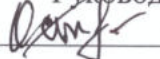


МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
«СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 32»

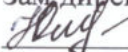
«Рассмотрено и утверждено»

Руководитель МО

 Оманова С.О.

«Согласовано»

Зам. директора по УВР

 Гаджиева Ю.У.

«Утверждено»

Директор МБОУ «СОШ № 32»

 Гасанова Д.А.

« 01 сентября 2023 »



Программа
Основы проектирования баз данных
в среде Visual FoxPro
(учебно-методическое пособие)

Разработала
учитель информатики
Ясулова Х.С.

Махачкала 2023

Учебно-методическое пособие посвящено изучению наиболее популярного и мощного пакета программ Visual FoxPro, предназначенного для построения профессиональных приложений, работающих с базами данных.

На конкретных примерах последовательно описан процесс создания основных элементов СУБД (таблицы и отношения между таблицами, формы, отчёты, запросы и т.д.). включены задания для самостоятельной работы, а также контрольные вопросы для самопроверки усвоения материала.

Пособие представляет интерес для студентов экономических специальностей, изучающих основы создания и функционирования информационных систем в экономике и желающих приобрести навыки работы в объектно-ориентированной, визуально-программируемой среде проектирования баз данных Visual FoxPro.

ОГЛАВЛЕНИЕ

Введение	4
Основные Понятия Теории Баз Данных	5
Лабораторная работа №1	9
Лабораторная работа №2	17
Лабораторная работа №3	23
Лабораторная работа №4	29
Лабораторная работа №5	36
Лабораторная работа №6	43
Лабораторная работа №7	50
Образцы заданий для создания БД	62
Список используемой литературы	70

ВВЕДЕНИЕ

В современном информационном обществе значимость информации как ресурса неоспорима. Информация имеет неопределимое свойство – будучи аккумулированной и обработанной с определенных позиций, она дает новые сведения, приводит к новым знаниям. Современный экономист должен уметь обрабатывать информацию, представлять ее в форме, удобной для обработки, хранения, передачи, а значит, должен владеть средствами проектирования информационных систем, ядром которых является база данных. Поэтому в настоящее время наблюдается бурный рост информационных технологий и стремительное развитие систем управления базами данных СУБД.

Наиболее популярной СУБД является Microsoft Visual FoxPro. В ней реализованы все атрибуты реляционных СУБД. В БД поддерживается целостность данных с помощью первичных ключей и связей между таблицами.

В данном пособии содержится теоретический материал, в котором освещаются вопросы проектирования и создания реляционных БД.

Лабораторные работы включают в себя следующие пункты:

- Теоретическая часть – содержит краткое описание предполагаемого объема работы и сопроводительную информацию.
- Цель работы – содержит перечень конкретных задач по выполнению лабораторной работы.
- Практическая часть – содержит пошаговое описание алгоритма выполняемой работы.
- Задание – содержит задание для самостоятельной работы и приобретение практических навыков.
- Контрольные вопросы – содержит перечень вопросов для самопроверки усвоения пройденного материала.

В Visual FoxPro вся информация содержится в БД и таблицах БД. В 1-й лабораторной работе рассмотрено создание БД и таблиц БД.

В БД могут быть реализованы связи между таблицами и определены условия целостности данных. Во 2-й лабораторной работе рассмотрено создание связей, определение отношений между таблицами, определение целостности данных.

Формы в Visual FoxPro используются для просмотра или ввода данных в таблицы. В 3-4-й лабораторных работах рассмотрено создание форм, как средств для более быстрого и удобного ввода данных в таблицы.

Отчеты используются для печати содержащейся в БД информации. В 5-й лабораторной работе рассмотрено создание отчета, редактирование отчета.

Запросы используются для осуществления выборки данных из одной или нескольких таблиц. В 6,8-й лабораторных работах рассмотрено создание запросов, а также создание запросов с помощью команд языка SQL.

Меню приложения используются для объединения всех создаваемых элементов БД в рамках проекта БД и представляет собой программу. В 7-й лабораторной работе рассмотрено создание меню приложения.

В пособии также содержится примеры описаний информационных систем, рассчитанные на самостоятельное изучение и реализацию на их основе соответствующих проектов СУБД.

ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ БАЗ ДАННЫХ

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

Информационная система - система, реализующая автоматизированный сбор, обработку и манипулирование данными и включающая технические средства обработки данных, программное обеспечение и соответствующий персонал.

Цель любой информационной системы - обработка данных об объектах реального мира. Основой информационной системы является база данных. **База данных (БД)** - это поименованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой 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.

Объектом называется элемент предметной области, информацию о котором мы сохраняем. Объект может быть реальным (например, человек, изделие или населенный пункт) и абстрактным (например, событие, счет покупателя или изучаемый студентами курс).

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро производить выборку с произвольным сочетанием признаков. Большое значение при этом приобретает структурирование данных.

Структурирование данных - это введение соглашений о способах представления данных.

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.

Система управления базами данных (СУБД) - это комплекс программных и языковых средств, предназначенных для создания, ведения и совместного применения баз данных многими пользователями.

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

Структурными элементами базы данных являются:

- **Поле** - элементарная единица логической организации данных, которая соответствует неделимой единице информации - реквизиту. Для описания поля используются следующие характеристики: имя, длина, тип и точность (для числовых данных). В структуре записи файла указываются поля, значения которых являются ключами: первичными и внешними.

- **Первичный ключ** - это одно или несколько полей, однозначно идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется простым, если из нескольких полей - составным ключом.

- **Внешний ключ** - это одно или несколько полей, которые выполняют роль поисковых или группировочных признаков. В отличие от первичного, значение внешнего ключа может повторяться в нескольких записях файла, то есть он не является уникальным. Если по значению первичного ключа может быть найден один единственный экземпляр записи, то по внешнему - несколько.

- **Файл (таблица)** - совокупность экземпляров записей одной структуры. Описание логической структуры записи файла содержит последовательность расположения полей записи и их основные характеристики,

– **Запись** - совокупность логически связанных полей.

– **Экземпляр записи** - отдельная реализация записи, содержащая конкретные значения ее полей. 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.

Основы реляционной модели данных были впервые изложены в статье научного сотрудника компании IBM доктора Е.Ф. Кодда в 1970 г. Наиболее распространенная трактовка реляционной модели данных принадлежит К. Дейту. Согласно Дейту, реляционная модель состоит из трех частей:

– *Структурной части.*

– *Целостной части.*

– *Манипуляционной части.*

Структурная часть описывает, какие объекты рассматриваются реляционной моделью. Постулируется, что единственной структурой данных, используемой в реляционной модели, являются нормализованные отношения.

Целостная часть описывает ограничения специального вида, которые должны выполняться для любых отношений в любых реляционных базах данных. Это целостность сущностей и целостность внешних ключей.

Манипуляционная часть описывает два эквивалентных способа манипулирования реляционными данными - реляционную алгебру и реляционное исчисление.

Фундаментальным понятием реляционной модели данных является понятие отношения. Реляционная модель описывает, какие данные могут храниться в реляционных базах данных, а также способы манипулирования такими данными.

В реляционных базах данных вся информация сведена в таблицы, строки и столбцы которых называются записями и полями соответственно. Эти таблицы получили название реляций (от слова relation - отношение), поэтому модель стала называться реляционной. Записи в таблицах не повторяются. Их уникальность обеспечивается первичным ключом, содержащим набор полей, однозначно определяющих запись. Для быстрого поиска в базе данных создаются индексы по одному или нескольким полям таблицы. Значения индексов хранятся в упорядоченном виде и содержат ссылки на записи таблицы. Для автоматической поддержки целостности связанных данных, находящихся в разных таблицах, используются первичные и внешние ключи.

Реляционная модель основана на теории множеств и математической логике. Такой фундамент обеспечивает математическую строгость реляционной модели данных.

Разработка приложений на основе СУБД предполагает подготовку решений по структуре БД. Эти решения непосредственно связаны с немашинной сферой - с описанием немашинной информационной базы, ее документов, содержащих необходимую информацию, а также с постановкой и алгоритмизацией задач по обработке этой информации.

На начальном этапе разработки структуры БД целесообразно построение информационно-логической модели, отражающей логическую структуру информации предметной области. Такая модель, отвечающая требованиям нормализации данных, является основой создания реляционных баз данных.

В соответствии с разработанной структурой базы данных осуществляется ее создание средствами СУБД на машинном носителе и ввод в эксплуатацию. Для обеспечения процессов создания БД и ее эксплуатации необходимо знание возможностей инструментальных средств СУБД. При этом следует руководствоваться рекомендациями по технологии использования средств СУБД. Такая технология должна определять все

необходимые процессы, включая первоначальный ввод, загрузку БД и контроль данных, выполнение операций по внесению изменений, реализацию запросов для получения нужных справок, восстановление БД и т.п. Одним из важнейших этапов этой технологии является подготовка экранных форм ввода-вывода для загрузки информации с документов внемашинной сферы в базу данных, корректировки данных и их просмотра.

Обработка данных средствами СУБД, то есть добавление, удаление, изменение и выборка данных производится при помощи языка запросов, встроенного алгоритмического языка и других средств СУБД.

Язык запросов SQL реализован в целом ряде популярных СУБД для различных типов ЭВМ либо как базовый, либо как альтернативный. В силу своего широкого использования является международным стандартом языка запросов. Язык SQL предоставляет развитые возможности как конечным пользователям, так и специалистам в области обработки данных.

В свою очередь, на основе реляционной модели были разработаны различные языки для доступа к реляционным данным, такие как SEQUEL, SQL, QUEL и другие. Фактическим промышленным стандартом в настоящее время стал язык SQL (Structured Query Language - язык структурированных запросов).

Конкретная реализация языка, может включать в себя более широкие возможности по сравнению со стандартизованными SQL, например, больше типов данных, большее количество команд, больше дополнительных опций имеющихся команд. Такие возможности делают работу с конкретной СУБД более эффективной, кроме того, такие нестандартные возможности языка проходят практическую апробацию и со временем могут быть включены в стандарт.

СУБД может иметь включающий или базовый язык программирования. В СУБД с включающим языком используется один из универсальных алгоритмических языков (C, Pascal и т.п.). Прикладная программа, написанная на включающем языке, может инициировать команды СУБД. В СУБД наряду с базовым языком применяется собственный алгоритмический язык, позволяющий кроме операций манипулирования данными выполнять различные вычисления и обработку данных.

В данном учебно-методическом пособии представлено систематическое введение в основы функционирования реляционных моделей данных и проектирования реляционных баз данных в среде **Visual FoxPro**.

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Лабораторная работа №1

Создание однотоабличной БД в среде Microsoft Visual FoxPro

Теоретическая часть

При создании приложения используется проект, который объединяет элементы приложения **Visual FoxPro** и группирует их по типам. База данных в **Visual FoxPro** – это совокупность таблиц, отношений между таблицами, индексов, триггеров, хранимых процедур. База данных является частью проекта, поэтому её целесообразно создавать в окне проекта. Таблицы составляют основу вашей базы данных. В них будет храниться вся необходимая информация. В дальнейшем данные в таблице будут дополняться новыми данными, редактироваться или исключаться из таблицы. Поля таблицы предназначены для хранения в них данных. Это могут быть числа, текстовая информация, даты, графические файлы и т.д. В **Visual FoxPro** допустимыми являются типы полей, указанные в таблице 1.

Таблица 1. Некоторые типы данных, используемых в СУБД FoxPro

Команда	Описание
CREATE	создание элементов БД (базы, таблиц, отчётов, запросов и т.д.)
MODIFY (STRUCTURE)	изменение элементов БД (базы, таблиц, отчётов, запросов и т.д.)
BROWSE, EDIT, CHANGE, APPEND	команды редактирования и просмотра содержимого таблиц
OPEN DATABASE	Открытие БД
CLOSE	закрытие элементов БД (базы, таблиц, отчётов, запросов и т.д.)
QUIT	Выход из Visual FoxPro

Таблица 2. Некоторые команды СУБД FoxPro

Тип	Описание	Пример
Integer	Целые числа	9846
Numeric	Данные, с фиксированной точкой	3.1456
		-56.235
Currency	Поле, содержащее числа денежного типа	23456.3422
		-237.0000
Character	Текст или числа, не требующие проведение расчётов	“Основы Visual FoxPro”
		“123”
		“01/07/04”
Logical	Поля, содержащие только одно из двух возможных значений (да/нет)	True; False
Date	Дата	01/07/04
DateTime	Дата и время	01/07/04 12:30:00 pm
Memo	Очень длинный текст или комбинация текста и чисел	

Цель работы

1. Создать проект *Информационные системы*
2. Создать БД *Штат*
3. Создать в БД *Штат* таблицу *Сотрудники*
4. Заполнить таблицу данными

Практическая часть

1. Создать в служебной папке *Мои документы* новую папку и присвоить ей имя, пример: *МояРабота* (указывать имя своей работы)
2. Запустить программу **Microsoft Visual FoxPro: Пуск/программы/Visual FoxPro**
3. Ознакомиться с элементами рабочего окна программы

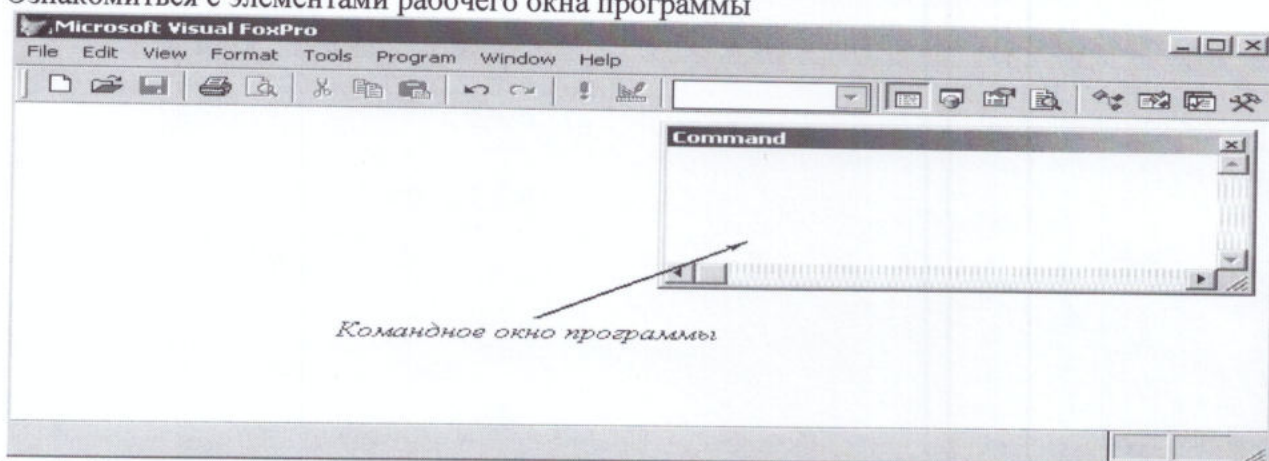


Рис.1. Главное окно Visual FoxPro 8.0

4. Создать новый проект: **File/New/Project/New File**, указать созданную нами ранее папку, присвоить имя проекту *Информационные системы* и сохранить. Все создаваемые в последующем элементы приложения будут храниться в проекте *Информационные системы*
5. БД создаётся аналогично: **File/New/Database/New file**, присвоить имя *Штат* и сохранить. Структура проекта и его элементы отражаются в окне программы **Project Manager** (Менеджер проекта)

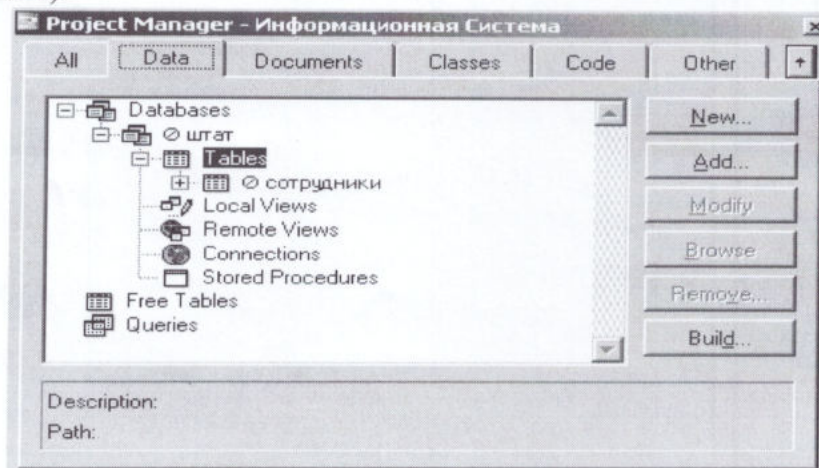


Рис.2. Окно Project Manager

6. Создать таблицу в БД *штат*: в окне **Project Manager** (см. рис.2.) щёлкнуть клавишей мыши на вкладке **Data/Databases/штат/Tables/New/New table**, присвоить имя *Сотрудники* и сохранить
7. В появившемся окне **Table Designer** (Конструктор таблиц) (см. рис.3.) на вкладке **Fields** (Поля) создать структуру таблицы в соответствии с таблицей 3.

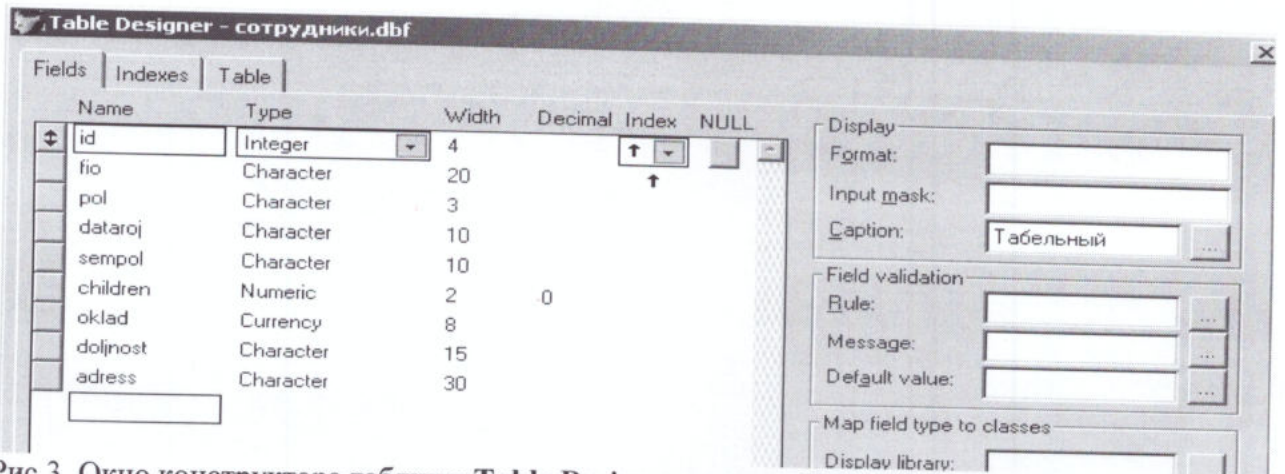


Рис.3. Окно конструктора таблицы **Table Designer** вкладка **Fields**

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
id	Integer	4	↑Ascending	Табельный номер
fio	Character	20		ФИО
pol	Character	3		Пол
dataroj	Character	10		Дата рождения
sempol	Character	10		Семейное положение
children	Numeric	2		Количество детей
oklad	Carrency	8		Оклад
doljnost	Character	15		Должность
adress	Character	30		Место жительства

Таблица 3. Определение полей таблицы *Сотрудники* в окне конструктора таблицы **Table Designer**

8. Перейти на вкладку **Indexes** (Индексы) окна конструктора таблицы **Table Designer** и присвоить созданному индексу значения в соответствии с таблицей 4. Это необходимо для создания ключевого поля

Order	Name	Type	Expression	Filter	Collate
↑	id	Primary	id		Machine

9. Для ввода и редактирования данных в таблице *Сотрудники* в командном окне программы введите команду APPEND и нажмите клавишу Enter

Командное окно программы **FoxPro** (см. рис.1.) предназначено для ввода команд с клавиатуры и последующего их выполнения. Любые действия и операции над элементами приложения в **СУБД FoxPro** могут осуществляться при помощи команд, вводимых в программное окно.

Сотрудники	
ФИО	Довгаль П.У.
Пол	муж
Дата рождения	17.05.1954
Семейное положение	женат
Количество детей	3
Оклад	9000.0000
Должность	инспектор
Место жительства	Гагарина 87 кв 132
Табельный номер	3
ФИО	Прутков И.С.
Пол	муж
Дата рождения	01.09.1971
Семейное положение	женат
Количество детей	1
Оклад	15000.0000
Должность	завотделом
Место жительства	Панина 198 кв 3

Рис.4. Просмотр и редактирование содержимого таблицы в режиме APPEND

10. После ввода данных (см. таблицу 4.) в командном окне программы введите команду BROWSE и нажмите клавишу Enter. Результат должен соответствовать рис.5.

ФИО	Пол	Дата рождения	Семейное положение	Количество детей	Оклад	Должность
Гуськов Г.Г.	муж	22.02.1976	холост	0	7000.0000	экономист
Довгаль П.У.	муж	17.05.1954	женат	3	9000.0000	инспектор
Прутков И.С.	муж	01.09.1971	женат	1	15000.0000	завотделом
Косыгин Н.И.	муж	28.03.1979	холост	0	7000.0000	экономист
Патриков Е.П.	муж	14.08.1980	холост	0	4500.0000	консультант
Белкина Б.Ю.	жен	24.11.1959	замужем	4	12000.0000	бухгалтер
Петренко Е.П.	жен	29.09.1981	незамужем	0	7000.0000	экономист
Петросова Г.Н.	жен	19.06.1976	замужем	2	4500.0000	консультант
Орлова К.Д.	жен	31.12.1977	незамужем	0	7000.0000	экономист
Косыгина А.К.	жен	01.10.1969	замужем	1	6000.0000	менеджер

Рис.5. Просмотр таблицы в режиме BROWSE

Таб. номер	ФИО	Пол	Дата рождения	Семейное положение	Количество детей	Оклад	Должность	Место жительства
1	Гуськов Г.Г.	муж	22.02.1976	холост	0	7000	экономист	Лаптиева 156 кв 43
2	Довгаль П.У.	муж	17.05.1954	женат	3	9000	инспектор	Гагарина 87 кв 132
3	Прутков И.С.	муж	01.09.1971	женат	1	15000	завотделом	Панина 198 кв 3
4	Косыгин Н.И.	муж	28.03.1979	холост	0	7000	экономист	Николаева 30 кв 77
5	Патриков Е.П.	муж	14.08.1980	холост	0	4500	консультант	Нурядилова 18г кв 84
6	Белкина Б.Ю.	жен	24.11.1959	замужем	4	12000	бухгалтер	Луначевского 64 кв 9
7	Петренко	жен	29.09.1981	незамужем	0	7000	экономист	Ленина 101 кв 15

Содержимое таблицы *Книги*2. Таблица *Разделы*

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
razdel	Character	15		Раздел

Определение полей таблицы *Разделы* в окне конструктора таблицы **Table Designer**

Раздел
Экономика
Информатика
Психология
Математика
Техника

Содержимое таблицы *Разделы*

3. Таблица *Издательство*

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
izdat	Character	20		Издательство
gorod	Character	15		Город

Определение полей таблицы *Издательство* в окне конструктора таблицы **Table Designer**

Издательство	Город
Финансы и статистика	Москва
Тригон	Санкт-Петербург
Питер	Санкт-Петербург
Наука	Москва
Машиностроение	Москва
Парадокс	Минск
БХВ-Петербург	Санкт-Петербург

Содержимое таблицы *Издательство*

4. Таблица *Автор книги*

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
kodavknigi	Integer	4		Код автора книги
kodknigi	Integer	4		Код книги
kodavtora	Integer	4		Код автора

Определение полей таблицы *Автор книги* в окне конструктора таблицы **Table Designer**

Код автора книги	Код книги	Код автора
1	1	1
2	1	2
3	2	3
4	3	4
5	4	5

6	5	6
7	6	7
8	7	8
9	8	9
10	9	10
11	10	11

Содержимое таблицы *Автор_книги*

5. Таблица *Авторы*

(наименование поля)	(тип поля)	(ширина поля)	(индексное поле)	(надпись, определяет заголовок поля)
Name	Type	Width	Index	Caption
kodavtora	Integer	4		Код автора
familiya	Character	20		Фамилия
imya	Character	15		Имя

Определение полей таблицы *Авторы* в окне конструктора таблицы **Table Designer**

Код автора	Фамилия	Имя
1	Фролов	Геннадий
2	Илюхин	Виктор
3	Попов	Владимир
4	Акимова	Светлана
5	Морис	Брюс
6	Кулагина	Галина
7	Омельченко	Людмила
8	Алан	Пиз
9	Мальцев	Анатолий
10	Горин	Дмитрий
11	Якокка	Ли

Содержимое таблицы *Авторы*

13. Для закрытия всех элементов приложения в командном окне программы введите команду CLOSE ALL

14. Для завершения работы с программой **Visual FoxPro** в командном окне программы введите команду QUIT.

Контрольные вопросы

1. Какие типы полей допустимы в **Visual FoxPro**
2. Какие команды встроенного языка **Visual FoxPro** вы знаете (перечислить)
3. Как создать проект
4. Что такое база данных в **Visual FoxPro**
5. Как создать базу данных
6. Как создать таблицу
7. Как создать ключевое поле
8. Как изменить структуру таблицы
9. Как вводить данные в таблицу
10. Как просмотреть содержимое таблицы
11. Как редактировать содержимое таблицы
12. Как добавить запись в таблицу
13. Как удалить запись из таблицы

определённую зарплату		
Запрос, показывающий сотрудников, имеющих более одного ребёнка и получающих зарплату менее 9000	Procedure	<pre>OPEN DATABASE \<Имя вашей папки>\штат.dbc DO "<Имя вашей папки>\зарплата дети.qpr" CLOSE DATABASES</pre>

Таблица 4. Наименование пунктов меню и определение их действий (Продолжение)

52. Сохранить созданное меню: пункт меню **File/Save As**, в появившемся диалоговом окне указать имя своей папки, куда следует сохранить файл, присвоить имя *Меню* и нажать кнопку **Save** (Сохранить)

53. Необходимо сгенерировать меню: пункт меню **Menu/Generate**, откроется диалоговое окно **Generate Menu** (Генерация меню), нажать кнопку **Generate** (Генерация)

54. После завершения генерации запустить программу меню на выполнение: в окне проекта **Project Manager** установить курсор на наименование созданного меню *Меню* и нажать кнопку **Run** (Запустить)

Контрольные вопросы

1. Что такое меню приложения
2. Какие типы меню вы знаете (перечислить)
3. Какого назначения кнопок конструктора меню **Menu Designer** (Конструктор меню)
4. Какие типы пунктов меню вы знаете (перечислить)
5. В чём отличие типа **Procedure** от типа **Command** пункта меню
6. Как определить комбинацию клавиш быстрого вызова пунктов меню
7. Как реализовать разделительные линии пунктов меню
8. Как расположить графическое изображение справа от команд пользовательского меню
9. Как создать подменю меню приложения
10. Как определить параметры меню (пункт меню **Visual FoxPro View**, команда **General Options**)
11. Как сгенерировать меню
12. Как запустить меню приложения

14. Как закрыть все приложения в Visual FoxPro

Лабораторная работа №2

Создание отношений между таблицами в многотабличной БД *Библиотека*

Теоретическая часть

Среди требований, предъявляемых к СУБД, основное место занимает возможность быстрого поиска необходимой информации. Средством, позволяющим решить эту проблему, является применение индексов. В **Visual FoxPro** для создания первичных ключей, определяющих отношения между таблицами и условия целостности данных также предназначены индексы. В этом случае индексы должны быть уникальными, то есть значения индексированного поля должны быть неповторяющимися (уникальными).

Для создания индекса таблицы используется вкладка **Indexes** (Индексы) окна конструктора таблиц **Table Designer**. Все индексы в **Visual FoxPro** имеют имена, задаваемые в поле **Name** (Имя). Для задания типа создаваемого индекса используется список **Type** (Тип).

Тип индекса	Описание
Regular (Обычный)	Создаётся индекс, в котором для каждой записи таблицы хранится значение индексного выражения. Если несколько записей имеют одинаковое значение индексного выражения, то каждое значение хранится отдельно и содержит ссылку на связанную с ней запись
Unique (Уникальный)	Создаётся индекс, в котором хранятся только неповторяющиеся значения индексного выражения. Если две или более записей содержат одинаковое значение индексного выражения, то будет храниться только одно значение и ссылка на первую из записей с одинаковым значением индексного выражения. Таблица может иметь несколько уникальных индексов
Candidate (Кандидат)	Создаётся уникальный индекс, который не содержит полей с пустыми значениями. Этот индекс обладает всеми качествами первичного ключа и не является им только по той причине, что таблица не может содержать более одного первичного ключа
Primary (Первичный)	Создаётся уникальный индекс, который используется для связывания таблиц и определения условий целостности данных. Поля, входящие в первичный ключ, не должны допускать ввода пустых значений. В отличие от уникального индекса, таблица может иметь только один первичный ключ

Таблица 1. Описание типов индекса

Обычно, в **Visual FoxPro** при создании форм, отчётов и запросов используются несколько таблиц, между которыми установлены постоянные отношения. Такие таблицы называются связанными. Из двух связанных таблиц одна является главной (родительской), а другая подчинённой (дочерней). При создании индексов для родительской таблицы должен быть определён ключ типа **Primary** (Первичный) или типа **Candidate** (Кандидат), а для дочерней таблицы – индекс для связи с родительской таблицей типа **Regular** (Обычный).

Типы отношений	Описание
Отношение “один-к-одному”	Каждая запись в одной таблице соответствует только одной записи в другой таблице
Отношение “один-ко-многим”	Наиболее распространённый тип отношений, каждой записи в одной таблице может соответствовать несколько записей в другой таблице
Отношение “много-к-одному”	Отношение “много-к-одному” можно сравнить с

	отношением “один-ко-многим”, рассматриваемое с другой точки зрения
Отношение “мноغو-ко-многим”	Нескольким записям одной таблицы может соответствовать несколько записей в другой таблице

Таблица 2. Типы отношений между таблицами

Одним из самых важных требований, предъявляемых к базам данных, является целостность данных, которую определяют установленные между таблицами отношения. Для определения целостности данных в **Visual FoxPro** используется окно построителя условий целостности данных **Referential Integrity Builder** (Построитель целостности данных), которое содержит перечень всех установленных отношений между таблицами (см. рис.3.).

Наименование опции	Описание
Cascade (Каскадное изменение)	При изменении значений полей первичного ключа или ключа-кандидата в родительской таблице автоматически осуществляется каскадное изменение всех соответствующих значений в дочерней таблице
Restrict (Запрет изменения)	Не позволяет изменять значения полей первичного ключа или ключа-кандидата в родительской таблице, если в дочерней таблице имеется хотя бы одна запись, содержащая ссылку на изменяемую запись
Ignore (Игнорировать)	Позволяет изменять значения полей первичного ключа или ключа-кандидата в родительской таблице независимо от существования связанных записей в дочерней таблице. Целостность данных при этом не поддерживается

Таблица 3. Описание действий **Visual FoxPro**, в зависимости от выбранной опции, при изменении значения первичного ключа или ключа типа **Candidate**

Наименование опции	Описание
Cascade (Каскад)	При удалении записи из родительской таблицы автоматически осуществляется каскадное удаление всех записей из дочерней таблицы, связанных с удаляемой записью
Restrict (Запрет)	Не позволяет удалять записи в родительской таблице, если в дочерней таблице имеется хотя бы одна запись, содержащая ссылку на удаляемую запись. При попытке удаления записи возникает ошибка, которую вы можете обработать программно
Ignore (Игнорировать)	Позволяет удалять записи в родительской таблице независимо от существования связанных записей в дочерней таблице. Целостность данных при этом не поддерживается

Таблица 4. Действия **Visual FoxPro**, в зависимости от выбранной опции, при удалении записи из родительской таблицы

Наименование опции	Описание
Restrict (Запрет)	Не позволяет вводить запись, если значение индексного выражения дочерней таблицы не соответствует одной из записей в родительской таблице
Ignore (Игнорировать)	При вводе данных в дочернюю таблицу не анализируется значение индексного выражения. Целостность данных при этом не поддерживается

Таблица 5. Действия **Visual FoxPro**, в зависимости от выбранной опции, при добавлении новой записи в родительскую таблицу

Цель работы

1. Создать первичный ключ для таблицы, создать индексы дочерних таблиц для связи с родительской таблицей
2. Создать отношения между таблицами
3. Определить условия целостности данных в базе данных проекта

Практическая часть

1. Запустить программу **Microsoft Visual FoxPro**
2. Открыть созданный проект: **File/Open/Информационная система**
3. Открыть БД *итат*: в окне **Project Manager** (Менеджер проекта) вкладка **Data/Databases/Библиотека/Open**
4. Необходимо модифицировать таблицы БД *Библиотека*, создать индексированные поля, а также первичные ключи для осуществления связей между таблицами. Для модификации (изменения структуры) таблицы в окне проекта **Project Manager** (Менеджер проекта) установить курсор на модифицируемую таблицу *Книги* и нажать кнопку **Modify** (Модифицировать) или в окне конструктора БД **Database Designer** (Конструктор базы данных) установить курсор в таблицу *Книги*, вызвать контекстное меню и выбрать команду **Modify** (Модифицировать). На экране откроется диалоговое окно **Table Designer** (Конструктор таблицы)
5. В окне **Table Designer** (Конструктор таблицы) перейти на вкладку **Indexes** (Индексы) (см. рис.1.) и ввести значения в соответствии с таблицей 6., нажать кнопку **ОК**, система попросит подтвердить сохранение изменений, нажать **Yes**

Order	Name	Type	Expression	Filter	Collate
↑	kodknigi	Primary	kodknigi		Machine
↑	razdel	Regular	razdel		Machine
↑	izdat	Regular	izdat		Machine

Таблица 6. Определение индексов таблицы *Книги* на вкладке **Indexes** (Индексы) окна конструктора таблицы **Table Designer**

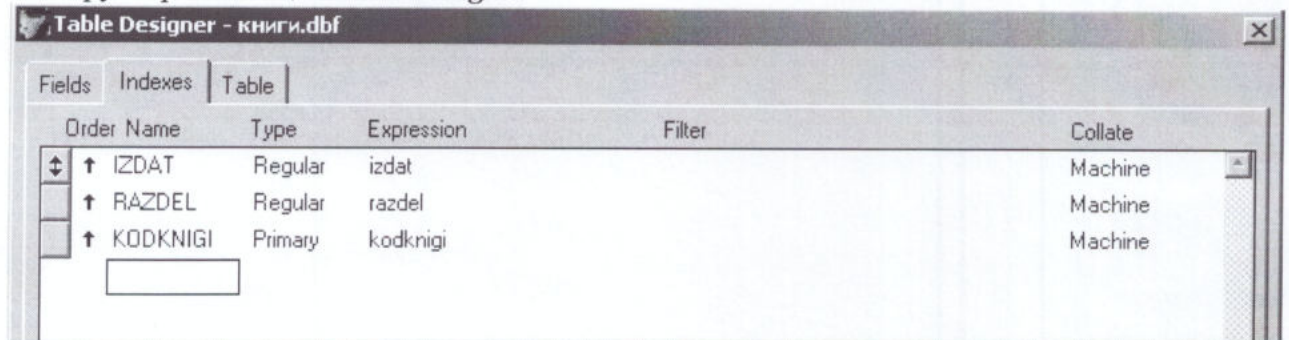


Рис.1. Вкладка **Indexes** (Индексы) окна **Table Designer** таблицы *Книги*

Индекс, с присвоенным типом **Primary** является первичным ключом таблицы

6. Аналогично внести изменения в таблицы БД *Библиотека*: *Разделы*, *Издательство*, *Автор_книги*, *Авторы* в соответствии с таблицами 7, 8, 9, 10

Order	Name	Type	Expression	Filter	Collate
↑	razdel	Primary	razdel		Machine

Таблица 7. Определение индексов таблицы *Разделы* на вкладке **Indexes** (Индексы) окна конструктора таблицы **Table Designer**

Order	Name	Type	Expression	Filter	Collate
-------	------	------	------------	--------	---------

↑	izdat	Primary	izdat		Machine
---	-------	---------	-------	--	---------

Таблица 8. Определение индексов таблицы *Издательство* на вкладке **Indexes** (Индексы) окна конструктора таблицы **Table Designer**

Order	Name	Type	Expression	Filter	Collate
↑	kodavknigi	Primary	kodavknigi		Machine

Таблица 9. Определение индексов таблицы *Автор_книги* на вкладке **Indexes** (Индексы) окна конструктора таблицы **Table Designer**

Order	Name	Type	Expression	Filter	Collate
↑	kodavtora	Primary	kodavtora		Machine

Таблица 10. Определение индексов таблицы *Авторы* на вкладке **Indexes** (Индексы) окна конструктора таблицы **Table Designer**

7. В окне **Database Designer** (Конструктор базы данных) выбрать родительскую таблицу *Разделы*. Таблицы в конструкторе БД обозначаются прямоугольниками, в нижней части которых после надписи **Indexes** (Индексы) расположен список индексов, созданных для данной таблицы. Первичный ключ в этом списке выделяется значком ключа, расположенным с левой стороны от наименования индекса (см. рис.2.). Установить курсор мыши на первичный ключ **razdel** таблицы *Разделы*. Нажать кнопку мыши и, не отпуская ее, переместить курсор мыши на индекс **razdel** дочерней таблицы *Книги*. Отпустить кнопку мыши. Созданные отношения между таблицами отображаются в виде линий (см. рис.2.). Аналогичным образом, создать связь между таблицами *Издательство* и *Книги* (Таблица *Издательство* – родительская, таблица *Книги* – дочерняя, первичный ключ – **izdat**)

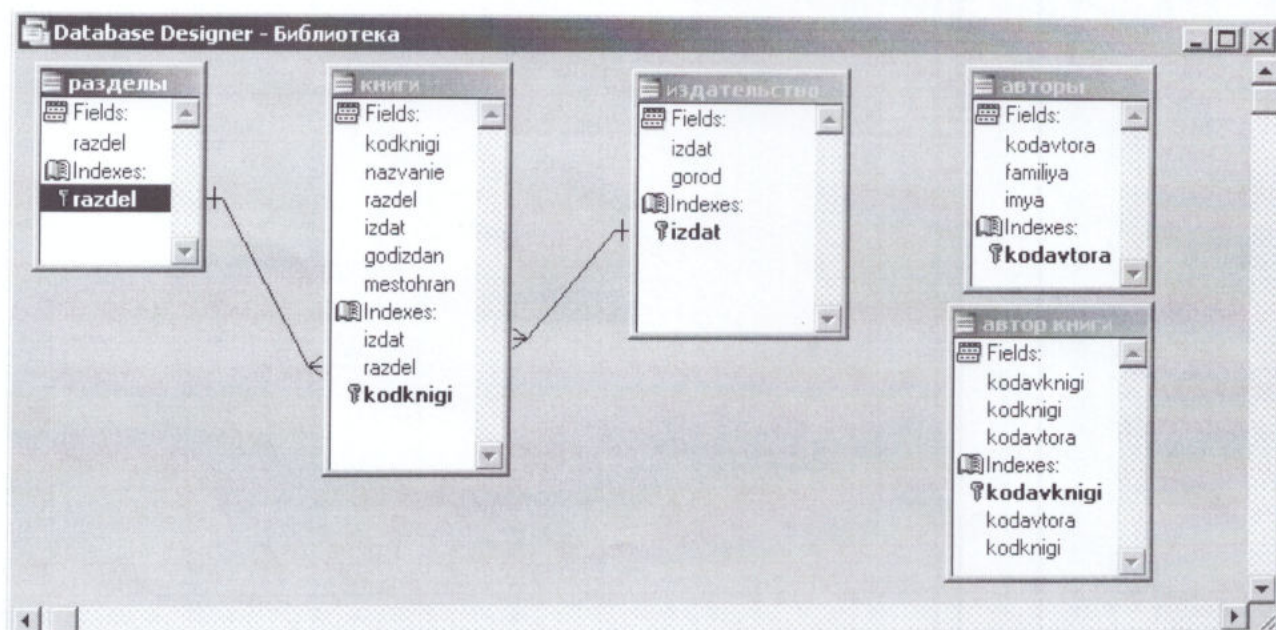


Рис.2. Отображение связей между таблицами в окне **Database Designer** (Конструктор базы данных)

8. Для редактирования отношений между связанными таблицами дважды щелкнуть левой кнопкой мыши на линии, появится диалоговое окно **Edit Relationship** (Редактирование отношения), в котором слева приведено наименование родительской таблицы и расположен раскрывающийся список индексов таблицы, справа размещена аналогичная информация о дочерней таблице. В этом диалоговом окне указан также тип установленного отношения между таблицами.

9. В окне **Edit Relationship** (Редактирование отношения) нажать кнопку **Referential Integrity** (Целостность данных), в появившемся диалоговом окне **Referential Integrity**

Builder (Построитель целостности данных) выбрать отношение **издательство-книги**. В полях **Update** (Изменить), **Delete** (Удалить), **Insert** (Заменить) установить тип действий **Restrict** (Запрет изменения) (см. таблица 3, 4, 5). Провести аналогичные действия для отношения **разделы-книги**. Результат описанных действий, которые необходимы для обеспечения целостности данных, представлен на рис.3.

10. Для сохранения выполненных действий нажать кнопку **OK**, система потребует подтверждение сохранения, нажать кнопку **Да**

11. Нажать кнопку **OK** для закрытия диалогового окна **Edit Relationship**

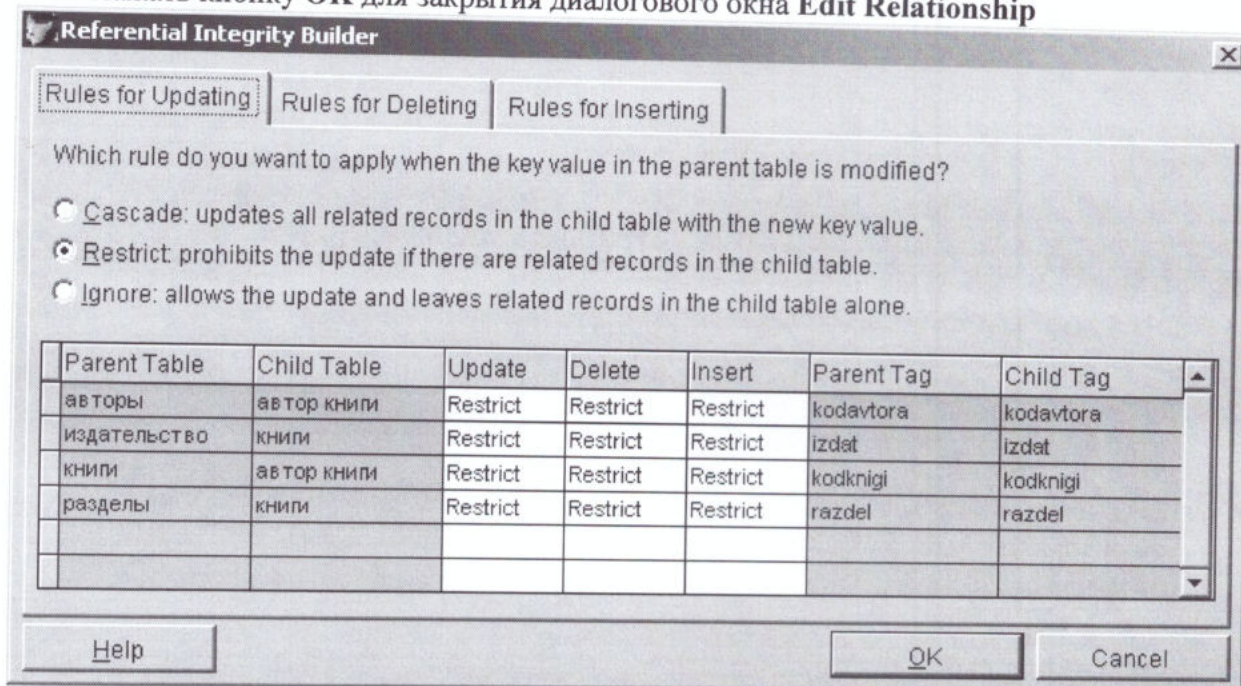


Рис.3. Диалоговое окно **Referential Integrity Builder** (Построитель целостности данных)

Задание

1. Создать связь между таблицами *Книги* и *Автор_книги*
2. Создать связь между таблицами *Авторы* и *Автор_книги*
3. Установить целостность данных в созданных отношениях: в полях **Update** (Изменить), **Delete** (Удалить), **Insert** (Заменить) установить тип действий **Restrict** (Запрет изменения)

Контрольные вопросы

1. Что такое индекс, типы индексов (перечислить)
2. Что такое первичный ключ
3. Какой тип индекса присваивается первичному ключу
4. Как создать индекс
5. Какие таблицы называются связанными
6. Какие типы отношений между таблицами вы знаете (перечислить)
7. Как создать отношение между таблицами в **Visual FoxPro**
8. Как редактировать отношения между таблицами в **Visual FoxPro**
9. Какие условия целостности данных в **Visual FoxPro** вы знаете
10. Как определить условие целостности данных
11. Как удалить созданное отношение между таблицами

Список используемой литературы

а) Основная литература

1. Попов А.А. Программирование в среде СУБД FoxPro 2.0 Построение систем обработки данных. –М.: Издательство “МАРТ”, 1996. -352 с.
2. Омельченко Л.Н. Самоучитель Visual FoxPro 8. –СПб.: БХВ-Петербург, 2003. -688 с.
3. Дейт К. Руководство по реляционной СУБД DB2. –М.: Финансы и статистика, 1988. -320 с.
4. Берещанский Д.Г. Практическое программирование на dBASE. –М.: Финансы и статистика, 1989. -192 с.
5. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. –М.: Мир, 1989. -360 с.
6. Карпова Б. Базы данных: модели, разработка, реализация. –СПб.: Питер, 2001.
7. Клементьев В.А. Знакомство с Visual FoxPro. –М.: 2002.
8. Каев А. Справка по языку FoxPro. –М.: 2002.

б) Дополнительная литература

9. Давыдова Е.М., Новгородова Н.А. Базы данных. – Томск: ТУСУР, 2007.
10. Дейт К. Руководство по реализации СУБД DB2. – М.: 1988.
11. Клепинин В.Б., Агафонова Т.П. Visual FoxPro 9.0. – СПб, 2008.
12. Кренке Д. Теория и практика построения баз данных. - М.: Питер, 2010, 858 с.
13. Ломтадзе В.В., Шишкина Л.П. Системы управления базами данных. Учебное пособие / - Иркутск: ИрГТУ, 2009. - 116 с.
14. Мейер М.М. Теория реляционных баз данных. Уч. пособие / - М.: Мир, 2011, 610 с.
15. Омельченко Л.Н., Шевякова Д.А. Visual FoxPro 9.0. – СПб: 2005.
16. Попов А.А. FoxPro 2.5/2.6, М, «ДЕСС КОМ», 2001, 660 с.
17. Пинтер Лес. Разработка приложения в Microsoft FoxPro 2.5. – М.: 1995.

в) Интернет-ресурсы

1. <http://www.bestreferat.ru>
2. <http://www.mysql.com>
3. <http://www.ord.com.ru>
4. <http://www.metod-kopilka.ru/page-test-8-6-2.html>
5. <http://www.5ballov.ru/> (07.05.2012).
6. <http://inf-fiz-mat.59311s023.edusite.ru/p14aa1.html>
7. <http://www.klyaksa.net/htm/uchitel/test7.htm>
8. <http://gplinform.ucoz.ru/tests>
9. <http://nsportal.ru/shkola/informatika-i-ikt/library/testy-po-teme-bazy-dannykh>

Лабораторная работа №3

Создание простейшей однотабличной формы в среде **Visual FoxPro**.

Теоретическая часть

В **Visual FoxPro** для просмотра, ввода и редактирования данных, хранящихся в таблицах, используются формы, являющиеся более наглядным средством представления информации. Важным преимуществом форм является, то, что они позволяют работать не с одной, а с несколькими связанными таблицами, что, в свою очередь, также увеличивает наглядность.

При создании форм в **Visual FoxPro** разработчик может использовать следующие средства:

- **Form Wizard** (мастер форм)
- **Form Builder** (построитель формы)
- **Builder** (построитель объектов формы)
- **Auto Format Builder** (построитель автоформата)
- **Form Designer** (конструктор форм).

Чтобы создать форму для одной или связанных таблиц с возможностью задания отображаемых в форме полей, стиля их отображения и указания типа кнопок управления, можно использовать **Form Wizard** (мастер создания форм).

Для самостоятельной разработки формы с заданными свойствами или изменения формы, созданной с помощью мастера, вам необходимо использовать **Form Designer** (конструктор форм).

Для облегчения размещения в конструкторе форм полей и надписей, оформленных в соответствии с выбранным стилем, можно использовать **Form Builder** (Построитель формы).

Цель работы

1. Создать форму *Сотрудники* с помощью **Form Wizard** (мастер форм)
2. Запустить форму *Сотрудники*;
3. Редактировать данные в таблице *Сотрудники* с помощью созданной формы *Сотрудники*.

Практическая часть

15. Запустить программу **Microsoft Visual FoxPro**
16. Открыть созданный проект: **File/Open/Информационная система**
17. Открыть БД *umat*: в окне **Project Manager** (Менеджер проекта) вкладка **Data/Databases/umat/Open**
18. Создать форму *Сотрудники*, щелкнув клавишей мыши на вкладке **Documents/Forms/New/Form Wizard/Form Wizard**, нажать клавишу **Ok**

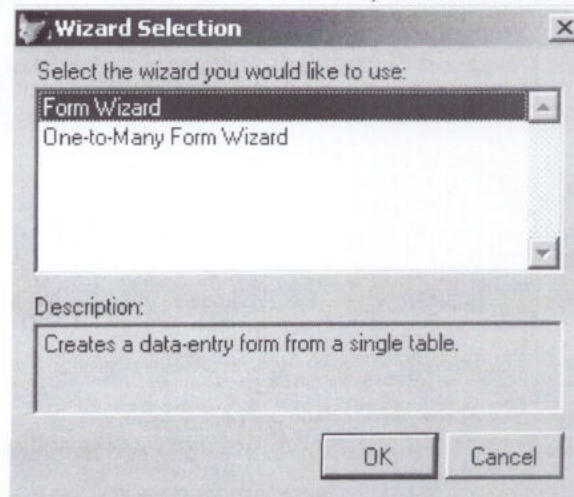


Рис.1. Диалоговое окно Wizard Selection

19. В появившемся окне **Form Wizard** в области **Databases and tables** (Базы данных и таблицы) выбрать необходимую БД *штат* и указать таблицу *Сотрудники*, для которой создается форма. Из области **Available fields** (Имеющиеся поля) выбрать поля, которые будут размещены в форме, в соответствии с рис.2, используя кнопки расположенные между списками для перехода к следующему шагу нажать кнопку **Next** (Далее)

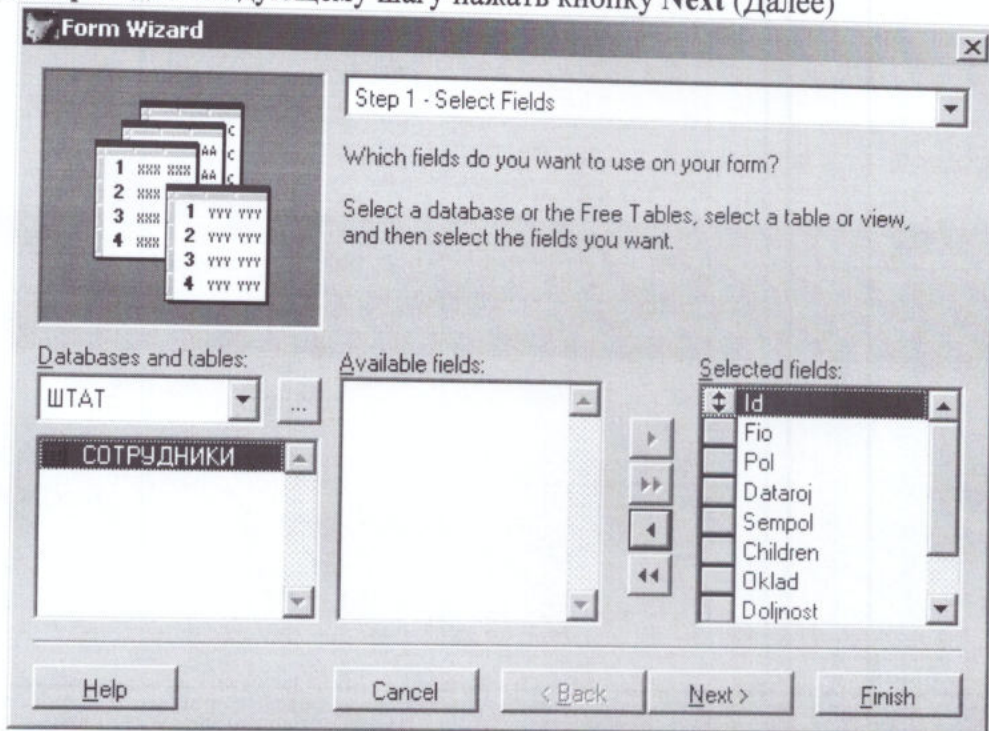


Рис.2. Диалоговое окно выбора полей для отображения.

20. В появившемся диалоговом окне в области **Style** установить стиль отображения объектов и в области **Button type** типы кнопок управления в соответствии с рис.3, нажать кнопку **Next**

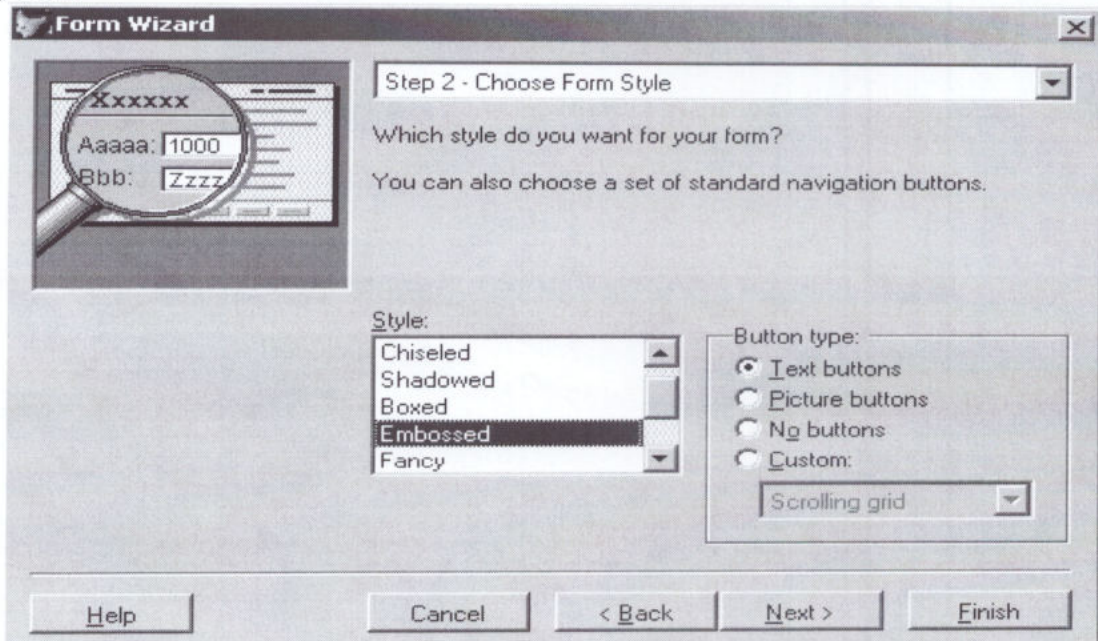


Рис.3. Окно выбора стиля отображения полей и управляющих кнопок.

21. В появившемся диалоговом окне задать критерий сортировки данных, указав поля по которым будет осуществляться упорядочивание в соответствии рис.4. нажать кнопку **Next**

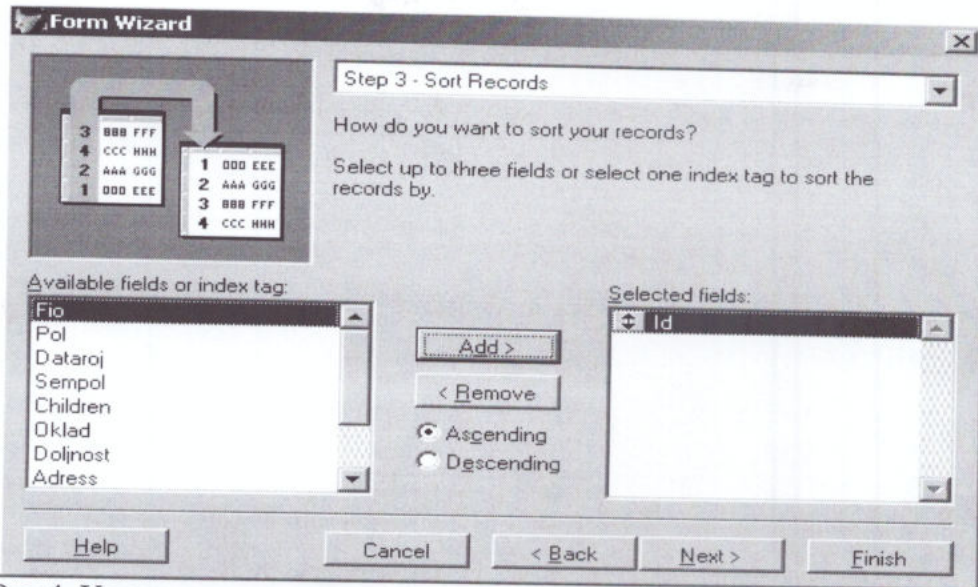


Рис.4. Установка критерия упорядочения данных

22. На заключительном шаге создания форм в области **Type a title for your form** (Тип заголовка формы) указать имя формы *Сотрудники*. Воспользовавшись кнопкой **Preview** (Просмотр) просмотрите, как будет выглядеть создаваемая форма, после просмотра нажмите кнопку **Return to Wizard**

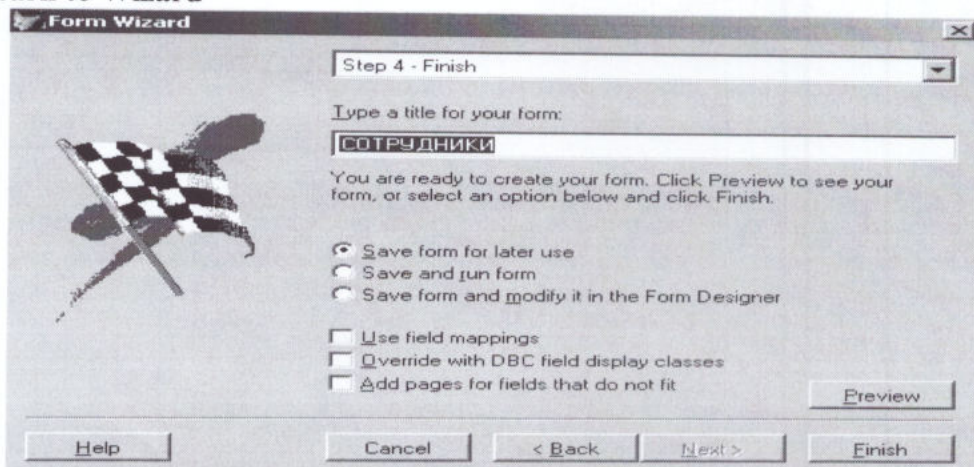


Рис.5. Окно задания заголовка формы.

Если что-то не так, вернитесь к предыдущим шагам, воспользовавшись кнопкой **Back**. Нажать кнопку **Finish** (Готово) и в появившемся диалоговом окне **Save As** (Сохранить как) указать папку, в которой будет размещена форма *Сотрудники*

23. Запустить в окне **Project Manager** форму *Сотрудники*: **Documents/Forms/Сотрудники/Run**

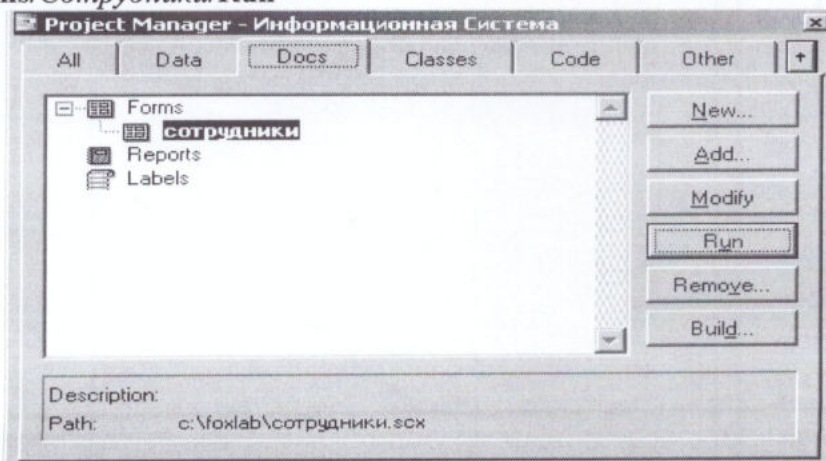


Рис.6. Окно Project Manager

Или с помощью меню программы **Visual FoxPro: Program/Do** задать имя формы *Сотрудники* и тип файла **Form**, нажать кнопку **Do**. С помощью кнопок (см. таблицу 1.), находящихся в нижней части формы можно вводить или редактировать записи в таблице сотрудники

Рис.7. Форма *Сотрудники*

24. Ввести в таблицу *Сотрудники* дополнительно шесть записей, одну удалить, две отредактировать

Top	Prev	Next	Bottom	Find	Save
Первая запись	Предыдущая запись	Следующая запись	Последняя запись	Найти	Сохранить
Print	Add	Edit	Delete	Exit	Revent
Вывести на печать	Добавить запись	Редактировать запись	Удалить запись	Выход	Отменить

Таблица 1. Кнопки управления и редактирования данных

Задание

Создать форму для таблицы *Книги* БД *Библиотека*

1. В диалоговом окне выбора полей для отображения (см. рис.2.) выбрать все поля таблицы *Книги*
2. В окне выбора стиля отображения полей и управляющих кнопок (см. рис.3.) в области **Style** (стиль) выбрать стиль **Embossed**, а в области **Button type** (Типы кнопок управления) выбрать пункт **No Buttons**
3. Установить критерий упорядочения данных по полю **кодкниги** (см. рис.4.)
4. В окне задания заголовка формы в области **Type a title for your form** (Тип заголовка формы) указать имя формы *Книги*. Просмотреть вид создаваемой формы нажатием кнопки **Preview** (Просмотр).
5. После сохранения запустить форму *Книги*, она должна иметь вид как на рис.8.

КНИГИ

Код книги: 1

Название: Практический курс программирования

Раздел: Информатика

Издательство: Наука

Год_издания: 1983

Место_хранения: 6-11

Рис.8. Форма *Книги***Контрольные вопросы**

1. Что такое форма
2. Какие средства можно использовать для создания форм в **Visual FoxPro**
3. Как создать форму с помощью **Form Wizard** (Мастер форм)
4. Как выбрать поля, которые будут отображены в форме с помощью **Form Wizard** (Мастер форм)
5. Как выбрать стиль отображения полей и управляющих кнопок в диалоговом окне **Form Wizard**
6. Как установить критерий упорядочения данных в диалоговом окне **Form Wizard**
7. Как предварительно посмотреть внешний вид создаваемой формы в диалоговом окне **Form Wizard**
8. Как запустить форму
9. Как вводить и редактировать данные с помощью формы

Лабораторная работа №4

Создание формы с помощью конструктора форм

Теоретическая часть

Любая форма в **Visual FoxPro** состоит из объектов, каждый из которых имеет характерные свойства. Для любого объекта вы можете указать действия, выполняемые написанной разработчиком программой при наступлении определённых событий. Процесс создания формы в конструкторе форм состоит в размещении в форме объектов и определении свойств, а также связанных с ними событий и выполняемых действий.

Процесс создания формы включает следующие действия:

- настройка параметров формы
- определение среды окружения, то есть выбор используемых в форме таблиц и установка связей между ними
- размещение в форме объектов: текста, полей различных типов, линий, рисунков, кнопок управления.

Цель работы

1. Настройка параметров формы
2. Ознакомление с объектами формы
3. Размещение объектов в форме с помощью **Form Builder** (Построитель форм)

Практическая часть

1. Запустить программу **Microsoft Visual FoxPro**
 2. Открыть проект *Информационная система*
 3. Открыть БД *штат*
 4. Выбрать в окне **Project Manager** вкладку **Documents/Forms/New/New Form**.
Откроется окно **Form Designer** (Конструктор форм) для создания новой формы
 5. Открыть окно окружения формы **Data Environment** (Среда окружения): пункт меню **View/ Data Environment**. Для размещения таблицы в среде окружения выбрать команду **Add** (Добавить) из меню **Data Environment** (Среда окружения) или вызвать контекстное меню на окне окружения формы **Data Environment** и выбрать пункт **Add**. В появившемся диалоговом окне **Add Table or View** (Добавить таблицу или представление данных), выбрать из списка таблиц открытой базы *штат* таблицу *Сотрудники*, для которой создаётся форма, и нажать кнопку **OK**
 6. Открыть окно свойств таблицы, размещённой в окне окружения **Data Environment**. Для этого установить на таблицу *Сотрудники* курсор, вызвать контекстное меню правой клавишей мыши и выбрать команду **Properties** (Свойства)
 7. В окне **Properties** (свойства) выделить свойство **Order** (Упорядочение). Для упорядочения данных в форме в поле коррекции свойства нажмите кнопку раскрытия списка и из списка индексов таблицы выберите индекс (id), по которому хотите упорядочить данные.
 8. Закройте окно определения среды окружения **Data Environment**.
 9. Для задания свойств формы установить курсор в форму, вызвать контекстное меню, выбрать команду **Properties** (Свойства). Откроется окно **Properties** (Свойства). В его верхнем списке, указывающем название объекта, для которого осуществляется настройка свойств содержится текст **Form1** (Формal)
 10. В окне **Properties** (Свойства) скорректировать свойство **Caption** (Надпись), введя в текстовое поле заголовков формы *Сотрудники тест*
 11. Свойство формы **AutoCenter** (Автоцентр) должно иметь значение **True** (Истина), чтобы форма располагалась в центре экрана
 12. Изменить свойства **FontName** (Наименование шрифта), указав шрифт **Times New Roman** и **FontSize** (Размер шрифта) указав размер шрифта **12**
- После того как вы определили параметры формы, разместили в окружении используемые таблицы, можно приступить к размещению объектов в форме. Осуществим размещение полей таблицы *Сотрудники* и надписей к ним, используя **Form Builder** (построитель формы)
13. Для запуска построителя форм установить курсор в форму, вызвать контекстное меню, выбрать команду **Builder** (Построитель), откроется диалоговое окно **Form Builder** (построитель формы), содержащее две вкладки: **Field Selection** (Выбор поля) и **Styles** (Стиль). В вкладке **Field Selection** (Выбор поля) из области **Databases and tables** (Базы данных и таблицы) выбрать БД *штат* и таблицу *Сотрудники*, затем из списка **Available fields** (Имеющиеся поля) перенести в **Selected fields** (Выбранные поля) все необходимые поля используя кнопки расположенные между списками. Перейти на вкладку **Styles** (Стиль), выбрать из списка стилей стиль **Embossed** и нажать кнопку **OK**. На форме будут размещены объекты формы (см. рис. 1.).
 14. Для размещения на создаваемой форме кнопок управления воспользуемся ранее созданной формой *Сотрудники*. Открыть форму *Сотрудники* в режиме редактирования: окно программы **Project Manager/Documents/Forms/сотрудники/Modify**. В появившемся окне **Form Designer** формы *Сотрудники* нажатием левой клавиши мыши выделить объект **BUTTONSET1** (блок кнопок управления, см. рис.2.) и нажать комбинацию клавиш **Ctrl+C** на клавиатуре или установить курсор в форму, вызвать контекстное меню, выбрать команду **Copy**. Закройте окно **Form Designer** формы *Сотрудники*.

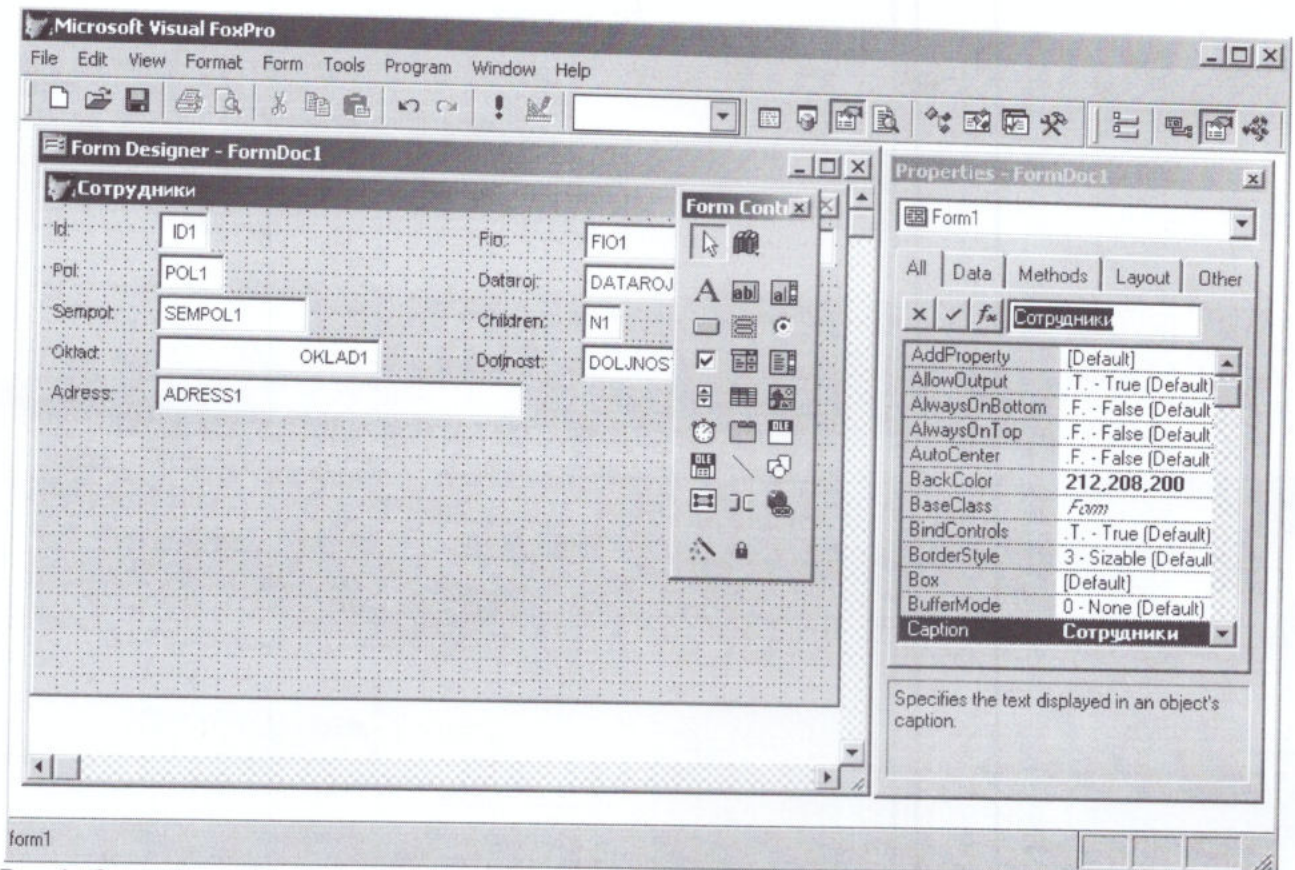


Рис.1. Окно **Form Designer** (Конструктор форм)

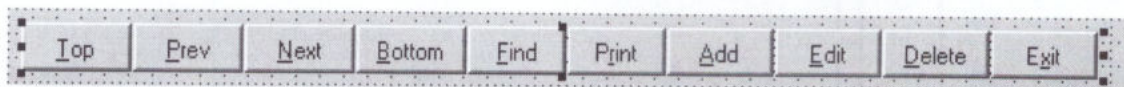


Рис.2. Объект - блок кнопок управления

15. В окне **Form Designer** формы *Сотрудники тест* установить курсор в форму, вызвать контекстное меню, выбрать команду **Paste** или нажать комбинацию клавиш **Ctrl+V**. Блок кнопок управления записями таблицы (см. рис.2.) в создаваемой форме будет скопирован.

16. Задать цвет фона формы. Выделить в окне **Properties** (Свойства) свойство формы **BackColor** (цвет фона), нажать расположенную с правой стороны поля редактирования свойства кнопку и в открывшемся диалоговом окне **Цвет** выберите цвет, который вы хотите использовать для фона

17. Расположить объекты формы в соответствии с рис.3.. Для перемещения объектов формы можно использовать метод перетаскивания мышью или кнопками управления курсором на клавиатуре, предварительно выделив необходимый объект, а также комбинацией клавиш **Ctrl+[кнопки управления курсором]**, использовать все выше перечисленные способы, выяснить их различия. Для изменения размеров формы или объектов формы необходимо выделить объект (форму), подвести курсор к углу объекта (формы), когда курсор примет вид разнонаправленной стрелки нажать левую клавишу мыши и удерживая её, изменить размеры объекта (формы)

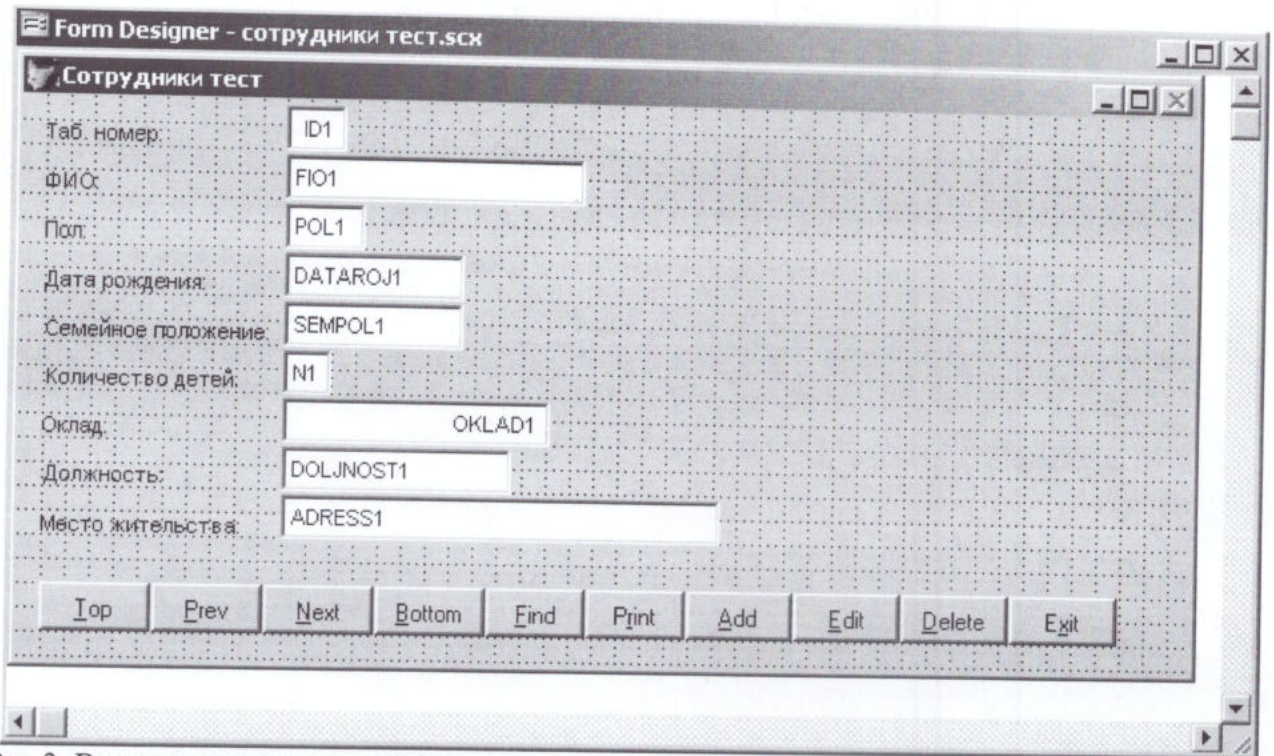



Рис.3. Вид отредактированной формы

18. Запустить отредактированную форму: установить курсор в форму, вызвать контекстное меню, выбрать команду **Run Form** или нажать кнопку [!] на панели инструментов. Система попросит сохранить созданную форму, нажать кнопку **Yes**, в появившемся окне указать папку и имя файла (*Сотрудники тест*), сохранить.

Задание

Создать кнопки навигации в форме *Книги*

1. Открыть форму *Книги* в окне **Form Designer** (Конструктор форм)
2. Нажать кнопку **Command Group** (Группа кнопок)  на панели инструментов **Form Controls** (элементы управления формы) и щёлкнуть в месте их предполагаемого размещения в форме
3. В Окне **Properties** (свойства) размещённого объекта выделить свойство **ButtonCount** (Количество кнопок) и указать необходимое количество размещаемых в форме кнопок, указав число **5**
4. Увеличить с помощью мыши размеры рамки, окружающей данный объект. Перейти в режим редактирования: установить на объект курсор, вызвать контекстное меню и выбрать команду **Edit** (Редактировать). Выделяя поочерёдно кнопки, переместить их, расположив горизонтально, в одну линию
5. Выйти из режима редактирования, щёлкнув вне области объекта **Command Group** (Группа кнопок)
6. Скорректировать размер рамки, окружающей объект: в свойстве **AutoSize** (Авторазмер) установить значение **True** (Истина)
7. Открыть окно свойств объекта **Command Group** (Группа кнопок), нажать кнопку раскрытия списка в верхней части данного окна, поочерёдно выбрать из списка элементы **Command1**, **Command2**, **Command3**, **Command4**, **Command5** и, используя свойство **Caption** (Надпись) задать названия кнопок соответственно: **Первая**, **Следующая**, **Предыдущая**, **Последняя**, **Выход** (см. рис.4.)

КНИГИ

Код книги: 1

Название: Практический курс программирования

Раздел: Информатика

Издательство: Наука

Год_издания: 1983

Место_хранения: 6-11

Первая Следующая Предыдущая Последняя Выход

Рис.4. Форма Книги

8. Определить команды, которые будут выполняться при нажатии на созданные кнопки. Установить на объект Command Group (Группа кнопок) курсор, вызвать контекстное меню и выбрать команду **Edit** (Редактировать), выделить кнопку **Первая**, вызвать контекстное меню и выбрать команду **Code** (Код программы), в появившемся окне ввести следующие команды:

Лабораторная работа №5

Создание и редактирование табличного отчёта в среде Visual FoxPro.

Теоретическая часть

Отчет – форматированное представление данных, выводимое на экран, принтер или в файл. Отчет, создаваемый в Visual FoxPro, может быть представлен в табличном виде или в свободной форме. Табличные отчеты используются для печати данных, представленных в виде списка. При создании отчета в Visual FoxPro разработчик может использовать следующие средства:

- **Report Wizard** – Мастер отчета. Позволяет быстро создать отчет, применяя сортировку, группировку данных и заданный разработчиком стиль оформления;
- **Report Designer** – Конструктор отчета. Позволяет разрабатывать собственные отчеты или модифицировать отчеты, созданные с помощью мастера;
- **Quick Report** – Быстрый отчет. Данное средство предназначено для размещения в конструкторе отчёта полей и задания среды окружения.

Цель работы

1. Создать однотобличный отчет *Сотрудники* с помощью **Report Wizard** (Мастер отчёта)
2. Отредактировать отчет в окне конструктора отчетов **Report Designer** (конструктор отчёта)

Практическая часть

1. Запустить программу **Microsoft Visual FoxPro**

2. Открыть проект *Информационная система*
3. Открыть БД проекта. Для этого необходимо на вкладке **Data** (Данные) установить курсор на её название и нажать кнопку **Open** (Открыть) окна проекта. При этом на стандартной панели инструментов в списке **Databases** (Базы данных) появится название открытой БД
4. Перейти на вкладку **Documents/Reports/New/New Report /Report Wizard**
5. В появившемся диалоговом окне **Wizard Selection** указать тип создаваемого отчета **Report Wizard** (Мастер отчетов) и нажать кнопку **OK**
6. Появляется первое диалоговое окно мастера отчёта (см. рис.1.). Выбрать из верхнего списка области **Databases and tables** (Базы данных и таблицы) базу данных *штат*, а из нижнего – таблицу *сотрудники*

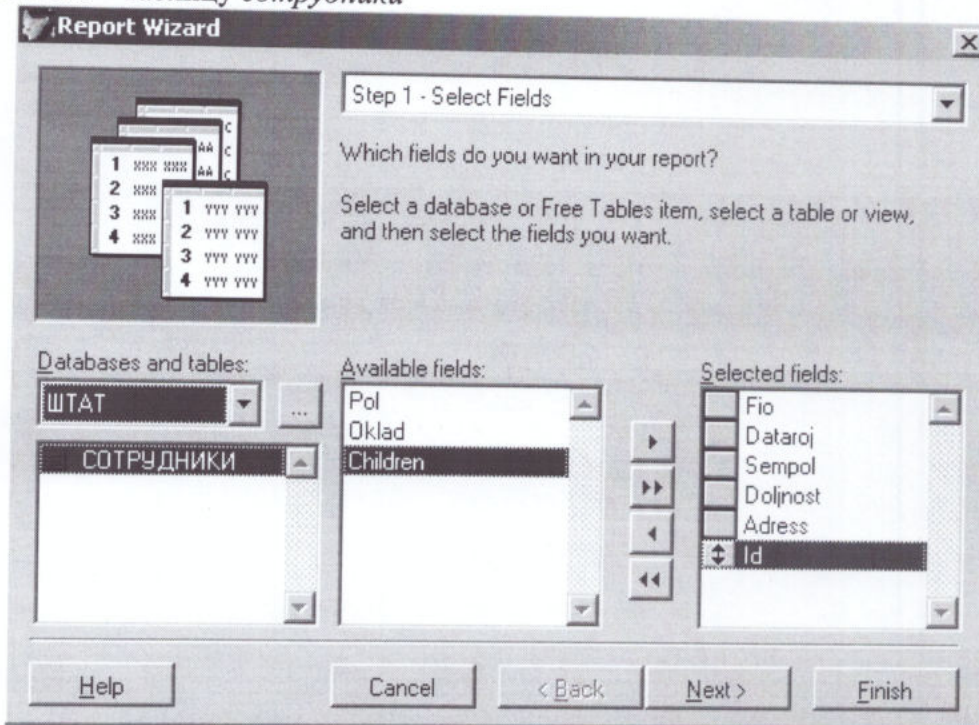


Рис.1. Диалоговое окно создания однотабличного отчета с помощью мастера.

7. Перенести из списка **Available fields** (Имеющиеся поля) в список **Selected fields** (Выбранные поля) поля, которые вы хотите разместить в создаваемом отчете (в соответствии с рис.1.). После формирования списка отображаемых в отчете полей нажать кнопку **Next** (Далее) для перехода к следующему шагу в создании отчета

8. В следующем окне мастера создания отчета указать поля, по которым будут осуществляться группировка данных в отчете (рис.2.)

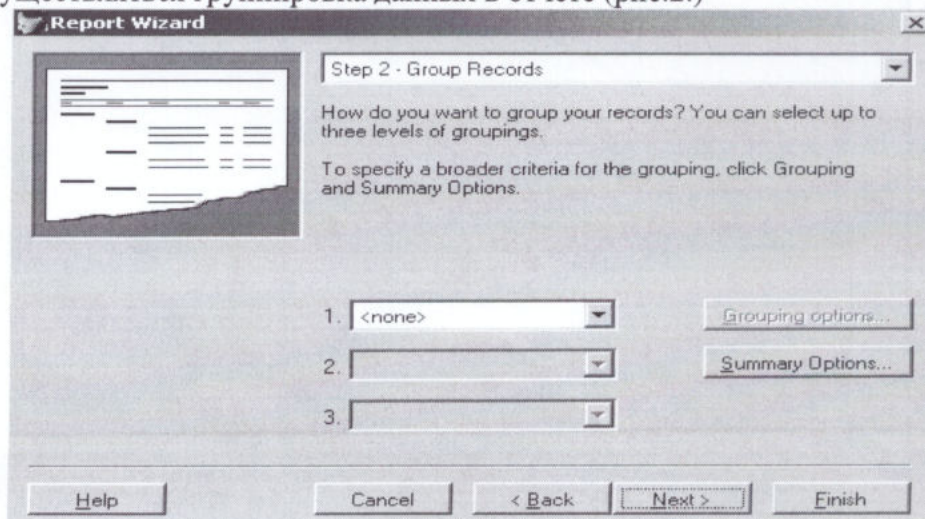


Рис.2. Диалоговое окно определения полей для группировки данных в отчете.

В центре диалогового окна расположены три раскрывающихся списка, позволяющих задать до трех группировок данных в отчете. Эти списки содержат все поля таблицы. Для осуществления группировки данных в отчете выбрать нужное поле из раскрывающегося списка 1. При создании второй и третьей группировки используются, соответственно, списки 2 и 3. В нашем примере группировка данных по полям не осуществляется, это диалоговое окно мастера отчетов необходимо пропустить и нажать кнопку **Next**

9. В следующем диалоговом окне мастера задается стиль отображения объектов в отчете (рис.3.). Список **Style** (Стиль) содержит несколько вариантов отображения объектов (полей, линий, заголовков и т.д.) в отчете. С помощью области просмотра в верхнем левом углу диалогового окна можно просмотреть тот стиль, который мы выбрали. Выбрав стиль, нажать кнопку

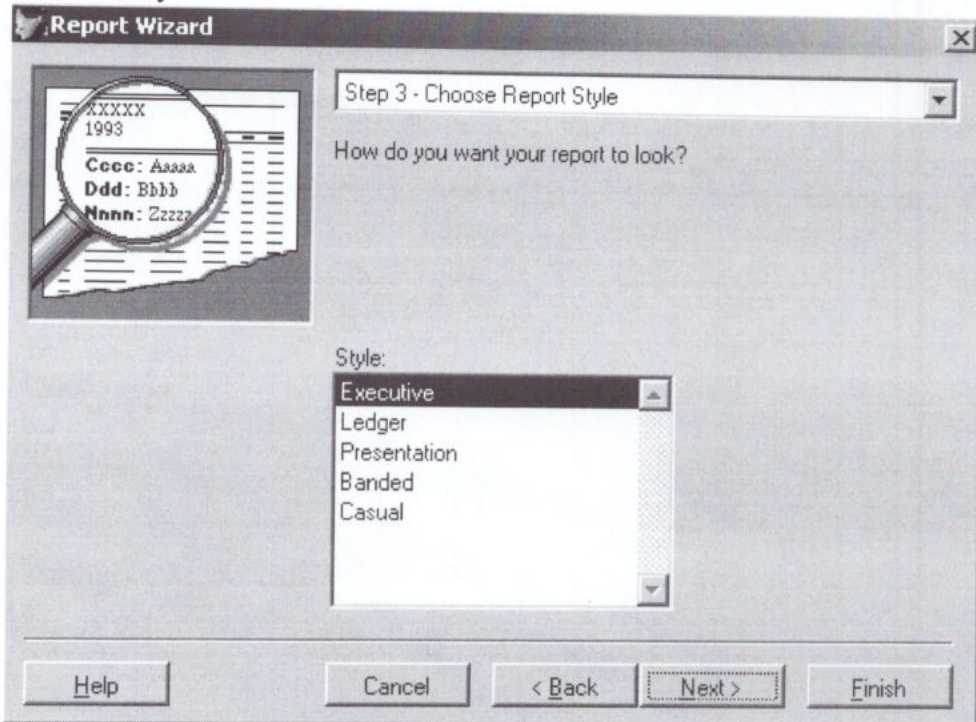


Рис.3. Диалоговое окно выбора стиля отображения объектов.

10. Указать порядок размещения объектов в отчете (рис.4.) и ориентацию страницы отчета. В области **Number of Columns** (Количество колонок) указать **1**, в области **Orientation** (Ориентация) указать **Portrait** (Книжная), в области **Field Layout** (Расположение полей) указать **Rows** (Строка). После установки требуемых опций, нажать кнопку **Next**

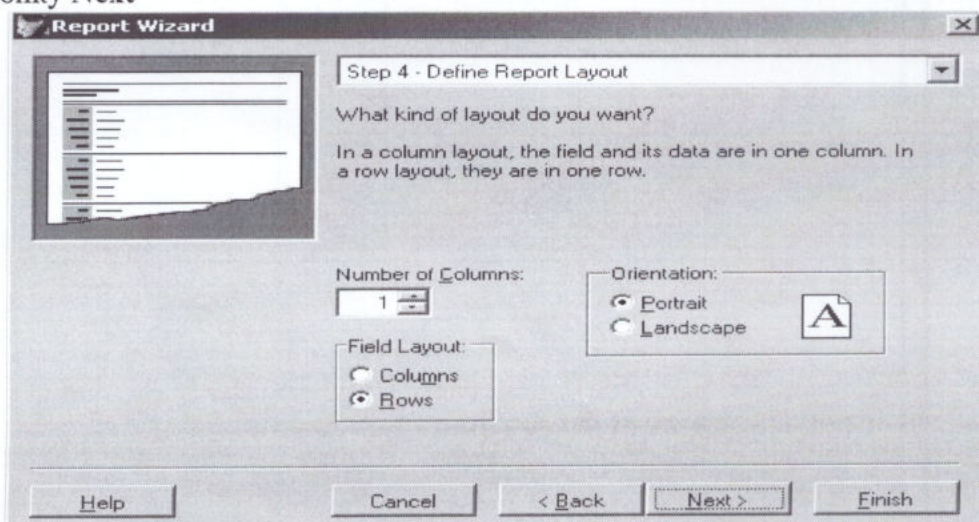


Рис.4. Диалоговое окно установки порядка размещения объектов.

11. Задать поле, по которому требуется упорядочение данных в отчете в соответствии с рис.5. Из списка **Available fields or index tag** (Выбранные поля и индексы) перенести в список **Selected fields** (Выбранные поля) поле **Fio** (упорядочивание данных по фамилии). Для переноса полей используйте кнопку **Add** (Добавить). Сформировав список полей, нажать кнопку **Next** (Далее);

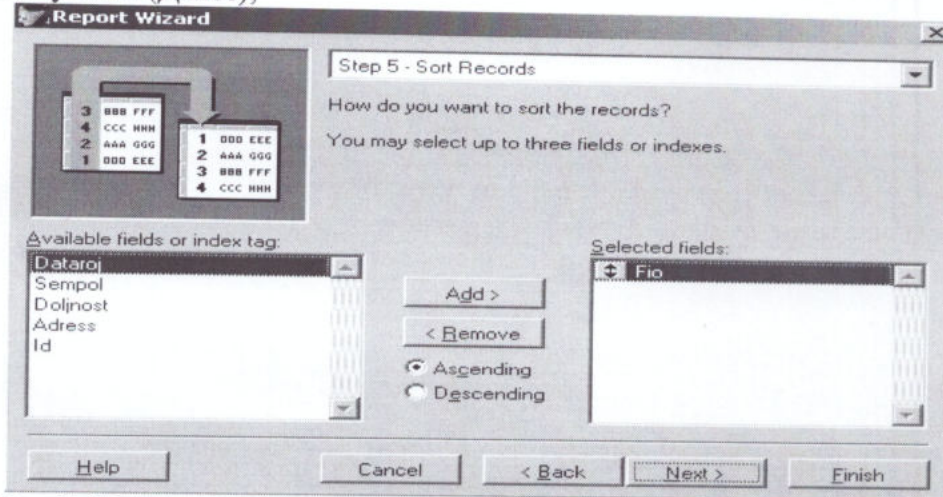
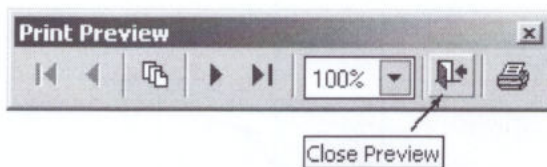


Рис.5. Диалоговое окно установки критерия упорядочения данных.

12. На заключительном шаге создания отчета в области **Type a title for your report** (Введите имя заголовка отчета) указать имя отчёта *Сотрудники*. Воспользовавшись кнопкой **Preview** (Просмотр) просмотреть, как будет выглядеть создаваемый отчет, после просмотра нажать кнопку **Close Preview** на панели **Print Preview** (см. рис.6.), если что-то не так, вернуться к предыдущим шагам, воспользовавшись кнопкой **Back**. Нажать кнопку **Finish** (Готово) и в появившемся диалоговом окне **Save As** (Сохранить как) указать папку, в которой будет размещен отчет *Сотрудники*.

Рис.6. Панель просмотра отчёта **Print Preview**

13. Просмотреть созданный отчёт: в окне **Project Manager** выбрать вкладку **Documents/Reports/Сотрудники/Preview** или команду **Preview** (Просмотр) из меню программы **View** (Вид)

Title (Титул)	В этой полосе размещается информация, появляющаяся перед основным отчётом (имя отчёта, сопроводительное письмо и т. д.) и называется титульной
Page Header (Верхний колонтитул)	Данные, помещённые в эту полосу, печатаются в начале каждой страницы (название отчёта, текущая дата и т. д.)
Detail (Детали)	Эта полоса содержит данные полей из таблицы или результат вычислений над ними
Page Footer (Нижний колонтитул)	В нижнем колонтитуле печатается название отчёта, дата, номер страницы, итоговые


значения по данным текущей страницы

14. Открыть отчёт *Сотрудники* в окне **Report Designer** (конструктор отчёта): в окне **Project Manager** выбрать вкладку **Documents/Reports/Сотрудники/Modify**

Таблица 1. Типы полос отчёта

15. Выделить все объекты в полосе **Title** (Титул) и перенести их полосе **Page Header** (Верхний колонтитул) используя метод перетаскивания мышью или кнопками управления курсором на клавиатуре, предварительно выделив необходимый объект, или комбинацией клавиш **Ctrl+[кнопки управления курсором]**

16. В полосе **Title** (Титул) напечатать следующий текст: *“Отчёт - форматированное представление данных, выводимое на экран, принтер или в файл. Табличный отчёт используется для печати данных, представленных в виде списка”*

17. Поместить набранный текст в прямоугольник со скругленными краями: нажать на панели инструментов **Report Controls** кнопку  и поместить данный объект в отчёт. Отредактировать его расположение в соответствии с рис.7.

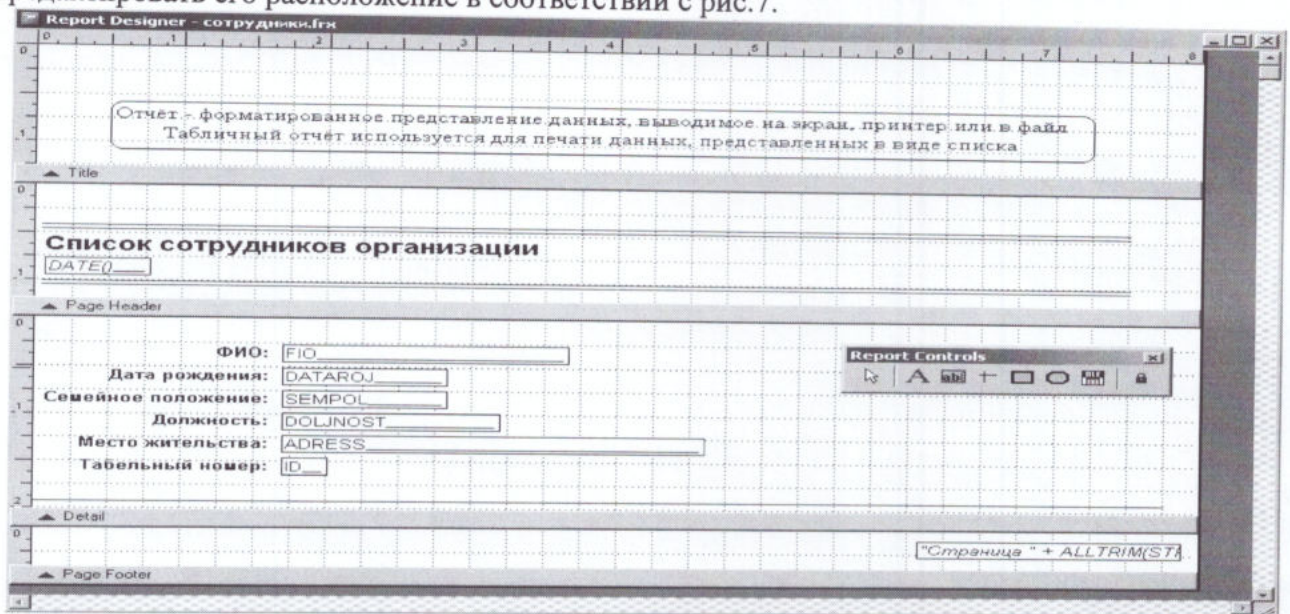


Рис.7. Окно **Report Designer** (Конструктор отчётов)

18. В процессе работы просматривайте результаты редактирования, используя пункт меню **View/Preview**

19. В полосе **Detail** (Детали), ниже поля **Табельный номер** провести линию, используя кнопку  панели инструментов **Report Controls** и отредактировать её расположение в отчёте в соответствии с рис.7.

20. В полосе **Page Footer** (Нижний колонтитул) переместить элемент `"Page " + ALLTRIM(STR(PAQ` в правый нижний угол в соответствии с рис.7.

21. Вызвать свойства элемента `"Page " + ALLTRIM(STR(PAQ` отчёта двойным щелчком мыши или через контекстное меню команда **Properties** и в появившемся диалоговом окне **Report Expression** (Выражение отчёта) в поле ввода **Expression** заменить слово *Page* на слово *Страница*, нажать **OK**

22. Просмотреть результат выполненной работы: пункт меню **View/Preview**

Задание

Создать отчет для таблицы *Книги* БД *Библиотека*

6. В диалоговом окне **Report Wizard** (Мастер отчетов) (см. рис.1.) выбрать следующие поля таблицы *Книги*: **Nazvanie, Razdel, Izdat, Godizdan**

7. В диалоговом окне выбора стиля отображения объектов (см. рис.3.) в поле области **Style** (Стиль) выбрать стиль **Executive**

8. В диалоговом окне установки порядка размещения объектов (см. рис.4.) в поле области **Number of Columns** (Количество колонок) указать **1**, в области **Field Layout** (Расположение полей) указать **Rows** (Строка)

9. В диалоговом окне установки критерия упорядочения данных (см. рис.5.) из списка **Available fields or index tag** (Выбранные поля и индексы) перенести в список **Selected fields** (Выбранные поля) поле **Nazvanie**

10. Просмотреть вид создаваемого отчета нажатием кнопки **Preview** (Просмотр).

11. После сохранения запустить отчет *Книги*, она должна иметь вид как на рис.8.

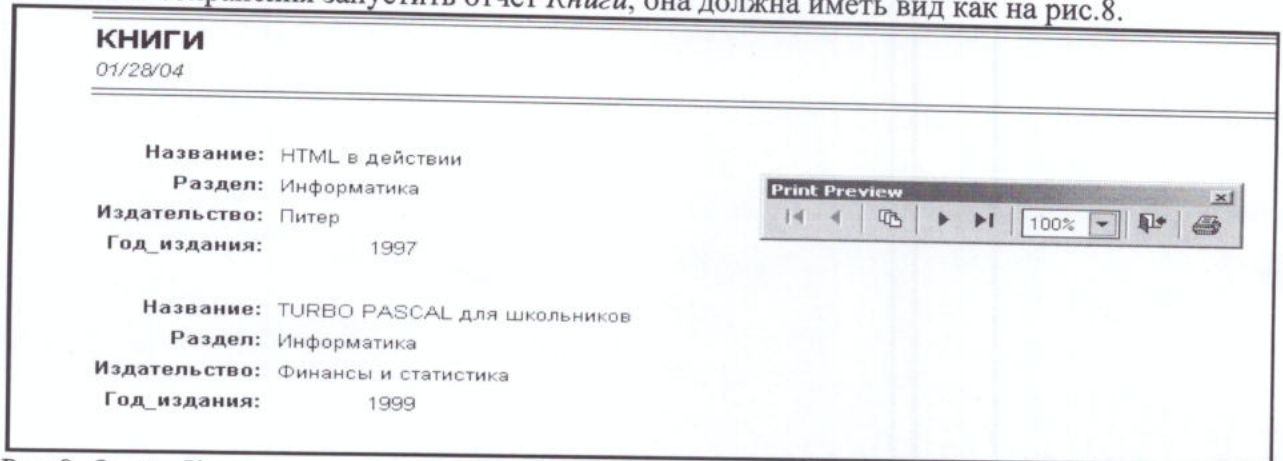


Рис.8. Отчёт *Книги*

Контрольные вопросы

1. Что такое отчёт
2. Какие средства можно использовать для создания отчётов в **Visual FoxPro**
3. Как можно отредактировать отчет
4. Какие существуют типы полос в окне **Report Wizard** (Конструктор Отчёта)
5. Как поместить объект в отчет, как изменить его расположение в отчёте
6. Какие объекты присутствуют на панели **Report Controls**
7. Как просмотреть созданный отчёт

Лабораторная работа №6

Создание запросов в среде Visual FoxPro

Теоретическая часть

Одним из основных назначений разработанного приложения является быстрый поиск информации в БД и получение ответов на разнообразные вопросы. Для этих целей в **Visual FoxPro** используются средства, называемые запросами.

При создании запросов в **Visual FoxPro** разработчик может использовать следующие средства:

- **Query Wizard** (Мастер запросов)
- **Query Designer** (Конструктор запросов)
- Команда **SELECT** языка **Visual FoxPro**

Результатом запроса является таблица, которую можно сохранить в массиве, создаваемой новой таблице, отобразить на экране в режиме **Browse** (Просмотр) или вывести в виде отчёта.

Для создания запросов можно использовать **Query Wizard** (Мастер запросов), который последовательно запрашивает наименования таблиц, используемых в запросе, перечень полей таблиц, критерий упорядочения и условия фильтрации данных. С помощью **Query Designer** (Конструктор запросов) можно формировать различной сложности критерии для выбора записей из одной или нескольких таблиц, над полями, выбираемыми из таблиц с помощью запросов, можно выполнять различные вычисления.






В верхней части окна **Query Designer** (Конструктор запросов) расположена панель, на которой отображаются используемые в запросе таблицы. Ниже находятся вкладки, предназначенные для выбора полей запроса и формирования условий выборки. Назначение

Вкладка	Назначение
Fields (Поля)	Позволяет указать поля исходных таблиц, выбираемые в результирующий запрос
Join (Объединение)	Позволяет задать условия объединения таблиц
Filter (Фильтр)	Позволяет определить фильтры, накладываемые для выбора записей
Order By (Упорядочение)	Позволяет задать критерии упорядочения данных
Group By (Группировка)	Позволяет задать условия группировки данных
Miscellaneous (Разное)	Позволяет задать дополнительные условия, такие как признак выборки повторяющихся значений, количество или процент выбора данных

этих вкладок приведено в таблице 1.. Открывая в окне **Query Designer** (Конструктор запросов) необходимые вкладки, можно выполнять следующие действия:

- Выбирать поля результирующей таблицы запроса
- Формировать вычисляемые поля
- Задавать критерии для выборки, группировки и упорядочения данных
- Указывать, куда выводить результаты выборки

Таблица 1. Назначение вкладок окна **Query Designer** (Конструктор запросов)

Команда меню	Кнопка	Назначение
Add Table (Добавить таблицу)		Добавляет в запрос новую таблицу
Remove Table (Удалить таблицу)		Удаляет выбранную таблицу из запроса
Remove Join Condition (Удалить условие объединения)		Удаляет условие объединения таблиц
Output Fields (Результирующие поля)		Открывает вкладку Fields для выбора полей результирующей таблицы
Join (Объединение)		Открывает вкладку Join для создания условия объединения таблиц
Filter (Фильтр)		Открывает вкладку Filter для задания фильтра
Order By (Упорядочение)		Открывает вкладку Order By для определения критерия упорядочения данных
Group By (Группировка)		Открывает вкладку Group By для определения условия группировки данных
Miscellaneous (Разное)		Открывает вкладку Miscellaneous для задания дополнительных параметров запроса
Query destination (Результат запроса)		Открывает диалоговое окно Query destination , в котором указывается куда выводить результат запроса
View SQL (Показать SQL)		Открывает диалоговое окно, в котором отображается SQL-оператор, соответствующий данному запросу
Maximize the table view (Максимизировать панель отображения)		Раскрывает панель отображения используемых в запросе таблиц на весь экран. Повторное нажатие возвращает панели


		первоначальный размер
Add Join (Добавить условие объединения)		Открывает диалоговое окно Join Condition для задание условия объединения таблиц
Comments (Комментарии)		Открывает диалоговое окно, в котором вы можете ввести краткое описание создаваемого запроса
Run Query (Выполнить запрос)		Запускает запрос на выполнение

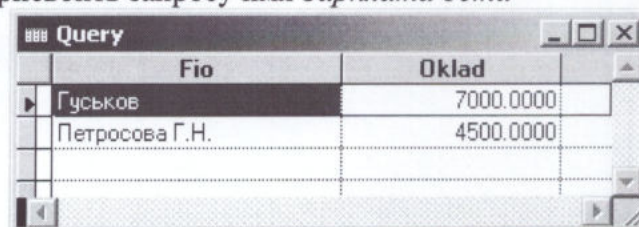
Таблица 2. Назначение команд меню **Query** и кнопок панели инструментов **Query Designer** окна **Query Designer** (Конструктор запросов)

Цель работы

1. создать запрос с помощью **Query Wizard** (Мастер запросов)
2. создать запрос с помощью **Query Designer** (Конструктор запросов)

Практическая часть

23. Запустить программу **Microsoft Visual FoxPro**
24. Открыть проект *Информационная система*
25. Открыть БД *Штам* проекта
26. Создать запрос о сотрудниках, имеющих более одного ребёнка и получающих зарплату менее 9000: в окне **Project Manager** щёлкнуть клавишей мыши на вкладке **Data/Queries/New/Query Wizard**, в появившемся диалоговом окне **Wizard Selection** (Выбор мастера) выбрать пункт **Query Wizard** (Мастер запросов) и нажать кнопку **OK**
27. В появившемся диалоговом окне выбора исходной таблицы и полей в области **Databases and tables** (Базы данных и таблицы) выбрать необходимую БД *штам* и указать таблицу *Сотрудники*. Из области **Available fields** (Имеющиеся поля) выбрать поля **Fio** и **Oklad**. Для перехода к следующему шагу нажать кнопку **Next** (Далее)
28. В появившемся диалоговом окне выбора условий выборки в области **Field** (Поле) верхней строки выбрать поле **Children** (Количество детей) таблицы *Сотрудники*, в области **Operator** (Оператор Условия) выбрать условие **more then**, в области **Value** (Выражение) установить значение **1**, в нижней строке внести следующие данные соответственно: в области **Field** (Поле) – поле **Oklad** (Зарплата), в области **Operator** (Оператор Условия) – **less than**, в области **Value** (Выражение) – **9000**. Нажать кнопку **Next** (Далее)
29. В появившемся диалоговом окне выбора условия сортировки данных из области **Available fields** (Имеющиеся поля) выбрать поле **Fio**. Нажать кнопку **Next** (Далее)
30. Нажать кнопку **Preview** (Просмотр) и просмотреть результат запроса (запрос будет выглядеть в виде простой таблицы см. рис.1.), закрыть таблицу запроса. Нажать кнопку **Finish** (Готово), в появившемся диалоговом окне **Save As** (Сохранить как) указать необходимую папку и присвоить запросу имя *Зарплата дети*



	Fio	Oklad
▶	Гуськов	7000.0000
	Петросова Г.Н.	4500.0000

Рис.1. Запрос *Зарплата дети*

31. Запустить запрос: в окне **Project Manager** щёлкнуть клавишей мыши на вкладке **Data/Queries/Зарплата дети/Run**
32. Закрыть запрос *Зарплата дети*
33. Создать новый запрос, показывающий количество сотрудников организации, получающих определённую зарплату: в окне **Project Manager** щёлкнуть клавишей мыши на

вкладке **Data/Queries/New/New Query**. В появившемся диалоговом окне **Add Table or View** (Добавить таблицу или представление данных) в области **Database** (База данных) выбрать БД *Штам*, в области **Tables in Database** таблицу *Сотрудники*. Нажать кнопку **Add**, при этом в верхней области окна **Query Designer** (конструктор запросов) (см. рис.2.), находящегося позади окна **Add Table or View** (Добавить таблицу или представление данных) появится выбранная таблица. Нажать кнопку **Close** (Закреть) для закрытия окна **Add Table or View**.

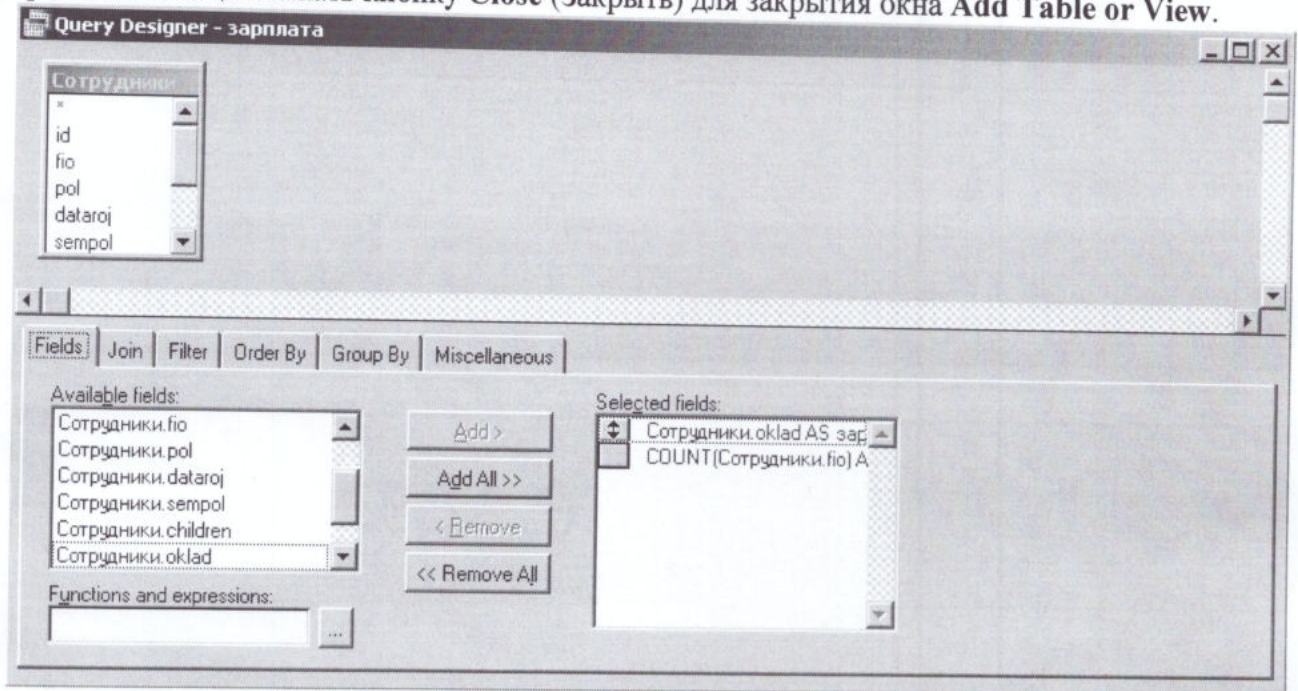


Рис.2. Окно **Query Designer** (конструктор запросов)

34. На вкладке **Fields** (Поля) окна **Query Designer** (конструктор запросов) в области **Available fields** (Имеющиеся поля) выделить поле *Сотрудники.oklad*, оно автоматически перенесётся в область **Function and expressions** (Функции и выражения), нажать расположенную справа от поля кнопку вызова построителя выражения, откроется диалоговое окно **Expression Builder** (Построитель выражения) (см. рис.3.)

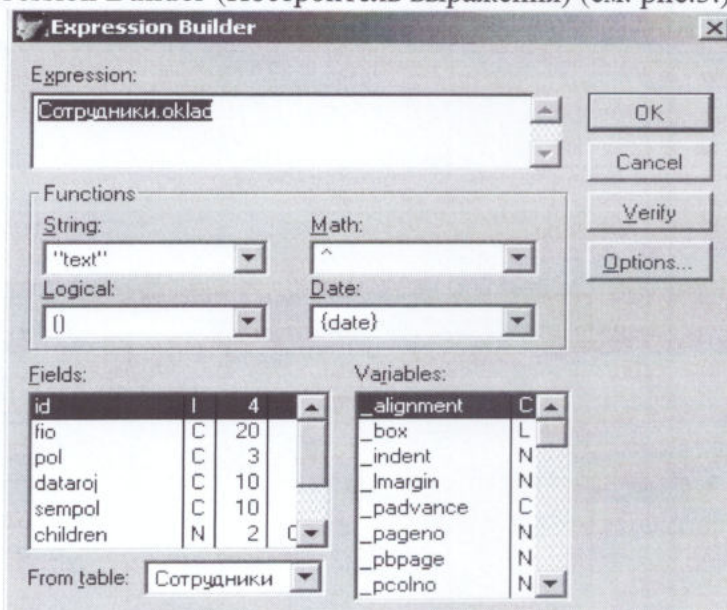


Рис.3. Диалоговое окно **Expression Builder** (Построитель выражения)

35. В поле ввода **Expression** (Выражение) диалогового окна **Expression Builder** (Построитель выражения) используя поля таблиц, расположенные в списке **Fields** (Поля),

функции области **Functions** (Функции), и ввод данных с клавиатуры сформировать следующее выражение: **Сотрудники.oklad AS Зарплата**

36. Для проверки правильности набранного выражения нажать кнопку **Verify** (Проверить), нажать кнопку **OK**

37. Для переноса созданного выражения из области **Function and expressions** (Функции и выражения) в область **Selected Fields** (Выбранные поля) нажать находящуюся между данными областями кнопку **Add** (Добавить)

38. Руководствуясь предыдущими пунктами 12-15 практической части лабораторной работы создать следующее выражение:

COUNT(Сотрудники.fio) AS Количество_сотрудников

39. Выбрать вкладку **Order By** (Упорядочение) окна **Query Designer** (конструктор запросов) и из области **Selected Fields** (Выбранные поля) в область **Ordering criteria** (Критерий упорядочения) перенести поле **Сотрудники.oklad**

40. Выбрать вкладку **Group By** (Группировка) окна **Query Designer** (конструктор запросов) и из области **Available fields** (Имеющиеся поля) в область **Grouped Fields** (Поля группировки) перенести поле **Сотрудники.oklad**

41. Выбрать пункт меню **Query** (Запрос) окна программы **Visual FoxPro**, выбрать команду **Comments** (Комментарии) (см. таблицу 2.). В появившемся окне **Comment** (Комментарий) в поле ввода области **Add Comment** (Добавить комментарий) ввести строку: **Запрос, показывающий количество сотрудников организации, получающих определённую зарплату.** Закрыть окно **Comment** (Комментарий)

42. Запустить созданный запрос: установить курсор в окне **Query Designer** (конструктор запросов), вызвать контекстное меню, выбрать команду **Run Query**, или нажать кнопку [!] на панели инструментов. Закрыть запрос

Зарплата	Количество_сотрудников
4500.0000	2
6000.0000	1
7000.0000	4
9000.0000	1
12000.0000	1
15000.0000	1

Рис.4. Запрос *Зарплата*

43. Просмотреть SQL-оператор, соответствующий запросу: установить курсор в окне **Query Designer** (конструктор запросов), вызвать контекстное меню, выбрать команду **View SQL** (Показать SQL), откроется диалоговое окно, в котором отображается SQL-оператор, соответствующий данному запросу. Закрыть окно **Query Designer** (конструктор запросов), система предложит сохранить запрос, нажать кнопку **Yes** (Да), в появившемся диалоговом окне **Save As** (Сохранить как) указать необходимую папку и присвоить запросу имя *Зарплата*

Задание

Создать многотабличный запрос, выводящий в таблицу названия, год издания и автора книг по информатике

1. Создать новый запрос, добавить таблицы *Авторы*, *Автор_книги* и *Книги* из БД *Библиотека* в окно **Query Designer** (конструктор запросов)

2. Из области **Available fields** (Имеющиеся поля) вкладки **Fields** (Поля) внести в область **Selected Fields** (Выбранные поля) используя построитель выражения **Expression Builder** следующие выражения:

Книги.nazvanie AS название_книги

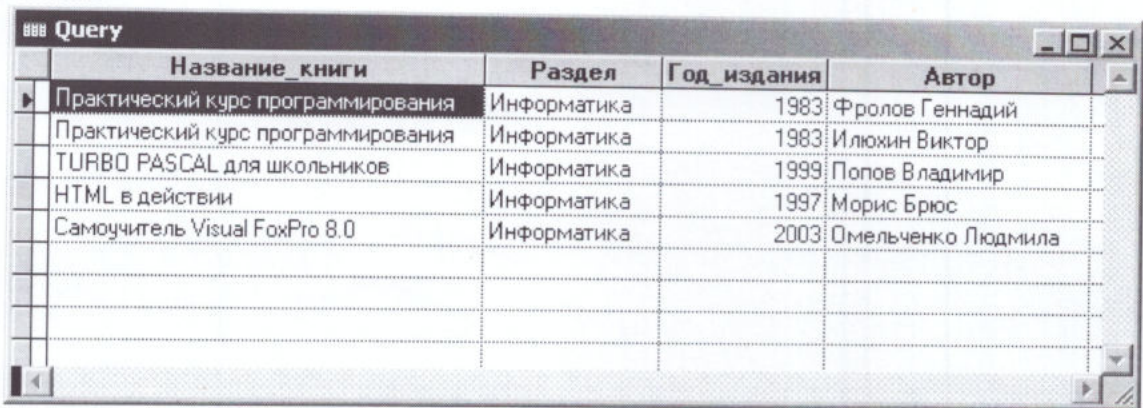
Книги.razdel AS раздел

Книги.godizdan AS год_издания

ALLTRIM(Авторы.familiya)+" "+ALLTRIM(Авторы.imya) AS автор

3. Необходимо задать условие выбора записей из таблицы (Книги только по информатике) по полю **razdel** (Раздел). Перейти на вкладку **Filter** (Фильтр), в области **Field Name** (Имя поля) нажать кнопку раскрытия списка и выбрать поле **Книги.razdel**, в области **Criteria** (Критерии) выбрать значение **==**, в области **Example** (Образец) ввести значение "Информатика"

4. Запустить запрос и посмотреть результат (см. рис.5.), закрыть запрос, закрыть окно **Query Designer** (конструктор запросов), сохранить запрос, присвоив ему имя *Книги по информатике*



Название_книги	Раздел	Год_издания	Автор
Практический курс программирования	Информатика	1983	Фролов Геннадий
Практический курс программирования	Информатика	1983	Илюхин Виктор
TURBO PASCAL для школьников	Информатика	1999	Попов Владимир
HTML в действии	Информатика	1997	Морис Брюс
Самоучитель Visual FoxPro 8.0	Информатика	2003	Омельченко Людмила

Рис.5. Запрос *Книги по информатике*

Контрольные вопросы

1. Что такое запрос
2. Какие действия можно выполнять в окне **Query Designer** (Конструктор запросов)
3. Какие вкладки находятся в окне **Query Designer** (перечислить)
4. Какого назначения команд меню **Query** (Запрос)
5. Какого назначения кнопок панели инструментов **Query Designer**
6. Как создать запрос в **Query Wizard** (Мастер запросов)
7. Для чего необходимо окно **Expression Builder** (Построитель выражения)
8. Опишите основные элементы окна **Expression Builder** (Построитель выражения)
9. Как создать многотабличный запрос
10. Для чего необходима команда **AS** языка **SQL**
11. Как задать условие выбора записей из таблицы
12. Как просмотреть **SQL**-оператор, соответствующий запросу
13. Как запустить запрос

Лабораторная работа №7

Создание меню приложения в среде **Visual FoxPro**

Теоретическая часть

В соответствии со стандартами **Windows** в любом приложении рекомендуется иметь строку меню, которая в **Visual FoxPro** содержит команды, предназначенные для вызова форм, формирования отчётов, запросов и т.д.

Строкой меню называется горизонтальное меню, располагаемое в верхней части экрана. Примером строки меню является основное меню **Visual FoxPro**, а также меню программ, работающих в среде **Windows**. Созданное в конструкторе **Menu Designer** (Конструктор меню) меню может замещать основное меню **Visual FoxPro** или добавляться к нему.

Для создания меню необходимо выполнить следующие действия:

- Открыть окно конструктора меню **Menu Designer** (Конструктор меню)

- Описать вид меню, текст, пункты меню и его атрибуты

- Сгенерировать меню. При этом создаётся программа, которая в результате запускается на выполнение

В **Visual FoxPro** имеется возможность создания меню в виде строки **Menu**, или всплывающего меню **Shortcut**, в котором основные пункты расположены по вертикали.

Кнопка	Назначение
Insert (Вставить)	Добавляет в меню новый пункт
Insert Bar (Вставить команды системного меню)	Открывает диалоговое окно Insert System Menu Bar , содержащее команды системного меню Visual FoxPro , позволяя разместить их в создаваемом пользовательском меню
Delete (Удалить)	Удаляет текущий пункт меню
Move Item (Переместить элемент)	Открывает одноимённое диалоговое окно, позволяющее указать пункт меню, в который переносится текущий подпункт
Preview (Просмотр)	Размещает создаваемое меню на экране, позволяя просмотреть его внешний вид

Таблица 1. Назначение кнопок конструктора меню **Menu Designer** (Конструктор меню)

Тип меню	Назначение
Command (Команда)	При выборе пункта меню данного типа будет выполняться связанная с ним команда
Pad Name (Наименование строки меню)	При выборе пункта меню никаких действий выполняться не будет. Как правило, используется в качестве дополнительного пояснения к меню
Submenu (Подменю)	При выборе пункта меню раскрывается связанное с данным пунктом ниспадающее меню
Procedure (Процедура)	При выборе пункта меню вызывается процедура, определённая для данного пункта меню

Таблица 2. Типы меню

Тип пункта меню	Действие
Submenu (Подменю)	Раскрывается связанное с данным пунктом меню ниспадающее меню
Procedure (Процедура)	Выполняется процедура, определённая в конструкторе меню
Command (Команда)	Выполняется команда, расположенная в поле рядом с типом пункта меню

Таблица 3. Типы пункта меню

Цель работы

1. Создать структуру меню **Menu** в окне **Menu Designer** (Конструктор меню)
2. Определить действия для пунктов меню
3. Сохранить, сгенерировать, запустить меню

Практическая часть

44. Запустить программу **Microsoft Visual FoxPro**
45. Открыть проект *Информационная система*
46. Открыть окно **Menu Designer** (Конструктор меню): в окне **Project Manager** щёлкнуть клавишей мыши на вкладке **Other/Menus/New**, в появившемся диалоговом окне **New Menu**

(Новое меню) выбрать пункт **Menu** (Меню), появится окно **Menu Designer** (Конструктор меню) (см. рис.1.)

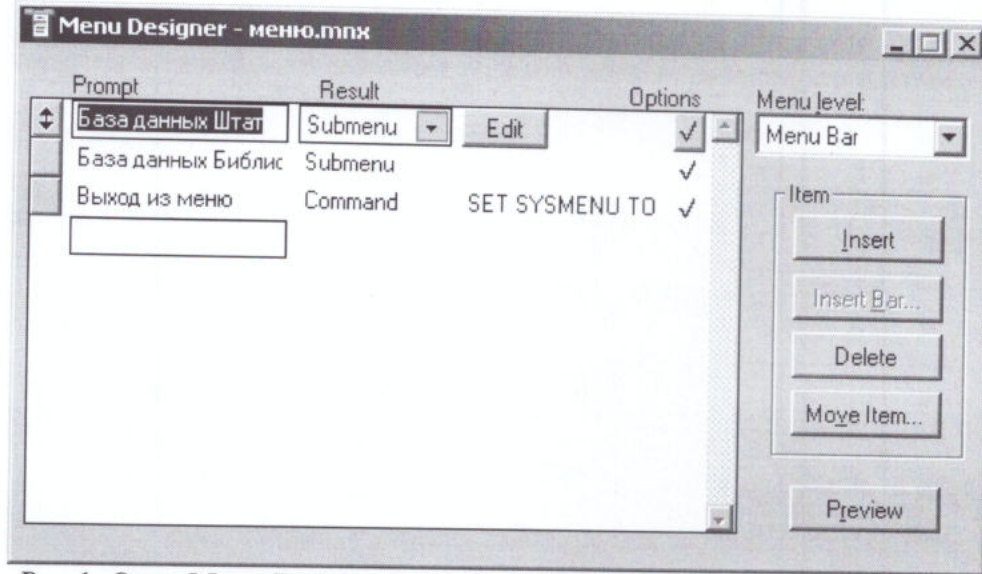


Рис.1. Окно **Menu Designer** (Конструктор меню)

Область конструктора меню, над которой расположены надписи **Prompt** (Приглашение), **Result** (Результат) и **Options** (Опции), предназначена для формирования меню. В поле **Prompt** (Приглашение) вводят наименования пунктов меню, раскрывающийся список **Result** (Результат) используется для указания типа пункта меню, а кнопка **Options** (Опции) открывает диалоговое окно **Prompt Options** (Опции элемента меню), в котором можно определить дополнительные параметры данного элемента меню. В списке **Menu level** (Уровень меню) указывается уровень текущего меню. Слева в конструкторе меню размещены кнопки (см. таблицу 1.)

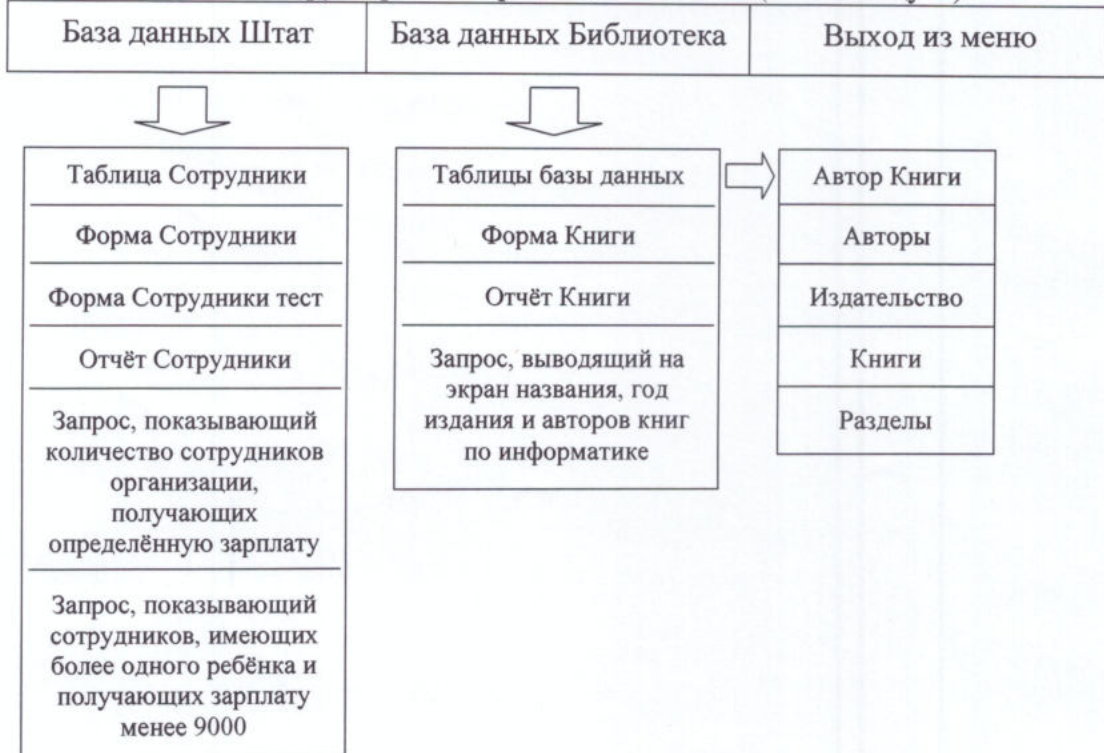


Рис.2. Схема создаваемого меню *Меню*

47. В поле **Prompt** (Приглашение) ввести наименование первого пункта меню – База данных Штат, нажать для перехода на следующее поле клавишу **Enter** или **Tab** на

клавиатуре, курсор окажется в списке **Result** (Результат) (см. рис.1.) Указать тип пункта меню **Submenu** (Подменю), нажать кнопку **Options** (Опции), откроется диалоговое окно **Prompt Options** (Опции элемента меню), в котором, в области **Shortcut** (Всплывающее меню) установить курсор в поле **Key Label** (Метка) и нажать комбинацию клавиш **Alt + F1** на клавиатуре. Мы определили комбинацию клавиш быстрого вызова пункта меню База данных Штат. Нажать кнопку **OK**

48. Перейти на следующую строку и ввести наименование второго пункта – База данных Библиотека, тип пункта меню **Submenu** (Подменю), определить комбинацию клавиш **Alt + F2** быстрого вызова пункта меню База данных Библиотека (см. пункт 4. лабораторной работы)

49. Перейти на следующую строку и ввести наименование третьего пункта – Выход из меню, в списке **Result** (Результат) указать тип пункта меню **Command** (Команда), в появившейся справа строке ввода ввести команду – SET SYSMENU TO DEFAULT. Определить комбинацию клавиш **Alt + F4** быстрого вызова действия пункта меню Выход из меню

50. Пункты меню База данных Библиотека и База данных Штат имеют свои подменю. Для создания подменю перейти на строку База данных Библиотека, нажать кнопку **Create** (Создать), находящуюся справа от списка **Result** (Результат), на экране появится пустое окно конструктора меню. Ввести в поле **Prompt** (Приглашение) наименование подпункта – Таблицы базы данных, в списке **Result** (Результат) указать тип пункта меню **Submenu** (Подменю), нажать кнопку **Create** (Создать), так как он тоже имеет подменю (см.рис.2.)

51. В появившемся пустом окне конструктора меню ввести в поле **Prompt** (Приглашение) наименование подпункта – Автор книги, в списке **Result** (Результат) указать тип пункта меню **Procedure** (Процедура), нажать кнопку **Create** (Создать) и в открывшемся окне редактирования процедуры ввести следующие команды:

Задание

Добавить в пункт меню – База данных Штат подпункты, определить действия для этих подпунктов меню в соответствии с таблицей 4.

Prompt (Приглашение)	Result (Результат)	Команды и процедуры
Таблица Сотрудники	Procedure	OPEN DATABASE \ <i>Имя вашей папки</i> \штат.dbc USE штат!сотрудники SELECT Сотрудники BROWSE LAST CLOSE DATABASES
Форма Сотрудники	Command	DO FORM \ <i>Имя вашей папки</i> \сотрудники.scx
Форма Сотрудники тест	Command	DO FORM \ <i>Имя вашей папки</i> \тест.scx
Отчёт Сотрудники	Command	REPORT FORM \ <i>Имя вашей папки</i> \сотрудники.frx PREVIEW

Таблица 4. Наименование пунктов меню и определение их действий

Prompt (Приглашение)	Result (Результат)	Команды и процедуры
Запрос, показывающий количество сотрудников организации, получающих	Procedure	OPEN DATABASE \ <i>Имя вашей папки</i> \штат.dbc DO "\iИмя вашей папки>\зарплата.qpr" CLOSE DATABASES

ОБРАЗЦЫ ЗАДАНИЙ ДЛЯ СОЗДАНИЯ БД

Информационная система “Библиотека”

Разработать систему, помогающую обслуживать библиотеку. Система должна предусматривать режимы ведения системного каталога, отражающего перечень областей знаний, по которым имеются книги в библиотеке. Внутри библиотеки области знаний в систематическом каталоге могут иметь уникальный внутренний номер и полное наименование. Каждая книга может содержать сведения из нескольких областей знаний. Каждая книга в библиотеке может присутствовать в нескольких экземплярах.

В библиотеке ведется картотека читателей.

На каждого читателя в картотеку заносятся следующие сведения:

- Фамилия, Имя, Отчество
- домашний адрес
- телефон (Будем считать, что у нас два телефона - рабочий и домашний)
- дата рождения

Каждому читателю присваивается уникальный номер читательского билета.

Каждый читатель может одновременно держать на руках не более 5 книг. Читатель не должен одновременно держать более одного экземпляра книги одного названия.

Каждая книга, хранящаяся в библиотеке, характеризуется следующими параметрами:

- уникальный шифр
- название
- фамилия автора (может отсутствовать)
- фамилии соавторов (могут отсутствовать)
- место издания (город)
- издательство
- год издания
- количество страниц.
- количество экземпляров книги в библиотеке

Каждая книга в библиотеке может присутствовать в нескольких экземплярах. Каждый экземпляр имеет следующие характеристики:

- уникальный инвентарный номер
- шифр книги, который совпадает с уникальным шифром из описания книг
- присутствие в библиотеке.

В случае отсутствия данного экземпляра книги должны быть записаны следующие сведения:

- номер билета читателя, который взял книгу
- дата выдачи книги
- дата возврата.

Каждый экземпляр книги может находиться на руках только у одного читателя.

Предусмотреть следующие ограничения на информацию в системе:

- Не может быть информации о соавторах, если нет информации об авторе.
- Книга может не иметь ни автора, ни соавтора.
- В библиотеке должны быть записаны читатели не моложе 17 лет.
- В библиотеке присутствуют книги, изданные начиная с 1960 по текущий год
- Каждый читатель может держать на руках не более 5 книг.
- Каждый читатель при регистрации в библиотеке должен дать телефон для связи он может быть рабочим или домашним.
- Каждая область знаний может содержать ссылки на множество книг, но каждая книга может относиться к различным областям.

Информационная система “Обеспечение учебного процесса”

Задано расписание занятий групп, в котором указаны следующие параметры

- день недели (занятия в субботу не проводятся)
- номер пары (занятия только для дневных факультетов, ограничить допустимый номер пары)
- фамилия преподавателя (выбирается из специальной таблицы, содержащей список преподавателей, с указанием их должностей и ученых степеней)
- название дисциплины (выбирается из таблицы дисциплин на семестр)
- тип занятий (выбирается из допустимого списка типов занятий: лекция, лабораторные работы, упражнения, курсовой проект)
- номер группы (выбирается из списка групп, который хранится в отдельной таблице)

По каждой дисциплине определено количество лабораторных работ, которое требуется выполнить в данном семестре количество домашних заданий по упражнениям и наличие курсового проекта.

В расписании на одной и той же паре в один и тот же день недели у одной группы не может быть нескольких занятий.

В расписание занятий должны быть внесены все запланированные занятия на данный семестр для всех групп.

Для каждого студента ведется электронный журнал, в котором занесены данные о сдаче студентом всех лабораторных, домашних заданий и курсовиков. При планировании домашних заданий отводить на каждое задание 2 календарных недели.

Информационная система “Железнодорожная станция”

Необходимо разработать информационную систему, обслуживающая пассажиров на предмет покупки билетов и получения информации о тех или иных составах. Система не привязана к конкретной станции и, тем самым, с любой станции можно купить билет на любой состав, как следующий через эту станцию, так и не следующий, но внесённый в систему. Система предусматривает возможность покупки билета на одно и то же место, если пассажир следует с промежуточной станции, а не со станции отправления состава, или до промежуточной станции.

Продажа билетов осуществляется от текущей даты и на 30 дней вперёд, для чего имеется таблица “Даты отправления составов”.

В системе ведётся картотека купленных билетов. По прохождении времени прибытия состава на станцию назначения, билеты проданные на этот состав удаляются из картотеки. Также удаляется и состав из таблицы “Даты отправления составов”.

На каждый билет в картотеку заносятся следующие сведения:

- ФИО пассажира
- Номер состава
- Номер вагона
- Номер места
- Дата отправления состава со станции следования
- Станция следования пассажира – станция посадки
- Станция назначения пассажира – конечная станция для пассажира
- Дата отправления пассажира
- Тип вагона (плацкартный, купейный или сидячий)
- Время отправления пассажира
- Время следования пассажира
- Время прибытия пассажира
- Цена билета

Каждому билету присваивается уникальный номер, так как на одно место может быть продано несколько билетов для пассажиров следующих через непересекающиеся станции. Как только покупается билет, место, указанное в билете, вносится в список занятых мест на

всём протяжении станций от станции следования пассажира до станции назначения. Соответственно, с прибытием состава на место назначения удаляются все занятые места, связанные с этим составом.

Каждое занятое место, характеризуется следующими параметрами:

- Номер билета
- Название станции
- Номер состава
- Номер вагона
- Номер места
- Дата отправления состава со станции следования

Каждому составу соответствует уникальный номер состава, а также следующие параметры:

- Станция следования состава
- Станция назначения состава
- Время отправления состава
- Время следования состава
- Время прибытия состава
- Дни недели, по которым отправляются составы

Для обеспечения целостности системы имеется таблица "Станции", в которой перечислены названия всех станций следования составов. Названия станций являются уникальными

Предусмотрены следующие ограничения на информацию в системе:

- В плацкартном вагоне 60 мест. В купейном и сидячем – соответственно 40 и 100.
- Время прибытия и отправления пассажира, а также тип вагона в билете, должны строго соответствовать реальному времени и типу вагона состава, на который продан билет. Цена билета высчитывается как сумма цен билетов между всеми станциями следования.
- Через одну станцию может проходить несколько составов, и каждый состав, естественно, проходит за время следования несколько станций.
- На одно и тоже место может быть куплено несколько билетов, если станции следования пассажиров, купивших эти билеты, не пересекаются.
- За один день может отправиться только один состав с одинаковым номером.
- Билеты продаются от текущей даты и до 30 дней вперёд.