ФГБОУ ВО «УдГУ» в г. Воткинске**



«УТВЕРЖДАЮ»
Зам. директора по УМР

Г.М. Смирнова

**Учебно-методическое пособие по дисциплине
«Базы данных»**

Для студентов направления подготовки
Прикладная информатика
09.03.03

Воткинск 2018

Разработчики:

Клинер С.В., ст. преподаватель
Кучерова Е.А., зав. кафедрой ИиИТ

Оглавление

Используемые сокращения	5
Введение.....	6
Часть I. Теоретические положения.....	8
1.1. Классификация баз данных	9
1.2. Структурные элементы базы данных	11
1.3. Модели БД.....	13
СРС1. Технология хранения, поиска и сортировки информации.....	15
1.4. Функциональные возможности СУБД.....	16
1.5. Команды для выполнения типовых операций	17
Типовая структура интерфейса	17
1.6. Этапы проектирования баз данных.....	19
1.7. Создание структуры таблиц базы данных	25
1.8. Ввод и редактирование данных	25
1.9. Обработка данных, содержащихся в таблицах	26
1.10. Вывод информации из базы данных.....	27
СРС2. Создание структуры многотабличной базы данных с учетом будущей профессии.	28
Часть II. Задания к лабораторным работам	31
Лабораторная работа № 1	31
Создание локальных баз данных.....	31
Задание 1. Создание таблиц в режиме ввода данных	31
Задание 2. Создание таблиц в режиме Конструктора.....	34
Задание 3. Создание таблиц в режиме Мастера.	37
Задание 4. Использование масок ввода	41
Задание 5. Создание форм	45
Задание 6. Создание диаграмм и кнопочных форм.....	50
Задание 7. Сортировка, поиск, фильтрация данных	53
Задание 8. Создание запросов на выборку.....	58
Задание 9. Вычисления в запросах	62
Задание 10. Создание отчетов	66
Задание 11. Создание индексов и ключей.....	70
Задание 12. Связывание таблиц	74
Тест	77
Лабораторная работа № 2	80
Задание 1. Создание составных форм, запросов, отчетов	80
Задание 2. Создание макросов	85
Создание открытых баз данных	88

Задание 3. Гиперссылки и объекты OLE.....	88
Задание 4. Экспорт и импорт данных.....	91
Задание 5. Макросы с условиями.....	94

Используемые сокращения

БД – база данных

СУБД – система управления базами данных

ИПС – информационно–поисковая система

ПК – персональный компьютер

ИКТ – информационно-коммуникационные технологии

СРС – самостоятельная работа студента

Введение

Современные информационные системы, основанные на концепции интеграции данных, характеризуются огромными объемами хранимых данных, сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей.

Методическое пособие «Работа с базами данных» составлено на основе рабочей программы по дисциплине «Базы данных», в соответствии с требованиями Государственного образовательного стандарта к минимуму содержания и уровню подготовки выпускника по дисциплине, в частности:

выпускник должен иметь представление

- об основных методах и средствах сбора, обработки, хранения, передачи и накопления информации;
- о программном обеспечении вычислительной техники, о компьютерных сетях и сетевых технологиях обработки и защиты информации;

знать

- основные понятия автоматизированной обработки информации, общий состав и структуру персональных данных ЭВМ и ВС;
- базовые системные программные продукты и пакеты прикладных программ;
- состав, функции и возможности использования специального программного обеспечения;

уметь

- использовать программное обеспечение, компьютерные и телекоммуникационные средства в профессиональной деятельности;
- оформлять документы по основным видам своей профессиональной деятельности в соответствии с требованиями документационного обеспечения управления;
- составлять и оформлять документы по товарным, денежным и расчетным операциям;

Учебно-методическое пособие содержит теоретические положения, 12 практических заданий, а также задания для самостоятельной работы студентов, тест, позволяющий студентам проверить свои знания.

Учебно-методическое пособие «Работа с базами данных» направлено на формирование представления о базах данных (БД) возможностях СУБД и их использовании, способствует развитию познавательной деятельности студентов, развитию у них творческого мышления, умений и навыков самостоятельной работы.

Основные функциональные возможности и технологические операции работы в СУБД в пособии рассматриваются на основе привязки к конкретному типу программного продукта – СУБД MS Access. Знания, полученные при изучении данного материала, являются базовыми при практическом знакомстве с любым новым видом СУБД.

Цель: познакомить с функциональными возможностями СУБД и общей методикой использования этих программных средств работе, связанной с организацией хранения и обработки данных.

После изучения дисциплины студенты должны знать:

- ✓ понятие и назначение базы данных, системы управления базами данных;
- ✓ различие архитектур баз данных: клиент-сервер и файл-сервер;
- ✓ структурные элементы базы данных;
- ✓ виды моделей данных;
- ✓ особенности и назначение реляционной модели;
- ✓ типы реальных связей информационных объектов;
- ✓ назначение нормализации отношений и виды форм;
- ✓ функциональные возможности СУБД;
- ✓ основные технологические этапы решения задач в СУБД;
- ✓ команды для выполнения типовых операций.

По окончании изучения дисциплины предусмотрена контрольная работа.

Часть I. Теоретические положения.

Цель любой информационной системы — обработка данных об объектах реального мира. В широком смысле слова база данных — это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Под *предметной областью* принято понимать часть реального мира, подлежащего изучению для организации управления и, в конечном счете, автоматизации, например, предприятие, ВУЗ и т.д.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро извлекать выборку с произвольным сочетанием признаков. Сделать это возможно, только если данные структурированы.

Структурирование — это введение соглашений о способах представления данных.

Неструктурированными называют данные, записанные, например, в текстовом файле.

Пользователями базы данных могут быть различные прикладные программы, программные комплексы, а также специалисты предметной области, выступающие в роли потребителей или источников данных, называемые *конечными пользователями*.

В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария — *системы управления базами данных*.

База данных (БД) — это поименованная совокупность структурированных данных, относящихся к определенной предметной области.

Система управления базами данных (СУБД) — это комплекс программных и языковых средств, необходимых для создания баз данных,

поддержания их в актуальном состоянии и организации поиска в них необходимой информации.

Централизованный характер управления данными в базе данных предполагает необходимость существования некоторого лица (группы лиц), на которое возлагаются функции администрирования данными, хранимыми в базе.

1.1. Классификация баз данных

По *технологии обработки* данных базы данных подразделяются на централизованные и распределенные.

Централизованная база данных хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе. Такой способ использования баз данных часто применяют в локальных сетях ПК.

Распределенная база данных состоит из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

По *способу доступа* к данным базы данных разделяются на базы данных с *локальным доступом* и базы данных с *удаленным (сетевым) доступом*.

Системы централизованных баз данных с сетевым доступом предполагают различные *архитектуры* подобных систем:

- файл-сервер;
- клиент-сервер.

Файл-сервер. Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют с функции

рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать также на рабочих станциях локальные БД, которые используются ими монопольно. Концепция файл-сервер условно отображена на рис. 1.

Клиент-сервер. В этой концепции подразумевается, что помимо хранения централизованной базы данных центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные (но не файлы) транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запросов SQL. Концепция клиент-сервер условно изображена на рис. 2.

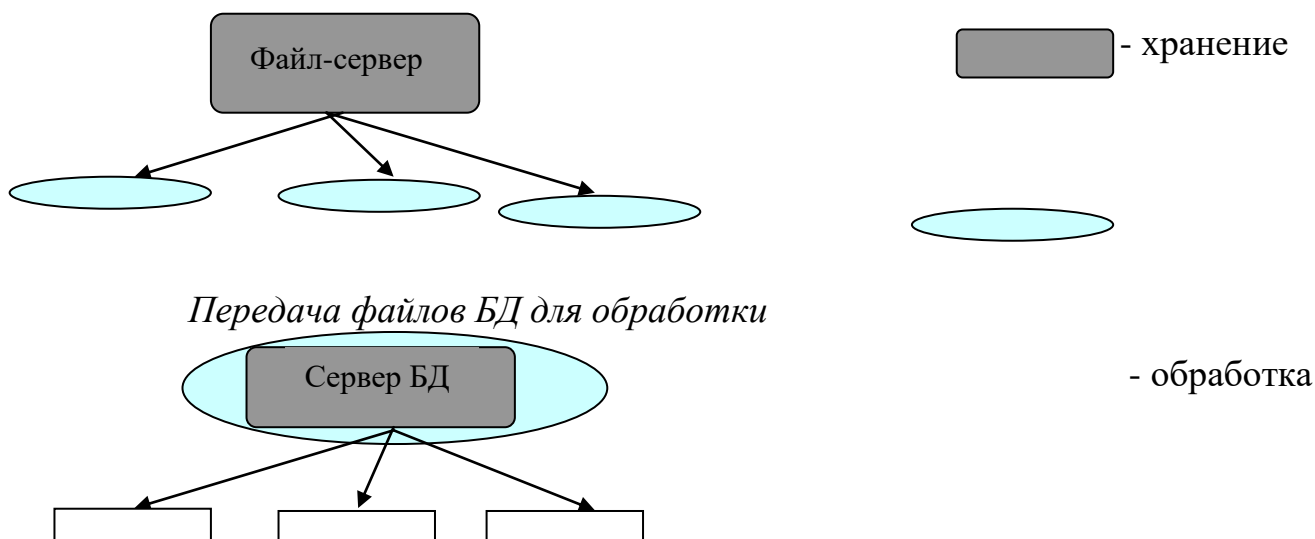


Рис. 1. Схема обработки информации в БД по принципу *файл-сервер*

Транспортировка извлеченных данных из БД

Рабочие станции

Рис. 2. Схема обработки информации в БД по принципу *клиент-сервер*

1.2. Структурные элементы базы данных

Понятие базы данных тесно связано с такими понятиями структурных элементов, как поле, запись, файл (таблица) (рис. 3).

Поле — элементарная единица логической организации данных, которая соответствует неделимой единице информации — реквизиту. Для описания поля используются следующие *характеристики*:

имя, например, Фамилия, Имя, Отчество, Дата рождения;

тип, например, символьный, числовой, календарный;

длина, например, 15 байт, причем будет определяться максимально возможным количеством символов;

точность для числовых данных, например, два десятичных знака для отображения дробной части числа.

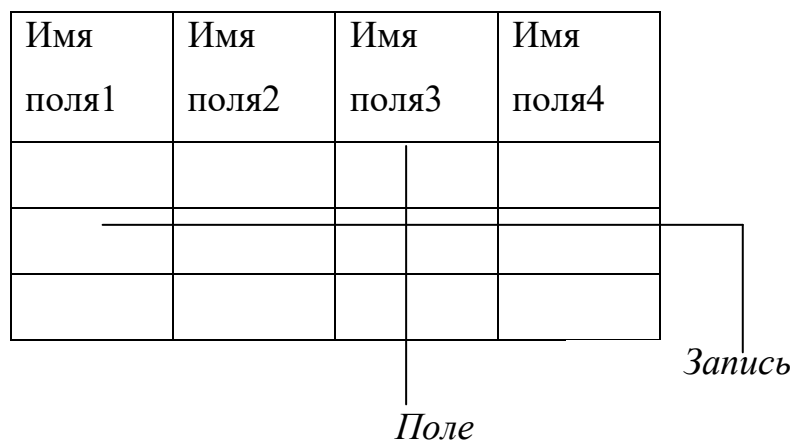


Рис. 3. Основные структурные элементы БД

Запись — совокупность логически связанных полей. Экземпляр записи — отдельная реализация записи, содержащая конкретные значения ее полей.

Файл (таблица) — совокупность экземпляров записей одной структуры. Описание логической структуры записи файла содержит последовательность расположения полей записи и их основные характеристики, как это показано на рис. 4.

Имя файла					
Поле		Признак ключа	Формат поля		
Имя (обозначение)	Полное наименование		Тип	Длина	Точность (для чисел)
имя 1					
...					
имя n					

Рис. 4. Описание логической структуры записи файла

В структуре записи файла указываются поля, значения которых являются *ключами первичными* (ПК), которые идентифицируют экземпляр записи, и *вторичными* (ВК), которые выполняют роль поисковых или группировочных признаков (по значению вторичного ключа можно найти несколько записей).

Пример 1. На **Рис. 5** приведен пример описания логической структуры записи файла (таблицы) СТУДЕНТ. Структура записи файла СТУДЕНТ линейная, она содержит записи фиксированной длины. Повторяющиеся группы значений полей в записи отсутствуют. Обращение к значению поля производится по его номеру.

Имя файла: СТУДЕНТ					
Поле		Признак ключа	Формат поля		
Обозначение	Наименование		Тип	Длина	Точность
Номер	№ личного дела	*	Симв	5	
Фамилия	Фамилия студента		Симв	15	
Имя	Имя студента		Симв	10	
Отчество	Отчество студента		Симв	15	
Дата	Дата рождения		Дата	8	

Рис. 5. Описание логической структуры записи файла СТУДЕНТ

1.3. Модели БД

Ядром любой базы данных является модель данных. Модель данных представляет собой множество структур данных, ограничений целостности и операций манипулирования данными. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

Модель данных — совокупность структур данных и операций их обработки.

СУБД основывается на использовании иерархической, сетевой или реляционной модели, на комбинации этих моделей или на некотором их подмножестве.

Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево).

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь. *Узел* — это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. Количество деревьев в базе данных определяется числом корневых записей.

К каждой записи базы данных существует только один (иерархический) путь от корневой записи.

В **сетевой** структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

Понятие *реляционный* (англ. *relation* — отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда.

Реляционные модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая *реляционная таблица* представляет собой двумерный массив и обладает следующими свойствами:

- каждый элемент таблицы — один элемент данных;
- все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Пример 2. Реляционной таблицей можно представить информацию о студентах, обучающихся в колледже (рис. 6).

Поименный №	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.92	111
16593	Петрова	Анна	Владимировна	15.03.93	102
16693	Анохин	Андрей	Борисович	14.04.93	111

Рис. 6. Пример реляционной таблицы

Отношения представлены в виде *таблиц*, строки которых соответствуют кортежам или *записям*, а столбцы — атрибутам отношений, доменам, *полям*.

Поле, каждое значение которого однозначно определяет соответствующую запись, называется *простым ключом* (ключевым полем).

Если записи однозначно определяются значениями нескольких полей, то такая таблица базы данных имеет *составной ключ*. В примере, показанном на рис. 6, ключевым полем таблицы является "Поименный №".

Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей); в противном случае нужно завести в структуру первой таблицы *внешний ключ* — ключ второй таблицы.

СРС 1. Технология хранения, поиска и сортировки информации

План СРС:

1. ИПС как средство обработки информации.
2. Компьютерные программы, автоматизирующие обработку больших объемов информации.
3. БД. Основные понятия БД. Классификация БД.
4. СУБД как средство ведения БД.
5. Основные объекты СУБД и их назначение.
6. Работа с СУБД: создание таблиц, установка связей, создание форм, запросов и отчетов.

Литература:

1. Боргено Я.Я., Кирсанова М.В. Офисные технологии: Учебное пособие.- М.:ИНФРА- М.; Новосибирск: Сибирское соглашение, 2006.- 256 с.
2. Введение в правовую информатику: Учебник для вузов/Под общ. Ред. Д.Б. Новикова, В.Л. Камынина. - М.: ООО НПО "Вычислительная математика и информатика", 2004. - 313 с.
3. Угринович «Информатика и информационные технологии» Базовый курс: Издательство "ДРОФА».

1.4. Функциональные возможности СУБД

Системой управления базами данных называют *программную систему*, предназначенную для создания на ЭВМ общей базы данных, используемой для решения множества задач. Подобные системы служат для поддержания базы данных в актуальном состоянии и обеспечивают эффективный доступ пользователей к содержащимся в ней данным в рамках предоставленных пользователям полномочий.

СУБД предназначена для централизованного управления базой данных в интересах всех работающих в этой системе.

По степени универсальности различают два класса СУБД:

- системы общего назначения;
- специализированные системы.

СУБД общего назначения не ориентированы на какую-либо предметную область или на информационные потребности какой-либо группы пользователей. Каждая система такого рода реализуется как программный продукт, способный функционировать на некоторой модели ЭВМ в определенной операционной системе, и поставляется многим пользователям как коммерческое изделие. Такие СУБД обладают средствами настройки на работу с конкретной базой данных. Использование СУБД общего назначения в качестве инструментального средства для создания автоматизированных информационных систем, основанных на технологии баз данных, позволяет существенно сокращать сроки разработки, экономить трудовые ресурсы. Этим СУБД присущи развитые функциональные возможности, и даже определенная функциональная избыточность.

Специализированные СУБД создаются в редких случаях при невозможности или нецелесообразности использования СУБД общего назначения.

СУБД общего назначения — это сложные программные комплексы, предназначенные для выполнения всей совокупности функций, связанных с созданием и эксплуатацией базы данных информационной системы.

Рынок программного обеспечения ПК располагает большим числом разнообразных по своим функциональным возможностям коммерческих систем управления базами данных общего назначения, а также средствами их окружения практически для всех массовых моделей машин и для различных операционных систем.

Используемые в настоящее время СУБД обладают средствами обеспечения целостности данных и надежной безопасности, что дает возможность разработчикам гарантировать большую безопасность данных при меньших затратах сил на низкоуровневое программирование. Продукты, функционирующие в среде WINDOWS, выгодно отличаются удобством пользовательского интерфейса и встроенными средствами повышения производительности. Некоторые СУБД-лидеров на рынке разработчиков информационных систем: dBASE, компании Borland International; Microsoft Access; Microsoft FoxPro корпорации Microsoft; Paradox компании Borland.

1.5. Команды для выполнения типовых операций

Типовая структура интерфейса

При работе с СУБД на экран выводятся рабочее поле и панель управления. Панель управления при этом включает меню, вспомогательную область управления и строку подсказки. Расположение этих областей на экране может быть произвольным и зависит от особенностей конкретной программы. Некоторые СУБД позволяют выводить на экран окно директив (командное окно) или строку команд. Познакомиться с видом экрана таких программных средств можно на примере окна СУБД Access.

Строка меню содержит основные режимы программы. Выбрав один из них, пользователь получает доступ к ниспадающему подменю, содержащему перечень входящих в него команд. В результате выбора некоторых команд ниспадающего меню появляются дополнительные подменю.

Вспомогательная область управления включает:

- строку состояния;
- панели инструментов;
- вертикальную и горизонтальную линейки прокрутки.

В *строке состояния* (статусной строке) пользователь найдет сведения о текущем режиме работы программы, имени файла текущей базы данных и т. п. *Панель инструментов* (пиктографическое меню) содержит определенное количество кнопок (пиктограмм), предназначенных для быстрой активизации выполнения определенных команд меню и функций программы. Чтобы представить на экране области таблицы базы данных, формы или отчета, которые на нем в настоящий момент не отображены, используют *вертикальную и горизонтальную линейки* прокрутки.

Строка подсказки предназначена для выдачи сообщений пользователю относительно его возможных действий в данный момент.

Важная особенность СУБД — использование буфера промежуточного хранения при выполнении ряда операций. Буфер используется при выполнении команд копирования и перемещения для временного хранения копируемых или перемещаемых данных, после чего они направляются по новому адресу. При удалении данных они также помещаются в буфер. Содержимое буфера сохраняется до тех пор, пока в него не будет записана новая порция данных.

Программы СУБД имеют достаточное количество команд, у каждой из которых возможны различные параметры (опции). Такая система команд совместно с дополнительными опциями образует *меню* со своими особенностями для каждого типа СУБД. Выбор определенной команды из меню производится одним из следующих двух способов:

- наведением курсора на выбранную в меню команду при помощи клавиш управления курсором и нажатием клавиши ввода;
- вводом с клавиатуры первой буквы выбранной команды.

Получить дополнительную информацию о командах, составляющих меню СУБД, и их использовании можно, войдя в режим помощи.

Несмотря на особенности СУБД, совокупность команд, предоставляемых в распоряжение пользователю некоторой усредненной системой управления базами данных, может быть разбита на следующие типовые группы:

- команды для работы с файлами;
- команды редактирования;
- команды форматирования;
- команды для работы с окнами;
- команды для работы в основных режимах СУБД (таблица, форма, запрос, отчет);
- получение справочной информации.

1.6. Этапы проектирования баз данных

При разработке БД можно выделить следующие этапы работы.

I этап. Постановка задачи.

На этом этапе формируется задание по созданию БД. В нем подробно описывается состав базы, назначение и цели ее создания, а также перечисляется, какие виды работ предполагается осуществлять в этой базе данных (отбор, дополнение, изменение данных, печать или вывод отчета и т.д.).

II этап. Анализ объекта.

На этом этапе рассматривается, из каких объектов может состоять БД, каковы свойства этих объектов. После разбиения БД на отдельные объекты необходимо рассмотреть свойства каждого из этих объектов, или, другими словами, установить, какими параметрами описывается каждый объект. Все эти сведения можно располагать в виде отдельных записей и таблиц. Далее необходимо рассмотреть тип данных каждой отдельной единицы записи. Сведения о типах данных также следует занести в составляемую таблицу.

III этап. Синтез модели.

На этом этапе по проведенному выше анализу необходимо выбрать определенную модель БД. Далее рассматриваются достоинства и недостатки каждой модели и сопоставляются с требованиями и задачами создаваемой БД. После такого анализа выбирают ту модель, которая сможет максимально обеспечить реализацию поставленной задачи. После выбора модели необходимо нарисовать ее схему с указанием связей между таблицами или узлами.

IV этап. Выбор способов представления информации и программного инструментария.

После создания модели необходимо, в зависимости от выбранного программного продукта, определить форму представления информации.

В большинстве СУБД данные можно хранить в двух видах:

- с использованием форм;
- без использования форм.

Форма – это созданный пользователем графический интерфейс для ввода данных в базу.

V этап. Синтез компьютерной модели объекта.

В процессе создания компьютерной модели можно выделить некоторые стадии, типичные для любой СУБД.

Стадия 1. Запуск СУБД, создание нового файла базы данных или открытие созданной ранее базы.

Стадия 2. Создание исходной таблицы или таблиц.

Создавая исходную таблицу, необходимо указать имя и тип каждого поля. Имена полей не должны повторяться внутри одной таблицы. В процессе работы с БД можно дополнять таблицу новыми полями. Созданную таблицу необходимо сохранить, дав ей имя, уникальное в пределах создаваемой базы.

При проектировании таблиц, рекомендуется руководствоваться следующими основными принципами:

1. Информация в таблице не должна дублироваться. Не должно быть повторений и между таблицами. Когда определенная информация хранится только в одной таблице, то и изменять ее придется только в одном месте. Это делает работу более эффективной, а также исключает возможность несовпадения информации в разных таблицах. Например, в одной таблице должны содержаться адреса и телефоны клиентов.

2. Каждая таблица должна содержать информацию только на одну тему. Сведения на каждую тему обрабатываются намного легче, если они содержатся в независимых друг от друга таблицах. Например, адреса и заказы клиентов лучше хранить в разных таблицах, с тем, чтобы при удалении заказа информация о клиенте осталась в базе данных.

3. Каждая таблица должна содержать необходимые поля. Каждое поле в таблице должно содержать отдельные сведения по теме таблицы. Например, в таблице с данными о клиенте могут содержаться поля с названием компании, адресом, городом, страной и номером телефона. При разработке полей для каждой таблицы необходимо помнить, что каждое поле должно быть связано с темой таблицы. Не рекомендуется включать в таблицу данные, которые являются результатом выражения. В таблице должна присутствовать вся необходимая информация. Информацию следует разбивать на наименьшие логические единицы (Например, поля "Имя" и "Фамилия", а не общее поле "Имя").

4. База данных должна иметь первичный ключ. Это необходимо для того, чтобы СУБД могла связать данные из разных таблиц, например, данные о клиенте и его заказы.

Стадия 3. Создание экранных форм.

Первоначально необходимо указать таблицу, на базе которой будет создаваться форма. Ее можно создавать при помощи мастера форм, указав, какой вид она должна иметь, или самостоятельно. При создании формы

можно указывать не все поля, которые содержит таблица, а только некоторые из них. Имя формы может совпадать с именем таблицы, на базе которой она создана. На основе одной таблицы можно создать несколько форм, которые могут отличаться видом или количеством используемых из данной таблицы полей. После создания форму необходимо сохранить. Созданную форму можно редактировать, изменяя местоположение, размеры и формат полей.

Стадия 4. Заполнение БД.

Процесс заполнения БД может проводиться в двух видах: в виде таблицы и в виде формы. Числовые и текстовые поля можно заполнять в виде таблицы, а поля типа MEMO и OLE – в виде формы.

VI этап. Работа с созданной базой данных.

Работа с БД включает в себя следующие действия:

- поиск необходимых сведений;
- сортировка данных;
- отбор данных;
- вывод на печать;
- изменение и дополнение данных.

Перед началом детального обсуждения способов проектирования баз данных необходимо отметить, что любая база данных является составной частью некой информационной системы (ИС), которая подразумевает не только хранение данных, но и их обработку. Поэтому, проектированию данных всегда сопутствует (а чаще предшествует) проектирование алгоритмов их использования. Здесь мы рассмотрим все этапы проектирования информационной системы: от функционального моделирования предметной области, до построения структуры реляционной базы данных.

При проектировании базы данных решаются две основные *проблемы*:

- каким образом отобразить объекты предметной области в абстрактные объекты модели данных, чтобы это отображение не

противоречило семантике предметной области и было, по возможности, лучшим (эффективным, удобным и т.д.)? Часто эту проблему называют проблемой логического проектирования баз данных;

- как обеспечить эффективность выполнения запросов к базе данных, т.е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создания каких дополнительных структур (например, индексов) потребовать и т. д.? Эту проблему обычно называют проблемой физического проектирования баз данных.

В случае реляционных баз данных трудно предложить какие-либо общие рецепты по части физического проектирования. Здесь слишком многое зависит от используемой СУБД. Поэтому мы ограничимся вопросами логического проектирования реляционных баз данных, которые существенны при использовании любой реляционной СУБД.

Будем считать, что проблема проектирования реляционной базы данных состоит в обоснованном принятии решений о том, из каких отношений должна состоять БД и какие атрибуты должны быть у этих отношений.

Рассмотрим классический подход, при котором весь процесс проектирования базы данных осуществляется в терминах реляционной модели данных методом последовательных приближений к удовлетворительному набору схем отношений. Исходной точкой является представление предметной области в виде одного или нескольких отношений, и на каждом шаге проектирования производится некоторый набор схем отношений, обладающих «улучшенными» свойствами. Процесс проектирования представляет собой *процесс нормализации* схем отношений, причем каждая следующая нормальная форма обладает свойствами, в некотором смысле, лучшими, чем предыдущая.

Каждой нормальной форме соответствует определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений. Примером может служить ограничение *первой нормальной формы* – значения всех атрибутов отношения атомарны. Поскольку требование *первой нормальной формы* является базовым требованием классической реляционной модели данных, мы будем считать, что исходный набор отношений уже соответствует этому требованию.

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- *первая нормальная форма (1NF);*
- *вторая нормальная форма (2NF);*
- *третья нормальная форма (3NF);*
- *нормальная форма Бойса-Кодда (BCNF);*
- *четвертая нормальная форма (4NF);*
- *пятая нормальная форма, или нормальная форма проекции-соединения (5NF или PJ/NF).*

Основные свойства нормальных форм состоят в следующем:

- каждая следующая нормальная форма в некотором смысле лучше предыдущей нормальной формы;
- при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.

В основе процесса проектирования лежит метод нормализации, т. е. декомпозиции отношения, находящегося в предыдущей нормальной форме, на два или более отношений, которые удовлетворяют требованиям следующей нормальной формы.

Первые шаги *процесса нормализации* учитывают функциональные зависимости между атрибутами отношений. Хотя мы и называем эти шаги первыми, именно они имеют основную практическую важность, поскольку

позволяют получить схему реляционной базы данных, в большинстве случаев удовлетворяющую потребности приложений.

1.7. Создание структуры таблиц базы данных

При формировании новой таблицы базы данных работа с СУБД начинается с создания структуры таблицы. Этот процесс включает определение перечня полей, из которых состоит каждая запись таблицы, а также типов и размеров полей.

Практически все используемые СУБД хранят данные следующих типов: текстовый (символьный), числовой, календарный, логический, примечание. Некоторые СУБД формируют поля специального типа, содержащие уникальные номера записей и используемые для определения ключа.

СУБД, предназначенные для работы в Windows, могут формировать поля типа объекта OLE, которые используются для хранения рисунков, графиков, таблиц.

Если обрабатываемая база данных включает несколько взаимосвязанных таблиц, то необходимо определение ключевого поля в каждой таблице, а также полей, с помощью которых будет организована связь между таблицами.

Создание структуры таблицы не связано с заполнением таблиц данными, поэтому эти две операции можно разнести во времени.

1.8. Ввод и редактирование данных

Заполнение таблиц данными возможно как непосредственным вводом данных, так и в результате выполнения программ и запросов.

Практически все СУБД позволяют вводить и корректировать данные в таблицах двумя способами:

- с помощью предоставляемой по умолчанию стандартной формы в виде таблицы;

- с помощью экранных форм, специально созданных для этого пользователем.

СУБД, работающие с Windows, позволяют вводить в созданные экранные формы рисунки, узоры, кнопки. Возможно построение форм, наиболее удобных для работы пользователя, включающих записи различных связанных таблиц базы данных. Пример экрана с формой ввода представлен на рис. 7.

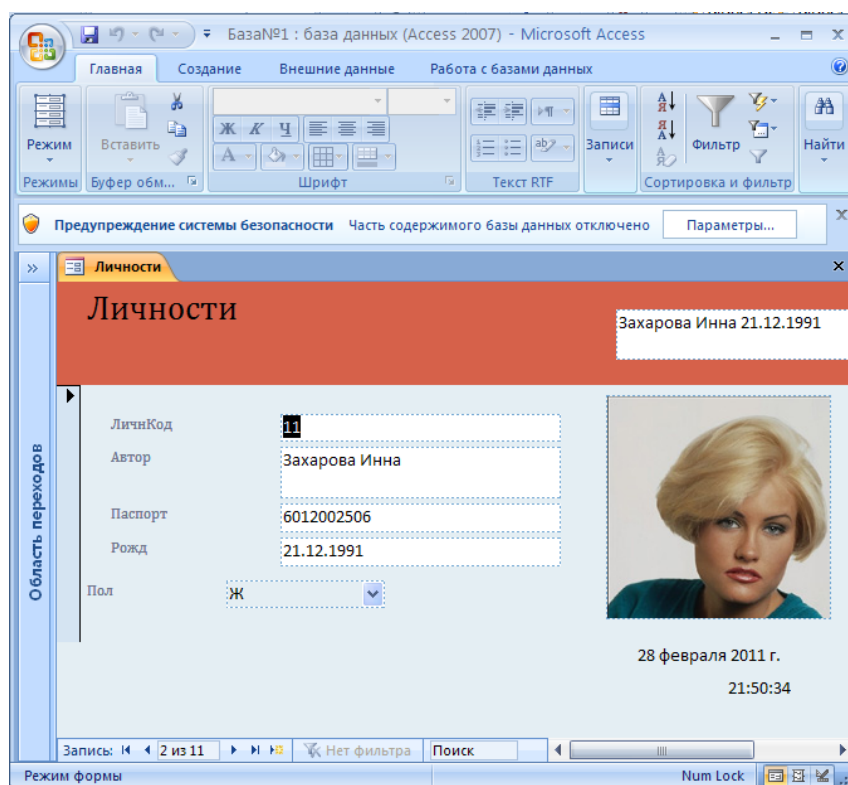


Рис. 7. Вид экрана СУБД с формой ввода для связанных таблиц базы данных.

1.9. Обработка данных, содержащихся в таблицах

Обрабатывать информацию, содержащуюся в таблицах базы данных, можно путем использования запросов или в процессе выполнения специально разработанной программы.

Конечный пользователь получает при работе с СУБД такое удобное средство обработки информации, как запросы. *Запрос* представляет собой инструкцию на отбор записей.

Большинство СУБД разрешают использовать запросы следующих типов:

- запрос-выборка, предназначенный для отбора данных, хранящихся в таблицах, и не изменяющий эти данные;
- запрос-изменение, предназначенный для изменения или перемещения данных; к этому типу запросов относятся: запрос на добавление записей, запрос на удаление записей, запрос на создание таблицы, запрос на обновление;
- запрос с параметром, позволяющий определить одно или несколько условий отбора во время выполнения запроса.

Самым распространенным типом запроса является запрос на выборку.

Результатом выполнения запроса является таблица с временным набором данных (динамический набор). Записи динамического набора могут включать поля из одной или нескольких таблиц базы данных. На основе запроса можно построить отчет или форму.

1.10. Вывод информации из базы данных

Практически любая СУБД позволяет вывести на экран и принтер информацию, содержащуюся в базе данных, из режимов таблицы или формы. Такой порядок вывода данных может использоваться только как черновой вариант, так как позволяет выводить данные только точно в таком же виде, в каком они содержатся в таблице или форме.

Каждый пользователь, работающий с СУБД, имеет возможность использования специальных средств построения *отчетов* для вывода данных. Используя специальные средства создания отчетов, пользователь получает следующие дополнительные возможности вывода данных:

- включать в отчет выборочную информацию из таблиц базы данных;
- добавлять информацию, не содержащуюся в базе данных;
- при необходимости выводить итоговые данные на основе информации базы данных;

- располагать выводимую в отчете информацию в любом, удобном для пользователя виде (вертикальное или горизонтальное расположение полей);
- включать в отчет информацию из разных связанных таблиц базы данных.

Вид экрана СУБД с отчетом, построенным по данным связанных таблиц базы данных, показан на рис. 8.

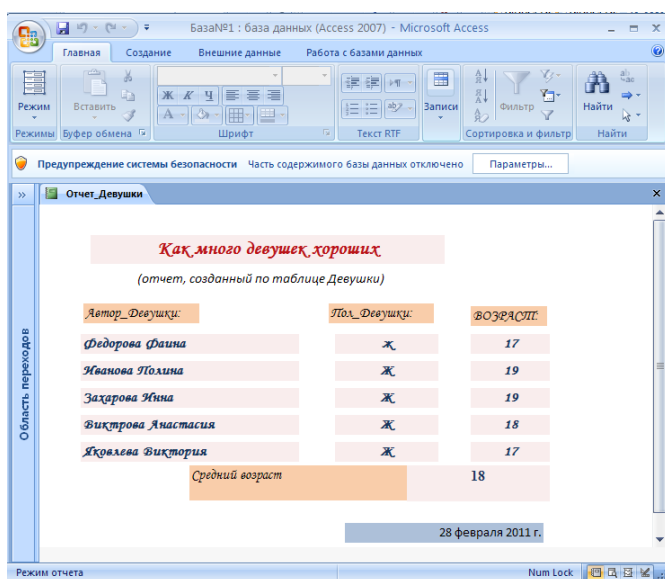


Рис. 8. Вид экрана СУБД с отчетом

СРС 2. Создание структуры многотабличной базы данных с учетом будущей профессии.

Таблица БД должна содержать не менее 15 записей для демонстрации наглядной обработки информации, хранящейся в БД.

1. Создать таблицу *Личности*. Предполагаемые поля: *Автор*, *Пол*, *Паспорт*.
2. Заполнить таблицу данными.
2. Создать новую таблицу *Адреса*, имена и типы предполагаемых полей, приведены на рис.1.

Личный код гражданина (Текст 5)
Почтовый индекс (Текст 6)
Название города (Текст 20)
Название улицы (Текст 20)
Номер дома (Текст 8)
Корпус (Текст 3)
Номер квартиры (Текст 5)

Рис.1

3. Создать таблицу *Блюда из мяса птицы*, имена и типы предполагаемых полей, приведены в таблице1:

Имя поля	Тип поля	Размер поля
<i>Автор</i>	Текстовое	25 символов
<i>Категория/_вид</i>	Текстовое	25 символов
<i>Стоимость</i>	Денежный	Число десятичных знаков 2
<i>Калорийность</i>	Число	Целое
<i>Белки</i>	Число	Целое
<i>Жиры</i>	Число	Целое
<i>Углеводы</i>	Число	Целое
<i>Ароматизаторы</i>	Число	Целое

4. Создать таблицу *Ингредиенты*, имена и типы предполагаемых полей, приведены в таблице2:

Имя поля	Тип поля	Размер поля
<i>Название блюда</i>	Текстовое	25 символов
<i>Картофель</i>	Число	<i>байт</i>
<i>Рис</i>	Число	<i>байт</i>
<i>Гречка</i>	Число	<i>байт</i>
<i>Мука</i>	Число	<i>байт</i>
<i>Перец болгарский</i>	Число	<i>байт</i>
<i>Макаронные изделия</i>	Число	<i>байт</i>
<i>Мясо</i>	Число	<i>целое</i>

<i>Яйца</i>	Число	<i>байт</i>
<i>Лук</i>	Число	<i>байт</i>
<i>Чеснок</i>	Число	<i>байт</i>
<i>Морковь</i>	Число	<i>байт</i>

5. Графически изобразить возможные связи между таблицами, предполагаемые связи изображены на рис.2.

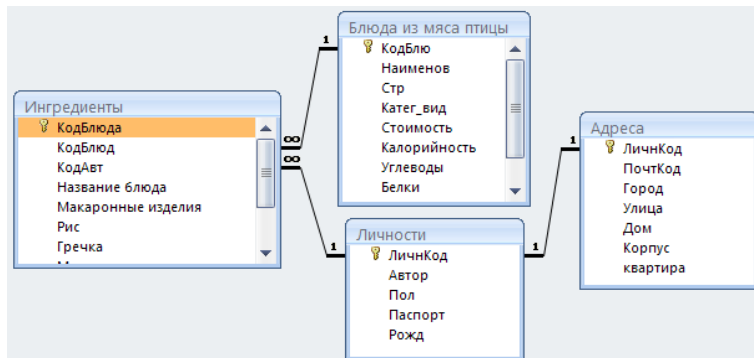


Рис.2

Литература:

1. Боргено Я.Я., Кирсанова М.В. Офисные технологии: Учебное пособие.- М.:ИНФРА- М.; Новосибирск: Сибирское соглашение, 2008.- 256 с.
2. Введение в правовую информатику: Учебник для вузов/Под общ. Ред. Д.Б. Новикова, В.Л. Камынина. - М.: ООО НПО "Вычислительная математика и информатика", 2007. - 313 с.
3. Информатика. Базовый курс / Симонович С.В. и др. - СПб : Издательство "Питер", 2006.
4. Макарова Н.В. Информатика. - М.: Финансы и статистика, 2006г.
5. Михеева Е.В. Практикум по информационным технологиям в профессиональной деятельности: учебное пособие.- М.: ТК Велби, изд-во Проспект, 2007.-288с.
6. Угринович «Информатика и информационные технологии» Базовый курс: Издательство "ДРОФА».

Часть II. Задания к лабораторным работам

Лабораторная работа № 1

Создание локальных баз данных

Задание 1. Создание таблиц в режиме ввода данных

Цель: Первоначальное знакомство с СУБД MS Access. Изучение способа создания таблиц в режиме ввода данных (режиме таблицы). Ввод и редактирование данных в таблицах.

1. Создать в папке Студенческие работы личную папку.

2. Запустить СУБД MS Access.

3. Создать файл базы данных.


3.1. Перейти в режим *Создания новой базы данных*.

3.2. В диалоговом окне *Файл новой базы данных* определить в качестве имени файла базы данных **База№1**, а в качестве ее местонахождения задать Вашу личную папку.

4. В открывшемся окне *База№1: база данных* **просмотреть все вкладки** и убедиться в том, что база данных пуста и не содержит ни одного объекта.

5. Выбрать вкладку *Таблицы* и создать новую таблицу в режиме *создания новой пустой таблицы*.

5.1. Задать названия полей (*Автор, Пол, Паспорт*), как показано на рис.1.1.

5.2. Закончить создание таблицы и закрыть ее (). Сохранить таблицу в активной базе данных под именем **Личности**.

6. Открыть таблицу **Личности** для работы с ней.

6.1. Ввести в таблицу данные, например, рис.1.2.

6.2. Научиться перемещаться по пространству таблицы (по строкам-записям и по столбцам - полям), пользуясь органами управления окна таблицы.

6.3. Сохранить введенные данные и закрыть таблицу.

6.4. Закрыть базу данных *База№1*.

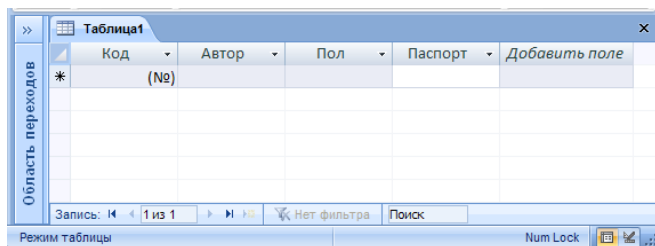


Рис.1.1

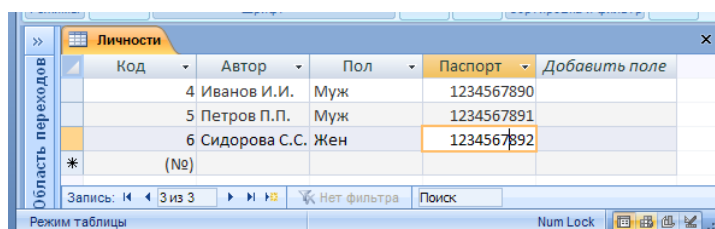


Рис.1.2

7. Изменить структуру таблицы *Личности*, переименовав поле *Код* в поле *Личный код*. В результате выполнения этого пункта таблица должна выглядеть так, как представлено на рис.1.3.

7.1. Открыть базу данных *База№1*.

7.2. Открыть таблицу *Личности*.

7.3. Переименовать поле *Код* в поле *Личный Код*.

7.5. Поменять местами поля *Пол* и *Паспорт*.

7.6. Сохранить введенные данные и закрыть таблицу.

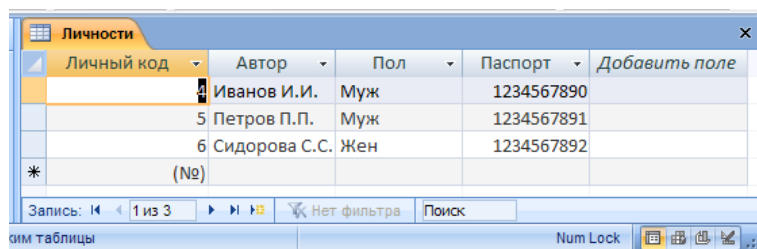


Рис.1.3

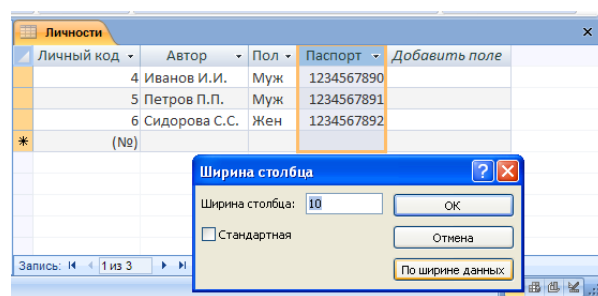


Рис.1.4

8. Изменить внешний вид таблицы с помощью контекстного меню. Пример выполнения представлен на рис.1.4.

8.1. Уменьшить ширину полей *Личный Код*, *Автор*, *Паспорт*, *Пол* следующим образом:

- поле *Автор, Паспорт* - по ширине данных,
- поля *Пол* - до ширины 5.5 символа,
- поле *Личный Код* - с помощью мыши («на глазок»), как

представлено на рис.1.4.

8.2. Отметить всю таблицу и установить для нее новый тип и размер шрифта. При необходимости изменить высоту строк, содержащих записи таблицы.

8.3. Скрыть (сделать неотображаемыми в таблице) поля *Личный Код* и *Пол*.

8.4. Восстановить (вновь сделать видимыми) поля *Личный Код* и *Пол*.

9. Отредактировать содержимое таблицы *Личности*.

9.1. Перейти к полю *Паспорт*.

9.2. Выделить первые две цифры (12) и скопировать их в буфер обмена.

9.3. С помощью команды *Заменить* заменить их на цифры 21.

9.4. Выделить содержимое полей *Пол* и *Паспорт* второй записи.

9.5. Скопировать содержимое этих полей в буфер обмена и вставить вместо соответствующих данных третьей записи.

9.6. Выделить первую запись (левее самого первого поля на полосе маркирования – черная стрелочка) и поместить (вырезать) ее в буфер обмена.

9.7. Вставить содержимое буфера обмена в конец таблицы, как новую запись.

9.8. Выделить вторую запись и удалить ее.

9.9. Закрывать файл базы данных и **Сохранить базу данных *База№1*** на диске для дальнейшего использования.

Задание 2. Создание таблиц в режиме Конструктора

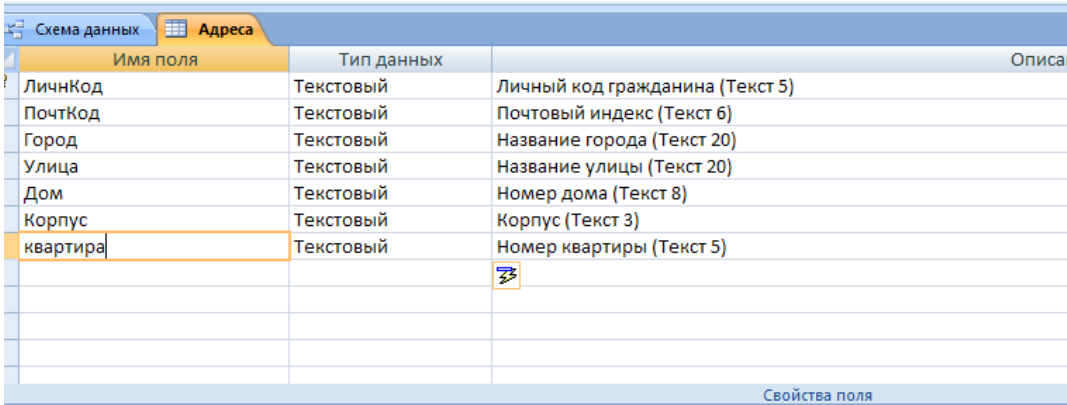
Цель: Изучение способов формирования структуры таблиц в режиме «Конструктора таблиц».

1. Запустить СУБД MS Access. Открыть базу данных *База№1*.

1.1. Просмотреть содержимое вкладки *Таблицы*. Убедиться, что в базе данных *База№1* содержится лишь одна таблица *Личности*.

2. Создать новую таблицу *Адреса*, пользуясь *Конструктором таблиц*.

2.1. Прототип этой таблицы, в котором представлены имена и типы полей, приведен на рис.2.1.



Имя поля	Тип данных	Описание
ЛичнКод	Текстовый	Личный код гражданина (Текст 5)
ПочтКод	Текстовый	Почтовый индекс (Текст 6)
Город	Текстовый	Название города (Текст 20)
Улица	Текстовый	Название улицы (Текст 20)
Дом	Текстовый	Номер дома (Текст 8)
Корпус	Текстовый	Корпус (Текст 3)
квартира	Текстовый	Номер квартиры (Текст 5)

Рис.2.1

2.2.1. При создании таблицы *Адреса* обязательно заполнять раздел *Описание*.

2.2.2. В разделе *Свойства поля* вводить только значения параметра «Размер поля» в соответствии с размерами полей, указанными в описании поля, например, для поля *Улица* тип поля - *текстовый*, а размер поля - 20 символов

2.3. Сохранить созданную таблицу в составе базы данных *База№1* под именем *Адреса*, не задавая ключевых полей.

3. Закрывать таблицу *Адреса*.

4. Открыть с помощью *Конструктора таблиц* таблицу *Личности*.

4.1. Пользуясь возможностями *Конструктора таблиц*, просмотреть *типы* и *размеры* всех полей таблицы *Личности*.

4.2. Добавить в таблицу *Личности* новое поле *Рожд*, имеющее тип *Дата/время* с кратким форматом даты (10.12.10)


4.3. Изменить типы для некоторых полей в таблице *Личности* (табл. 2.2.), сняв при необходимости значение *Ключевое поле*.

Таблица 2.2

Имя поля	Тип поля	Размер поля
<i>Личный Код</i>	Текстовое	5 символов
<i>Автор</i>	Текстовое	25 символов
<i>Пол</i>	Текстовое	3 символа
<i>Паспорт</i>	Текстовое	12 символов
<i>Рожд</i>	Дата/Время	Краткий формат даты

4.4. Сохранить измененную таблицу *Личности* под старым именем в составе базы данных *База№1*.

4.5. Закрыть таблицы *Личности* и *Адреса*.

5. Открыть окно «Схема данных» (Меню «Сервис» – «Схема данных» или кнопка ).

5.1. В окне *Добавление таблицы* из предлагаемого списка, состоящего из двух ранее созданных таблиц, поочередно выбрать каждую таблицу и добавить ее в схему данных. Закрыть окно *Добавление таблицы*.

5.2. Просмотреть схему данных, состоящую из двух *несвязанных* таблиц *Личности* и *Адреса*, которые входят в состав базы данных *База№1*. Установить размеры индивидуальных окон для каждой таблицы и для всей схемы данных, такие, как показаны на рис.2.2.

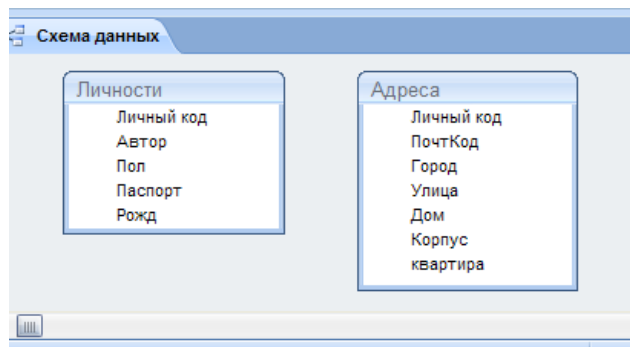


Рис.2.2

5.3. Сохранить созданную структуру и закрыть окно *Схема данных*.

6. Сохранить базу данных на диске для дальнейшего использования.

Задание 3. Создание таблиц в режиме Мастера.

Цель: Изучение способа формирования таблиц в режиме «Мастера таблиц». Проверка вводимых данных с использованием Условий на значения.

Использование Мастера таблиц

1. Запустить СУБД MS Access. Открыть базу данных *База№1*.

1.1. Просмотреть содержимое вкладки *Таблицы*. Убедиться, что в базе данных *База№1* содержатся две таблицы *Личности* и *Адреса*.

2. Создать таблицу *Блюда из мяса птицы и Ингредиенты*, пользуясь *Мастером таблиц*.

2.1. Перейти в режим *Создания таблицы*. В окне *Таблицы* выбрать режим создания таблицы с помощью *Шаблоны таблиц*.

2.2. В диалоговом окне *Создание таблицы* из группы таблиц, предназначенных для делового применения, выбрать в качестве образца таблицу *События*.

2.3. **В качестве образцов полей оставить в новой таблице поля *Код*, *Название* и *Место*, остальные поля удалить.**

2.4. Определить для новой таблицы имя ***Блюда из мяса птицы***.

2.5. Не задавая связей ни с какими имеющимися таблицами, закончить создание таблицы, указав, что после создания таблицы предполагается изменить структуру созданной таблицы (скорректировать названия полей и уточнить их тип и размер).

Задание Условий на значения

2.6. Внести в структуру таблицы ***Блюда из мяса птицы*** следующие изменения.

2.6.1. Для поля *Код*:

- изменить имя поля на *КодБлюда*;
- изменить тип поля на *Текстовый*;
- установить размер поля в 5 символов;

- сделать поле НЕ ключевым;
- изменить подпись поля на «Код Блюда» (с пробелом между словами в отличие от имени поля Код Блюда).

2.6.2. Для поля *Название*:

- изменить имя поля на Наименов;
- изменить тип поля на Текстовый;
- установить размер поля в 15 символов;
- изменить подпись поля на «Название Блюда» (с пробелом между словами).

2.6.3. Для поля *Место*:

- изменить имя поля на *Стр*;
- изменить тип поля на Текстовый;
- установить размер поля в 10 символов;
- подпись поля «*Страна*».

2.6.4. Добавить поля:

Имя поля	Тип поля	Размер поля
<i>Автор</i>	Текстовое	25 символов
<i>Катег_вид</i>	Текстовое	25 символов
<i>Стоимость</i>	Денежный	Число десятичных знаков 2
<i>Калорийность</i>	Число	Целое
<i>Белки</i>	Число	Целое
<i>Жиры</i>	Число	Целое
<i>Углеводы</i>	Число	Целое
<i>Ароматизаторы</i>	Число	Целое

- установить такое Условие на значение поля *Калорийность*, чтобы возможными значениями этого поля были: 100, 150, 200, 250, 300. Условие записать в виде логического выражения, связывающего допустимые значения;

● задать в качестве сообщения об ошибке следующий текст:
«Недопустимое Значение в поле "Калорийность".

2.6.4. Для всех полей таблицы *Блюда из мяса птицы* заполнять раздел *Описание*.

2.6.5. Сохранить сделанные изменения в структуре таблицы *Блюда из мяса птицы* и закрыть ее.

3. Просмотреть все созданные ранее таблицы (*Личности* и *Адреса*), пользуясь *Конструктором таблиц*, создать таблицу *Ингредиенты*:

Имя поля	Тип поля	Размер поля
<i>Название блюда</i>	Текстовое	25 символов
<i>Картофель</i>	Число	<i>байт</i>
<i>Рис</i>	Число	<i>байт</i>
<i>Гречка</i>	Число	<i>байт</i>
<i>Мука</i>	Число	<i>байт</i>
<i>Перец болгарский</i>	Число	<i>байт</i>
<i>Макаронные изделия</i>	Число	<i>байт</i>
<i>Мясо</i>	Число	<i>целое</i>
<i>Яйца</i>	Число	<i>байт</i>
<i>Лук</i>	Число	<i>байт</i>
<i>Чеснок</i>	Число	<i>байт</i>
<i>Морковь</i>	Число	<i>байт</i>

и проверить выполнение для этих таблиц следующих требований.

3.1. Все поля обеих таблиц должны иметь соответствующие подписи.

3.2. Все поля, несущие кодовые данные (например, поле *Личный Код* и т.п.), должны иметь тип *Текстовый* и размер 5 символов.

3.3. Для числовых таблиц должны выполняться следующие требования.

а) Поля должны иметь тип *Числовой*, размер *байт*.

б) Поля должны иметь *Значения по умолчанию 0*.

в) *Условия на значения* полей-ингредиентов должны допускать ввод значений веса в граммах, лежащих в диапазоне от **0** до **1000** и включающих значение по умолчанию **0**.

3.4. Для поля *Рожд* (таблица *Личности*) сформировать и записать в соответствующих позициях раздела *Свойства поля* Значение по умолчанию и Условие на значение, которые позволят выполнить следующие требования:

- поле *Рожд* должно допускать ввод дат рождения лиц (авторов).
- значением по умолчанию, свидетельствующим об отсутствии данных, должно быть значение «1 января 1111 года» (В последствии обратите внимание на отображение данных, относящихся к XII веку).

3.5. Все поля таблицы *Адреса* за исключением поля *Корпус* должны быть *Обязательными полями*.

4. Сохранить созданные таблицы в базе данных *База№1* и закрыть базу данных.

Задание 4. Использование масок ввода

Цель: Знакомство с использованием Масок ввода для повышения достоверности вводимых данных.

1. Открыть базу данных *База№1*.

1.1. Проверить наличие и убедиться в сохранности и работоспособности всех созданных ранее таблиц данной базы.

1.2. Продолжить модификацию таблиц базы данных, начатую в предыдущем задании, обеспечив выполнение для перечисленных ниже полей, следующих требований.

Таблица *Адреса*

1.2.1. Для поля *Личный Код* должна быть определена маска ввода, позволяющая обеспечить выполнение следующее:

- количество символов - **5**
- первый символ - **буква**
- второй символ - **«-»**
- три последних символа - **цифры**
- при вводе все символы должны преобразовываться в **прописные**
- маска должна содержать вторую секцию **«;0»** (см. справку по маскам ввода).

1.2.2. Для поля *ПочтКод* должна быть установлена маска ввода, требующая обязательного ввода шести цифр без знака, пробелы не допускаются (поле *ПочтКод* является текстовым, но вводиться в него должны только **цифровые символы**) (**>000000**).

1.2.3. Для поля *Город* должна быть задана такая маска, чтобы при вводе символы автоматически преобразовывались в **прописные** (**>????????????**).

1.2.4. Повторить п. 1.2.3 для поля *Улица*, используя не маску ввода, а формат поля (ввести в Формат поля символ «>»).

1.2.5. Создать новое поле *Телефон*, в котором будет храниться по образцу **(8-) 911-567-44-55** (" (8-) "000\ -000\ -0000 .

1.2.6. Для поля *Телефон* после задания маски ввода определить и записать в соответствующей позиции *Свойств поля* минимально необходимый размер текстового поля.

Таблица Личности

1.2.7. Создать новое поле *Прим*, предназначенное для хранения произвольного текста примечания (например, автобиографии произвольного размера), выбрав для него подходящий тип поля, и указав, что оно допускает ввод «пустых строк». Для поля *Прим*, должны быть заданы Подпись и Описание поля, а в качестве значения по умолчанию определена «пустая строка».

1.2.8. Поле *Автор* должно удовлетворять следующим требованиям:

ТипТекстовый.

Размер.....25 символов.

Маска.....автоматически вводит прописные буквы. (>???????)

1.2.9. Поле *Пол* должно удовлетворять следующим требованиям:

ТипТекстовый

Размер.....1 символ

Маска.....автоматически вводит прописные буквы, допускает только **М** и **Ж** (русские) и **M** и **F** (английские) - (>L, условие на значение: "M" Or "Ж" Or "M" Or "F").

1.2.10. Поле *Паспорт* должно удовлетворять следующим требованиям:

ТипТекстовый

Размер.....15 символов


Маска.....позволяет ввести запись, содержащую серию и № паспорта, разделенные пробелами, по образцу **40 01 123456 (00\ 00\ 000000)**.

1.2.11. Все созданные поля должны в обязательном порядке иметь заполненные Описания и Подписи.

1.3. Сохранить измененные таблицы под старыми именами в составе базы данных **База№1**.

1.4. Закрывать все таблицы.

2. Открыть окно «Схема данных» (Меню «Сервис» – «Схема данных»).

2.1. Активизировать режим *Добавление таблицы* () и добавить в схему данных вновь созданные таблицы. Схема данных должна выглядеть так, как показано на рис.4.1.

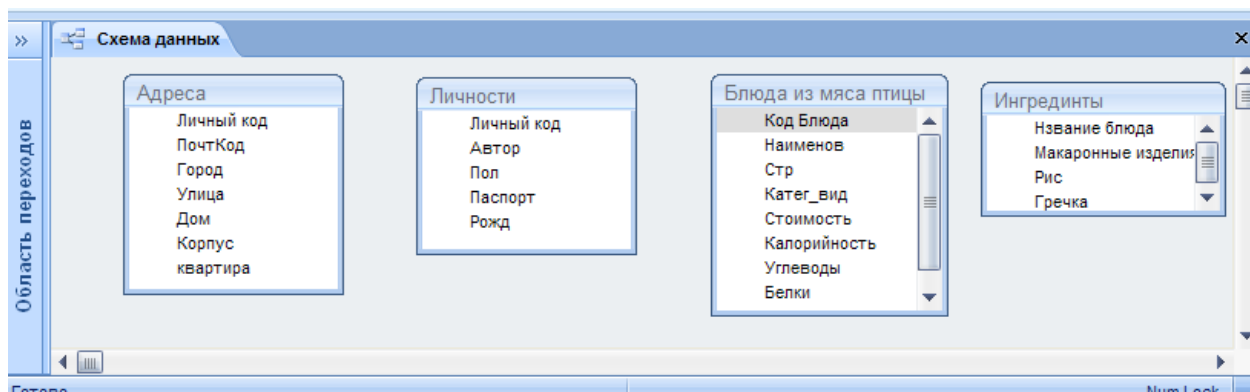


Рис.4.1

2.2. Просмотреть схему данных, состоящую из четырех несвязанных таблиц *Личности*, *Адреса*, *Блюда из мяса птицы* и *Ингредиенты*, которые входят в состав базы данных **База№1**. Установить размеры индивидуальных окон для каждой таблицы и для всей схемы данных так, чтобы были видны все поля таблиц.

2.3. Сохранить созданную структуру и закрыть окно *Схема данных*.

3. Закрывать все таблицы базы данных **База№1**.

4. Заполнить базу **База№1** данными.

4.1. Поочередно открывая все таблицы базы данных **База№1**, ввести перечисленные ниже данные.

4.1.1. В таблицу *Личности* ввести данные о 5-7 авторах блюд.

4.1.2. В таблицу *Адреса* ввести полный набор данных об адресах 5-7 авторов блюд.

4.1.3. В таблицу *Блюда из мяса птицы* ввести данные о 15-20 блюдах.

4.1.4. В таблицу *Ингредиенты* ввести данные о компонентах блюд.

5. Проверить корректность введенных данных и в случае необходимости отредактировать их.

5.1. Убедиться в том, что все используемые маски ввода используются правильно, для чего попытаться ввести в поля заведомо некорректные данные.

6. Продемонстрировать преподавателю схему данных и корректно заполненные таблицы.

7. Сохранить базу данных для дальнейшего использования.

Задание 5. Создание форм

Цель: Изучение технологий создания и изменения простых форм с использованием «Мастера форм», «Конструктора форм» и «Автоформ».

Использование Мастера форм

1. Открыть базу данных База№1.

1.1. Проверить наличие и убедиться в сохранности и работоспособности всех созданных ранее таблиц данной базы.

2. Создать форму Форма Личности.

2.1. В окне *Создание - Другие формы* указать использование режима *Мастера форм* и определить таблицу *Личности* в качестве источника.

2.2. В первом диалоговом окне *Создание формы* выбрать все поля из таблицы *Личности* для представления в создаваемой форме.

2.3. Во втором диалоговом окне выбрать внешний вид формы *В один столбец*.

2.4. В третьем диалоговом окне просмотреть предлагаемые стили оформления и выбрать *Обычная* или *Официальная*.

2.5. В четвертом диалоговом окне задать в качестве имени формы *«Форма Личности»*, а для дальнейшей работы определить режим *Изменение макета формы*.

2.6. Перейти в режим *Конструктора форм* (на вкладке с именем формы вызвать контекстное меню и выбрать режим *Конструктор*). Выполнить следующие действия:

2.6.1. Открыть области для заголовка формы и примечания формы. Разместить в них текст, как показано на рис.5.1.

2.6.2. Поместить в области примечания текущую дату и время. Для этого воспользоваться вкладкой *Конструктор – «Дата и время»* и перетащить *Дату* и *Время* в область *Примечания* формы.

2.6.3. Создать в заголовке формы вычисляемое поле, отображающее имя и фамилию.

а) Пользуясь панелью инструментов *Панель элементов*, разместить в области заголовка новое поле.

б) Выбрав команду *Свойства* в контекстном меню и записать в свойстве *Данные* (появляется окно Свойств справа) формулу **=[Автор] & " " & [Рожд]**, пользуясь построителем выражений (...).

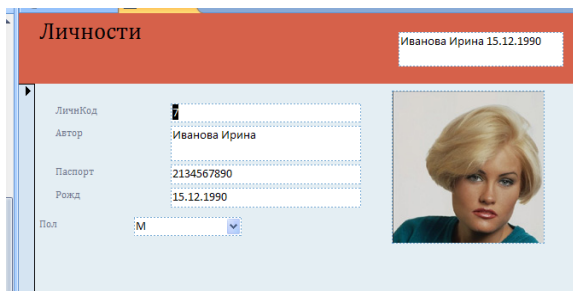


Рис.5.1

2.6.4. Создать поле со списком *Пол*.

а) Удалить из формы поле *Пол*.

б) Щелкнуть мышью на кнопках *Мастер* и *Поле со списком* панели инструментов *Панель элементов*, затем перетащить из списка полей поле *Пол* в область данных. При этом будет активизировано окно мастера *Создание полей со списком*, в котором необходимо проделать следующие действия:

- ◆ активизировать переключатель *Будет введен фиксированный набор значений*;
- ◆ заполнить Столбец1 значениями, которые может принимать поле *Пол*: М, Ж, М, F;
- ◆ сохранить значение, которое содержит поле со списком в поле *Пол*;
- ◆ задать подпись поля «Пол».

2.6.5. Поместить в форму рисунок (рис. 5.1).

а) С помощью любого доступного графического редактора (например, **MS Paint**) создать растровый рисунок, сохранив его в рабочей папке.

б) На панели инструментов *Панель элементов* выбрать элемент управления *Рисунок*.

в) В открывшемся диалоговом окне выбрать созданный ранее файл рисунка и корректно разместить его в рамке, выбрав пункт *Свойства* определить вариант размещения (например, *По размеру рамки*).

3. Сохранить форму. Перейти в *режим работы с формой*.

4. Ввести в таблицу *Личности* две строки реальных данных, пользуясь созданной формой. Проверить, как работают созданные для таблиц *Условия на значения* и *Маски ввода*.

5. Создать форму *Форма Адреса*.

5.1. В окне *Создание - Другие формы* указать использование режима *Мастера форм* и определить таблицу *Адреса* в качестве источника.

5.2. Во втором диалоговом окне *Создание формы* выбрать из доступных полей таблицы *Адреса* все поля и в следующем диалоге выбрать *Ленточный* вариант размещения полей.

5.3. Выбрать в качестве стиля оформления формы вариант *Поток* или *Стандартная*.

5.4. Задать для создаваемой формы имя *Форма Адреса* и указать последующий переход в режим ввода данных.

5.5. Изменить в созданной форме заголовки столбцов (рис.5.2).

5.6. В области примечаний формы разместить текущую дату и время в формате, представленном на рис.5.2

Личный код	Почтовый Код	Город	Улица	Дом	Корпус	квартира

Рис.5.2

5.7.

Закончить создание формы, проверить ее внешний вид, перейдя в режим представления *Формы*,

сохранить ее как один из элементов

базы данных *База№1*.

6. Воспользоваться созданной формой и ввести в таблицу *Адреса* три новые записи. При вводе проверить, как работают *Условия на значения* и *Маски ввода*.

Использование Конструктора форм

7. Создать форму *ФормаАдреса1*.

7.1. Начать создание новой формы, для чего в окне *Новая форма* указать режим использования *Конструктора форм* и выбрать *Адреса* в качестве таблицы, для которой создается форма.

7.2. Открыть *Список полей* таблицы *Адреса* с помощью меню «Добавить поля».

7.3. Пользуясь технологией «Drag and Drop» перенести в область данных формы все поля таблицы *Адреса*.

7.4. Пользуясь контекстным меню – «Заголовок/примечание формы» сделать доступными в форме области для создания заголовка и примечания. Увеличить их размер и разместить в этих областях тексты заголовка и примечания, как это сделано на рис.5.3.

7.5. Пользуясь элементами управления цветом и оформлением областей текста и полей таблицы, размещенными на панели форматирования, оформить поля таблицы и подписи, приблизительно как показано на рис.5.3:

7.6. Перемещая подписи и поля таблицы по поверхности формы, разместить их так, как показано на рис.5.3.

7.6.1. Для перемещения полей и подписей рекомендуется пользоваться выделением не отдельных полей, а групп полей или групп подписей (помещение в группу осуществляется щелчком мыши при нажатой клавише **Shift**).

7.6.2. Изменить размеры отображаемых частей полей и подписей.

7.6.3. Выровнять поля и подписи, как показано на рис.5.3, пользуясь контекстным меню и командой «Выровнять» подобрать «Вид выравнивания» для выделенных полей (или групп полей) и подписей.

7.6.4. Изменить общий размер формы и привести его к виду рис.5.3.

7.7. Завершить создание формы и дать ей имя **ФормаАдреса1**.

8. Ввести в таблицу *Адреса* две строки данных, пользуясь созданной формой. Проверить, как работают созданные для таблиц *Условия на значения* и *Маски ввода*.

Задание 6. Создание диаграмм и кнопочных форм

Цель: Изучение технологии создания диаграмм в формах и кнопочных (управляющих) форм.

Создание диаграмм

1. Открыть базу данных *База№1*.

1.1. Проверить работоспособность всех созданных ранее таблиц.

2. Создать круговую диаграмму, показывающую распределение ингредиентов по блюдам из мяса птицы (рис.6.1).

2.1. Открыть вкладку форм, выбрать тип формы - *Диаграмма* и в качестве источника указать таблицу *Ингредиенты*.

2.2. В качестве полей с данными определить *Лук*, в качестве полей категорий определить *Название блюда*.

2.3. Изменить тип диаграммы – *Круговая, объемная разрезная круговая диаграмма*.

2.5. Сохранить созданную форму под именем *График*.

2.7. Изменить данные в таблице *Ингредиенты* и пронаблюдать соответствующие изменения на диаграмме.

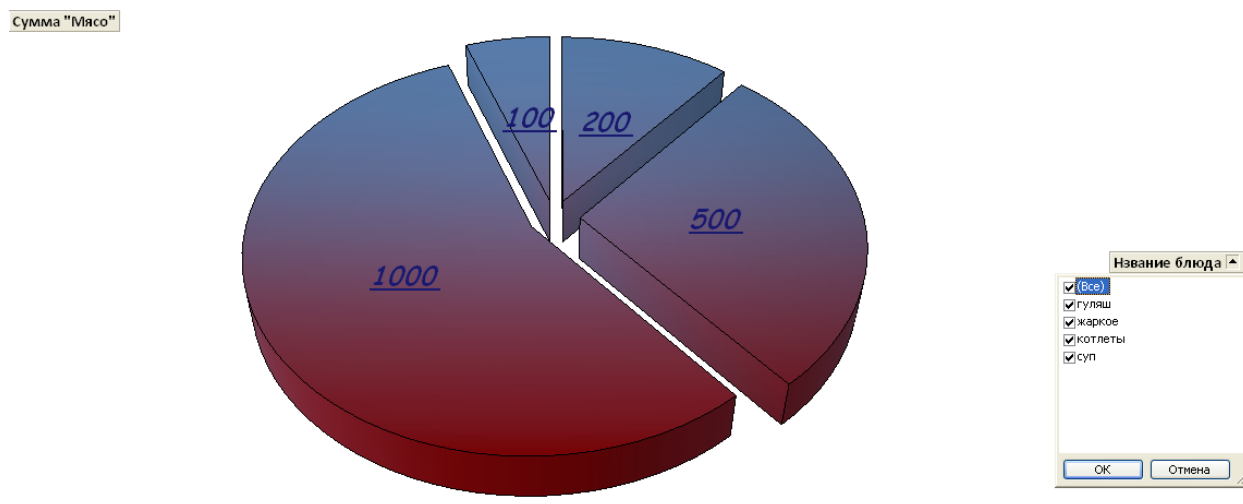


Рис.6.1

Создание кнопочных форм

3. Создать кнопочную форму по образцу, приведенному на рис.6.2.

3.1. Открыть диалоговое окно *Диспетчера кнопочных форм* (На вкладке **Работа с базами данных** в группе **Работа с базами данных** нажмите кнопку **Диспетчер кнопочных форм**).

ПРИМЕЧАНИЕ. Если выводится приглашение создать кнопочную форму, нажмите кнопку **Да**.

1. Нажмите кнопку **Создать**.

На экране появится диалоговое окно **Создание**.

2. Введите имя новой кнопочной формы **Мясные блюда** и нажмите кнопку **ОК**.

Имя новой кнопочной формы добавляется в список поля **Страницы кнопочной формы**.

3. Выберите имя новой кнопочной формы **Мясные блюда** и нажмите кнопку **Изменить**.

Будет открыто диалоговое окно **Изменение страницы кнопочной формы**.

4. Нажмите кнопку **Создать**.

Откройте диалоговое окно **Изменение элемента кнопочной формы**.

5. В поле **Текст** введите имя для первой кнопки кнопочной формы «**Личности**», а затем выберите для нее команду из списка **Команда (Открыть форму для добавления)** и в качестве источника определить форму **Форма Личности**.

ПРИМЕЧАНИЯ

↓ Чтобы открыть кнопочную форму одновременно с базой данных, выберите имя кнопочной формы в диалоговом окне **Диспетчер кнопочных форм**, а затем щелкните пункт **По умолчанию**.

↓ При создании кнопочной формы с помощью диспетчера кнопочных форм в приложении Access создается таблица «**Элементы кнопочной формы**», в которой описывается, что отображено на кнопках

формы, и какие действия они выполняют. Если изменить кнопочную форму вручную, приложение может перестать работать.

3.2. Повторить аналогичную последовательность действий для форм **Форма Адреса, Форма Адреса1 и График.**

3.3. Добавить новую кнопку *Выход* для закрытия файла базы данных.

3.4. Откорректировать созданную кнопочную форму в режиме Конструктора форм в соответствии с рис.6.2.

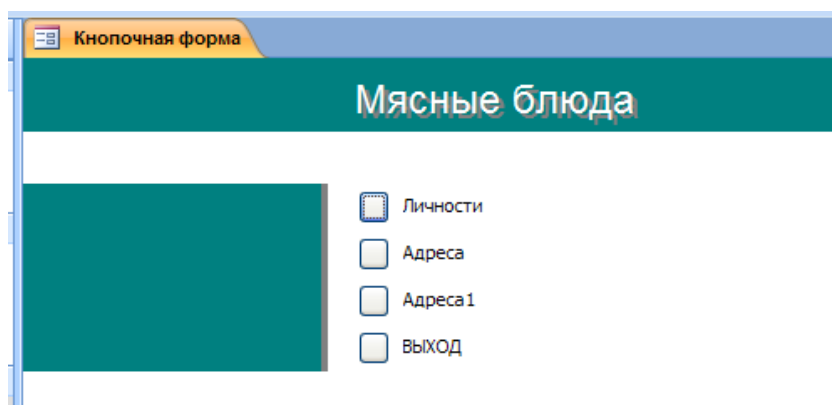


Рис.6.2

4. Сохранить базу данных *База№1* со всеми ее таблицами и формами.

Задание 7. Сортировка, поиск, фильтрация данных

Цель: Изучение способов работы с данными: сортировки, поиска и замены, фильтрации.

1. Открыть базу данных *База№1*.

1.1. Проверить наличие и убедиться в сохранности и работоспособности всех созданных ранее таблиц данной базы.

1.2. Открыть таблицу *Ингредиенты* в режиме таблицы.

Сортировка данных

2. Отсортировать данные по одному полю.

2.1. Просмотреть содержимое таблицы Личности и убедиться, что в ней есть достаточное количество разнообразных записей (10-15 записей).

2.2. Активизировать панель инструментов *Таблица в режиме таблицы*, в которой присутствуют кнопки управления режимами сортировки и фильтрации данных (рис.7.1).



Рис.7.1

Используя пункты меню, отсортировать записи таблицы *Ингредиенты* последовательно по пяти параметрам: Название блюда, Мясо, Лук, Чеснок, Морковь.

2.3. Отсортировать записи таблицы *Ингредиенты* последовательно по трем параметрам (Название блюда, Лук, Мясо) в обратном порядке, используя кнопки панели инструментов.

3. Отсортировать данные по нескольким полям

Пользуясь режимом сортировки, и выделяя совместно несколько смежных полей, выполнить сортировку данных таблицы ***Блюда из мяса птицы*** по возрастанию значений в нескольких полях одновременно, а именно:

- по Наименов и по Стр,
- по Стр и по Автор,
- по Калорийности и по Углеводам,
- по Углеводам и по Белкам,
- по Белкам и по Жирам
- по Жирам и по Ароматизаторам.

Примечание. Для правильного выполнения п.2.2. предварительно убедиться, что в таблице имеются необходимые повторяющиеся данные, например одинаковые Катег_вид.

Поиск и замена данных

4. Познакомиться с возможностями поиска данных в таблицах базы данных, пользуясь режимом поиска данных («Найти»), а также параметрами диалогового окна *Поиск и Замена*.

4.1. Осуществить поиск значений внутри поля

4.1.1. Найти все блюда с названием, например, «**суп**»

4.1.2. Найти блюда, у которых вес Мяса, например, =500.

4.1.3. Найти в таблице ***Личности*** всех авторов, у которых фамилия начинается с символов «Иван», например: Иванов, Иванова, Иванчиков, Иваненко и т.п.

4.1.4. Найти всех лиц, у которых внутри фамилии встречаются символы «ров», например Федоров, Викторова и т.п. Если в таблице нет таких фамилий, то необходимо ввести их, разместив в различных записях.

4.2. Осуществить поиск значений во всей таблице ***Личности***:

4.2.1. Указав область поиска – вся таблица, найти все текстовые поля любых записей, в которых в любом контексте встречается комбинация из двух символов «10», например, должны быть обнаружены поля, содержащие

дату рождения **10.12.92**, личный код **Л-010**, номер паспорта 20 01 **710790** и т.д.

4.3. Осуществить поиск с различными вариантами просмотра:

4.3.1. Установить курсор на запись, расположенную в середине таблицы. Используя параметры диалогового окна *Найти*, выполнить поиск данных по п.4.2.1 (содержащих символы «**10**»), соответственно в областях таблицы, расположенных от курсора до начала (первого поля первой записи) и от курсора до конца (последнего поля последней записи).

5. Познакомиться с возможностями замены данных в таблицах базы данных.

5.1. Используя режим замены («Заменить») поменять во всей таблице *Ингредиенты* в поле *Мясо* все значения веса, например, 255 на 0,255. Объясните почему получился 0, а не 0,255?

5.2. Пользуясь заменой, восстановить одно из исправленных значений в поле *Мясо*.

5.3. Во всей таблице *Личности* все инициалы заменить на соответствующие имена, например, И.И. – Иван или Е.С. - Екатерина.

Фильтрация данных

6. Познакомиться с возможностями фильтрации данных в таблицах базы данных.

6.1. Используя возможности **нормальной фильтрации** (меню «Сортировка и фильтр»–«Фильтр»–«Изменить фильтр» и «Применить фильтр»), отфильтровать (выделить) данные, удовлетворяющие следующим критериям отбора:

- лицо с конкретным личным кодом, например **Л-005** или **2**,
- лицо с конкретным именем, например **Ирина** ,
- лица с конкретной датой рождения, например 25.10.1993,
- лица с конкретным значением паспорта, например **20 08 787878**

(номер и серию паспорта вводим без пробелов - 2008787878).

6.2. Выполнить фильтрацию, задавая в качестве критериев отбора данных значения нескольких полей, а именно:

- лиц мужского пола (**М**) и определенным номером паспорта,
- лиц женского пола (**Ж**) с определенной датой рождения, например **10.12.1993**,
- лиц с конкретным именем (**Ирина**), определенным номером паспорта (**2008787878**) и определенной датой рождения (**10.12.1993**).

6.3. Используя **фильтрацию по выделенному значению** и последовательно выделяя значения необходимых полей, отфильтровать следующие данные:

- лиц женского пола (**Ж**),
- лиц мужского пола (**М**) с определенной датой рождения,
- лиц с датой рождения, не равной определенной дате.

6.4. Используя **расширенную фильтрацию** («Сортировка и фильтр» - «Дополнительно» – «Расширенный фильтр»), сформулировать и записать в образце запроса критерии для выполнения следующих вариантов фильтрации данных по таблице **Блюда из мяса птицы**:

- блюда какой страны содержат **более 1000 ккал и более 100 гр** жиров,

- блюда с определенным наименованием,

а по таблице **Личности**:

- лиц, родившихся после определенной даты (**12.10.93**),
- лиц, по гороскопу относящихся к «Стрельцам» и родившихся в **1993 г.**, (**23.11.93 до 22.12.93**),

- лиц с определенной фамилией (**Иванова**), которым сегодня от **18 до 22** лет.

6.4.1. Отфильтровать данные, для которых необходимо сформулировать критерий фильтрации, содержащий логические функции (И, ИЛИ) над значениями полей:

Задание 8. Создание запросов на выборку

Цель: Изучение техники построения запросов на выборку к одиночной таблице и запросов на создание таблицы.

1. Открыть базу данных *База№1*.

1.1. Проверить наличие и убедиться в сохранности и работоспособности всех созданных ранее таблиц данной базы.

2. Задать критерии выборки в запросе.

2.1. Открыть окно *Создание – Конструктор запросов* и добавить в него таблицу *Личности*.

2.2. Определить в качестве типа запроса - *Выборка*.

2.3. Определить столбцы и заполнить строки запросной формы так, чтобы в результате запроса увидеть четыре поля (*Автор, Пол, Дата рождения, Паспорт*) всех записей, входящих в таблицу *Личности*. Пример подготовленного запроса представлен на рис.8.1.

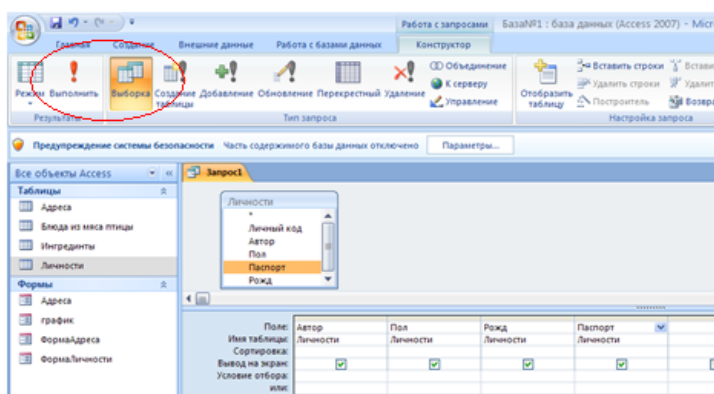


Рис.8.1

2.3.1. Выполнить запрос и просмотреть его результат.

2.3.2. Пользуясь контекстным меню или кнопкой меню «Режим», расположенной на панели инструментов вкладка Главная просмотреть созданный запрос в трех возможных формах его представления:

- в окне Конструктора запросов (*QBE-Query By Example*),
- в режиме таблицы,
- в режиме SQL (*Structured Query Language*).

2.4. Сформулировать критерии выборки, создать и выполнить следующие запросы:

- лиц с личным кодом, например, **Л-003, Л-005, Л-007,**
- лиц с личным кодом, «большим» чем, например, **Л-007,**
- лиц с личным кодом в интервале, например, от **Л-003** до **Л-007.**

2.5. Сформулировать критерии выборки, создать и выполнить следующие запросы:

- лиц личным кодом, например, **Л-003, Л-005, Л-007** и датой рождения <10.10.1993,
- лиц с датой рождения <10.10.1993 и личным кодом, «большим» чем, например, **Л-007,**
- лиц женского пола (**Ж** и **Ф**) с личным кодом в интервале, например, от **Л-003** до **Л-007.**

3. Осуществить выборку с последующей обработкой результатов:

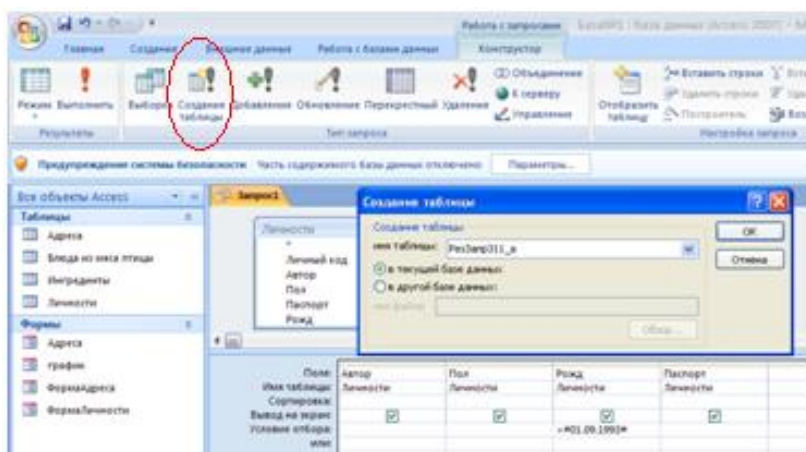
3.1. Выполнить выборку с последующей сортировкой результатов:

3.1.1. Сформулировать критерии выборки, определить варианты сортировки результатов выборки, создать и выполнить следующие запросы:

- лиц с личным кодом в интервале, например, от **Л-003** до **Л-007,** отсортированными по убыванию,
- лиц мужского пола (**М-русск.** и **М-латин.**) с личным кодом в интервале, например, от **Л-003** до **Л-013,** датой рождения >01.09.1993. Результаты выборки должны быть отсортированы сначала в порядке возрастания кода, затем в порядке убывания даты рождения.

3.2. Выполнить выборку с сохранением результатов в новой таблице:

3.2.1. Выбрать режим *Создание таблицы* для размещения результатов запроса в новой таблице. Выполнить последовательно запросы п.3.1.1 и сохранить их результат в новой таблице **РезЗан311_а** и **РезЗан311_б,** которые будут размещены на вкладке *Таблицы* базы данных **База№1.**



3.2.2. Сохранить сформулированные в п.3.1.1. запросы под именами *Запрос_31_а* и *Запрос_31_б* на вкладке *Запросы* базы данных *База№1*.

3.3. Выполнить выборку с последующей группировкой результатов:

3.3.1. Осуществить выборку данных о лицах женского (**Ж**) и мужского (**М**) пола, включив в результат следующие поля *Пол*, *Автор*, *код*, *дата рождения*. Выполнить выборку, просмотреть и проверить результат.

3.3.2. Выполнить выборку по критериям п.3.3.1 с группировкой результатов по полу. Над данными других полей необходимо выполнить следующие виды обработки:

- для поля *Автор* подсчитать **количество** лиц мужского и женского пола,
- для поля *Рожд* определить **среднюю дату рождения** мужчин и женщин,
- для поля *Личный Код* определить **наибольший и наименьший** личный код для мужчин и женщин соответственно.

Примечание. Для выполнения запросов п.3.3.2 необходимо указать на необходимость выполнения операций группировки результатов (Σ - итоги) и определить групповые операции (определение количества **Count**, вычисление среднего **Avg**, нахождения наибольшего **Max** или наименьшего **Min** значений).

3.3.3. Сохранить запрос под именем *Запрос_333*. Сохранить результат запроса в таблице под именем *РезЗан333*. Результат выполнения запроса представлен на рис.8.2.

Count-Автор	Avg-Рожд	Пол	Min-Личный код	Max-Личный код
1	33222	М	7	7
2	33550	Ж	03	9
*				

Рис.8.2

3.3.4. Выполнить п.3.3.2 с группировкой по полу лиц, чей код превышает значение 6. Сравнить результаты, полученные в п.3.3.4 и 3.3.2.

4. Создать запрос на выборку лиц, имеющих одинаковые имена.

4.1. В качестве источника указать таблицу *Личности*.

4.3. Поиск повторяющихся данных осуществлять по полю *Автор*, например, Иванова* (* - любое количество любых символов, т.к. после фамилии сразу вписано имя автора).

4.4. Дополнительно для отображения в результате запроса указать поле *Личный код*.

4.5. Сохранить запрос под именем *Запр_Однофамильцы*.

4.6. Выполнить запрос и убедиться в правильности его работы. В случае отсутствия повторяющихся данных в таблице *Личности* необходимо изменить данные в поле *Автор* и повторить запрос еще раз.

4.7. Открыть созданный запрос в режиме *Конструктора* и проанализировать его содержание.

5. Сохранить базу данных *База№1* со всеми ее таблицами и формами.

Задание 9. Вычисления в запросах

Цель: Знакомство с возможностями создания и использования вычисляемых полей в запросах.

1. Открыть базу данных *База№1*.

1.1. Проверить наличие и убедиться в сохранности и работоспособности всех созданных ранее таблиц данной базы.

2. Создать запрос, позволяющий вывести фамилии людей, сгруппированных по признаку пола, для которых в результирующей таблице будут представлены два значения, относящиеся к дате рождения - в кратком формате даты и возраст.

2.1. Создать в запросной *QBE* форме новое поле - для возраста. В этом поле следует записать выражение, связывающее дату рождения и возраст ($\text{Round}((\text{Date}()-[\text{Личности}]![\text{Рожд}])/365;0)$).

2.2. Повторить выполнение запроса из п.2.1, предварительно обеспечив выполнение следующих условий:

- вычисляемое поле с возрастом должно располагаться рядом (справа) со своим аналогом;
- поля, представленные в результате выборки, должны иметь соответствующие подписи;
- количество десятичных знаков после запятой в числовых полях должно быть равно 2.

Для выполнения этих условий необходимо, находясь в запросной *QBE* форме, активизировать диалоговое окно *Страница свойств* для конкретного поля и на вкладке *Общие* этого окна задать необходимые значения параметров поля (в том числе и подписи). Образец результата запроса представлен на рис.9.1.

Автор	Рожд	ВОЗРАСТ	Пол	Паспорт
Ванечкина Ирина	21.12.1989	21,16	Ж	
Федорова Фаина	25.09.1993	17,39	Ж	55 25 358915
Иванова Полина	25.06.1992	18,65	Ж	31 31 351264
Петров Павел	01.06.1985	25,72	М	21 34 567891
Иванова Ирина	15.12.1990	20,18	М	21 34 567890
*	11.11.1111			55

Рис.9.1

3. Создать запрос, в котором будут представлены сведения о дате рождения и возрасте студента, как показано на рис.9.2.

3.1. Для выполнения задания необходимо записать выражение, вычитающее год даты рождения из года текущей даты. Для этого, в свою очередь, следует воспользоваться стандартными функциями **Date()** и **Year()**. Созданное выражение должно быть записано в соответствующем поле запросной *QBE* формы (Возраст: CDate([«date»])-[Рожд]).

Автор	Рожд	Возраст
Ванечкина Ирина	21.12.1989	7723
Федорова Фаина	25.09.1993	6349
Иванова Полина	25.06.1992	6806
Петров Павел	01.06.1985	9387
Иванова Ирина	15.12.1990	7364
*	11.11.1111	

Рис.9.2

3.2. Выполнить запрос, удостовериться в правильности его работы и сохранить запрос под именем **Возраст ручной**.

4. Создать запрос, в котором непосредственно перед его выполнением будут запрашиваться определенные в *QBE* форме параметры.

4.1. На основе сохраненного запроса **Возраст ручной**, создать запрос, позволяющий получить все сведения для конкретного лица, фамилия которого будет запрошена как значение параметра непосредственно перед выполнением запроса. В качестве комментария в диалоге «Введите значение параметра» должен быть текст «**Фамилия?**».

4.2. На основе предыдущего запроса, создать новый запрос, который позволит выполнять действия только с конкретными фамилиями. Используйте оператор **Like** для ввода в параметре вместо букв имени символов шаблона * и ? (**Like [Фамилия?] & "*"").**

5. Создать запрос на обновление таблицы.

5.1. Открыть новую запросную форму и определить в качестве источника таблицу *Ингредиенты*.

5.2. Определить тип запроса – *Запрос на обновление*.

5.3. В поле *Обновление* сформулировать выражение, удваивающее количество вес лука для всех блюд ($[\text{Ингредиенты}]![\text{лук}]*2$).

5.4. Выполнить запрос и проверить правильность его работы.

6. Создать перекрестный запрос на основе таблицы *Личности*, иллюстрирующий распределение мужчин и женщин по годам рождения.

6.1. Указать в окне *Новый запрос* вариант *Перекрестный запрос*.

6.2. Выбрать в качестве заголовка строк поле *Пол*, в качестве заголовка столбцов поле *Рожд*, а в качестве значения - поле *Фамилия* с функцией **Count** (подсчет значений).

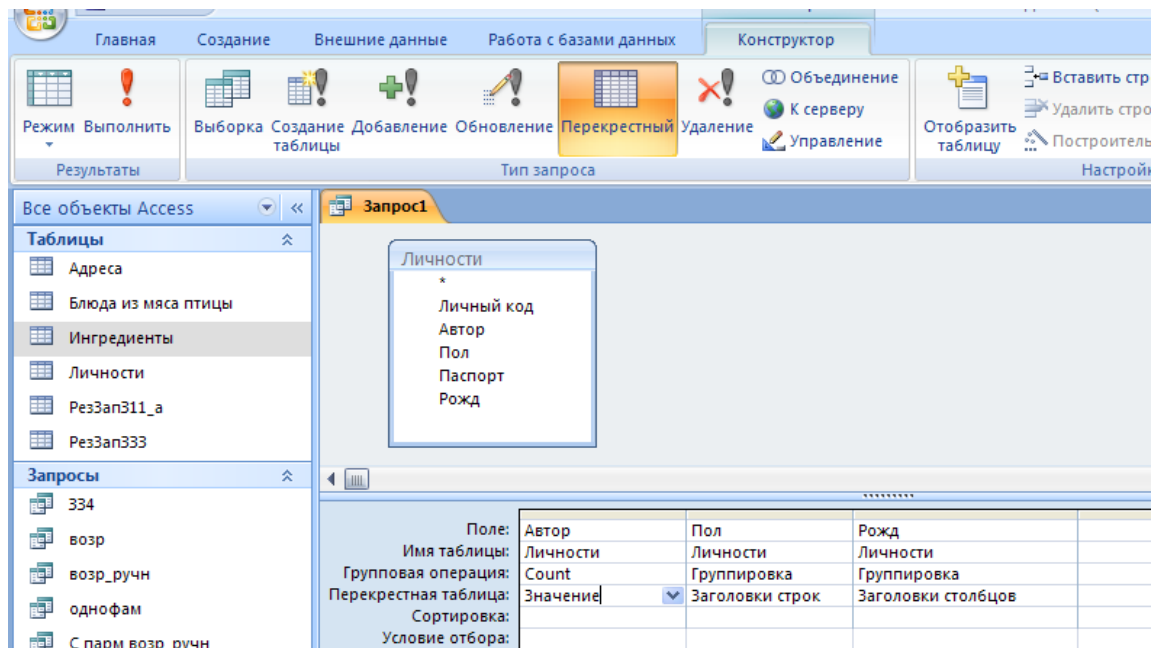


Рис.9.3

6.3. Сохранить запрос под именем *Перекрестный запрос*.

6.4. Выполнить запрос и проверить правильность его работы.

6.5. В режиме *Конструктора* изменить этот запрос так, чтобы в заголовках столбцов выводились не даты, а года рождения (Выражение1: `Format([Рожд];"уууу")`).

7. Сохранить базу данных *База№1* со всеми ее таблицами, формами и запросами.

Задание 10. Создание отчетов

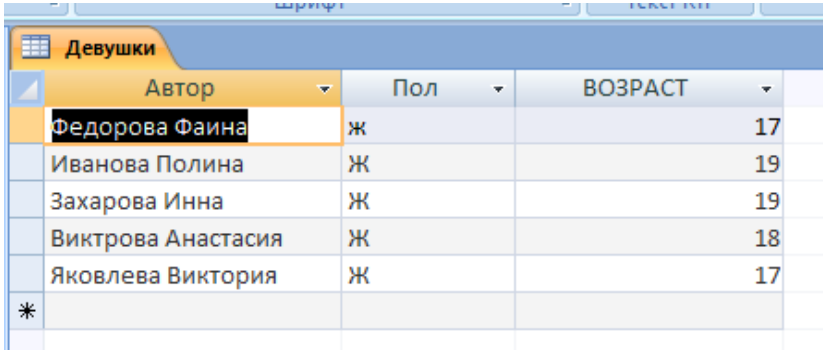
Цель: Знакомство с созданием простых отчетов к одиночным таблицам.

1. Открыть базу данных *База№1*.

1.1. Проверить наличие и убедиться в сохранности и работоспособности всех созданных ранее таблиц данной базы.

1.2. Создать и выполнить запрос, позволяющий извлечь в отдельную таблицу *Девушки* (рис.10.1) лиц женского пола в возрасте от 16 до 20 лет.

1.3. Создать и выполнить запрос, позволяющий извлечь в отдельную таблицу *Дедушки* лиц мужского пола в возрасте от 50 до 80 лет. Таблица должна содержать поля *Автор*, *Пол*, *Рожд*. При отсутствии в таблице *Личности* подходящих данных, предварительно ввести 5 записей, удовлетворяющих критерию выборки.



Автор	Пол	ВОЗРАСТ
Федорова Фаина	ж	17
Иванова Полина	Ж	19
Захарова Инна	Ж	19
Виктрова Анастасия	Ж	18
Яковлева Виктория	Ж	17
*		

Рис.10.1

2. Создать отчет *Отчет_Девушки*.

2.1. Перейдя в режим создания нового отчета, активизировать *Конструктор отчетов* для создания отчета на основе таблицы *Девушки*.

2.2. Пользуясь *Списком полей*, выбрать из таблицы поля, которые необходимо включить в отчет, а именно *Автор*, *Пол*, и вычисляемое поле *Возраст*, созданное на основе поля *Рожд*.

2.3. Разместить выбранные поля и подписи к ним, а также заголовок и колонтитул отчета на пространстве листа так, как показано на рис.10.2

2.4. Перейдя в режим просмотра отчета, убедиться в его сходстве с образцом.

Қақ много девушек хороших
(отчет, созданный по таблице Девушки)

Автор_Девушки:	Пол_Девушки:	ВОЗРАСТ:
Федорова Фаина	Ж	17
Иванова Полина	Ж	19
Захарова Инна	Ж	19
Виқтрова Анастасия	Ж	18
Яқовлева Виктория	Ж	17

12 февраля 2011 г.

Рис.10.2

2.5. Поместить в примечании отчета итоговое поле, в котором рассчитывается средний возраст лиц, данные о которых приведены в отчете.

Девушки2

Заголовок отчета

Қақ много девушек хороших

Верхний колонтитул

(отчет, созданный по таблице Девушки)

Область данных

Автор_Девушки:	Пол_Девушки:	ВОЗРАСТ:
Автор_Девушки	Пол_Девушки	ВОЗРАСТ

Нижний колонтитул

=Date()

Примечание отчета

Средний возраст	=Avg([ВОЗРАСТ])
-----------------	-----------------

2.6. Сохранить отчет под именем *Отчет_Девушки*.

3. Создать отчет *Отчет_Дедушки*.

Примечание. В данном пункте описана вся последовательность действий по:

- созданию структуры отчета (поля, попадающие в отчет);

- **обработке данных**, выбираемых в отчет (группировка и сортировка);
- **предварительному оформлению** отчета.

Эти разнородные действия выполняются последовательно, и на каждом этапе демонстрируется результат.

Рекомендуется попутно просмотреть и другие возможные варианты, чтобы сравнить их. Если Вы выбрали неверный вариант, всегда можно вернуться на один или несколько шагов назад и повторить действия. Можно также прервать работу *Мастера* и повторить действия с самого начала.

3.1. Создать отчет, представленный на рис.10.3. Для этого на вкладке *Отчеты* указать режим использования *Мастера отчетов* и выполнить следующее:

- определить имя таблицы-источника *Дедушки*;
- выбрать все поля таблицы (*Автор, Пол, Рожд, Возраст*);
- выбрать *режим группировки* на основе данных поля *Рожд*;
- определить интервал группировки по *годам рождения*;
- задать *сортировку* записей по алфавиту фамилий в пределах группы;
- выбрать в качестве варианта вида макета *Блок*;
- выбрать стиль отчета *Строгий*;
- задать имя отчета *Отчет_Дедушки*;
- определить для дальнейшей работы с отчетом режим *Изменения макета отчета*.

3.2. Представленные на рис.10.3 заголовки и подзаголовки групп не могут быть созданы *Мастером* и создаются с помощью *Конструктора отчетов* после завершения работы *Мастера*.

Для окончательного оформления отчета необходимо вернуться к *Конструктору отчетов* и выполнить следующие действия:

- создать заголовки отчета,
- создать подзаголовки групп по году рождения,
- изменить названия столбцов отчета,
- изменить размеры полей,

- создать итоговое поле в примечании группы, подсчитывающее количество лиц в группе.

Дедушки				
Отчет "Дедушки"				
<i>Рожд по годам</i>	<i>Автор</i>	<i>Пол</i>	<i>Рожд</i>	<i>ВОЗРАСТ</i>
<u>Заголовок группы, организованный по 1950</u>			<u>году рождения</u>	
1950	Петров Василий	М	05.05.1950	61
<u>Заголовок группы, организованный по 1955</u>			<u>году рождения</u>	
1955	Евдокимов Егор	М	08.03.1955	56
1955	Семенов Федор	М	22.11.1955	55

Рис.10.3

Примечание. Для того чтобы включить в подзаголовок цифру года рождения, необходимо воспользоваться *Построителем выражений*. На основе значения поля *Рожд* и функций преобразования дат создать выражение, выделяющее год из поля даты =Format\$([Рожд];"yyуу";0;0), а затем разместить его в вычисляемом поле, соединяющем две текстовые строки в подзаголовке группы.

4. Создать отчет *Отчет_Девушки2* на основе таблицы *Девушки*, который позволит получить:

- сведения о девушках в возрасте от 16 до 20 лет с указанием их фамилий и имен, возраста (не даты рождения), пол.
- итоговые значения о количестве лиц в каждой возрастной группе.
- итоговое значение о среднем возрасте всех лиц, представленных в отчете.

Данные в отчете должны быть сгруппированы по возрастным группам, в каждой из которых должны присутствовать только девушки одного года рождения.

5. Сохранить базу данных на диске для дальнейшего использования.

Задание 11. Создание индексов и ключей

Цель: Знакомство с технологией создания и использования простых и составных индексов и первичных ключей.

1. Открыть базу данных *База№1*.

1.1. Проверить наличие и убедиться в сохранности и работоспособности всех созданных ранее таблиц данной базы.

2. Создать первичные ключи в ранее сконструированных таблицах *Личности* и *Адреса*.

2.1. Открыть таблицу *Личности* в режиме *Конструктора таблиц*. Пользуясь контекстным меню поля *ЛичнКод* отметить его как ключевое поле.

2.1.1. Находясь в окне *Личности: таблица*, вызвать меню «Индексы» и просмотреть содержимое открывшегося окна.

2.2. Открыть таблицу *Адреса* в режиме *Конструктора таблиц*. Пользуясь окном *Индексы*, создать первичный ключ для данной таблицы на основе поля *ЛичнКод*.

Примечание. Если в процессе определения ключевых полей появляются ошибки, свидетельствующие о невозможности создания ключа (повторяющиеся или отсутствующие значения ключевого поля), то следует внимательно просмотреть значения будущих ключевых полей в *режиме таблицы*, отсортированных по этому полю и внести в таблицу необходимые исправления.

3. Создать составные индексы.

3.1. Открыть для таблицы *Адреса*, находящейся в режиме *Конструктора таблиц*, окно *Индексы* и просмотреть его содержимое.

3.2. Создать в таблице *Личности* новый индекс *Пол*, предназначенный для ускорения процесса поиска и выполнения запросов по данному полю.

3.3. Создать в таблице *Адреса* новый составной индекс *ГорУлиДом*, включив в его состав поля *Город*, *Улица*, *Дом*.

3.4. Находясь в окне *Индексы* и переходя от строки *PrimaryKey* к *ГорУлиДом*, просмотреть расположенные ниже *Свойства поля* и дать объяснения увиденным свойствам индексов. Окно *Индексы* для таблицы *Адреса* будет выглядеть примерно так, как показано на рис.11.1.

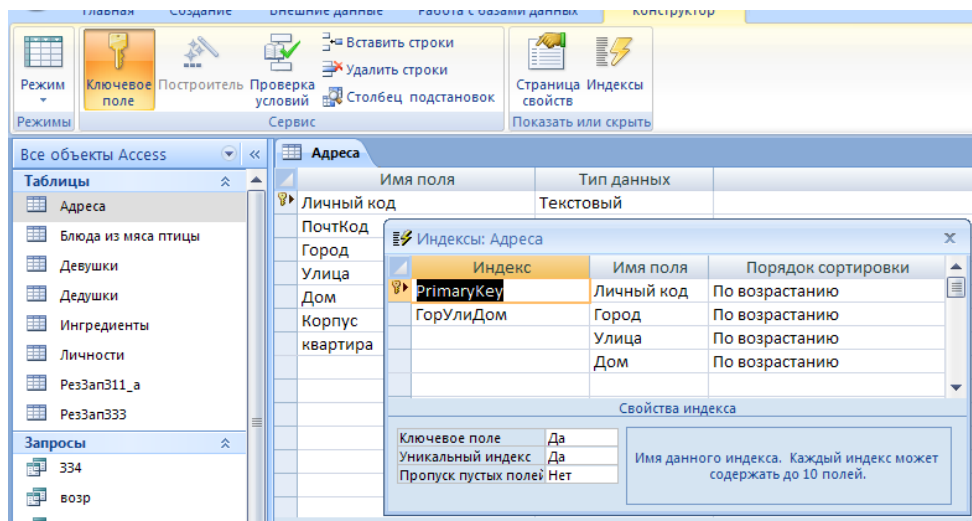


Рис.11.1

3.4.1. Для получения сведений о свойствах индексов прочитайте пояснения, размещенные в области *Свойства индекса* окна *Индексы*, а также раздел подсказки (**F1**), посвященный этому вопросу.

3.5. Создать составной индекс *ФамРожд*, включающий поля *Автор*, *Рожд*.

3.6. Открыть для таблицы *Адреса*, находящейся в режиме *Конструктора таблиц*, окно *Индексы* и просмотреть его содержимое.

3.6.1. Добавить в составной индекс *ГорУлиДом* поле *Квартира*.

3.6.2. Сделать составной индекс *ГорУлиДом* уникальным. Данный индекс необходим для обеспечения логической достоверности данных (предполагается, что не может быть двух студентов, проживающих в одной квартире).

3.6.3. Проверить работоспособность созданного индекса, введя два одинаковых адреса.

3.7. В таблице *Блюда из мяса птицы* для последующего создания индексов сделать следующие коррективы:

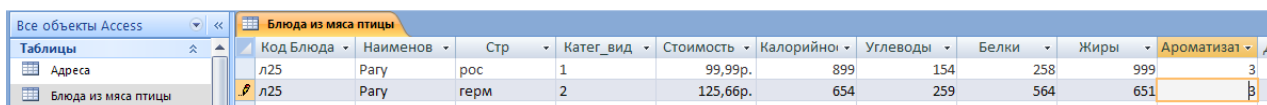
3.7.1. Ввести новое числовое поле *Ароматизаторы*, имеющее минимальный размер, допускающее ввод только следующих значений [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] и заполнить его данными.

3.7.2. Создать составной индекс *КодНаиАром*, включающий поля *Код Блюда*, *Наименов*, *Ароматизаторы*.

Заметьте, что *Код Блюда* не будет первичным ключом таблицы, каким, например, в таблицах *Личности* и *Адреса* является поле *ЛичныйКод*.

3.7.3. Установить такие свойства индекса *КодНаиАром*, чтобы этот составной индекс стал ключом таблицы и не допускал повторяющихся значений ключа (комбинаций значений полей, составляющих этот ключ).

3.8. Перейти в *режим таблицы* и ввести в таблицу *Блюда из мяса птицы* несколько повторяющихся значений, например:



Код Блюда	Наименов	Стр	Катег_вид	Стоимость	Калорийно	Углеводы	Белки	Жиры	Ароматизат
л25	Рагу	рос	1	99,99р.	899	154	258	999	3
л25	Рагу	герм	2	125,66р.	654	259	564	651	3

Рис.11.2

Проверить как СУБД MS Access реагирует на повторяющиеся значения ключей и попытки оставить пустым любое из полей, составляющих ключ.

3.9. Для таблицы *Ингредиенты* создать составной индекс *НаимМясо* и придать ему свойства ключа таблицы. В его состав должны войти поля *Наименование блюда* и *Мясо*.

3.9.1. Для проверки правильности созданного ключа ввести в таблицу *Ингредиенты* данные о блюде, содержащим разное количество мяса. Убедитесь в невозможности ввести две различные по всему множеству полей записи с одинаковыми значениями составных ключей.

3.9.2. Ввести в таблицу *Ингредиенты* 5 корректных записей о блюдах, упомянутых в таблице *Блюда из мяса птицы*.

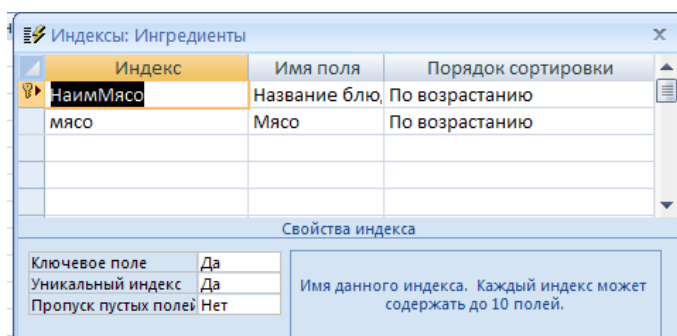


Рис.11.3

4. Ввести в каждую из имеющихся таблиц по 5 новых записей с различными (допустимыми - 3 записи и недопустимыми - 2 записи) значениями уникальных составных ключей и убедиться в том, что правильно определенные ключи:

- не позволяют вводить логически неверные данные (например, о блюде, содержащем разное количество мяса, что не допускается технологией приготовления блюда);

- не позволяет присвоить двум людям (хоть и имеющим одинаковые данные) один личный учетный код.

4.1. Проверить правильность работы всех таблиц базы данных, просмотреть их содержимое и сохранить все таблицы в базе данных **База№1**.

Примечание. Созданные в данной работе индексы, особенно составные, создавались с целью демонстрации возможности использования механизма индексации для обеспечения логической достоверности данных.

Основная цель использования индексов состоит в существенном ускорении поиска и выборки данных при использовании индексированных полей в запросах. В данном задании выборка не выполняется, поскольку на базе данных **База№1** в силу ее чрезвычайно малого объема невозможно заметить разницу в скорости выполнения выборки с использованием индексов и без их использования.

Приведенные варианты индексов служат демонстрационным целям, и в следующих заданиях в структуру ключевых полей и индексов будут внесены существенные изменения, создающие новую систему ключевых полей, предназначенную в первую очередь для организации связей между таблицами с целью их совместного использования.

Задание 12. Связывание таблиц

Цель: Знакомство с использованием ключевых полей для связывания таблиц. Изучение типов связей.

1. Открыть базу данных *База№1*.

1.1. Проверить наличие и убедиться в сохранности и работоспособности всех созданных ранее таблиц данной базы.

1.2. Удалить все индексы, созданные ранее для всех таблиц базы данных *База№1*. Отменить все признаки *первичных ключей* для всех таблиц базы данных.

2. Создать новую систему *первичных ключей* для таблиц по следующему образцу (табл. 12.1).

Таблица 12.1

Таблица	Первичный ключ
<i>Личности</i>	<i>ЛичнКод</i>
<i>Адреса</i>	<i>ЛичнКод</i>
<i>Блюда из мяса птицы</i>	<i>КодБлю</i>
<i>Ингредиенты</i>	<i>КодБлюда</i>

2.1. В тех полях таблиц, которые становятся ключевыми, удалить повторяющиеся записи.

2.2. В ранее созданных полях *Личный Код* и *КодБлюда* для всех таблиц базы данных внимательно проверить совпадение масок, использовавшихся при вводе данных. Так, например, маска вида «>L\ -000;0» и маска «>L\ -000» сохраняют данные в разных видах - с сохранением в таблице символа «-» и без него, соответственно. Очевидно, что это приведет к несовпадению данных, хранящихся в таблицах, использующих разные маски, и невозможности установить связи между этими таблицами.

2.3. В таблице *Ингредиенты* для поля *КодБлюда*, которое будет уникально определять каждую запись таблицы и использоваться как *первичный ключ*, установить тип поля - *Счетчик*.

2.4. Открыть окно *Схема данных* и разместить таблицы так, как показано на рис.12.1 (сначала без связей).

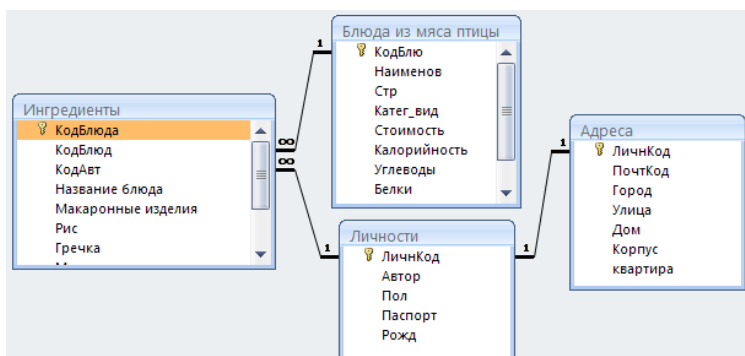


Рис.12.1

3. Установить связи между таблицами *Личности*, *Адреса*, *Блюда из мяса птицы*, *Ингредиенты*.

3.1. Внимательно прочитать все подсказки, комментарии и пояснения, касающиеся связывания таблиц и параметров объединения.

3.2. Выбрав таблицу *Ингредиенты*, с помощью мыши «перетащить» поле *КодБлюд* на поле *КодБлю* таблицы *Блюда из мяса птицы*, указав тем самым необходимость установления связи между таблицами по этим полям.

3.3. В открывшемся окне *Связи* определить тип и параметры связи между таблицами *Ингредиенты* и *Блюда из мяса птицы*.

3.3.1. Между полями *Блюда из мяса птицы.КодБлю* и *Ингредиенты.КодБлюд* должна быть установлена связь типа *Один-ко-Многим* (рис.12.2).

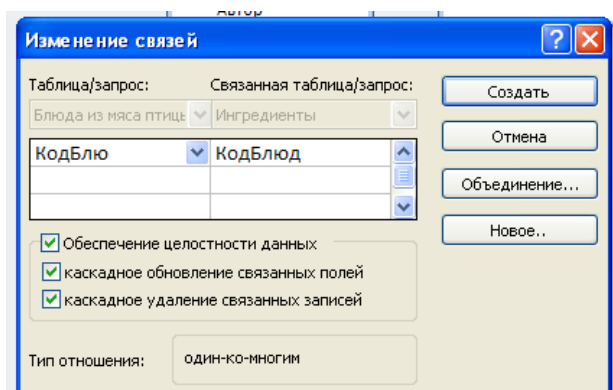


Рис.12.2

3.3.2. Установить режим *Обеспечения целостности данных: Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*. Для установки этих режимов необходимо активизировать соответствующие переключатели в группе *Обеспечение целостности* окна *Связи*.

3.3.3. Прочитать и законспектировать краткую и полную справку MS Access по режиму *Обеспечения целостности данных*, а также режимам *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей*.

3.3.4. Нажав кнопку *Объединение*, открыть окно *Параметры объединения* и выбрать такой режим объединения данных различных таблиц, при котором из связанных таблиц будут выбираться только те записи, для которых совпадают значения полей связи.

3.3.5. Прочитать и законспектировать краткую и полную справку MS Access по каждому из трех возможных значений параметра объединения.

3.4. Установить связь между таблицами *Личности* и *Ингредиенты* через поля *КодАвт* и *ЛичнКод* соответственно. Тип отношения между этими таблицами - *Один-ко-Многим*, где стороной *Много* выступает таблица *Ингредиенты*. Обеспечить целостность данных при обновлении и удалении записей. Установить объединение таблиц - по первому типу (когда из связанных таблиц выбираются только записи, для которых совпадают значения полей связи).

3.5. Установить связь типа *Один-к-Одному* между соответствующими полями таблиц *Личности* и *Адреса* с обеспечением целостности данных при обновлении и удалении записей.

3.6. Активизировать и просмотреть окно *Схема данных*. Его внешний вид должен соответствовать образцу, приведенному на рис.12.1.

4. **Используя подтаблицы** просмотреть связанные данные для таблицы *Личности*.

5. **Сохранить все таблицы** и базу данных *База№1*.

Тест

Вариант 1

Задание #1 База данных представляет собой:

Выберите один из 4 вариантов ответа:

- 1) текстовый документ большого объема
- 2) электронная таблица
- 3) совокупность специальным образом организованных взаимосвязанных данных
- 4) совокупность программ и языковых средств для создания, ведения и использования данных

Задание #2 Основными объектами базы данных являются:

Выберите несколько из 6 вариантов ответа:

- 1) таблицы
- 2) журналы
- 3) формы
- 4) запросы
- 5) сводки
- 6) отчеты

Задание #3 Поле *ДатаОтгрузки*, по вашему мнению, должно иметь следующий тип:

Выберите один из 4 вариантов ответа:

- 1) числовой
- 2) текстовый
- 3) счетчик
- 4) дата и время

Задание #4 Основное назначение *запроса* заключается в:

Выберите один из 4 вариантов ответа:

- 1) выборке данных
- 2) сортировке данных
- 3) хранении данных
- 4) фильтрации данных

Задание #5 В терминологии баз данных столбец таблицы представляет собой:

Выберите один из 4 вариантов ответа:

- 1) объект
- 2) поле
- 3) запись
- 4) ключ

Задание #6 Основное отличие полей *вторичного ключа* от обычных полей таблицы:

Выберите один из 4 вариантов ответа:

- 1) типом данных
- 2) способом отображения данных
- 3) способностью однозначно идентифицировать запись
- 4) помогают при поиске и сортировке данных

Задание #7 Базы данных могут быть следующих типов:

Выберите несколько из 6 вариантов ответа:

- 1) кадровые
- 2) иерархические
- 3) кольцевые
- 4) сетевые
- 5) библиотечные
- 6) реляционные

Задание #8 Подтвердите или опровергните высказывание - *Одно поле должно содержать данные разных типов.*

Выберите один из 4 вариантов ответа:

- 1) данные в одном поле могут быть разных типов
- 2) данные в одном поле должны быть однотипными
- 3) данные в одном поле могут быть разных типов при определенных условиях
- 4) типы полям не задаются

Вариант 2

Задание #1 Основными типами данных СУБД MS Access являются:

Выберите несколько из 6 вариантов ответа:

- 1) текстовый
- 2) дата и время
- 3) числовой
- 4) денежный
- 5) дробный
- 6) счетчик

Задание #2 СУБД представляет собой:

Выберите один из 4 вариантов ответа:

- 1) файл документа
- 2) файл электронной таблицы
- 3) совокупность специальным образом организованных взаимосвязанных данных
- 4) совокупность программ и языковых средств для создания, ведения и использования баз данных

Задание #3 Поле *Адрес*, по вашему мнению, должно иметь следующий тип:

Выберите один из 5 вариантов ответа:

- 1) числовой
- 2) дата и время
- 3) счетчик
- 4) текстовый
- 5) логический

Задание #4 Основное назначение *таблицы в БД* заключается в:

Выберите один из 4 вариантов ответа:

- 1) отбор данных
- 2) сортировка данных
- 3) хранение данных
- 4) фильтрация данных

Задание #5 В терминологии баз данных строка таблицы представляет собой:

Выберите один из 4 вариантов ответа:

- 1) объект
- 2) поле
- 3) запись
- 4) ключ

Задание #6 Основное отличие *ключевого поля (первичный ключ)* от обычных полей таблицы:

Выберите один из 4 вариантов ответа:

- 1) типом данных
- 2) способом отображения данных
- 3) способностью однозначно идентифицировать запись
- 4) возможность предотвращать удаление записей

Задание #7 Между таблицами в базе данных могут существовать следующие типы отношений (связей):

Выберите несколько из 3 вариантов ответа:

- 1) один-к-одному
- 2) многие-ко-многим
- 3) один-ко-многим

Задание #8 Подтвердите или опровергните высказывание - *Одна запись может содержать данные разных типов.*

Выберите один из 4 вариантов ответа:

- 1) да
- 2) нет
- 3) может, при определенных условиях
- 4) типы данным не задаются

Ключи к тесту:

Вариант 1	Баллы	Вариант 2	Баллы
1) (1 б.) Верные ответы: 3;	1	1) (1 б.) Верные ответы: 1; 2; 3; 4; 6;	0,4
			0,6
			0,8
			1
2) (1 б.) Верные ответы: 1; 3; 4; 6;	0,4	2) (1 б.) Верные ответы: 4;	1
	0,6		
	0,8		
	1		
3) (1 б.) Верные ответы: 4;	1	3) (1 б.) Верные ответы: 4;	1
4) (1 б.) Верные ответы: 1;	1	4) (1 б.) Верные ответы: 3;	1
5) (1 б.) Верные ответы: 2;	1	5) (1 б.) Верные ответы: 3;	1
6) (1 б.) Верные ответы: 4;	1	6) (1 б.) Верные ответы: 3;	1
7) (1 б.) Верные ответы: 2; 4; 6;	0,3	7) (1 б.) Верные ответы: 1; 2; 3;	0,3
	0,7		0,7
	1		1
8) (1 б.) Верные ответы: 2;	1	8) (1 б.) Верные ответы: 1;	1
	10,8		10,8

Критерий оценки

Баллы	Оценка
10,8-8,4	5
8,3-6,4	4
6,3-4,1	3
4-0	2

Лабораторная работа № 2

Задание 1. Создание составных форм, запросов, отчетов

Цель: Изучение способов создания составных форм, запросов и отчетов для связанных таблиц.

Создание составных форм

1. Создать базу данных **База№2**, которая содержит следующие таблицы:

1.1. **Студенты** (поля: *КодСтуд*, *ФамСтуд*, *ИмяСтуд*, *ПолСтуд*, *РождСтуд*).

1.2. **Родители** (поля: *КодСтуд*, *ФамРод*, *ИмяРод*, *ОтчРод*, *ПолРод*, *ТелРод*).

1.3. **Оценки** (поля: *КодСтуд*, *БаллХим*, *БаллИнф*, *БаллМат*, *Семестр*).

2. Ввести для каждого студента данные о двух родителях или трех ближайших родственниках.

3. Определить в качестве первичного ключа в таблице **Студенты** поле *КодСтуд*.

4. Открыть окно **Схема данных** и определить связи между созданными таблицами так, как представлено на рис.2.1.

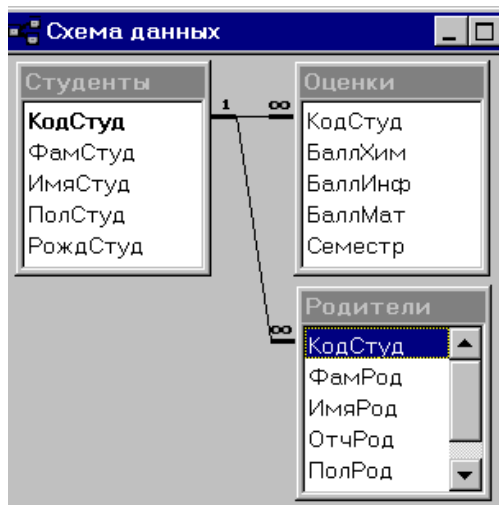


Рис.2.1

5. Сохранить все таблицы и сделать резервную копию базы данных **База№2**

6. Создать составную форму (для двух связанных таблиц *Студенты* и *Родители*).

6.1. На основе таблицы *Родители*, пользуясь режимом *Автоформа табличная*, создать *подчиненную* форму, содержащую следующие сведения: код студента, фамилии и имена обоих родителей, их пол и номер телефона.

6.2. На основе таблицы *Студенты* создать *главную* форму, содержащую все сведения о студентах.

6.3. Разместить *подчиненную форму* внутри *главной*, как показано на рис.2.2. Для этого использовать *режим Конструктора* для *главной формы* и «перетащить» в нее предварительно созданную *подчиненную форму* из вкладки *Формы* окна базы данных (также можно использовать элемент управления *Подчиненная форма/отчет* на панели инструментов *Панель Элементов*). Для детальной подгонки всех элементов форм воспользоваться услугами *Конструктора форм*, последовательно улучшая общий вид составной формы.

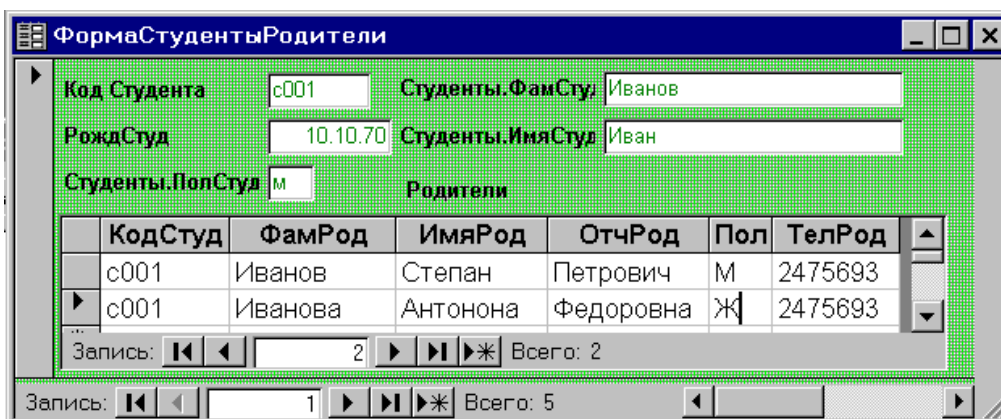


Рис.2.2

7. Заполнить таблицы *Студенты* и *Родители* правдоподобными данными для пяти студентов, пользуясь созданной формой.

8. Удалить запись о третьем по счету студенте. Пронаблюдать, как выполняется удаление соответствующих записей о родителях из таблицы *Родители*. Просмотреть таблицы *Родители* и *Студенты* в режиме таблицы. Убедиться в том, что данные удалены правильно.

9. Создать составную форму (для двух связанных таблиц *Студенты* и *Оценки*), используя главную форму *Студенты* и предварительно созданную подчиненную форму *Оценки*. Общий вид созданной формы должен соответствовать рис.2.3.

10. Создать две командные кнопки в примечании формы *Студенты* с надписями *Оценки* и *Родители*, нажатие на которые приведет к открытию одноименных форм.

10.1. Активизировать в форме *Студенты* область примечаний.

10.2. Щелкнуть мышью на элементах управления *Мастер* и *Кнопка* и разместить кнопку в примечании формы.

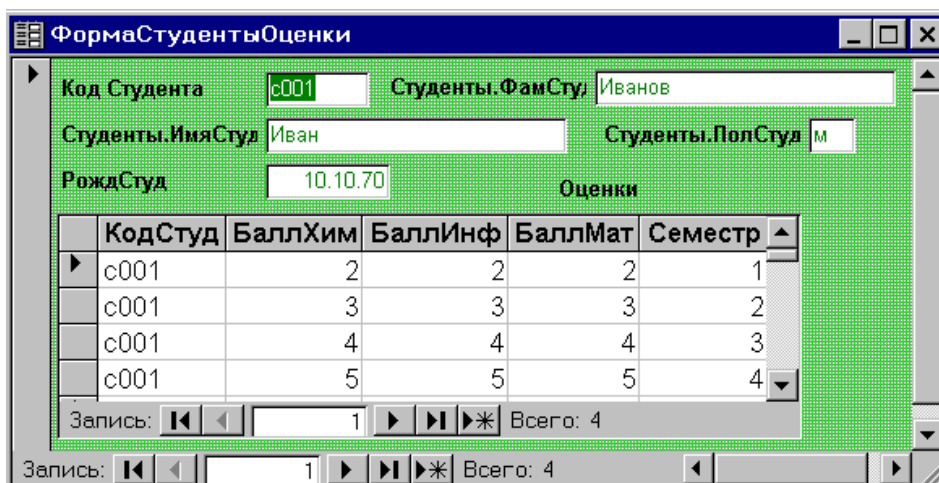



Рис.2.3

10.3. В окне мастера *Создание кнопок* указать *Категорию* - *Работа с формой* и *Действия* - *Открытие формы*.

10.4. Выбрать форму, открываемую нажатием кнопки (*Оценки* или *Родители*), указать, что отбор сведений не требуется, разместить на кнопках текст «Оценки» или «Родители».

10.5. Перейти в *режим формы* и проверить работоспособность кнопок.

10.6. Для созданной кнопки в контекстном меню выбрать пункт *Свойства* и вкладку *События*. В разделе *Нажатие кнопки* просмотреть назначенную ей процедуру обработки событий (кнопка ).

Создание составных запросов

11. Создать запрос **СтудОценки**, с помощью которого будут извлекаться данные из двух связанных таблиц **Студенты** и **Оценки**. В выборку должны попасть: фамилия и имя студента, его оценка по информатике и номер семестра. В качестве параметров запроса должны выступать фамилия студента и номер семестра. Образец запроса представлен на рис.2.4:

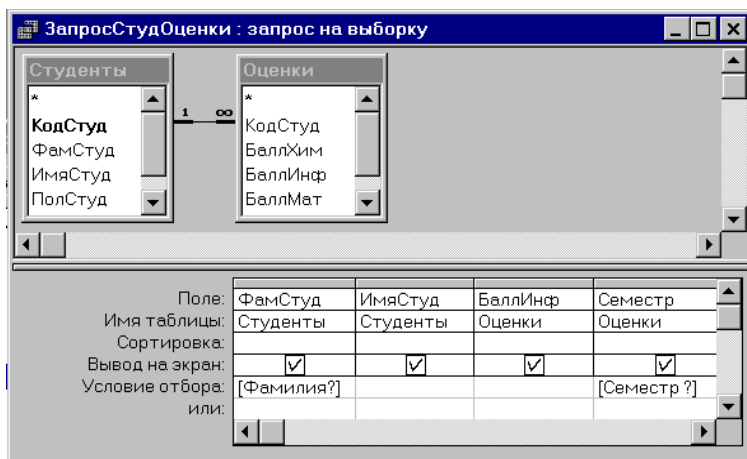


Рис.2.4

11.1. Выполнить запрос и убедиться в правильности его работы.

Создание составных отчетов

12. Создать отчет для двух связанных таблиц **Студенты** и **Родители**. В этот отчет должны попасть такие данные как фамилия, имя и дата рождения студента, а также фамилии, имена и отчества обоих его родителей и номера их телефонов. Внешний вид отчета, включая заголовок и общее оформление, должен соответствовать приведенному на рис.2.5 образцу.

12.1. Для создания составного отчета первоначально нужно создать *подчиненный* отчет для таблицы стороны *Много* - таблицы **Родители**. Для создания такого отчета рекомендуется использовать режим создания отчета *Автоотчет*, а затем скорректировать его с использованием Конструктора отчетов.

12.2. Создать *главный* отчет для таблицы стороны *Один* - таблицы **Студенты**, используя Мастер отчетов. На последнем шаге мастера перейти

к работе в режиме конструктора и «внедрить» в главный отчет, созданный ранее отчет *Родители*.

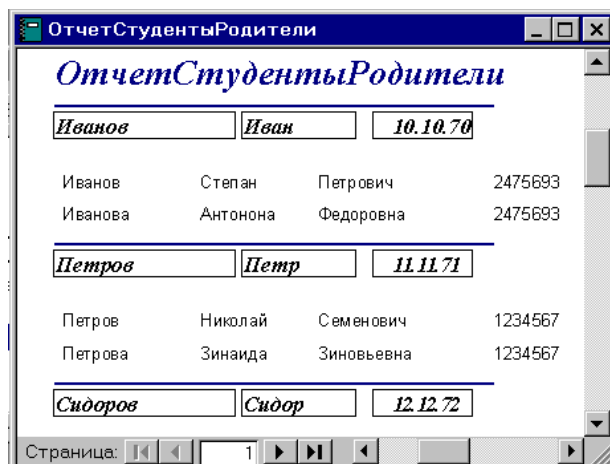


Рис.2.5

12.3. Общий вид итогового отчета представлен на рис. 2.5:

13. Сохранить базу данных для дальнейшего использования.

Задание 2. Создание макросов

Цель: Знакомство с созданием простых макросов и их использованием для автоматизации работы базы данных.

1. Открыть базу данных *База№2*.

2. Создать макрос *Макрос1*, обеспечивающий выполнение следующих действий (рис.2.6):

- выдачу сообщения «НАЧАЛО РАБОТЫ МАКРОСА». Окно, в котором выводится текст, озаглавить как «Окно начала макроса»;
- выполнение запроса *СтудОценки*, созданного ранее. В качестве первого аргумента - *Имени запроса* - выбрать *СтудОценки*. В качестве режима просмотра выбрать режим *Таблица*;
- выдачу сообщения «КОНЕЦ РАБОТЫ МАКРОСА». Окно, в котором выводится текст, озаглавить как «Окно окончания макроса».

Примечание. Для команды *Сообщение* обязательными параметрами, задаваемыми в нижней части диалогового окна (рис.2.6) являются аргумент *Сообщение*, в окне которого записывается выводимый текст и аргумент *Заголовок*, в окне которого записывается текст заголовка окна.

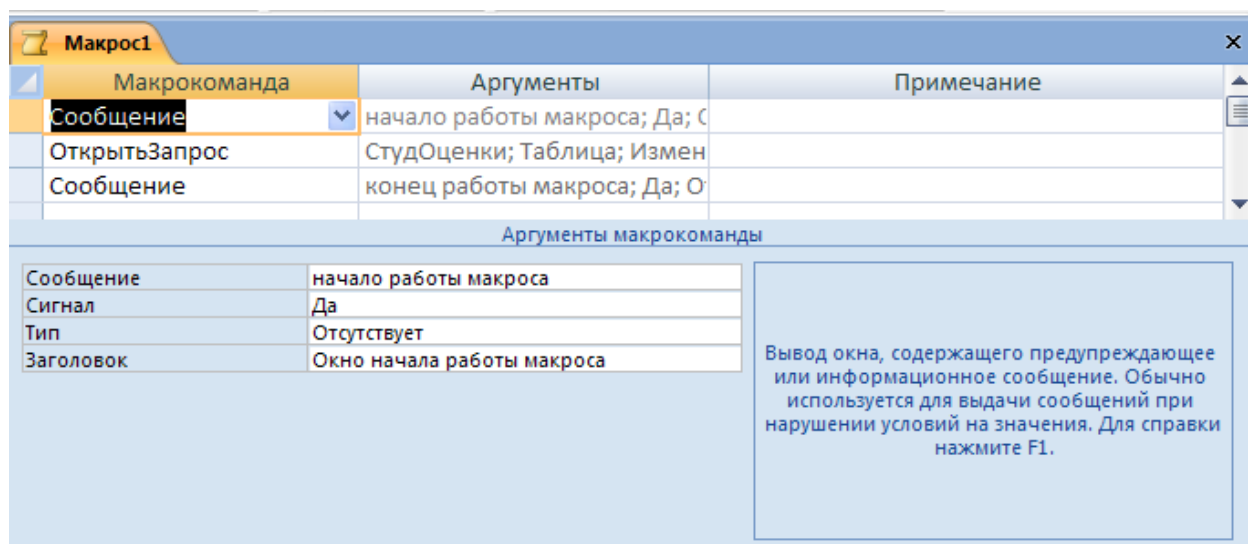


Рис.2.6

3. Выполнить макрос *Макрос1*. Убедиться в правильности выполнения.

4. Установить с помощью кнопки на панели макросов режим пошагового выполнения и, выполнив макрос по шагам, пронаблюдать сообщения, выдаваемые на каждом шаге.

5. Создать макрос *Макрос2*.

5.1. Добавить в макрос *Макрос1*, после второй команды еще две команды:

- для выдачи информационного сообщения *ПАУЗА* после появления таблицы с результатом запроса;
- для закрытия окна с результатом выполнения запроса.

5.2. Сохранить макрос под новым именем *Макрос2* (Рис.2.7).

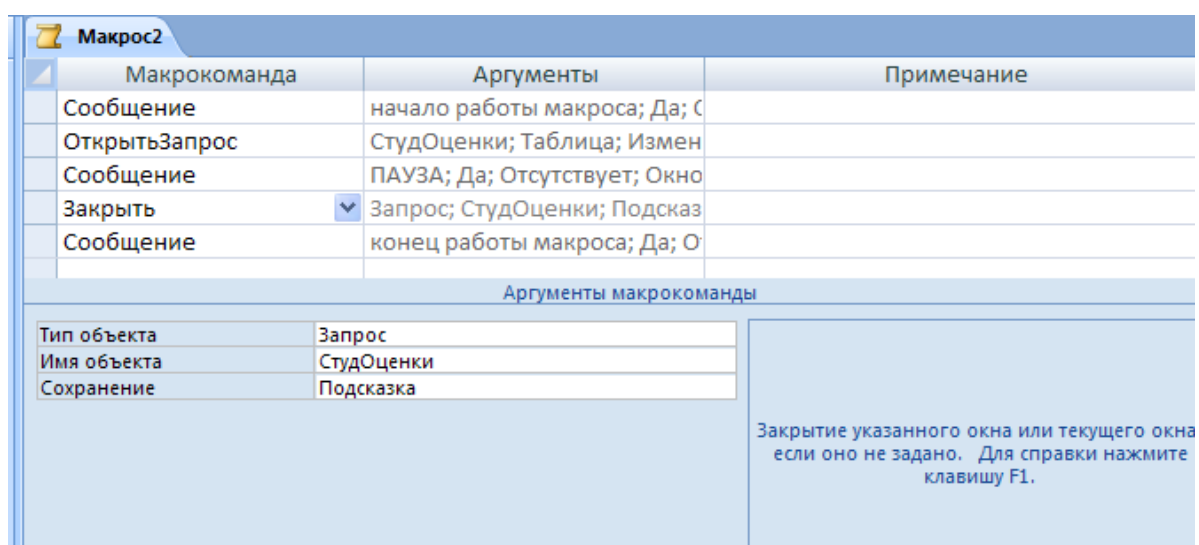


Рис.2.7

6. Выполнить «Макрос2». Убедиться в правильности выполнения.

7. Преобразовать макрос в модуль VBA.

7.1. Перейти на вкладку *Макросы* и выделить *Макрос2*.

7.2. Осуществить преобразование макроса *Макрос2* в модуль VBA (меню «Создание»–«Макрос»–«Модуль») и проанализировать его структуру.

7.4. Запустить созданный модуль (клавиша F5).

8. Создать две командные кнопки в заголовке формы *Студенты* с подписями «Родители» и «Оценки», нажатие на которые вызовет открытие соответствующих форм (рис.2.7).

8.1. Создать макросы *Родители* и *Оценки*, обеспечивающие открытие соответствующих форм.

8.1.1. На вкладке *Макросы* перейти в режим создания макросов.

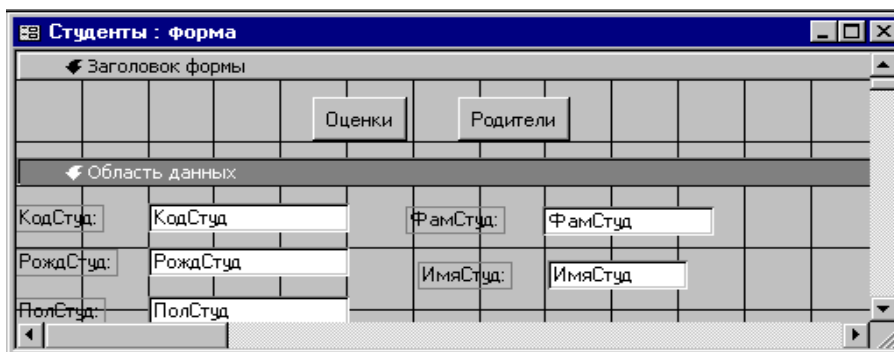


Рис.2.7

8.1.2. Перетащить из вкладки *Формы* окна базы данных форму *Родители* в область *Макрокоманда* и проверить заполнение аргументов макрокоманды.

8.1.3. Сохранить макрос под именем *Родители*.

8.1.4. Проверить работоспособность макроса.

8.1.5. Повторить п.8.1.1-8.1.4 для формы *Оценки*.

8.2. Создать командные кнопки с надписями «Родители» и «Оценки», к которым присоединить соответствующие макросы.

8.2.1. Открыть форму *Студенты* в режиме конструктора и активизировать в ней область заголовка.

8.2.2. Перетащить с вкладки *Макросы* окна базы данных макросы *Родители* и *Оценки* в область заголовка формы *Студенты*.

8.2.3. Отформатировать кнопки, пользуясь контекстным меню «Размер»-«По размеру данных».

8.2.4. Проверить работоспособность кнопок.

9. Сохранить базу данных на диске для дальнейшего использования.

Создание открытых баз данных

Задание 3. Гиперссылки и объекты OLE

Цель: Изучение методов работы с гиперссылками и объектами, созданными внешними приложениями.

1. **Открыть** базу данных *База.№2* и убедиться в наличии таблицы *Студенты*, содержащей в себе не менее шести записей.

2. **Создать** в таблице *Студенты* новое поле *ДопИнформ*, имеющее тип *Гиперссылка*.

3. **Заполнить** поле *ДопИнформ* следующими данными:

3.1. Для первого студента в *режиме таблицы* ввести в поле *ДопИнформ* ссылку на сайт колледжа **<http://www.omkpt.ru/>**.

3.2. Для второго студента создать *гиперссылку* на отделение *сервиса*.

3.2.1. В диалоговом окне *Изменение гиперссылки* (контекстное меню «Гиперссылка»-«Изменить гиперссылку») в поле *Текст* ввести “*Отделение сервиса*”, а в качестве имени web-страницы ввести адрес - **<http://omkpt.ru/qualifications/catering>**.

3.3. Для третьего студента разместить ссылку на страницу сайта *Достижения колледжа*.

3.3.1. В диалоговом окне *Изменение гиперссылки* (контекстное меню «Гиперссылка»-«Изменить гиперссылку») в поле *Текст* ввести “*Достижения колледжа*”, а в качестве имени web-страницы ввести адрес - **<http://omkpt.ru/awards>**.

3.4. Создать для следующих трех студентов файлы с визитными карточками и установить на них гиперссылки.

3.4.1. В текстовом редакторе **MS Word** создать три документа с именами *Визитка1*, *Визитка2* и *Визитка3* по приведенному на рис.2.8 образцу.

3.4.2. Для каждой из трех записей в диалоговом окне *Изменение гиперссылки* в поле *Текст* ввести «*Визитка*», а ссылку на файл создать, воспользовавшись окном *Связать с файлом*.

4. Убедиться в работоспособности всех шести созданных гиперссылок (рис.2.9), поочередно обратившись к страницам сайта колледжа и к документам Word.

Фамилия	Абдулов
Имя	Александр
ССУЗ	ОмКПТ
Отделение	Сервиса
Группа	331

Рис.2.8

ДопИнф
Визитка1
http://omkpt.ru/
отделение сервиса
Достижения колледжа
Визитка2
Визитка3

Рис.2.9

5. Создать в таблице *Студенты* новое поле *Визитка*, имеющее тип *поле объекта OLE*.

6. Заполнить поле *Визитка* файлами *Визитка1*, *Визитка2* и *Визитка3* для четвертой, пятой и шестой записей соответственно.

6.1. Открыть окно *Вставка объекта*, воспользовавшись контекстным меню «Вставить объект», выбрать создание объекта из файла и указать путь к файлу *Визитка1.doc* (кнопка *Обзор*).

6.2. Повторить аналогичные действия для файлов *Визитка2* и *Визитка 3*.

6.3. Последовательно открыть внедренные объекты двойным щелчком на надписи *Документ Microsoft Word* (рис.2.10) и убедиться в их работоспособности.

	Визитка
	Документ Microsoft Word
	Документ Microsoft Word
	Документ Microsoft Word

Рис. 2.10

7. Изменить форму *Студенты* добавив в нее поле *ДопИнформ* и поле *Визитка* в соответствии с рис.2.11 (Конструктор – Вставить поле).

8. Просмотреть содержимое созданных полей через форму.

9. Открыть файл *Визитка1* через гиперссылку и изменить для соответствующего студента номер группы. Пронаблюдать, отобразились ли изменения в поле *Визитка*.

10. Сделать вывод о возможностях и сферах применения полей гиперссылок и полей объектов OLE.

11. Сохранить базу данных для дальнейшего использования.

Студенты

Оценки Родители

КодСтуд
007

РождСтуд
12.08.1993

ПолСтуд
ж

ФамСтуд
Иванова

ИмяСтуд
Виктория

ДопИнф:
[Визитка1](#)

Визитка:

Фамилия	Абдулов
Имя	Александр
ССУЗ	ОмКПТ
Отделение	Сервиса
Группа	331

Рис.2.11

Задание 4. Экспорт и импорт данных

Цель: Знакомство с технологиями экспорта и импорта данных в MS Access.

1. Открыть базу данных *База№2*

2. Экспортировать данные из таблицы *Студенты* в web-страницу.

2.1. Активизировать таблицу *Студенты*.

2.2. Пользуясь меню «Внешние данные»-«Экспорт»-«Дополнительно» вызвать диалоговое окно *Экспорт объекта - Документ HTML*.

Примечание. Для подключения автоматического форматирования MS Access установить флажки по рисунку 2.12, на запрос о *Шаблоне HTML* нажать кнопку *OK*.

Выберите место назначения для экспортируемых данных

Укажите имя и формат целевого файла.

Имя файла:

Укажите параметры экспорта.

Экспортировать данные с макетом и форматированием.
Выберите этот параметр для сохранения сведений о форматировании и макете при экспорте таблицы, запроса, формы отчета.

Открыть целевой файл после завершения операции экспорта.
Выберите этот параметр для просмотра результатов операции экспорта. Параметр доступен только при экспорте форматированных данных.

Экспортировать только выбранные записи.
Выберите этот параметр для экспорта только выбранных записей. Параметр доступен только при экспорте

2.4. Сохранить созданную web-страницу в рабочей папке и просмотреть ее содержимое с помощью любого доступного браузера (см. рис.2.13).

КодСтуд	ФамСтуд	ИмяСтуд	ПолСтуд	РождСтуд	ДопИнф	Визитка
007	Иванова	Виктория	ж	12.08.1993	Визитка1	
s001	Иванов	Павел	М	01.05.1990	http://omkpt.ru/	
s002	Петрова	Елена	Ж	16.01.1989	отделение сервиса	
s004	Ермолаева	Дарья	Ж	05.12.1989	Достижения колледжа	
s005	Суслов	Владимир	М	28.09.1988	Визитка2	
s006	Романов	Игорь	М	02.02.1990	Визитка3	

Рис. 2.13

2.5. Просмотреть созданную web-страницу в Блокноте в режиме HTML. Обратить внимание на теги таблицы (TABLE, TD, TR).

3. Экспортировать данные из отчета *Студенты* в web-страницу (рис.2.14).

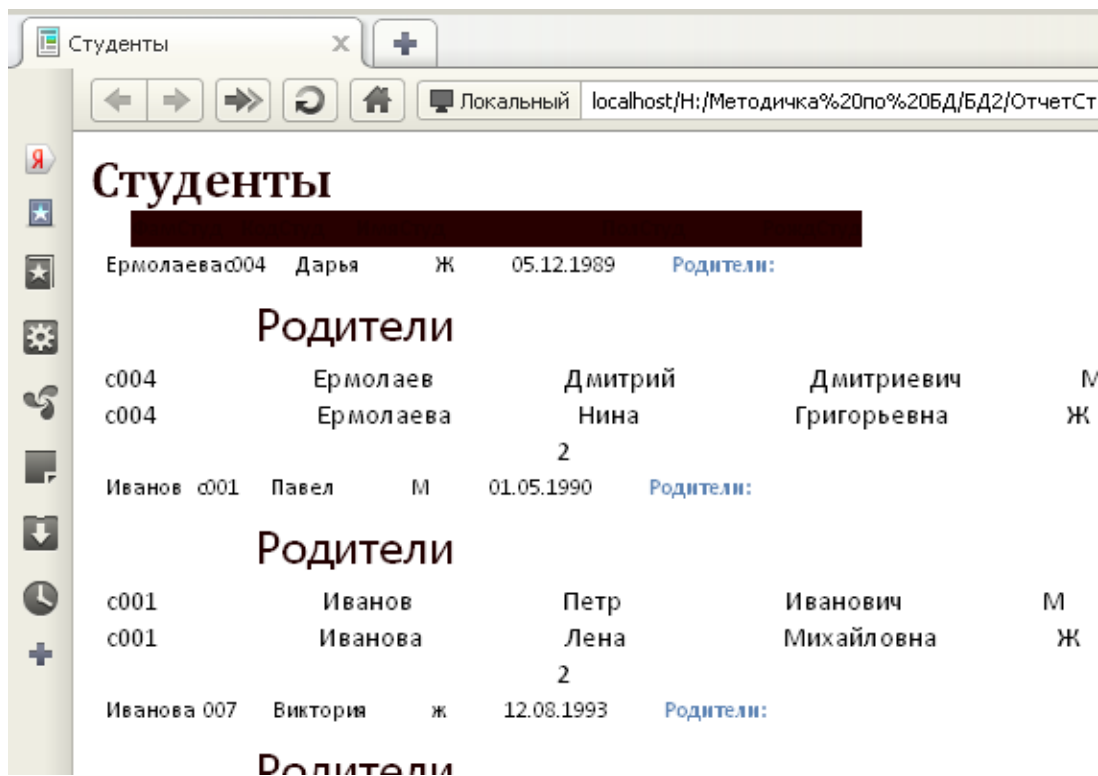


Рис. 2.14

3.1. С помощью режима создания отчета *Автоотчет* создать отчет *СтудWeb* на основе таблицы *Студенты*.

3.2. С помощью *Конструктора отчетов* изменить отчет *СтудWeb* так, чтобы он занимал не менее трех страниц (*Макет – В столбик*).

3.3. Экспортировать отчет *СтудWeb* в web-страницы, пользуясь окном *Экспорт объекта*.

3.4. Пользуясь созданными гиперссылками перехода научиться перемещаться между web-страницами, обратив внимание на их имена и количество.

4. Импортировать данные из созданной в п.2 web-страницы *Студенты* в новую таблицу.

4.1. Открыть окно импорта данных (меню «Внешние данные»-«Импорт»-«Дополнительно»).

4.2. С помощью кнопки *Обзор* выбрать web-страницу ***Студенты.html***.

4.3. Выбрать сохранение в новой таблице.

4.4. На следующем шаге, в открывшемся диалоговом окне указать, что первая строка содержит заголовки столбцов.

4.5. Далее, перемещаясь по полям, выбрать соответствующие типы данных для каждого из полей.

4.6. Определить в качестве ключевого поля поле *КодСтуд.*

4.7. На последнем шаге дать новой таблице имя ***Студенты_Импорт***.

4.8. Открыть таблицу ***Студенты_Импорт*** и сравнить ее с исходной таблицей ***Студенты***.

5. Сохранить базу данных для дальнейшего использования.

Задание 5. Макросы с условиями

Цель: Использование групп макросов и макросов с условиями для автоматизации работы базы данных.

1. Для базы данных *База№1* создать макрос *МакАдр_1*, который выполняет следующие действия:

- выдает информационное сообщение о начале работы;
- открывает форму *Адреса*;
- выдает об этом информационное сообщение и приостанавливает работу до нажатия *Кнопки*;
- открывает отчет *Девушки*;
- выдает информационное сообщение об окончании работы.

1.1. Дать этому макросу имя *МакАдр_1* и сохранить его.

1.2. Выполнить макрос *МакАдр_1* несколько раз и попытаться проследить последовательность действий. Обратит внимание на то, что форма *Адреса* и отчет *Девушки* остаются открытыми после работы макроса.

2. Создать макрос *МакАдр_2*, выполняющий следующие действия:

- выдачу сообщения о начале работы;
- открытие формы *Адреса*;
- подачу звукового сигнала;
- выдачу *информационного* сообщения о предстоящем закрытии формы *Адреса*;
- закрытие формы *Адреса*;
- открытие отчета *Девушки*;
- выдачу *предупреждающего* сообщения о предстоящем закрытии отчета;
- закрытие отчета *Девушки*.

2.1. Дать этому макросу имя *МакАдр_2* и сохранить его.

2.2. Выполнить макрос *МакАдр_2* в пошаговом и в автоматическом режимах.

3. Создать макрос *МакУсл_1* с командой безусловного перехода.

3.1. Для макроса *МакАдр_1*, пользуясь кнопками *Панели инструментов Макрос*, открыть столбец *Имя Макроса* (рис.5.1) и для каждой строки (команды) задать имена aaa, bbb, sss, ddd, eee и т.д.

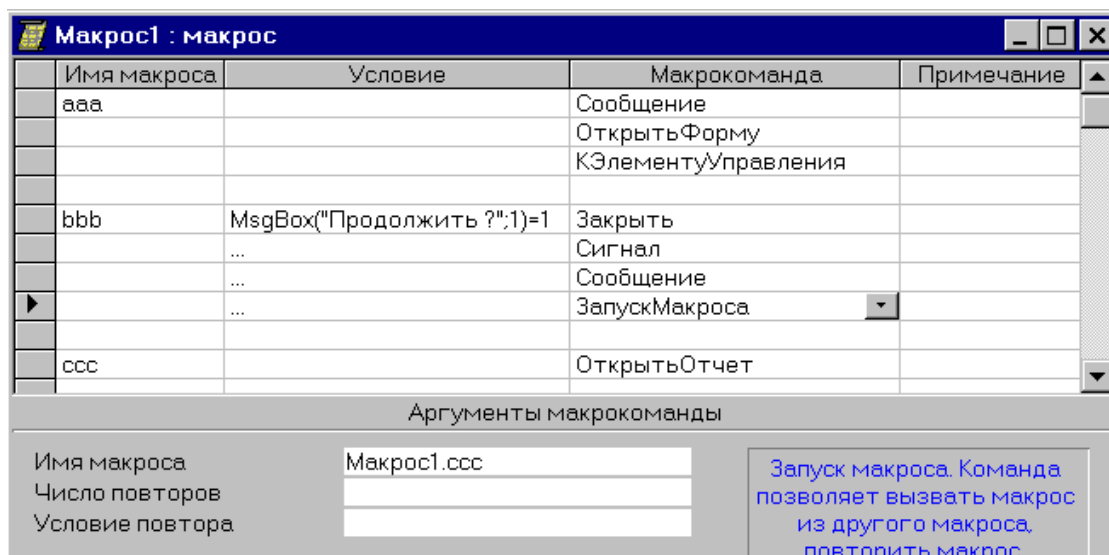


Рис.5.1

3.2. Вставить после первой команды *Сообщение* команду *Запуск макроса* с параметром eee, которая обеспечивает безусловный переход к команде eee после выполнения первой команды. Выполнить макрос и убедиться в правильности его работы.

3.3. Дать этому макросу имя *МакУсл_1* и сохранить его.

4. Добавить в окне работы с макросами столбец *Условие* с помощью кнопок *Панели инструментов Макросы*.

5. Создать макрос *МакУсл_2* (рис.5.1) с командой *условного перехода*, выполняющий следующие действия:

- выдачу стартового сообщения;
- открытие формы *Адреса*;
- переход к элементу управления - полю *ПочтКод*, представленному в форме *Адреса*;
- закрытие формы *Адреса*;

- подачу звукового сигнала;
- выдачу сообщения о завершении работы.

6.1. В качестве условия для выполнения закрытия формы *Адреса* определить значение функции **MsgBox(“Продолжить ?”);1=1**. (Значение 1 функция **MsgBox()** принимает при нажатии кнопки, представленной в диалоговом окне, отображаемом этой функцией).

6.2. Каждой группе команд дать имена aaa, bbb, ccc (рис.5.1). Сохранить созданный макрос, не выполняя его.

6.3. Вызвать форму *Адреса* и добавить в область примечания формы кнопку с надписью «Продолжение макроса», как показано на рис.5.2.

6.3.1. Для созданной кнопки в контекстном меню выбрать пункт *Свойства* и вкладку *События*. В разделе *Нажатие кнопки* указать необходимость выполнения команды bbb макроса *МакУсл_2*.

6.3.2. Закрыть все вкладки и сохранить созданную форму.

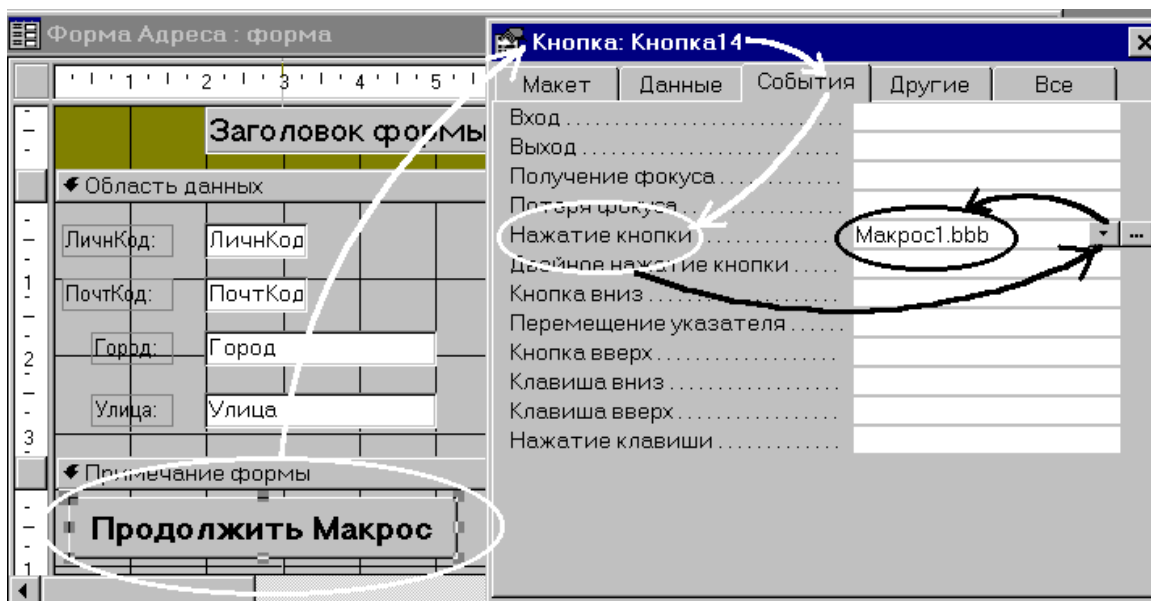


Рис.5

.2

6.4. Выполнить макрос *МакУсл_2* в пошаговом режиме. Убедиться, что логика работы макроса верна и все его команды выполняются в нужной последовательности. Если макрос выполняется неправильно, проанализировать все сообщения, получаемые на каждом шаге, выявить причину и внести в макрос необходимые изменения.

6.5. Дополнить созданный макрос еще одной командой (с именем ссс), которая обеспечит открытие отчета *Девушки* и выдачу итогового сообщения об окончании работы.

7. Сохранить базу данных на диске для дальнейшего использования.