

МИНИСТЕРСТВО ОБРАЗОВАНИЯ КРАСНОЯРСКОГО КРАЯ
КРАЕВОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«КРАСНОЯРСКИЙ КОЛЛЕДЖ ОТРАСЛЕВЫХ ТЕХНОЛОГИЙ
И ПРЕДПРИНИМАТЕЛЬСТВА»

РАССМОТРЕНО

методической комиссией
протокол № 06 от «24» июня 2021 г.

УТВЕРЖДЕНО

Директор КГБПОУ «Красноярский колледж
отраслевых технологий и предпринимательства»

_____/Н. В. Журова/
Приказ № 01-91-1п от «30» июня 2021 г.

АДАптиРОВАННАЯ ПРОГРАММА ПОДГОТОВКИ
СПЕЦИАЛИСТОВ СРЕДНЕГО ЗВЕНА

09.02.07 Информационные системы и программирование

на базе среднего общего образования

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ И ЛАБОРАТОРНЫХ РАБОТ

ОП.08 Основы проектирования баз данных

И. о. зам. директора по УР _____/Е. В. Миля
Подпись ФИО

Красноярск, 2021 г.

Организация-разработчик: Краевое государственное бюджетное профессиональное образовательное учреждение «Красноярский колледж отраслевых технологий и предпринимательства»

Разработчик: Лавренков Семен Сергеевич, преподаватель

СОДЕРЖАНИЕ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
2. ОБЩИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ И
ОФОРМЛЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ И ЛАБОРАТОРНЫХ РАБОТ
3. МЕТОДИКА ПРОВЕДЕНИЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ И
ЛАБОРАТОРНЫХ РАБОТ
4. СОДЕРЖАНИЕ ПРАКТИЧЕСКИХ ЗАНЯТИЙ И ЛАБОРАТОРНЫХ РАБОТ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические указания к проведению практических занятий и лабораторных работ по учебной дисциплине ОП.08 Основы проектирования баз данных предназначены для обучающихся СПО по специальности 09.02.07 Информационные системы и программирование.

Уровень профессиональной подготовки по специальности 09.02.07 Информационные системы и программирование, определяемый ФГОС СПО, предусматривает владение практическими навыками выбора материалов для профессиональной деятельности.

Формируемые компетенции, реализуемые в процессе выполнения практических занятий и лабораторных работ:

умения	проектировать реляционную базу данных; использовать язык запросов для программного извлечения сведений из баз данных
знания	основы теории баз данных; модели данных; особенности реляционной модели и проектирование баз данных; изобразительные средства, используемые в ER- моделировании; основы реляционной алгебры; принципы проектирования баз данных; обеспечение непротиворечивости и целостности данных; средства проектирования структур баз данных; язык запросов SQL

Выполнение лабораторных работ способствует формированию и развитию общих и профессиональных компетенций:

Код	Наименование общих компетенций
ОК 1	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 2	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 4	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста
ОК 9	Использовать информационные технологии в профессиональной деятельности
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языке

2. ОБЩИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ И ОФОРМЛЕНИЮ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ РАБОТ

Лабораторные (практические) работы выполняются обучающимися по графику, составленному в соответствии с рабочей программой учебной дисциплины ОП 08 Основы проектирования баз данных.

Учебная дисциплина ОП 08 Основы проектирования баз данных зависит от содержания лабораторных (практических) работ, которые соответствуют более глубокому освоению дисциплины, закреплению теоретических знаний и прививают обучающимся практические навыки самостоятельной работы.

Задача лабораторных работ (практических занятий) – закрепить теоретические знания обучающихся.

Согласно учебного плана по специальности и программы учебной дисциплины на лабораторные (практические) занятия обучающихся выделено 30 академических часов, из них:

№ и наименование разделов (тем)	Количество часов
Тема 2. Взаимосвязи в моделях и реляционный подход к построению моделей	4
Тема 3 Этапы проектирования баз данных	8
Тема 4 Проектирование структур баз данных	10
Тема 5. Организация запросов SQL	8
ИТОГО:	30

3. МЕТОДИКА ПРОВЕДЕНИЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ И ЛАБОРАТОРНЫХ РАБОТ

Целью практических занятий является приобретение практических навыков работы с базами данных, а именно создание проектов БД, проектирование реляционных БД, проведение сортировок в БД, создание сортировок и выборок, создание различных форм и запросов, разработка пользовательского интерфейса БД.

Целью лабораторных работ является отработка обучающимися практических навыков по разработке БД, управлению БД, проведению поисков необходимой информации в БД, модернизированную и сопровождению БД.

Исходя, из поставленных целей в работе будут решаться следующие задачи:

Закрепление знаний по:

- ✓ основам теории баз данных;
- ✓ моделям данных;
- ✓ особенностям реляционной модели и проектированию баз данных;
- ✓ изобразительным средствам, используемым в ER- моделировании;
- ✓ основам реляционной алгебры;
- ✓ принципам проектирования баз данных;
- ✓ обеспечению непротиворечивости и целостности данных;
- ✓ средствам проектирования структур баз данных;
- ✓ языкам запросов SQL

Ознакомиться:

- ✓ с проектированием баз данных;
- ✓ с сопровождением баз данных;

При выполнении лабораторной работы формируются навыки:

- ✓ работы новыми и готовыми базами данных.

Научиться пользоваться:

- ✓ различным программным обеспечением для разработки и сопровождения баз данных;
- ✓ различными способами сортировки и выборки данных.

4. СОДЕРЖАНИЕ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ РАБОТ

Тема 2. Взаимосвязи в моделях и реляционный подход к построению моделей

Лабораторное занятие № 1

Тема: Задание ключей. Создание основных объектов БД (2 часа)

Цель:

1. Задание ключей, создание основных объектов БД.

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: компьютер, операционная система Windows 7, MS Access 2010

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Краткие теоретические сведения:

База данных состоит из следующих компонентов:

Таблицы – основные объекты базы данных (БД). В них хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц. Сведения по разным вопросам следует хранить в разных таблицах. В БД столбцы называются *полями*, а строки – *записями*. Для работы таблицу необходимо открыть. Перед окончанием работы ее следует закрыть, предварительно сохранив все изменения.

Запрос – это средство, с помощью которого извлекается из базы данных информация, отвечающая определенным критериям. Результаты запроса представляют не все записи из таблицы, а только те, которые удовлетворяют запросу.

Формы – Обеспечивают более наглядную работу с таблицами, с помощью форм в базу вводят новые данные или просматривают имеющиеся.

Отчеты – средство представления данных таблиц. Отчеты могут быть оформлены надлежащим образом и распечатаны в том виде, в котором требуется пользователю.

Макросы – набор из одной или более макрокоманд, выполняющих определенные операции (открытие форм, печать отчетов)

Модули – это программы, написанные на языке программирования VisualBasic.

Таблицы

С таблицами можно работать в двух режимах — **таблицы** и **конструктора**.

В режиме таблицы можно начать ввод данных сразу, структура таблицы при этом будет создаваться автоматически. Полям присваиваются имена с последовательными номерами («Поле1», «Поле2» и т. д.).

На вкладке **Создание** в группе **Таблицы** щелкните **Таблица**. 

Приложение Access создаст базу данных с пустой таблицей с именем «Таблица1» и откроет эту таблицу в режиме таблицы. При этом будет выделена первая пустая ячейка столбца **Добавить поле..**

5. Чтобы добавить данные, начните их вводить в первую пустую ячейку или вставьте данные из другого

6. Для переименования столбца (поля) дважды щелкните заголовок столбца и введите новое имя. Рекомендуется присваивать полям значимые имена, чтобы при просмотре области **списка полей** было понятно, что содержится в каждом поле.

7. Для перемещения столбца сначала щелкните его заголовок, чтобы выделить столбец, а затем перетащите столбец в нужное место.

8. Кроме того, можно выбрать сразу несколько смежных столбцов, а затем одновременно перетащить их в новое место. Чтобы выбрать несколько последовательно расположенных столбцов, щелкните заголовок первого столбца, а затем, удерживая нажатой клавишу SHIFT, щелкните заголовок последнего столбца.

Создание таблицы в режиме конструктора

В режиме конструктора сначала создается структура новой таблицы. Затем можно переключиться в режим таблицы для ввода данных или ввести данные, используя другой метод, например, вставку или импорт.

4. На вкладке **Создание** в группе **Таблицы** щелкните **Конструктор таблиц**. 

5. Для каждого поля в таблице введите имя в столбце **Имя поля**, а затем в списке **Тип данных** выберите тип данных.

6. Когда все необходимые поля будут добавлены, сохраните таблицу:



4 Щелкните значок **Кнопка MicrosoftOffice**, а затем выберите команду **Сохранить**.

Порядок выполнения практической работы:

1. Изучить теоретический материал.

2. Выполнить предлагаемые задания.

3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:

- номер, наименование практической работы и тему;
- ответы на контрольные вопросы;
- выводы.

4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

ЗАДАНИЕ 1. СОЗДАНИЕ СТРУКТУРЫ БД.

1.1. Запустить программу MicrosoftAccess для создания новой базы данных.

Для этого следует нажать кнопки сначала **Пуск**, затем **Программы**, среди которых нужно выбрать **MicrosoftAccess**.

1.2. Создать новую базу данных под названием МОДЕЛИ, а в скобках указать Фамилию ученика.

Для создания базы данных войдите в **Пуск/Программы/MicrosoftAccess**. После чего откроется диалоговое окно на экране монитора **Создание новой базы данных**. Нужно выбрать пункт **Новая база данных** и нажать **ОК**, в появившемся окне выбрать диск **С:/Мои документы** и в строке **Имя файла** следует ввести слово **Модели (Фамилию ученика)** и затем нажать кнопку **Создать**. После чего открывается **окно базы данных** под названием **Модели (Фамилия ученика)**. В верхней части окна базы данных представлено меню, где перечислены объекты базы данных:

- **Таблицы**,
- **Запросы**,
- **Формы**,
- **Отчеты**,
- **Макросы**, - **Модули**.

1.3. Создать структуру новой таблицы.

Для создания структуры новой таблицы в **окне базы данных** следует щелкнуть мышью по кнопке **Таблицы**, затем по кнопке **Создать**. После чего **Система** откроет диалоговое окно, в котором будет предложено воспользоваться одним из следующих способов создания таблицы:

- **Режим таблицы**,
- **Конструктор**,
- **Мастер таблиц**,
- **Импорт таблиц**,
- **Связь с таблицами**.

Нужно выделить способ **Конструктор**, так как будем конструировать свою таблицу, и нажав **Ок**, структура **Таблицы** открывается в режиме конструктора, куда необходимо ввести

- **Имя поля**,
- **Тип данных**, - **Описание**.

1.4. Ввести в новую таблицу имена полей и их спецификации так, как это показано в приведенной ниже таблице.

ИМЯ ПОЛЯ	ТИП ДАННЫХ	ОПИСАНИЕ
Код модели	Числовой	Код модели по заводскому заказу
Модель	Текстовый	Тип салона
Мощность	Текстовый	Мощность двигателя (квт/л.с.)

Количество дверей	Числовой	Двухдверный или четырехдверный Салон
Коробка передач	Текстовый	Автоматическая или ручная
Обивка	Текстовый	Велюр, кожа, ткань
Другое оснащение	Мемо	Дополнительные аксессуары
Изображение	Поле объекта OLE	Рисунок, представляющий модель
Заводская цена	Денежный	Заводская продажная нетто-цена
Транспортные издержки	Денежный	Издержки на доставку
Предпродаж-ные издержки	Денежный	Издержки на предпродажную Подготовку

1.5. Ввести свойства полей новой таблицы так, как это показано ниже (приведены только те свойства полей, значения которых отличаются от стандартных).

Поле	Размер Поля	Формат Поля	Число десятичных Знаков	Значение По Умолчанию	Обязательно Поле	Индексированное поле
Код модели	Длинное целое	Основной	0		Да	Да(Не допускаются совпадения)
Модель	20				Да	Да(совпадения допуск)
Мощность	10					Нет
Количество Дверей	Целое	Основной	0	4		Нет
Коробка Передач	15			Ручная		Нет
Обивка	20					Нет
Другое Оснащение	Мемо					
Изображение					Нет	
Заводская Цена		Денежный	Авто		Да	Нет
Транспортные Издержки		Денежный	Авто		Да	Нет
Предпродажные Издержки		Денежный	Авто	105	Да	Нет

1.6. Установить первичный ключ таблицы.

Для этого необходимо при выделении поля **Код модели** нажать на панели инструментов Ключевое поле.

1.7. Сохранить созданную Структуру таблицы под именем Предлагаемые модели и закрыть таблицу.

1.8. Открыть снова таблицу Предлагаемые модели в режиме Конструктор и отредактировать ее:

- ввести новую строку под названием Цвет перед строкой Количество дверей.

Для вставки нового поля перед существующим необходимо выбрать область маркировки поля слева от существующего поля, далее нажать кнопку **Вставить строку** на панели инструментов. В новую строку ввести название имя поля –Цвет, тип данных текстовый, описание – цвет кузова.

- изменить название поля Мощность на название Мощность двигателя.

-поменять местами поля таблицы Цвет и Количество дверей.

Для того, чтобы поменять местами предлагаемые поля, необходимо перед полем Цвет создать пустую строку, как это было указано выше, затем вырезать строку Количество дверей и вставить ее в созданную пустую строку, используя панель инструментов.

Контрольные вопросы:

1. Опишите этапы создания таблицы в конструкторе?
2. Какие типы данных в БД вы знаете? 3. Какие параметры форматирования таблиц вы можете применить в БД?

Отчет

Сохранить БД в своей папке под названием Proba.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10 баллов	Защита своей работы персонально – 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №2

Тема: Создание проекта БД. Создание БД. Редактирование и модификация таблиц (2 часа)

Цель:

1. Знакомство реляционной БД, освоение принципов проектирования БД

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: компьютер

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Краткие теоретические сведения:

Основные типы полей, используемые в Microsoft Visual FoxPro приведены в таблице

Тип поля	Содержимое
Character (Символьный)	Любые символы. Максимальный размер символьного поля – 254 символа

Numeric (Числовой)	Включает цифры от 0 до 9, знак (необязательно), дробную часть (необязательно)
Float (Вещественный)	Числа, имеющие дробную часть(представляются в формате с плавающей точкой), определенные с обычной точностью
Double (Двойной Вещественный сдвойной точностью)	Числа, имеющие дробную часть(представляются в формате с плавающей точкой), определенные с двойной точностью
Integer (Числовой, целочисленный)	Целые числа
Data (Дата)	Дата, определяющая день, месяц и год. Ввод в это поле контролируется VisualFoxPro автоматически
DateTime (Дата и время)	Дата и время. По умолчанию формат вводимого значения имеет вид мм/дд/ггчч:мм:сс
Logical (Логический)	Логические данные. Может принимать только .T. – истина, .F. – ложь
Мемо (Текстовое поле произвольной длины)	Примечания, размер которых неограничен. Может содержать любую символьную информацию, размер которой ограничивается лишь объемом доступного дискового пространства. Фактически – это указатель на блок данных в файле с расширением fpt (файлы dbf и fpt одноименны). Для заполнения данного поля нужно дважды щелкнуть мышью по данному полю (или CTRL+Home), после чего в открывшемся окне набрать нужный текст примечаний. Закрыв окно редактирования (CTRL+W), значение поля будет сохранено, а слово мемо в поле базы данных изменится на Мемо.
General (Общие, двоичное поле произвольной длины)	Поле, используется для хранения ссылки на объект. Под объектом понимается любой текстовый файл, звуковой файл, диаграмма, иллюстрация или другой файл, созданный с помощью приложения, поддерживающего технологию OLE. Для заполнения данного поля нужно двойным щелчком мыши на нем открыть окно редактирования. После чего выбрать пункт меню Edit(правка)-insertobject(вставить объект) и в
	открытом выбранном приложении создать нужный объект (например, рисунок). Закрыв окно приложения и окно редактирования поля, поле gen будет заменено на Gen, значение будет сохранено
Currency (Денежный)	Поле для представления денежных сумм, используется для хранения чисел с четырьмя знаками после запятой
Memobinary	Двоичный текст произвольной длины. Содержит любые данные поля мемо, которые можно использовать, не изменяя кодовую страницу
Characterbinary	Двоичный текст длиной до 254 символов. Содержит любые символьные данные, которые используются, без изменения кодовой страницы

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование практической работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

Запустим Microsoft Visual FoxPro

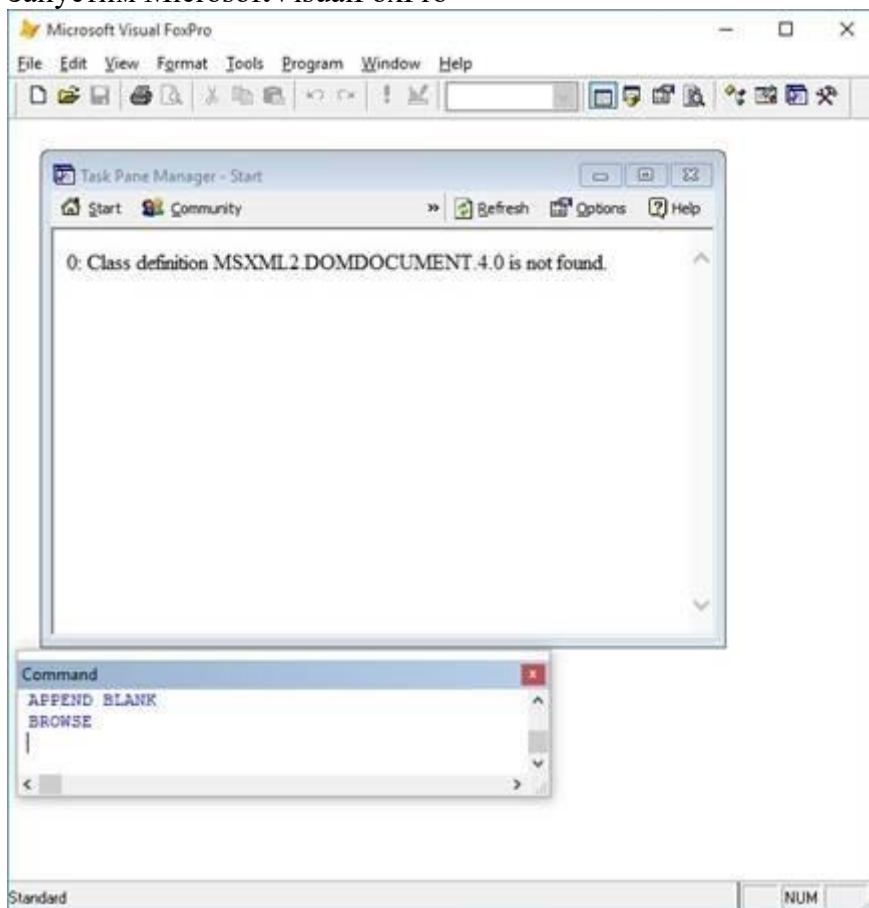


Рисунок 1 У нас появится 2 окна

1. Command – В этом окне мы будем вводить команды
 2. TaskPaneManager – окно с инструкцией Прежде чем начать работать в FoxPro надо ознакомиться с основными командами для создания “Базы данных”
 - Create<имя> — создать
 - Modifystructure – изменить структуру
 - Use< имя> — открыть подготовить
 - Select< имя> — выбрать
 - Browse – открыть визуализировать
 - Appendblank – добавить запись
- Начнём создания нашей первой Базы данных. Для этого напомним Create<имя> в окне команд.
У нас откроется окно структуры базы данны

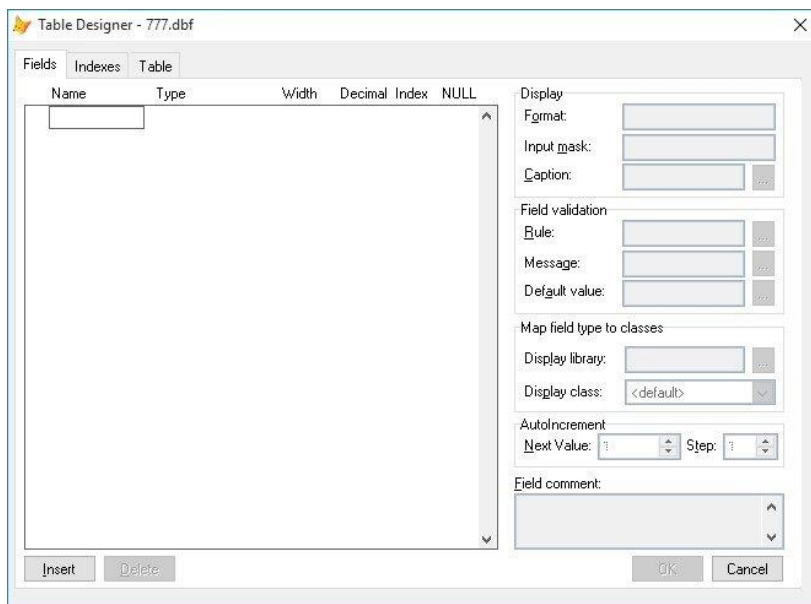


Рисунок 2

Name это название будущего столбца

Type это тип столбца

Width – ширина столбца

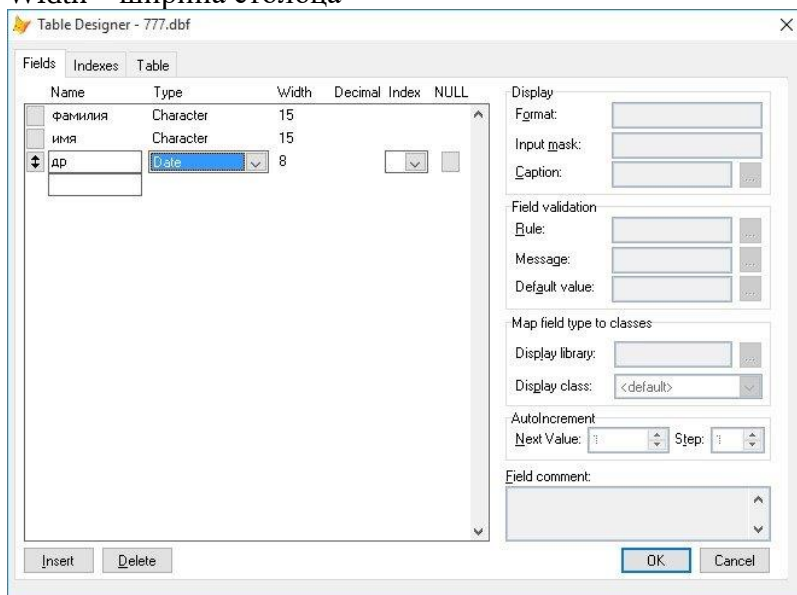


Рисунок 3

Основные типы полей, используемые в Microsoft Visual FoxPro приведены в таблице

Тип поля	Содержимое
Character (Символьный)	Любые символы. Максимальный размер символьного поля – 254 символа
Numeric (Числовой)	Включает цифры от 0 до 9, знак (необязательно), дробную часть (необязательно)
Float (Вещественный)	Числа, имеющие дробную часть (представляются в формате с плавающей точкой), определенные с обычной точностью

Double (Двойной Вещественный с двойной точностью)	Числа, имеющие дробную часть(представляются в формате с плавающей точкой), определенные с двойной точностью
Integer (Числовой, целочисленный)	Целые числа
Data (Дата)	Дата, определяющая день, месяц и год. Ввод в это поле контролируется VisualFoxPro автоматически
DateTime (Дата и время)	Дата и время. По умолчанию формат вводимого значения имеет вид мм/дд/ггчч:мм:сс
Logical (Логический)	Логические данные. Может принимать только .T. – истина, .F. – ложь
Мемо (Текстовое поле произвольной длины)	Примечания, размер которых неограничен. Может содержать любую символьную информацию, размер которой ограничивается лишь объемом доступного дискового пространства. Фактически – это указатель на блок данных в файле с расширением fpt (файлы dbf и fpt одноименны). Для заполнения данного поля нужно дважды щелкнуть мышью по данному полю (или CTRL+Home), после чего в открывшемся окне набрать нужный текст примечаний. Закрыв окно редактирования (CTRL+W), значение поля будет сохранено, а слово мемо в поле базы данных изменится на Мемо.
General (Общие, двоичное поле произвольной длины)	Поле, используется для хранения ссылки на объект. Под объектом понимается любой текстовый файл, звуковой файл, диаграмма, иллюстрация или другой файл, созданный с помощью приложения, поддерживающего технологию OLE. Для заполнения данного поля нужно двойным щелчком мыши на нем открыть окно редактирования. После чего выбрать пункт меню Edit(правка)-insertobject(вставить объект) и в открытом выбранном приложении создать нужный объект (например, рисунок). Закрыв окно приложения и окно редактирования поля, поле gen будет заменено на Gen, значение будет сохранено
Currency (Денежный)	Поле для представления денежных сумм, используется для хранения чисел с четырьмя знаками после запятой
Memobinary	Двоичный текст произвольной длины. Содержит любые данные поля мемо, которые можно использовать, не изменяя кодовую страницу
Characterbinary	Двоичный текст длиной до 254 символов. Содержит любые символьные данные, которые используются, без изменения кодовой страницы

Чтобы визуализировать нашу базу данных надо ввести Browse



Рисунок 4 Далее у нас откроется наша таблица но редактировать мы там нечего не можем. Для этого пишем команду Appendblankкаждыйраз когда хотим добавить строчку и опять повторяем команду Browse

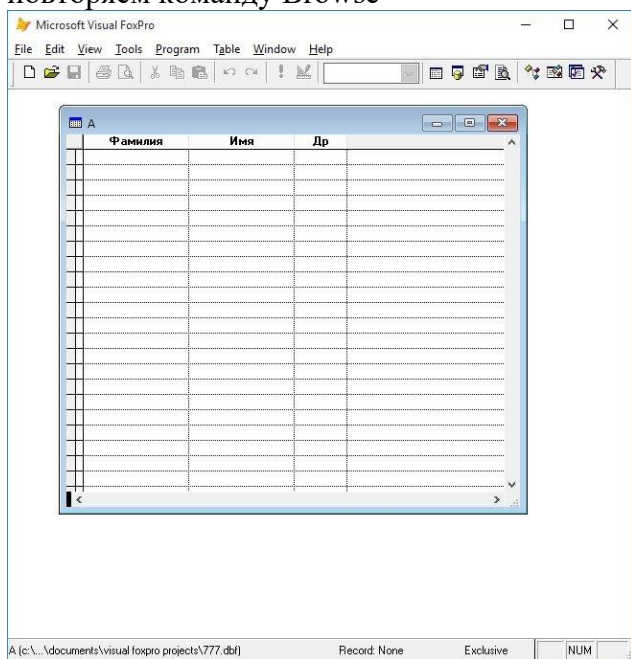


Рисунок 5



Рисунок 6

Всё готово теперь мы можем вносить данных в нашу базу данных. Важно – дата пишется мм/дд/гг

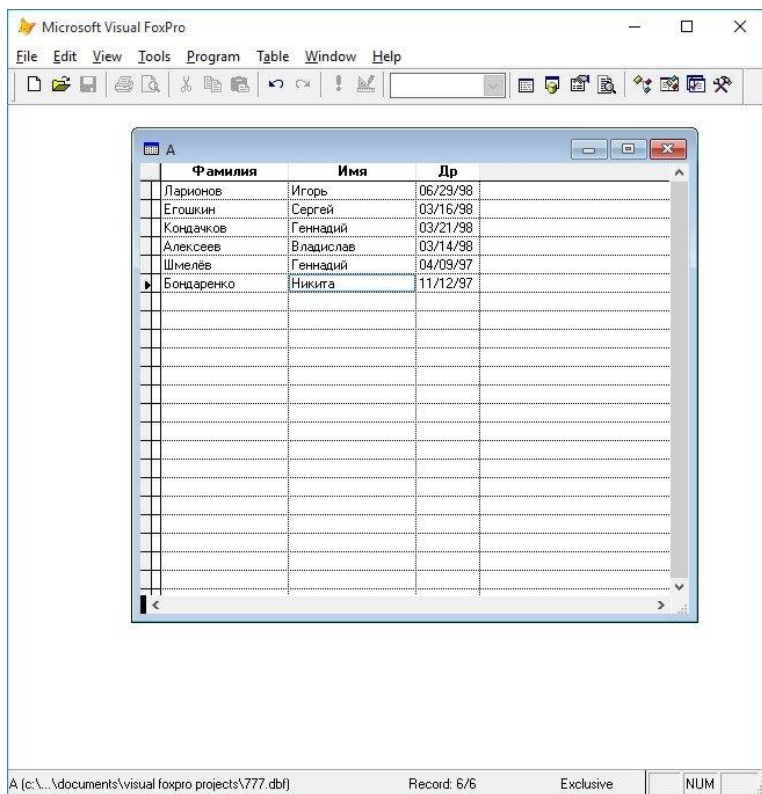


Рисунок 7

Контрольные вопросы:

1. Как создать таблицу в FoxPro?
2. Какие типы полей используются в FoxPro?

ОТЧЕТ

Сохраните работу.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Тема 3 Этапы проектирования баз данных

Лабораторное занятие №3

Тема: Нормализация реляционной БД, освоение принципов проектирования БД (2 часа)

Цель:

1. Научить обучающихся использовать офисные программы при оформлении документации;
2. Закрепить навыки работы в офисных программах;
3. Воспитать аккуратность и самостоятельность в работе.

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: компьютер

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

1. Создайте таблицу в соответствии с заданным образцом. Значения в выделенных полях

Отчет

Сохранить работу в своей папке под названием Лабораторная работа № 3

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №4

Тема: Преобразование реляционной БД в сущности и связи (2 часа)

Цель:

1. Преобразование реляционной БД

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: компьютер, Windows, MS Access

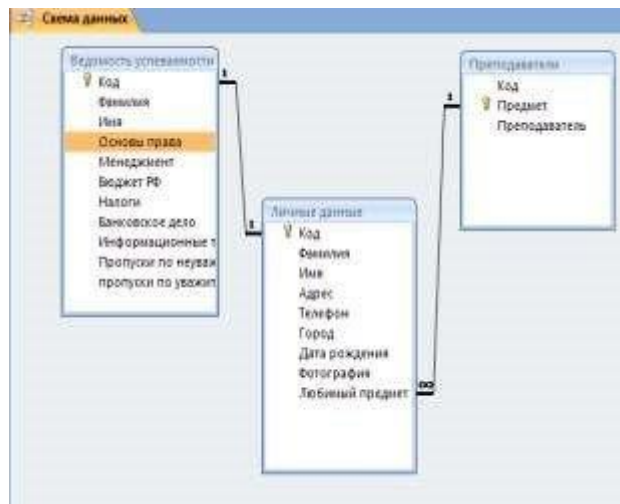
Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Краткие теоретические сведения:

Связи между таблицами

Для полноценной работы Базы данных нужно наладить связи между таблицами, сделать их зависимыми друг от друга, чтобы по данным одной таблицы можно было узнать данные другой таблицы.



Связь между таблицами осуществляется с помощью числового идентификатора. Взяв идентификатор из одной таблицы, можно найти данные в другой таблице. Для установки связи между двумя таблицами необходимо в группе **Показать или скрыть**, расположенной на вкладке **Работа с базами данных**, выбрать команду **Схема данных**. Появится окно **Схема данных**, в котором отображаются зависимости между таблицами.

Добавить новую связь можно, дважды щелкнув левой кнопкой мыши по окну или выбрав соответствующую команду

на вкладке **Работа с базами данных** → **Схема данных** → **Конструктор** → **Изменить связь**. Появится окно **Изменение связей**, в котором нужно указать таблицы между которыми устанавливаются связи.

Как же влияет установление связи в таблицах на работу MicrosoftAccess?



Программ контролирует ввод данных в связанные таблицы по следующим правилам:

- запрещено добавлять значение, отсутствующее в исходной таблице, в связанную таблицу. То есть, запрещен ввод нового любимого предмета в таблицу **Личные данные**, если в таблице **Преподаватели** этот предмет отсутствует;
- запрещено удаление в исходной таблице записей, на которые есть ссылка в связанной таблице;
- при изменении записей в исходной таблице

меняются записи в связанной таблице.

Связанные записи можно удалять только через исходную таблицу.

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:

- номер, наименование практической работы и тему;
- ответы на контрольные вопросы;
- выводы.

4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

СОЗДАНИЕ ЗАПРОСА НА ВЫБОРКУ

В этом упражнении мы создадим запрос на выборку жестких дисков, имевших емкость не менее 8 Гбайт при цене менее 150 условных единиц. Результирующая таблица должна содержать также адрес поставщика и номер его телефона.

1. Запустите программу Microsoft Access-2010 (*Пуск* → *Программы* → *Microsoft Access*).
2. В окне MicrosoftAccess включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке *ОК*.
3. В окне *Комплектующие: база данных* откройте панель *Запросы*. Дважды щелкните на значке *Создание запроса* в режиме *Конструктора* - откроется бланк запроса по образцу. Одновременно с ним откроется диалоговое окно *Добавление таблицы*.
4. В окне *Добавление таблицы* выберите таблицу *Поставщики* и щелкните на кнопке *Добавить*. Закройте окно *Добавление таблицы*.
5. В списке полей таблицы *Поставщики* выберите поля, включаемые в результирующую таблицу: *Компонент*, *Модель*, *Цена оптовая*, *Поставщик*, *Телефон*. Выбор производите двойными щелчками на именах полей.
6. Задайте условие отбора для поля *Компонент*, в соответствующую строку введите *Жесткий диск*. Из таблицы будут выбираться не все изделия, а только жесткие диски.
7. Задайте условие отбора для поля *Цена оптовая*. В соответствующую строку введите *< 150*. Из таблицы будут выбираться только изделия, имеющие цену менее 150 условных единиц.
8. Нам еще надо задать условие отбора по основному потребительскому параметру - емкости жесткого диска. Однако в таблице *Поставщики* такого поля нет. С другой стороны, в ней есть поле *Модель*, которое однозначно определяет параметры изделия. Благодаря тому, что по полю *Модель* у нас установлена связь с таблицей *Комплектующие*, мы получаем возможность ввести в запрос поле *Основной параметр*, взяв его из другой таблицы.

Добавьте список полей таблицы *Комплектующие* в верхнюю часть бланка запроса по образцу. Для этого щелкните правой кнопкой мыши в верхней области бланка и в

открывшемся контекстном меню выберите пункт *Добавить таблицу* - откроется уже знакомое нам окно *Добавление таблицы*. Выберите в нем таблицу *Комплектующие*.

14. Двойным щелчком на поле *Основной параметр* в списке полей таблицы *Комплектующие* введите это поле в бланк запроса по образцу.

15. В строке *Условие отбора* столбца *Основной параметр* введите условие >8 (емкость диска - более 8 Гбайт).

16. Закройте бланк запроса по образцу. При закрытии запроса введите его имя - *Выбор комплектующих*.

17. В окне *Комплектующие: база данных* откройте только что созданный запрос и рассмотрите результирующую таблицу. Ее содержательность зависит от того, что было введено в таблицы *Комплектующие* и *Поставщики* при их наполнении в упражнении 1. Если ни одно изделие не соответствует условию отбора и получившаяся результирующая таблица не имеет данных, откройте базовые таблицы и наполните их модельными данными, позволяющими проверить работу запроса.

18. По окончании исследований закройте все открытые объекты и завершите работу с программой MicrosoftAccess.

СОЗДАНИЕ ЗАПРОСОВ "С ПАРАМЕТРОМ"

Выше мы рассмотрели, как действует условие отбора, но должны отметить его существенный недостаток. Пользователь базы данных работает с запросами, которые ему подготовил разработчик. Если, например, разработчик предусмотрел запрос, отбирающий изделия, имеющие цену менее 150 условных единиц, то пользователь базы уже не в состоянии отобрать изделия, цена которых менее 140 условных единиц, поскольку у него нет соответствующего запроса.

Специальный тип запросов, называемый запросами "с параметром", позволяет пользователю самому ввести критерий отбора данных на этапе запуска запроса. Этот прием обеспечивает гибкость работы с базой.

Создадим простой запрос, позволяющий отбирать процессоры, предельную цену которых пользователь может задать сам при запуске запроса.

1. Запустите программу MicrosoftAccess - 2010 (*Пуск* → *Программы* → *Microsoft Access*).

2. В окне MicrosoftAccess включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке *ОК*.

3. В окне *Комплектующие: база данных* откройте панель *Запросы*. Дважды щелкните на значке *Создание запроса* в режиме *Конструктора* - откроется бланк запроса по образцу.

4. Согласно упражнению 3, создайте запрос на выборку, основанный на таблице *Поставщики*, в который войдут следующие поля:

- *Компонент*;
- *Модель*;
- *Цена оптовая*; • *Поставщик*; • *Телефон*.

7. В строке *Условие отбора* поля *Компонент* введите *Процессор*.

8. Строку *Условие отбора* для поля *Цена оптовая* надо заполнить таким образом, чтобы при запуске запроса пользователь получал предложение ввести нужное значение.

Текст, обращенный к пользователю, должен быть заключен в квадратные скобки. Если бы мы хотели отобрать процессоры, цена которых больше 100 условных единиц, мы бы написали >100 . Если бы нам были нужны процессоры дешевле 80 единиц, мы бы написали <80 . Но если мы хотим дать пользователю возможность выбора, мы должны написать $<[Введите максимальную цену]$.

11. Закройте запрос. При закрытии сохраните его под именем *Выбор комплектующих*.

12. В окне *Комплектующие: база данных* откройте панель *Запросы* и запустите запрос *Выбор комплектующих* - на экране появится диалоговое окно *Введите значение параметра*.

13. Введите какое-либо число и щелкните на кнопке *ОК*. В зависимости от того, что реально содержится в таблице *Поставщики*, по результатам запроса будет сформирована результирующая таблица.

14. Закройте все объекты базы данных. Закройте программу MicrosoftAccess. **Вывод.** Мы научились формировать запросы "с параметром" и узнали, что в основе этого вида

запросов лежат запросы на выборку, у которых в поле *Условие отбора* записан заключенный в квадратные скобки текст, обращенный к пользователю.

СОЗДАНИЕ ИТОГОВОГО ЗАПРОСА

Если полностью заполнить данными таблицу *Комплекующие*, введя параметры всех компонентов, входящих в сборочную спецификацию персонального компьютера, то можно узнать, во что обходится себестоимость комплекующих узлов. Запросы, выполняющие вычисления по всем записям для какого-либо числового поля, называются *итоговыми запросами*. В итоговом запросе может рассчитываться сумма значений или величина среднего значения по всем ячейкам поля, может выбираться максимальное или минимальное значение данных в поле или выполняться иная *итоговая функция*. Итоговые запросы, как и запросы на выборку, готовятся с помощью бланка запроса по образцу.

Предположим, что малое предприятие собирает компьютеры трех классов: "Элитный", "Деловой" и "Экономичный". Несмотря на то, что архитектура всех компьютеров близка, их компоненты заметно отличаются по цене и техническим параметрам. Соответственно, имеются значительные отличия в цене этих трех моделей, что важно для захвата разных секторов рынка. Наша задача - подготовить итоговый отчет, с помощью которого можно определять цену каждой из модели компьютеров и динамично ее отслеживать при изменении входящих компонентов или их поставщиков.

7. Запустите программу Microsoft Access-2010 (*Пуск → Программы → Microsoft Access*).

8. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплекующие* и щелкните на кнопке *ОК*.

9. В окне *Комплекующие: база данных* откройте панель *Таблицы*. Выберите таблицу *Комплекующие*.

10. Щелчком на значке *Конструктор* откройте таблицу в режиме проектирования - нам это необходимо для создания дополнительного поля *Класс*, в котором будут храниться данные о том, для какого класса изделий предназначены компоненты.

11. В начало структуры таблицы вставьте новое поле. Для этого выделите первое поле *Компонент* и нажмите клавишу *INSERT*.

12. Введите имя нового поля - *Класс* - и определите его тип - *Текстовый*.

9. Закройте окно *Конструктора*. При закрытии подтвердите необходимость изменить структуру таблицы.

10. Откройте таблицу *Комплекующие* и наполните ее содержанием, введя для каждого класса данные по следующим изделиям:

- материнская плата;
- процессор;
- оперативная память;
- жесткий диск;
- корпус;
- дисковод CD-ROM;
- дисковод гибких дисков;
- видеоадаптер;
- звуковая карта; • клавиатура; • мышь.

Цены на эти изделия для каждого класса проставьте произвольно. Прочие поля таблицы можно не заполнять - в формировании итогового запроса они участвовать не будут.

16. Закройте таблицу *Комплекующие*.

17. Откройте панель *Запросы* щелчком на одноименной кнопке окна *Комплекующие: база данных*.

18. Выполните двойной щелчок на значке *Создание запроса в режиме конструктора*. В открывшемся диалоговом окне *Добавление таблицы* выберите таблицу *Комплекующие*, на основе которой будет разрабатываться итоговый запрос. Закройте окно *Добавление таблицы*.

19. В бланк запроса по образцу введите следующие поля таблицы *Комплекующие*: *Класс*, *Компонент*, *Цена*.

20. Для поля *Класс* включите сортировку по возрастанию. Включите также сортировку по полю *Цена*, но на этот раз - по убыванию.

21. На панели инструментов MicrosoftAccess щелкните на кнопке *Групповые операции* или воспользуйтесь строкой меню (*Вид* → *Групповые операции*). Эта команда необходима для создания в нижней части бланка строки *Групповые операции*. Именно на ее базе и создаются итоговые вычисления. Все поля, отобранные для запроса, получают в этой строке значение *Группировка*.

22. Для поля, по которому производится группировка записей (в нашем случае - *Класс*), оставьте в строке *Групповые операции* значение *Группировка*. Для остальных полей щелкните на этой строке - появится кнопка раскрывающегося списка, из которого можно выбрать итоговую функцию для расчета значений в данном поле.

20. Для поля *Цена* выберите итоговую функцию *Sum* для определения стоимости изделия как суммы стоимостей комплектующих.

21. Для поля *Компонент* выберите итоговую функцию *Count*, определяющую общее количество записей, вошедших в группу. В нашем случае это количество узлов, из которых собран компьютер.

22. Закройте бланк запроса по образцу и дайте ему имя *Расчет стоимости изделия*. Запустите запрос и убедитесь, что он правильно работает.

23. Закройте все объекты базы данных. Завершите работу с программой MicrosoftAccess.

Контрольные вопросы:

1. Для чего необходима связь в таблицах?
2. Как создать запрос на выборку данных?
3. Как создать запрос с «параметром»?
4. Как создать итоговый запрос?

Отчет

Сохранить работу

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №5

Тема Проектирование реляционной БД. Нормализация таблиц (2 часа)

Цель:

1. Преобразование реляционной БД.

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: компьютер, операционная система Windows , MS Access

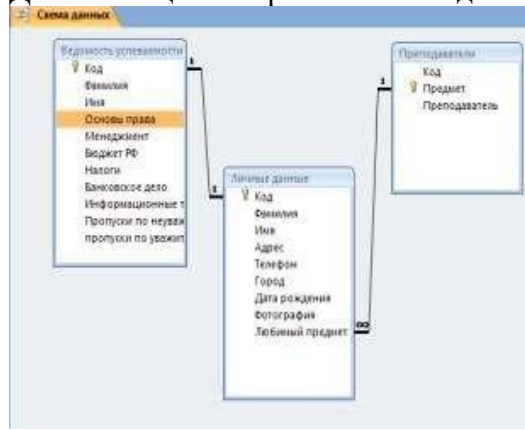
Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Краткие теоретические сведения:

Связи между таблицами

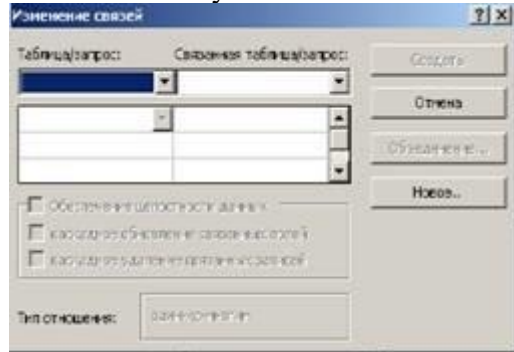
Для полноценной работы Базы данных нужно наладить связи между таблицами, сделать их зависимыми друг от друга, чтобы по данным одной таблицы можно было узнать данные другой таблицы.



Связь между таблицами осуществляется с помощью числового идентификатора. Взяв идентификатор из одной таблицы, можно найти данные в другой таблице. Для установки связи между двумя таблицами необходимо в группе **Показать или скрыть**, расположенной на вкладке **Работа с базами данных**, выбрать команду **Схема данных**. Появится окно **Схема данных**, в котором отображаются зависимости между таблицами.

Добавить новую связь можно, дважды щелкнув левой кнопкой мыши по окну или выбрав соответствующую команду на вкладке **Работа с базами данных** → **Схема данных** → **Конструктор** → **Изменить связь**. Появится окно **Изменение связей**, в котором нужно указать таблицы между которыми устанавливаются связи.

Как же влияет установление связи в таблицах на работу Microsoft Access?



Программа контролирует ввод данных в связанные таблицы по следующим правилам:

- запрещено добавлять значение, отсутствующее в исходной таблице, в связанную таблицу. То есть, запрещен ввод нового любимого предмета в таблицу **Личные данные**, если в таблице **Преподаватели** этот предмет отсутствует;
- запрещено удаление в исходной таблице записей, на которые есть ссылка в связанной таблице;
- при изменении записей в исходной таблице

меняются записи в связанной таблице.

Связанные записи можно удалять только через исходную таблицу.

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование практической работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

СОЗДАНИЕ ЗАПРОСА НА ВЫБОРКУ

В этом упражнении мы создадим запрос на выборку жестких дисков, имевших емкость не менее 8 Гбайт при цене менее 150 условных единиц. Результирующая таблица должна содержать также адрес поставщика и номер его телефона.

1. Запустите программу Microsoft Access-2010 (**Пуск** → **Программы** → **Microsoft Access**).
2. В окне Microsoft Access включите переключатель **Открыть базу данных**, выберите ранее созданную базу **Комплекующие** и щелкните на кнопке **ОК**.
3. В окне **Комплекующие: база данных** откройте панель **Запросы**. Дважды щелкните на значке **Создание запроса** в режиме **Конструктора** - откроется бланк запроса по образцу. Одновременно с ним откроется диалоговое окно **Добавление таблицы**.

4. В окне *Добавление таблицы* выберите таблицу *Поставщики* и щелкните на кнопке *Добавить*. Закройте окно *Добавление таблицы*.
5. В списке полей таблицы *Поставщики* выберите поля, включаемые в результирующую таблицу: *Компонент*, *Модель*, *Цена оптовая*, *Поставщик*, *Телефон*. Выбор производите двойными щелчками на именах полей.
6. Задайте условие отбора для поля *Компонент*, в соответствующую строку введите *Жесткий диск*. Из таблицы будут выбираться не все изделия, а только жесткие диски.
7. Задайте условие отбора для поля *Цена оптовая*. В соответствующую строку введите < 150 . Из таблицы будут выбираться только изделия, имеющие цену менее 150 условных единиц.
8. Нам еще надо задать условие отбора по основному потребительскому параметру - емкости жесткого диска. Однако в таблице *Поставщики* такого поля нет. С другой стороны, в ней есть поле *Модель*, которое однозначно определяет параметры изделия. Благодаря тому, что по полю *Модель* у нас установлена связь с таблицей *Комплектующие*, мы получаем возможность ввести в запрос поле *Основной параметр*, взяв его из другой таблицы. Добавьте список полей таблицы *Комплектующие* в верхнюю часть бланка запроса по образцу. Для этого щелкните правой кнопкой мыши в верхней области бланка и в открывшемся контекстном меню выберите пункт *Добавить таблицу* - откроется уже знакомое нам окно *Добавление таблицы*. Выберите в нем таблицу *Комплектующие*.
9. Двойным щелчком на поле *Основной параметр* в списке полей таблицы *Комплектующие* введите это поле в бланк запроса по образцу.
10. В строке *Условие отбора* столбца *Основной параметр* введите условие >8 (емкость диска - более 8 Гбайт).
11. Закройте бланк запроса по образцу. При закрытии запроса введите его имя - *Выбор комплектующих*.
12. В окне *Комплектующие: база данных* откройте только что созданный запрос и рассмотрите результирующую таблицу. Ее содержательность зависит от того, что было введено в таблицы *Комплектующие* и *Поставщики* при их наполнении в упражнении 1. Если ни одно изделие не соответствует условию отбора и получившаяся результирующая таблица не имеет данных, откройте базовые таблицы и наполните их модельными данными, позволяющими проверить работу запроса.
13. По окончании исследований закройте все открытые объекты и завершите работу с программой Microsoft Access.

СОЗДАНИЕ ЗАПРОСОВ "С ПАРАМЕТРОМ"

Выше мы рассмотрели, как действует условие отбора, но должны отметить его существенный недостаток. Пользователь базы данных работает с запросами, которые ему подготовил разработчик. Если, например, разработчик предусмотрел запрос, отбирающий изделия, имеющие цену менее 150 условных единиц, то пользователь базы уже не в состоянии отобрать изделия, цена которых менее 140 условных единиц, поскольку у него нет соответствующего запроса.

Специальный тип запросов, называемый запросами "с параметром", позволяет пользователю самому ввести критерий отбора данных на этапе запуска запроса. Этот прием обеспечивает гибкость работы с базой.

Создадим простой запрос, позволяющий отбирать процессоры, предельную цену которых пользователь может задать сам при запуске запроса.

1. Запустите программу Microsoft Access - 2010 (*Пуск* → *Программы* → *Microsoft Access*).
2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке *ОК*.
3. В окне *Комплектующие: база данных* откройте панель *Запросы*. Дважды щелкните на значке *Создание запроса* в режиме *Конструктора* - откроется бланк запроса по образцу.
4. Согласно упражнению 3, создайте запрос на выборку, основанный на таблице *Поставщики*, в который войдут следующие поля:
 - *Компонент*;
 - *Модель*;
 - *Цена оптовая*; • *Поставщик*; • *Телефон*.

5. В строке *Условие отбора* поля *Компонент* введите *Процессор*.
6. Строку *Условие отбора* для поля *Цена оптовая* надо заполнить таким образом, чтобы при запуске запроса пользователь получал предложение ввести нужное значение. Текст, обращенный к пользователю, должен быть заключен в квадратные скобки. Если бы мы хотели отобрать процессоры, цена которых больше 100 условных единиц, мы бы написали >100. Если бы нам были нужны процессоры дешевле 80 единиц, мы бы написали < 80. Но если мы хотим дать пользователю возможность выбора, мы должны написать < [Введите максимальную цену].
7. Закройте запрос. При закрытии сохраните его под именем *Выбор комплектующих*.
8. В окне *Комплектующие: база данных* откройте панель *Запросы* и запустите запрос *Выбор комплектующих* - на экране появится диалоговое окно *Введите значение параметра*.
9. Введите какое-либо число и щелкните на кнопке *ОК*. В зависимости от того, что реально содержится в таблице *Поставщики*, по результатам запроса будет сформирована результирующая таблица.
10. Закройте все объекты базы данных. Закройте программу Microsoft Access. **Вывод.** Мы научились формировать запросы "с параметром" и узнали, что в основе этого вида запросов лежат запросы на выборку, у которых в поле *Условие отбора* записан заключенный в квадратные скобки текст, обращенный к пользователю.

СОЗДАНИЕ ИТОГОВОГО ЗАПРОСА

Если полностью заполнить данными таблицу *Комплектующие*, введя параметры всех компонентов, входящих в сборочную спецификацию персонального компьютера, то можно узнать, во что обходится себестоимость комплектующих узлов. Запросы, выполняющие вычисления по всем записям для какого-либо числового поля, называются *итоговыми запросами*. В итоговом запросе может рассчитываться сумма значений или величина среднего значения по всем ячейкам поля, может выбираться максимальное или минимальное значение данных в поле или выполняться иная *итоговая функция*. Итоговые запросы, как и запросы на выборку, готовятся с помощью бланка запроса по образцу.

Предположим, что малое предприятие собирает компьютеры трех классов: "Элитный", "Деловой" и "Экономичный". Несмотря на то, что архитектура всех компьютеров близка, их компоненты заметно отличаются по цене и техническим параметрам. Соответственно, имеются значительные отличия в цене этих трех моделей, что важно для захвата разных секторов рынка. Наша задача - подготовить итоговый отчет, с помощью которого можно определять цену каждой из модели компьютеров и динамично ее отслеживать при изменении входящих компонентов или их поставщиков.

1. Запустите программу Microsoft Access-2010 (*Пуск → Программы → Microsoft Access*).
2. В окне Microsoft Access включите переключатель *Открыть базу данных*, выберите ранее созданную базу *Комплектующие* и щелкните на кнопке *ОК*.
3. В окне *Комплектующие: база данных* откройте панель *Таблицы*. Выберите таблицу *Комплектующие*.
4. Щелчком на значке *Конструктор* откройте таблицу в режиме проектирования - нам это необходимо для создания дополнительного поля *Класс*, в котором будут храниться данные о том, для какого класса изделий предназначены компоненты.
5. В начало структуры таблицы вставьте новое поле. Для этого выделите первое поле *Компонент* и нажмите клавишу *INSERT*.
6. Введите имя нового поля - *Класс* - и определите его тип - *Текстовый*.
7. Закройте окно *Конструктора*. При закрытии подтвердите необходимость изменить структуру таблицы.
8. Откройте таблицу *Комплектующие* и наполните ее содержанием, введя для каждого класса данные по следующим изделиям:
 - материнская плата;
 - процессор;
 - оперативная память;
 - жесткий диск;
 - корпус;

- дисковод CD-ROM;
- дисковод гибких дисков;
- видеоадаптер;
- звуковая карта; • клавиатура; • мышь.

Цены на эти изделия для каждого класса проставьте произвольно. Прочие поля таблицы можно не заполнять - в формировании итогового запроса они участвовать не будут.

9. Закройте таблицу *Комплектующие*.

10. Откройте панель *Запросы* щелчком на одноименной кнопке окна *Комплектующие: база данных*.

11. Выполните двойной щелчок на значке *Создание запроса в режиме конструктора*. В открывшемся диалоговом окне *Добавление таблицы* выберите таблицу *Комплектующие*, на основе которой будет разрабатываться итоговый запрос. Закройте окно *Добавление таблицы*.

12. В бланк запроса по образцу введите следующие поля таблицы *Комплектующие*: *Класс*, *Компонент*, *Цена*.

13. Для поля *Класс* включите сортировку по возрастанию. Включите также сортировку по полю *Цена*, но на этот раз - по убыванию.

14. На панели инструментов Microsoft Access щелкните на кнопке *Групповые операции* или воспользуйтесь строкой меню (*Вид → Групповые операции*). Эта команда необходима для создания в нижней части бланка строки *Групповые операции*. Именно на ее базе и создаются итоговые вычисления. Все поля, отобранные для запроса, получают в этой строке значение *Группировка*.

15. Для поля, по которому производится группировка записей (в нашем случае - *Класс*), оставьте в строке *Групповые операции* значение *Группировка*. Для остальных полей щелкните на этой строке - появится кнопка раскрывающегося списка, из которого можно выбрать итоговую функцию для расчета значений в данном поле.

16. Для поля *Цена* выберите итоговую функцию *Sum* для определения стоимости изделия как суммы стоимостей комплектующих.

17. Для поля *Компонент* выберите итоговую функцию *Count*, определяющую общее количество записей, вошедших в группу. В нашем случае это количество узлов, из которых собран компьютер.

18. Закройте бланк запроса по образцу и дайте ему имя *Расчет стоимости изделия*. Запустите запрос и убедитесь, что он правильно работает.

19. Закройте все объекты базы данных. Завершите работу с программой Microsoft Access.

Контрольные вопросы:

1. Для чего необходима связь в таблицах?
2. Как создать запрос на выборку данных?
3. Как создать запрос с «параметром»? 4. Как создать итоговый запрос?

Отчет

Сохранить работу

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10	Защита своей работы персонально – 10 баллов	Менее 20 баллов – оценка «2»

	баллов		
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №6

Тема:Проведение сортировки и фильтрации данных. Поиск данных по одному и нескольким полям. Поиск данных в таблице(2 часа)

Цель занятия:

1. Получить некоторые практические навыки по созданию меню в программе VisualFoxPro.

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: Компьютер, операционная система Windows, FoxPro

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Команды для осуществления сортировки:

Sortto<имя новой БД>on<имя поля> — сортирует и создаёт новую базу данных с уже отсортированными данными

Ключи команды: [/A | /D] [/C]

[/A] — задает возрастающий порядок для данного поля

[/D] — убывающий порядок

SET DATE — Задает формат отображаемых выражений типа даты и DateTime.

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование практической работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

Сортировка базы данных Задание:

1. Отсортировать базу данных

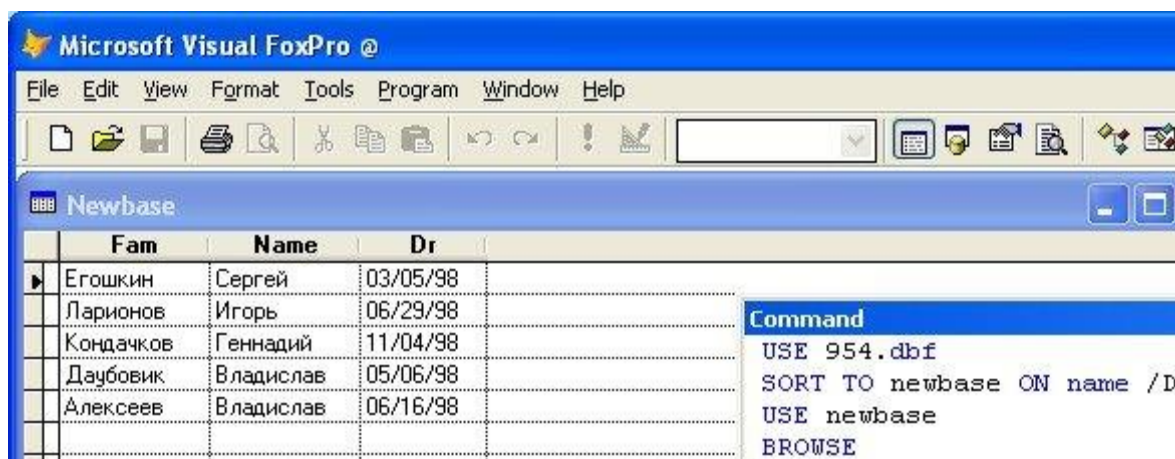
1. Вызываем базу которую мы будем сортировать

```
Command
USE 954.dbf
```

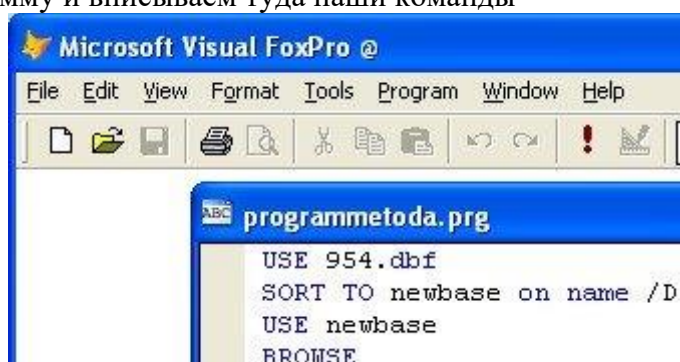
2. Сортируем базу 954.dbf по параметру “name” на убывание при этом создаём новую базу под названием “newbase”

```
Command
USE 954.dbf
SORT TO newbase ON name /D
```

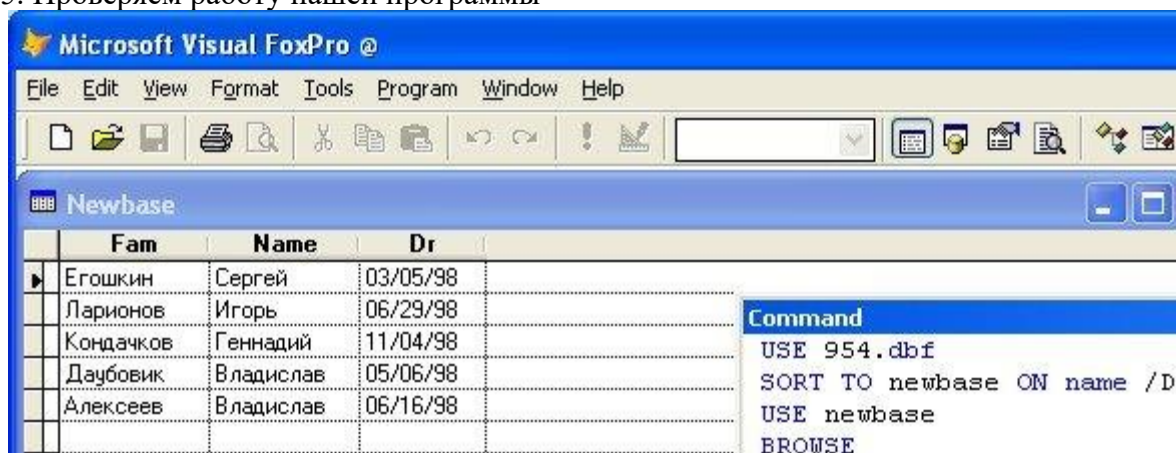
3. Вызываем базу созданную нами и проверяем сортировку



Как мы видим поле “name” прошло сортировку в порядке убывания 4. Теперь создаём программу и вписываем туда наши команды



5. Проверяем работу нашей программы



Чтобы проверить работоспособность программы нужно открыть пункт меню Program в нём выбрать Do... далее выбираем нашу программу и пишем ACTIVATE MENU <Название меню>

Создание поиска

Запускаем программу меню:

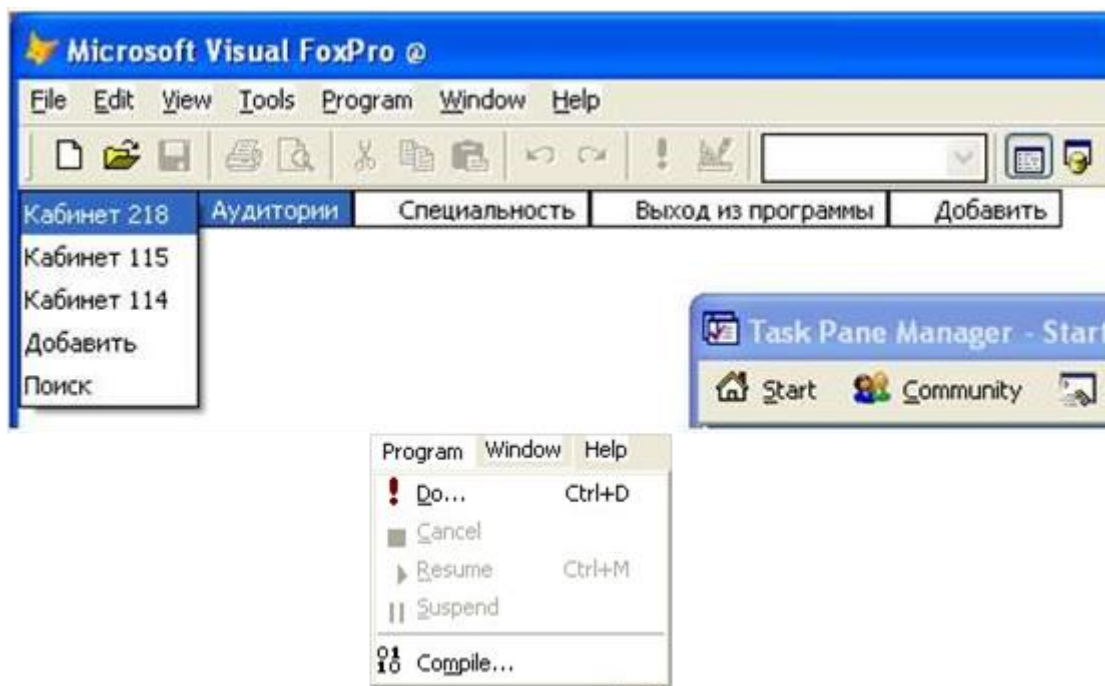


Рис.1(Наше меню.)

Далее нажимаем на кнопку «Поиск» дабы нам необходимо запустить нашу программу:

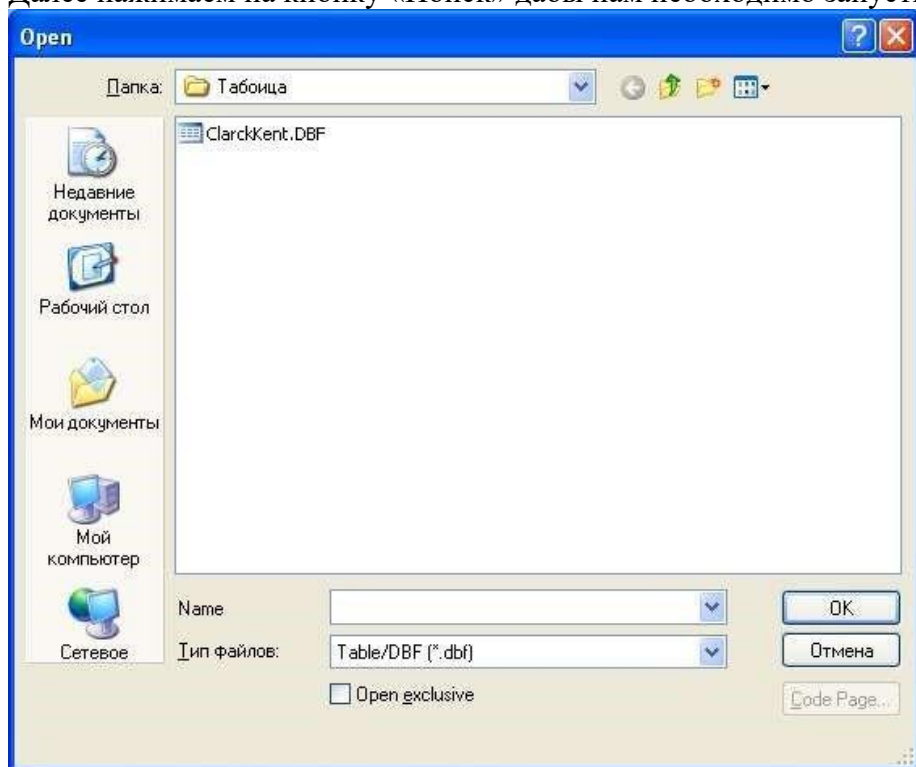


Рис.2(Открывается поиск нашей таблицы.) Выбираем.

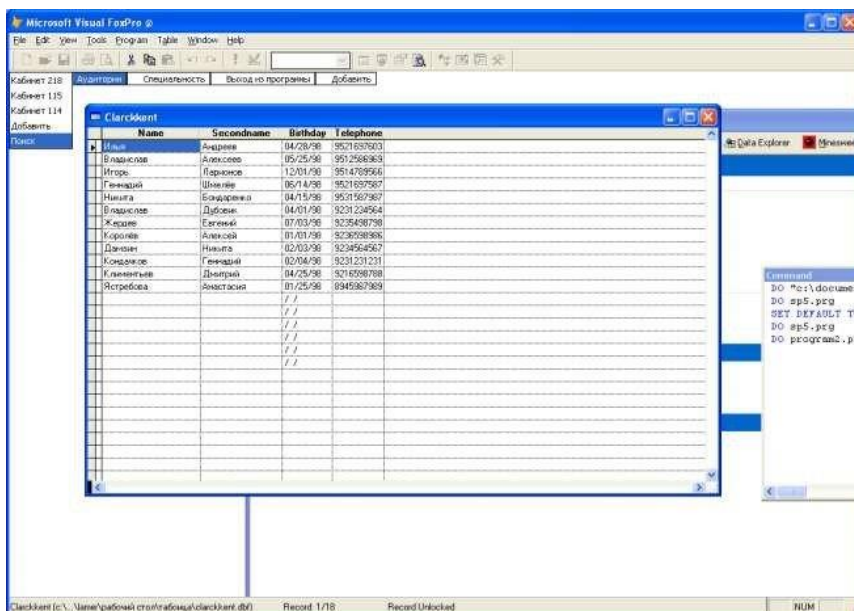


Рис.3(Открылась таблица.)
Заходим в пункт меню «Поиск»

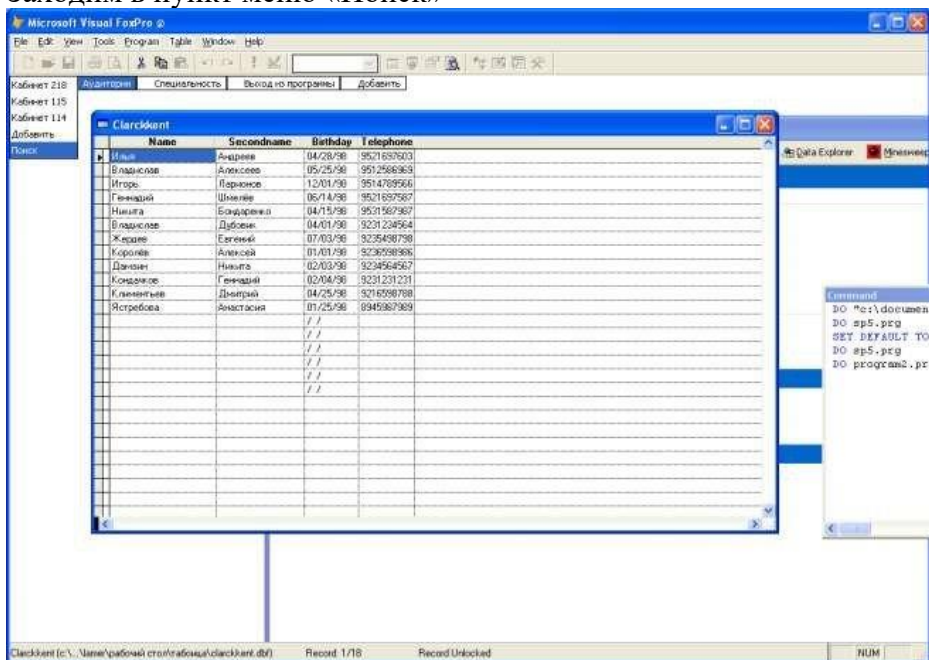


Рис.4(Ну и соответственно видим нашу выполненную работу.) И немного расскажем, как мы это сделали:

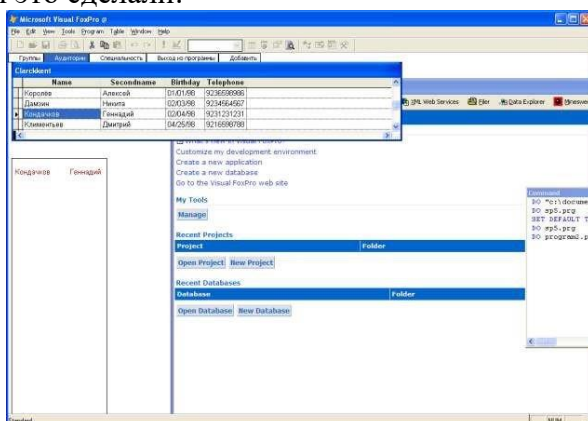


Рис.5(Добавление пункта)

Прописываем в нашем меню пункт «Поиск» чтобы он обращался к нашей программе, которая будет выполнять поиск.

Ну, и конечно же, наша сама программа выполняющая поиск:

```

define popup B
define bar 1 of B prompt "кабинет 218"
define bar 2 of B prompt "кабинет 115"
define bar 3 of B prompt "кабинет 114"
define bar 4 of B prompt "добавить"
define bar 5 of B prompt "Поиск"
ON SELECTION PAD f2 OF Andreev activate popup B
on selection bar 1 of B do sp1
on selection bar 2 of B do sp1
on selection bar 3 of B do sp2
on selection bar 4 of B do sp4
on selection bar 5 of B do sp5

```

Рис.6(Программа)

Контрольные вопросы:

1. Для чего предназначена сортировка?
2. Как осуществляется сортировка в FoxPro? 3. Как осуществить поиск в БД?

ОТЧЕТ

Сохранить работу.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10 баллов	Защита своей работы персонально – 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Тема 4 Проектирование структур баз данных

Лабораторное занятие №7

Тема:Работа с переменными. Написание программного файла и работа с табличными файлами. Заполнение массива из табличного файла. Заполнение табличного файла из массива. (2 часа)

Цель занятия:

1. Преобразование реляционной БД.

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: омпьютер, операционная система Windows, FoxPro

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:

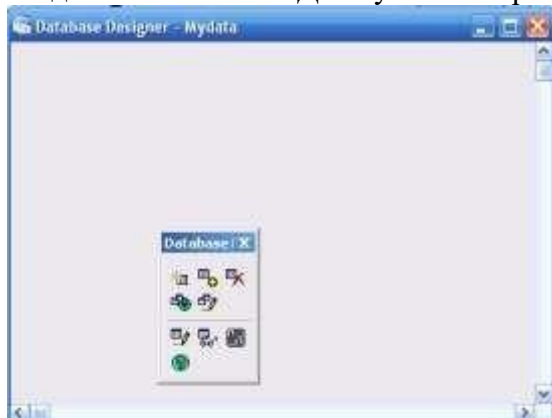
- номер, наименование практической работы
 - и тему;
 - ответы на контрольные вопросы; • выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

«Создание базы данных и заполнение таблиц»

1. Разработка структуры базы данных.

Для создания новой БД выберите пункт меню **File/New**, в окне **New** установите флажок **Database** и нажмите кнопку **NewFile**. Для сохранения БД укажите путь к своей папке и введите имя новой БД - Mydata. Откроется окно Дизайнера БД.



Для работы с базой данных используется панель инструментов Дизайнера БД.
Назначение кнопок:

- создать новую таблицу
- добавить существующую таблицу
- удалить таблицу из БД
- создать новое удаленное представление
- создать новое локальное представление
- изменить структуру и свойства таблицы
- просмотр таблицы
- редактирование хранимых процедур
- создание удаленной БД

2. Разработка структур таблиц базы данных.

Создайте в БД первую таблицу (Типы книг) с именем **tip**.

Поля таблицы **tip**

Имя поля	Тип поля, размер	Описание (св-во Caption)
Kodt	Integer (AutoInc)	Код типа
Name	Character, 30	Наименование типа

Создайте вторую таблицу (Книги) с именем **knig**.

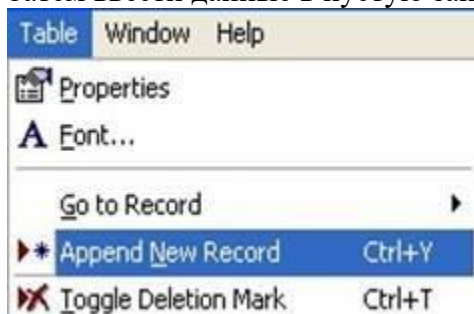
Поля таблицы **knig**

Имя поля	Тип поля, размер	Описание (св-во Caption)
Kodt	integer	Код типа
Kodkn	Integer (AutoInc)	Код книги
Autor	Character, 30	Автор
Nazvanie	Character, 40	Название
Cena	Numeric, 10,2	Цена

Kol	Integer	Количество
-----	---------	------------

3. *Ввод данных в таблицы.*

Для добавления записей необходимо выбрать нужную таблицу и нажать на кнопку **Browse**. Чтобы добавить пустую запись, нужно выбрать в меню **Table** команду **AppendNewRecord**. Затем ввести данные в пустую запись.



Второй способ – набрать команду **AppendBlank** в командном окне.

Введите в таблицу **Tip** следующие данные:

Kodt	Name
1	Художественная
2	Научная
3	Журналы
4	Газеты
5	Публицистическая

Введите в таблицу **knig** следующие данные:

Kodt	Kodkn	Autor	Nazvanie	Cena	Kol
1	1	Пушкин А.С.	Евгений Онегин	150.00	2
1	2	Есенин С.А.	Лирика	95.00	7
3	3		Мир ПК №1	50.00	5
3	4		Мир ПК №2	55.00	5
5	5	Герхард С.	Мир фантастики	100.00	4
2	6	Крылов Е.В.	Мы и закон	90.00	3
4	7		168 часов № 55	15.00	5
2	8	Иванов А.М.	Молекулярная физика	125.00	1
1	9	Кузнецов А.А.	Преступление века	199.00	2
5	10	Розенберг Е.С.	Закон добра	255.00	3
2	11	Прохоров В. И.	Жизнь	96.00	10
2	12	Лейман А.П.	Математический анализ	180.00	12
4	13		Комсомольская правда	25.00	2

4. Покажите результат преподавателю.

5. Завершите работу приложения. 4. *Выполнение самостоятельного задания.*

По аналогии с выполненным заданием создайте базу данных предложенного вам варианта. Заполните созданные таблицы контрольными данными (введите новые записи) - в главной таблице не менее 5 записей, в дочерней – не менее 10.

«Ввод, редактирование и удаление данных» Ввод записей в таблицу.

Откройте в своей папке созданную ранее БД Mydata, которая содержит две таблицы – **Tip** и **Knig**.

Добавьте в таблицу **Tip** новую запись – поле **kodt=6**, поле **name= 'Учебники'**. Откройте таблицу **Knig**. Добавьте в нее дополнительно две произвольные записи по типу книг Учебники (**kodt=6**).

1. *Редактирование данных в таблице.*

Перейдите в таблицу **Tip**. Выполните в командном окне команду:

REPLACE name WITH 'Научно-фантастическая' FOR name='Научная' *Просмотрите результат замены данных в таблице – команда BROWSE.*

Аналогично замените данные 'Учебники' на 'Учебники и задачники'. **Выполненную команду и результат скопируйте и вставьте в файл отчета.**

Перейдите в таблицу **Knig**. Выполните в командном окне следующие команды по замене данных:

- заменить поле nazvanie значением 'Молекулярная химия', в записи со значением этого поля 'Химия';
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- заменить поле kol значением 3 у всех записей со значением 2;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- заменить поле cena значением 30.00 у всех записей со значением этого поля, меньшим 30;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- заменить поле cena значением 20.00 у записи со значением поля nazvanie ='168 часов № 55';
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет).**

2. Удаление данных в таблице.

Перейдите в таблицу **Knig**. Выполните в командном окне команду:

DELETE ALL FOR kol<4

Проверьте результат выполнения команды, просмотрев данные; (дополните отчет).
Сколько записей было удалено с помощью этой команды?

Выполните команду:

RECALL all

Проверьте результат выполнения команды, просмотрев данные; (дополните отчет).
Что произошло в ранее удаленными записями?

Аналогичным образом выполните следующие команды по удалению данных:

- удалить все записи, у которых значение поля kodt равно 1;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Восстановите удаленные записи;
- удалить все записи, у которых значение поля kol равно 5;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Восстановите удаленные записи;
- удалить все записи, у которых значение поля cena больше 100;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Восстановите удаленные записи;
- удалить все записи, у которых значение поля kol больше 8 и значение поля cena меньше 100;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- Восстановите все удаленные записи.

3. Вывод данных на экран дисплея.

Перейдите в таблицу **Knig**. Установите указатель на запись, у которой поле kodkn=6.

Выполните в командном окне команду: DISPLAY

Проверьте результат выполнения команды, просмотрев данные; (дополните отчет).

Выполните в командном окне команду: DISPLAY ALL

Проверьте результат выполнения команды, просмотрев данные; (дополните отчет).

Используя аналогичным образом команды вывода данных, выполните следующие действия:

- Вывести на экран поля autor и nazvanie всех записей;
 - **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
 - Вывести на экран поля autor, nazvanie и kol всех записей, у которых количество больше 9;
 - **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
 - Вывести на экран поля autor, nazvanie и cena всех записей, у которых цена меньше 120;
 - **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
 - Вывести на экран поля autor, nazvanie, cena и kol всех записей, у которых поле kodt=6;
 - **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
 - Вывести на экран структуру текущей таблицы;
 - **проверьте результат выполнения команды, просмотрев данные; (дополните отчет).**
- Выполните аналогичные действия, используя команду LIST.**

4. Перемещение по записям таблицы.

Перейдите в таблицу **Knig**. Используя команды перемещения, выполните следующие действия:

- переместите с помощью команды указатель на последнюю запись в таблице; **• проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- **проверьте значение функции eof() (для вывода значения на экран выполните команду ? eof()); (дополните отчет);**
- переместите с помощью команды указатель на первую запись в таблице;
- **проверьте результат выполнения команды, просмотрев данные;**
- **проверьте значение функции bof() (для вывода значения на экран выполните команду ? bof()); (дополните отчет);**
- переместите с помощью команды указатель на запись с номером 5;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- переместите с помощью команды указатель на предыдущую запись;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- переместите с помощью команды указатель на 3 записи вниз;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет);**
- переместите с помощью команды указатель на 4 записи вверх;
- **проверьте результат выполнения команды, просмотрев данные; (дополните отчет).**

5. Покажите файл с результатами преподавателю.

6. Выполнение самостоятельного задания.

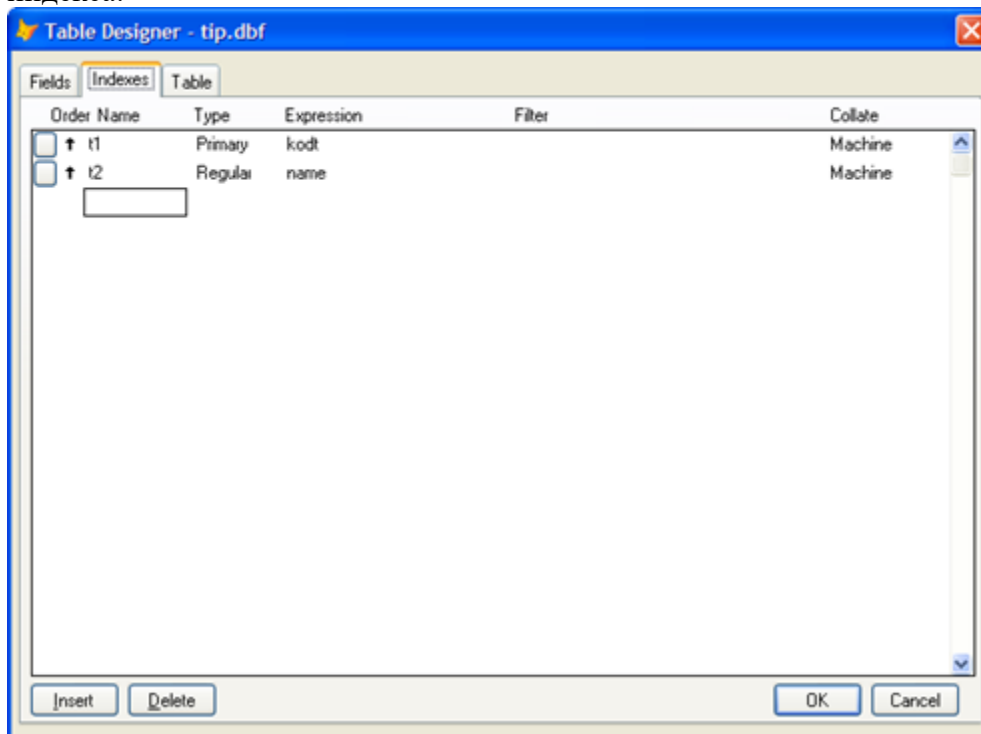
Откройте базу данных вашего варианта, созданную ранее. Добавьте в таблицы недостающие данные.

Выполните в командном окне тренировочные команды по замене данных, удалению и восстановлению записей, выводу записей на экран, перемещению по записям таблицы.

«Создание индексов и определение отношений между таблицами»

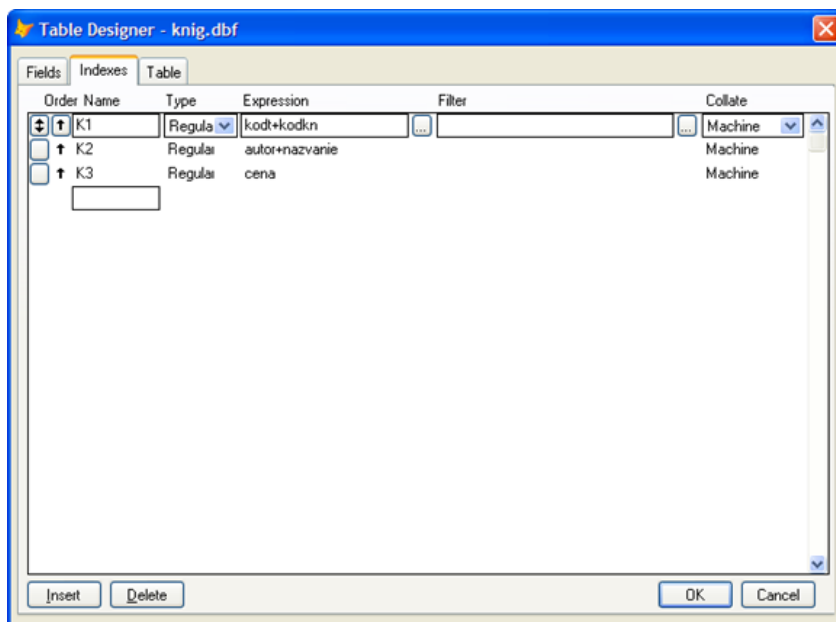
1. Создание индексов с помощью конструктора.

Откройте в своей папке созданную ранее БД Mydata. Для таблицы Tip откройте конструктор таблиц и в окне конструктора перейдите на вкладку Indexes. Создайте для данной таблицы два индекса:



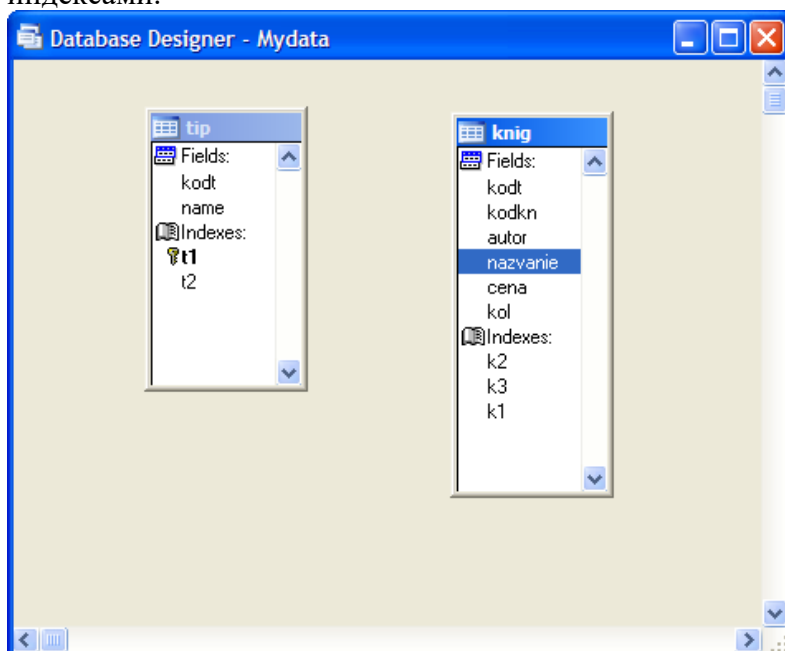
Сохраните внесенные изменения. Обратите внимание, что созданные индексы появились в макете таблицы в конструкторе БД.

Аналогичным образом создайте индексы для таблицы Knig:

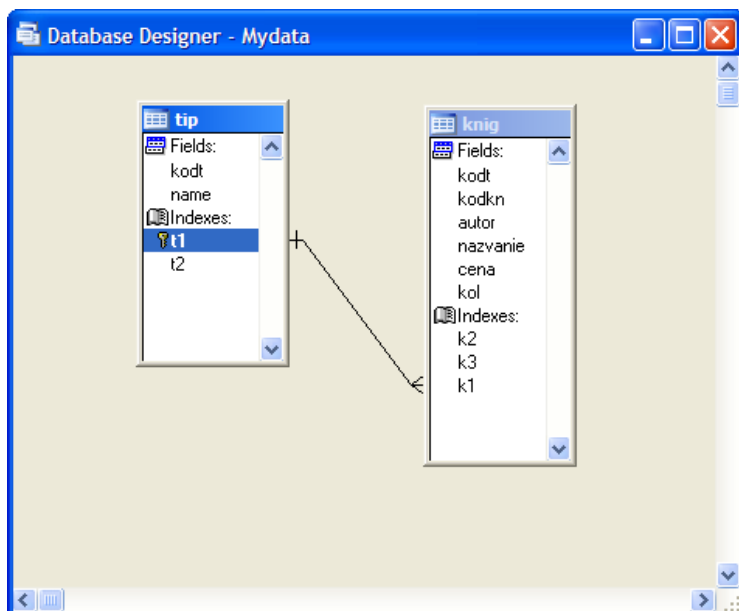


2. Установление отношений между таблицами.

Перейдите в конструктор БД, в котором находятся таблицы Tip и Knig, с созданными индексами.



Установите долговременную связь между таблицами, перетаскив ключевой индекс главной таблицы на соответствующий индекс дочерней.



3. Контроль целостности данных

Для созданной связи выберите пункт EditReferentialIntegrity контекстного меню и установите каскадные правила обновления и каскадные правила удаления, а также правила вставки Restrict.

4. Выполнение самостоятельного задания

Откройте базу данных вашего варианта, созданную ранее. Создайте для каждой таблицы необходимые индексы. Для главной таблицы должен быть обязательно создан первичный (ключевой) индекс.

Установите долговременные отношения между таблицами, используя созданные индексы. Осуществите контроль целостности данных.

5. Покажите результат преподавателю.

«Использование индексов»

1. Создание дополнительных индексов.

Откройте в своей папке созданную ранее БД Mydata. Для таблицы Knig создайте дополнительные индексы:

- Индекс k4 - по полю kodkn;
- Индекс k5 - по полю kol;
- Индекс k6 - по полям autor и kol; 2. **Выбор активного индекса в процессе выполнения.**

Откройте таблицу Knig. Выполните команды, которые по очереди подключают все индексы (setorderto ...).

После подключения очередного индекса, просмотрите результат и добавьте команду и результат в отчет.

3. Обновление индексов.

Откройте таблицу Knig в монопольном режиме. Выполните для данной таблицы переиндексирование.

Аналогично выполните обновление индексов для таблицы Tip. 4. **Выполнение самостоятельного задания.**

Откройте базу данных вашего варианта, созданную ранее. Создайте для каждой таблицы дополнительные индексы по разным полям.

Выполните подключение индексов в процессе работы и просмотрите результаты.

5. Покажите файл с результатами преподавателю.

Контрольные вопросы:

1. Что такое индекс?
2. Опишите этапы создания БД?
3. Перечислите основные элементы БД?

Отчет

Сохраните работу.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10 баллов	Защита своей работы персонально – 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №8

Тема:Добавление записей в табличный файл из двумерного массива. Работа с командами ввода-вывода. Использование функций для работы с массивами**(2 часа)**

Цель занятия:

1. Создать, отредактировать запись в таблице

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: омпьютер, операционная система Windows, FoxPro

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование практической работы и тему; • ответы на контрольные вопросы;
- выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

Добавление записи в таблицу через команды

1. Нам необходимо в первую очередь, конечно же открыть нашу Базу Данных.
 2. Далее нам необходимо создать программу, через которую, будет все работать.
 3. Далее необходимо наше «Меню» которое мы сделали в «3 работе по FoxPro». Оно нам пригодится для того, чтобы добавить новый пункт, который нужен в нашей практической работе.
 4. Подготовить отчет.
 5. Написать вывод о проделанной работе.
- Для начала нам нужно открыть нашу Базу Данных:
Для этого вам необходимо написать команду в окне «Commands» команду «Browse» и нажать Enter:

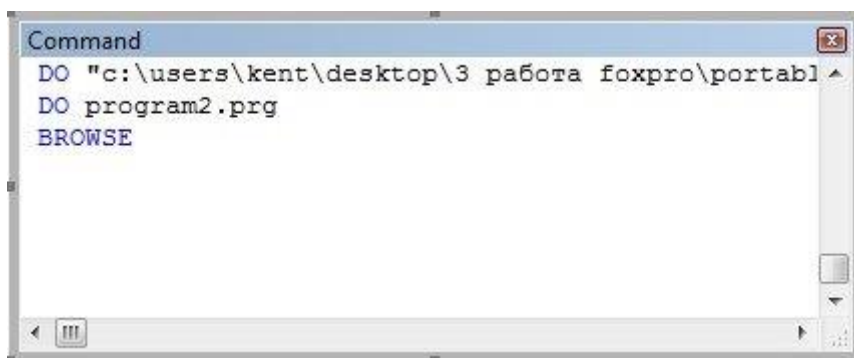


Рис.1(Команда Browse)

Далее у вас откроется окно, для выбора вашей Базы Данных:

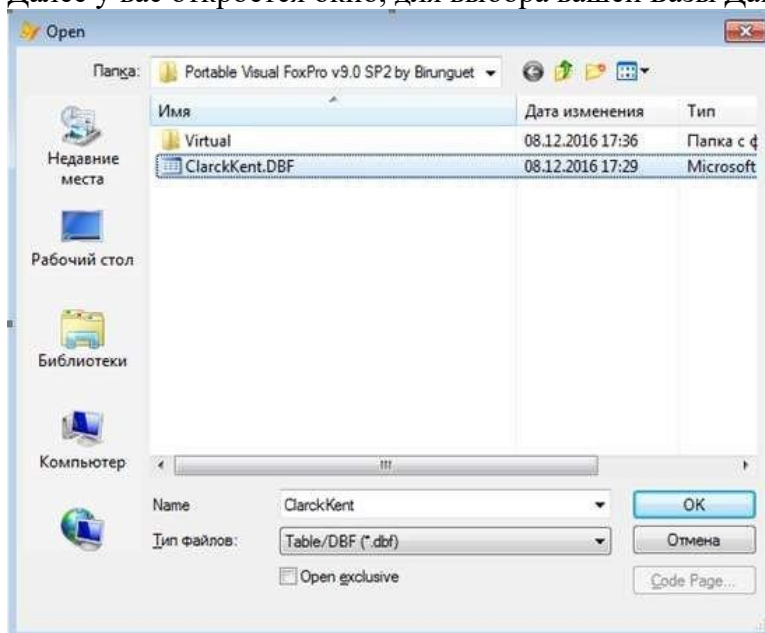


Рис.2(Выполнение команды «Browse»)

Нажимаем «ОК» и видим нашу таблицу с нашими данными:

Clarckkent				
	Name	Secondname	Birthday	Telephone
▶	Илья	Андреев	04/28/98	9521697603
	Владислав	Алексеев	05/25/98	9512586969
	Игорь	Ларионов	12/01/98	9514789566
	Геннадий	Шмелёв	06/14/98	9521697587
	Никита	Бондаренко	04/15/98	9531587987
	Владислав	Дубовик	04/01/98	9231234564
	Жердев	Евгений	07/03/98	9235498798
	Королёв	Алексей	01/01/98	9236598986
	Дамзин	Никита	02/03/98	9234564567
	Кондачков	Геннадий	02/04/98	9231231231
	Климентьев	Дмитрий	04/25/98	9216598788
	Ястребова	Анастасия	01/25/98	8945987989

Рис.3(Наша таблица)

Чтобы создать новый файл, необходимо: New -> Programm -> New file:



Рис.4(Создание)

После того, как мы создали программу, необходимо вписать эти параметры, у меня она будет называться «sp4».

Команда USE у каждого по-своему, она выбирает ваш путь, к вашей же таблице:

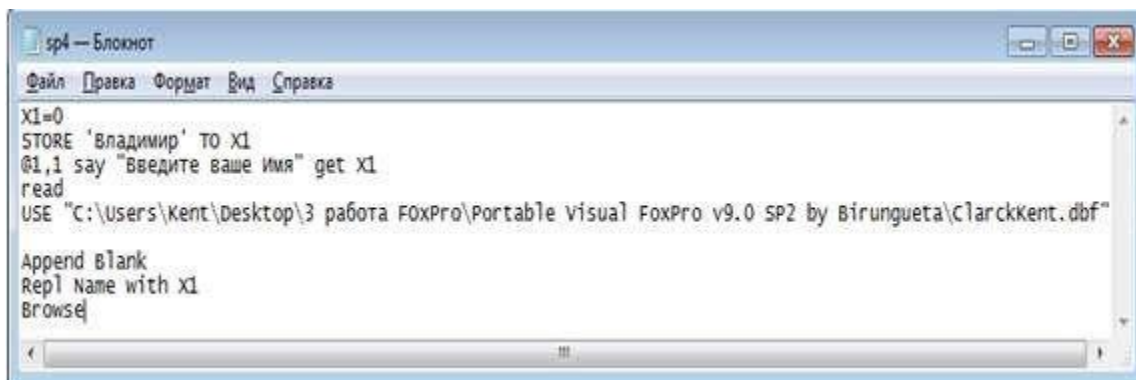


Рис.5(Programm)

Но после этого, не запускайте ее, потому что мы не определили ее в нашем меню.

Для того, чтобы добавить эту программу в само меню, вам необходимо проделать несколько действий:

Для начала зайти в программу, где находится сама меню.

Давайте добавим новый пункт меню, который будет носить имя «Добавить».

Сначала впишем ее, чтобы она отображалась: Я добавлю ее в выпадающую кнопку «Аудитории»:

```
define bar 1 of В prompt "кабинет 218"
define bar 2 of В prompt "кабинет 115"
define bar 3 of В prompt "кабинет 114"
define bar 4 of В prompt "Добавить"
ON SELECTION PAD f2 OF Andreev activate popup В |
```

Рис.6(Кнопка «Добавить»)

Теперь сделаем, чтобы наша кнопка «Добавить» обращалась к нашей программе «sp4»:

```

on selection bar 1 of B do sp1
on selection bar 2 of B do sp1
on selection bar 3 of B do sp2
on selection bar 4 of B do sp4

```

Рис.7(Добавление программы для работоспособности)
И конечно же, проверим работоспособность, для этого:

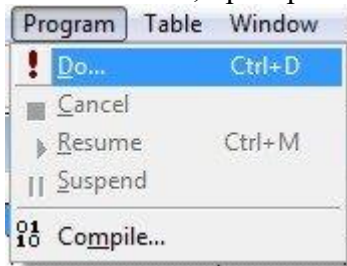


Рис.8(Проверка)

После откроется окно, для того, чтобы вы выбрали свою программу:

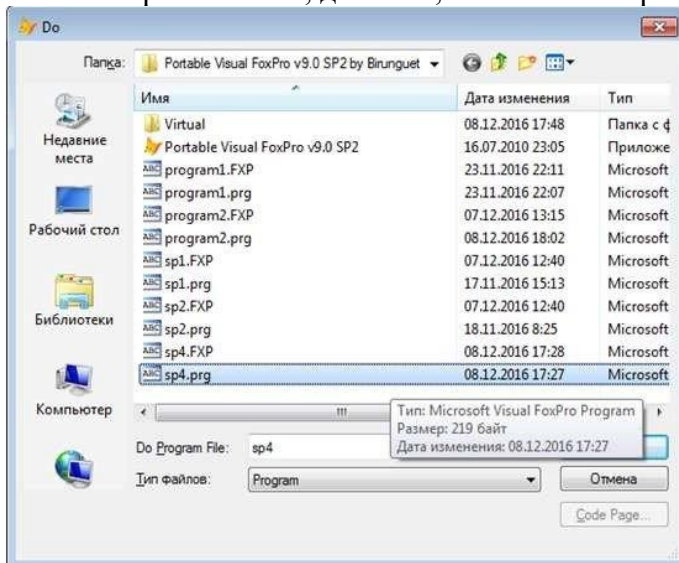


Рис.9(Выбор программы)

После выбора видим:

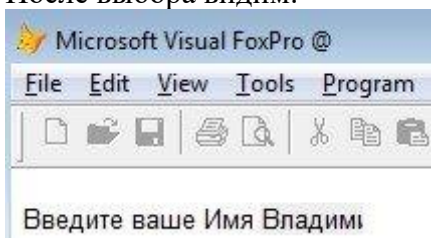


Рис.10(Выполнение программы)

Вам надо стереть то, что написано в поле, и ввести то, что нужно вам.

Ввели, и смотрим, что получилось:

Clarckkent				
	Name	Secondname	Birthday	Telephone
	Илья	Андреев	04/28/98	9521697603
	Владислав	Алексеев	05/25/98	9512586969
	Игорь	Ларионов	12/01/98	9514789566
	Геннадий	Шмелёв	06/14/98	9521697587
	Никита	Бондаренко	04/15/98	9531587987
	Владислав	Дубовик	04/01/98	9231234564
	Жердев	Евгений	07/03/98	9235498798
	Королёв	Алексей	01/01/98	9236598986
	Дамзин	Никита	02/03/98	9234564567
	Кондачков	Геннадий	02/04/98	9231231231
	Климентьев	Дмитрий	04/25/98	9216598788
	Ястребова	Анастасия	01/25/98	8945987989
	Владимир		/ /	

Рис.11(Проверка работоспособности)

Контрольные вопросы:

1. Что такое запись в таблице?
2. Как можно добавить запись в таблицу?
3. Какие параметры можно настроить для записи?

Отчет

Сохраните работу.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10 баллов	Защита своей работы персонально – 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №9

Тема:Создание меню различных видов. Модификация и управление меню(2 часа)

Цель занятия:

1. Получить некоторые практические навыки по созданию меню в программе VisualFoxPro

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: компьютер, выход в Интернет

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

раткие теоретические сведения: Команды для создания меню:

DefineMenu<имя меню>color r/w – Создание меню

DEFINE PAD <имя пункта> OF <имя меню>PROMPT «название» – Создание горизонтального пункта

ON PAD <имя пункта> OF <имя меню> ACTIVATE POPUP <имя меню раскрывающегося> – Активация горизонтального пункта

DEFINE POPUP <имя меню раскрывающегося> – Создание раскрывающегося пункта

DEFINE BAR 1 OF <имя меню раскрывающегося> PROMPT «название» –Создание вертикального пункта

ON SELECTION BAR 1 OF <имя меню раскрывающегося>do<имя программы>– Активация вертикального пункта

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:

- номер, наименование практической работы и тему;
- ответы на контрольные вопросы;
- выводы.

4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

Создание меню

1. Создать меню;
2. Создать в меню 3 PAD;
3. Создать в каждом PAD POPUP;
4. Создать в каждом POPUP 3 BAR;
5. 1 PAD и 2 PAD, а также в 3 PAD 1 и 2 BAR могут называться как угодно;
6. В 3 PAD 3 BAR должен обязательно называться.
7. Написать программу для BAR «Выход»; 1. Для начала создадим меню.

```
DEFINE MENU Дамзин_Меню COLOR r/w
```

Рисунок – 1 Создание меню.

2. После создания меню нужно добавить горизонтальные пункты.

```
DEFINE PAD p1 OF Дамзин_Меню PROMPT "Вид"  
DEFINE PAD p2 OF Дамзин_Меню PROMPT "файл"  
DEFINE PAD p3 OF Дамзин_Меню PROMPT "Настройки"
```

Рисунок – 2 Создание горизонтальных пунктов.

3. Создадим раскрывающийся пункт.

```
DEFINE POPUP pop1  
DEFINE POPUP pop2  
DEFINE POPUP pop3
```

Рисунок – 3 Создание раскрывающихся пунктов.

4. Теперь нам нужно чтобы при нажатии на горизонтальный пункт появлялся раскрывающийся пункт.

```
ON PAD p1 OF Дамзин_Меню ACTIVATE POPUP pop1  
ON PAD p2 OF Дамзин_Меню ACTIVATE POPUP pop2  
ON PAD p3 OF Дамзин_Меню ACTIVATE POPUP pop3
```

Рисунок – 4 Активация раскрывающихся пунктов.

5. Далее примемся за создание вертикальных пунктов в каждом раскрывающемся пункте.

```

DEFINE BAR 1 OF pop1 prompt "Окно"
DEFINE BAR 2 OF pop1 prompt "Окно (без рамки) "
DEFINE BAR 3 OF pop1 prompt "Полный экран"
DEFINE BAR 1 OF pop2 prompt "Экспорт"
DEFINE BAR 2 OF pop2 prompt "Импорт"
DEFINE BAR 3 OF pop2 prompt "Открыть"
DEFINE BAR 1 OF pop3 prompt "Инструменты"
DEFINE BAR 2 OF pop3 prompt "Информация"
DEFINE BAR 3 OF pop3 prompt "Выход"

```

Рисунок – 5 Создание вертикальных пунктов.

6. Напишем программу для закрытия нашего меню.

```

ON SELECTION BAR 3 OF pop3 do exitprog
PROCEDURE exitprog
* activate window wihod
CLOSE ALL
in_stat=.f.
DEACTIVATE POPUP m_menu
quit |

```

Рисунок – 6 Алгоритм закрытия меню.

7. Теперь нам нужно все это сохранить в файл *.prg (По умолчанию сохраняется в C:\Users\Имя пользователя\Documents\VisualFoxProProjects НА WIN 8.1!!!).

```

SAVE TO ProgramMetoda.prg
DO programmetoda.prg

```

Рисунок – 7 Создание файла *.prg.

8. Через пункт меню file выбираем пункт open и открываем наш файл *.prg и в него уже копируем все из командной строки и сохраняем комбинацией клавиш Ctrl+S.

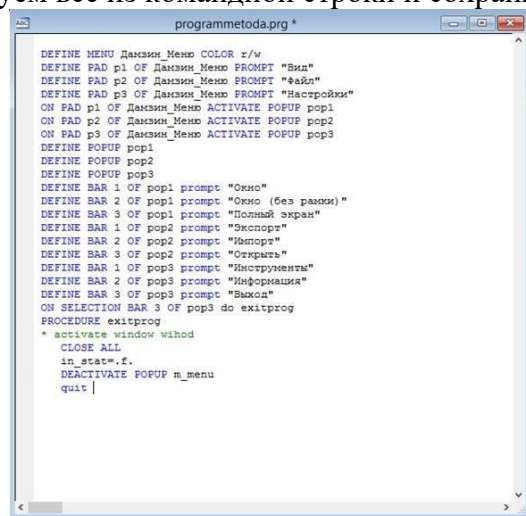


Рисунок – 8 Общий результат работы.

Чтобы проверить работоспособность программы нужно открыть пункт меню Program в нём выбрать Do... далее выбираем нашу программу и пишем ACTIVATE MENU <Название меню>.

Контрольные вопросы:

1. Что такое меню?
2. Перечислите этапы разработки меню в FoxPro?

Отчет

Сохраните работу.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно	Точность	Защита работы	70 б – оценка «5»

выполнена работа- 10 баллов	выполнения работы – 10 баллов	аудиторно – 20 баллов	40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10 баллов	Защита своей работы персонально – 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №10

Тема:Создание рабочих и системных окон. Добавление элементов управления рабочим окном (2 часа)

Цель занятия:

1. Создать в программе меню дополнительные пункты в меню которые будут открывать новые окна с информацией о разработчике и базой данных

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: Компьютер, операционная система Windows , FoxPro

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Краткие теоретические сведения:

Добавление новых пунктов:

```
DEFINE BAR 6 OF M_MENU PROMPT "Infoprogram"
DEFINE BAR 7 OF M_MENU PROMPT "openbar"

ON SELECTION BAR 6 OF M_MENU do Infoprogram
ON SELECTION BAR 7 OF M_MENU do openbar
```

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:
 - номер, наименование практической работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

Создание дополнительных окон в VisualFoxPro1. Добавляем в наше меню новые пункты

```
DEFINE BAR 6 OF M_MENU PROMPT "Infoprogram"
DEFINE BAR 7 OF M_MENU PROMPT "openbar"

ON SELECTION BAR 6 OF M_MENU do Infoprogram
ON SELECTION BAR 7 OF M_MENU do openbar
```

Рисунок 1

2. Добавляем в меню команды выполнения наших программ для новых окон (О создателе и база данных)

```
PROCEDURE infoprogr
DO prograam_test.prg

PROCEDURE openbar
DO oknotest.prg
```

Рисунок 2

3. Пишем код программы для открытия базы данных в новом окне.

```
DEACTIVATE WINDOW all
RELEASE WINDOWS all
USE "c:\as\baza.dbf" SHARED
DEFINE WINDOW f2 from 1,1 TO 10,120 COLOR r/b FONT 'arial'

ACTIVATE WINDOW f2
BROWSE
```

Рисунок 3

4. Пишем код для программы о создателе.

(Если задать шрифт стандартный, то русский текст не будет отображаться корректно)

```
DEFINE WINDOW f3 from 12,1 TO 15,35 COLOR r*/w* FONT 'arial'

ACTIVATE WINDOW f3
@1,1 say 'Created by Absolyutly Company'
```

Рисунок 4

Проверяем работоспособность наших пунктов в меню



Рисунок 5

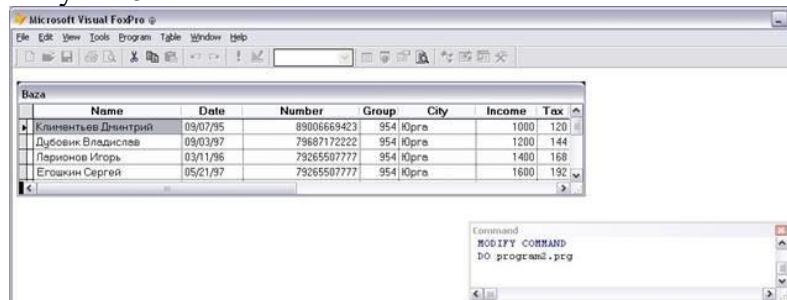


Рисунок 6

Контрольные вопросы:

1. Что такое окно программы?
2. Какие возможности по работе с окном дает FoxPro?

Отчет

Сохраните работу.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно	Точность	Защита работы	70 б – оценка «5»

выполнена работа- 10 баллов	выполнения работы – 10 баллов	аудиторно – 20 баллов	40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10 баллов	Защита своей работы персонально – 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №11

Тема: Создание файла проекта базы данных. Создание интерфейса входной формы. Использование исполняемого файла проекта БД, приемы создания и управления. (2 часа)

Цель занятия:

1. Преобразование реляционной БД

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: компьютер, выход в Интернет

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:

- номер, наименование практической работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

1. Создание проекта. Исследование созданного проекта.

Создайте проект любым из имеющихся способов (посредством главного меню или с помощью команды). При сохранении введите имя проекта:

ВНИМАНИЕ! Имя вашего проекта – **Myproj_N**, где N – ваш номер по журналу (например, если ваш номер 3, то имя проекта - **Myproj_3**).

Изучите разделы проекта и просмотрите доступные функциональные кнопки проекта.

2. Добавление файлов в проект.

Задание 1.

Перейдите на вкладку **Data** и выберите в списке пункт **Database**. Добавьте в проект базу данных **Mydata**, созданную ранее. Обратите внимание, что при установке курсора на добавленную БД, в нижней части проекта отображается полный путь к файлу БД. **Убедитесь, что путь соответствует вашей папке!**

Проверьте работу функциональных кнопок на данной вкладке.

На вкладке **Data** выберите пункт **FreeTables** (свободные таблицы) и добавьте в этот раздел свободную таблицу вашего варианта, созданную на первом практическом занятии. **Убедитесь, что путь таблицы соответствует вашей папке!** Проверьте работу функциональных кнопок при работе со свободными таблицами.

Скопируйте скрин-шот с вкладкой Data в файл отчета.

Задание 2.

Перейдите на вкладку **Documents**. Изучите активные функциональные кнопки.

Задание 3.

Перейдите на вкладку **Code**. Изучите активные функциональные кнопки. Выберите в списке пункт **Program**. Добавьте все разработанные вами программы в проект.

Убедитесь, что пути программных файлов соответствуют вашей папке! Проверьте работу функциональных кнопок при работе с программными файлами. Используя кнопки, откройте любую программу для изменения, попробуйте запустить программу и т.д.

Используя кнопки проекта, создайте новую программу **prog20**, которая решает задачу, выданного вам варианта (на карточке).

Текст созданной программы и результат ее выполнения добавьте в отчет. Проверьте наличие созданной программы в проекте. Если программа отсутствует, то самостоятельно добавьте ее в проект. Скопируйте скрин-шот с вкладкой Code в файл отчета.

Задание 4.

Перейдите на вкладку **Other**. Изучите активные функциональные кнопки. Выберите в списке пункт **OtherFiles**. Добавьте в проект несколько картинок (файлы с расширением .bmp, .jpg и т.д.).

Скопируйте скрин-шот с вкладкой Other в файл отчета.

3. Покажите результат преподавателю. Закройте созданный проект. После выхода в операционную систему проверьте наличие в вашей папке файлов с

Для возможности проверки преподавателем ваших выполненных заданий, необходимо создать файл отчета в среде MicrosoftWord, в который скопировать все выполненные команды и скрин-шоты с результатами.

вашим именем Myproj_N.pjx и Myproj_N.pjt.

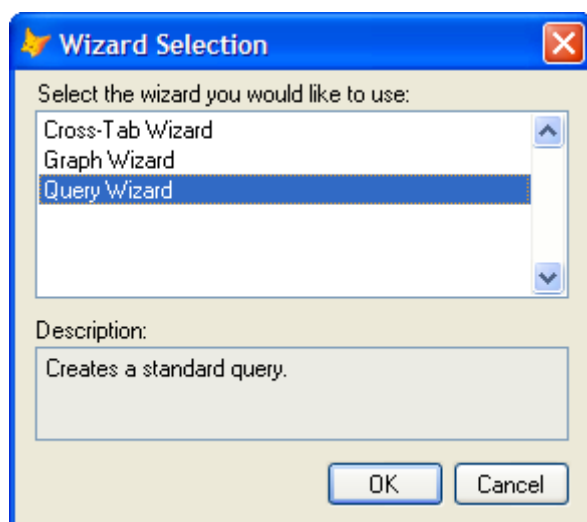
«Создание запросов»

Откройте свой проект, созданный ранее.

1. Создание запроса с помощью мастера из главного меню.

Задание: создать запрос на основе таблицы **knig**, содержащий поля **Название** и **Автор**.

В главном меню выберите пункт **File – New** и в окне **New** выберите создание запроса с помощью мастера. В окне **WizardSelection** выберите мастер запросов:



Выполните шаги мастера:

- 1 **шаг** – из таблицы **Knig** выберите поля **nazvanie** и **autor**.
- 2 **шаг** – здесь отсутствует, так как запрос создается на основе одной таблицы.
- 3 **шаг** – оставьте без изменения.
- 4 **шаг** – установите сортировку по полю **nazvanie** по возрастанию.
- 5 **шаг** – завершите создание запроса.

Сохраните запрос под именем **knig1**. Проверьте наличие созданного запроса в своей папке.

2. Запуск запроса на выполнение разными способами.

В командном окне выполните команду **DO knig1.qpr** и посмотрите созданный запрос.

Добавьте в свой отчет скрин-шот с командой и результатом ее выполнения.

Затем попробуйте запустить запрос, используя опции главного меню – **Program – Do**.

Добавьте созданный запрос в свой проект. Для этого перейдите в проекте на вкладку **Data** и выберите раздел **QUERIES**, нажмите кнопку **Add**, выберите имя созданного вами запроса и нажмите на кнопку **Ok**.

Запустите запрос из проекта – кнопка **Run**.

3. Создание запроса с помощью команды CREATE QUERY в режиме конструктора.

Задание: создать запрос на основе таблицы **knig**, содержащий поля **Название**, **Автор** и **Цена** и записи с ценой больше 100.

В командном окне выполните команду **CREATE QUERY**. В результате запускается конструктор запросов.

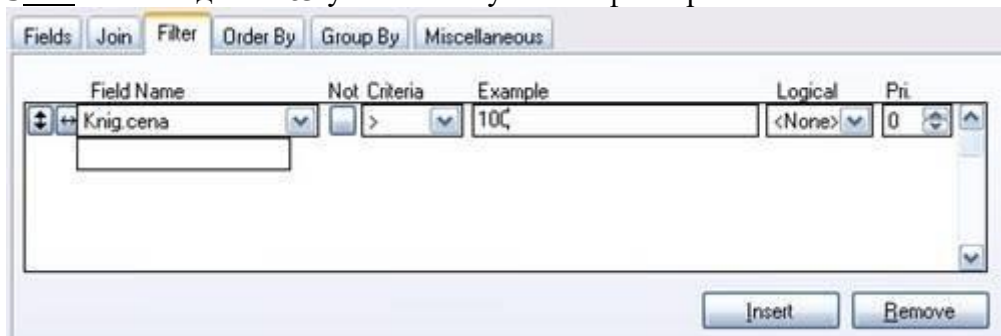
1 шаг. Выберите таблицу, на основе которой будет создан запрос – **knig**, нажмите кнопку **Add** и закройте окно **AddTableorView**.

Для удобства работы в конструкторе запросов используется панель инструментов



2 шаг. На вкладке **Fields** выберите поля, включаемые в запрос (см. задание). Для этого выберите нужное поле и нажмите кнопку **Add**.

3 шаг. На вкладке **Filter** установите условие фильтра:



Для создания используйте разворачивающиеся списки.

4 шаг. На вкладке **OrderBy** выберите условие сортировки – по полям **autor** и **pazvanie**.

5 шаг. Для просмотра запроса нажмите кнопку  на панели инструментов. *Добавьте в свой отчет скрин-шот с результатом выполнения запроса.*

6 шаг. С помощью кнопки  панели инструментов просмотрите текст программы, которая создает запрос. *Добавьте в свой отчет скрин-шот с программным кодом.*

7 шаг. Закройте окно конструктора запросов. Сохраните запрос под именем **knig_cen**. Добавьте созданный запрос в свой проект.

Добавьте созданный запрос в свой проект.

Запустите созданный запрос любым способом и просмотрите результат.

Добавьте в свой отчет скрин-шот с результатом выполнения запроса.

4. Изменение существующего запроса.

Выберите в проекте имя созданного вами запроса **knig_cen** - и нажмите на кнопку **Modify**. В результате открывается конструктор запросов.

На вкладке **Filter** измените условие отбора записей: **cena<100**. Сохраните запрос и просмотрите результат.

Добавьте в свой отчет скрин-шот с результатом выполнения запроса.

5. Создание запроса из проекта с помощью конструктора.

Перейдите в проекте на вкладку **Data** и выберите раздел **QUERIES**, нажмите кнопку **New**. В появившемся окне нажмите кнопку **NewQuery**.

Далее аналогично предыдущему запросу создайте запрос: на основе таблицы **Tip**. В запрос выберите поле **name** и установите сортировку по этому же полю. Сохраните запрос с именем **tip1** и просмотрите его.

Добавьте в свой отчет скрин-шот с результатом выполнения запроса.

6. Создание запроса на основе двух таблиц с помощью мастера.

Задание: создать запрос на основе двух таблиц - *tip* и *knig*, содержащий поля

Наименование типа, Название книги, Автор и Цена.

Перейдите в проекте на вкладку **Data** и выберите раздел **QUERIES**, нажмите кнопку **New**. В появившемся окне нажмите кнопку **QueryWizard**. В результате будет запущен мастер создания запросов.

1 шаг – Выбор полей. Из таблицы **Tip** выберите поле **name**, из таблицы **Knig** выберите поля **Autor, Nazvanie, Cena**.

2 шаг – Установление связи между таблицами. Нажмите кнопку **Add** для выбора связи.

2а шаг – Записи, включаемые в запрос. Оставляем без изменения.

3 шаг –

Фильтрация записей. Для эксперимента установите фильтр для таблицы **Tip** и нажмите кнопку

только записи, у которых **kodt** равен 1. Очистите условие фильтра и перейдите на следующий шаг.

4 шаг – Сортировка записей. Установите сортировку по полю **name**.

4а шаг – Лимит записей. Оставляем без изменения.

5 шаг – Финиш. Нажмите кнопку **Finish** и сохраните запрос с именем **tip_kn**.

Запустите запрос для просмотра. Обратите внимание, что в нем содержатся данные из двух таблиц, что очень удобно для просмотра.

Добавьте в свой отчет скрин-шот с результатом выполнения запроса.

Нажмите в проекте кнопку **Modify** для созданного запроса и перейдите на вкладку **Join**. Проверьте, правильно ли установлена связь между таблицами.

7. Покажите результат преподавателю. Закройте свой проект и осуществите выход из системы.

Контрольные вопросы:

1. Что такое проект в FoxPro? 2. Какие виды проектов существуют? 3. Каким способом можно создать проект в БД?

Отчет

Сохраните работу.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10 баллов	Защита своей работы персонально – 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №12

Тема:Создание формы. Управление внешним видом формы. (2 часа)

Цель занятия:

1. Преобразование реляционной БД.

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: Компьютер, операционная система Windows, FoxPro

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Краткие теоретические сведения:

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:

- номер, наименование практической работы и тему;
- ответы на контрольные вопросы;
- выводы.

4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

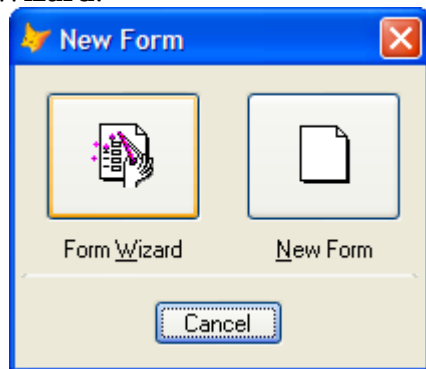
Задания для выполнения практической работы:

«Создание форм с помощью мастера» Откройте свой проект, созданный ранее.

1. Создание однотабличной формы.

Задание: создать с помощью мастера форму на основе таблицы **izd**.

Перейдите в проекте на вкладку **Documents** и выберите раздел **Forms**. Для создания новой формы нажмите кнопку **New** в проекте. В появившемся окне **NewForm** нажмите кнопку **FormWizard**:

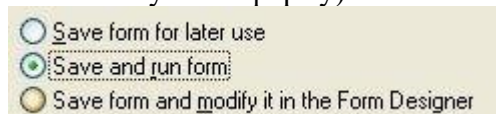


В появившемся окне **WizardSelection** выберите пункт **FormWizard** для создания однотабличной формы:



Выполните шаги мастера, выполняя все его инструкции.

- 1 шаг – для таблицы **Izd** выберите все поля.
- 2 шаг – в качестве оформления формы выберите стиль **Shadowed** и тип кнопок **Custom**.
- 3 шаг – установите сортировку по полю **n_izd** по возрастанию.
- 4 шаг – введите заголовок формы – **Данные об издательствах**, просмотрите результат с помощью кнопки **Preview**. Перед сохранением выберите в окне опцию **Save and run form** (Сохранить и запустить форму).



После этого нажмите кнопку **Finish**.

В появившемся окне **SaveAs** укажите путь к своей папке и задайте имя новой формы – **izd1**.

Если все действия были выполнены верно, то форма будет сохранена и запущена. Проверьте как осуществляется вывод данных в форме, используя полосу прокрутки. Проверьте работу кнопок.

Добавьте в таблицу новую запись с данными: код издательства – **1**, название издательства – **Форум**, адрес – **Москва**, ф.и.о. руководителя – **Кузьмичев А.С.**

2. Модификация формы.

Задание: внести изменения в форму **izd1**, используя конструктор.

Откройте созданную форму **izd1** в режиме конструктора.

- *изменение цвета объектов формы:*

Активируйте панель инструментов **ColorPalette** с помощью пункта меню **View – Toolbars**. С помощью кнопок данной панели измените цвет надписи «Данные об издательствах» и цвет фона формы. Запустите форму, сохраните ее и проверьте результат внесенных изменений.

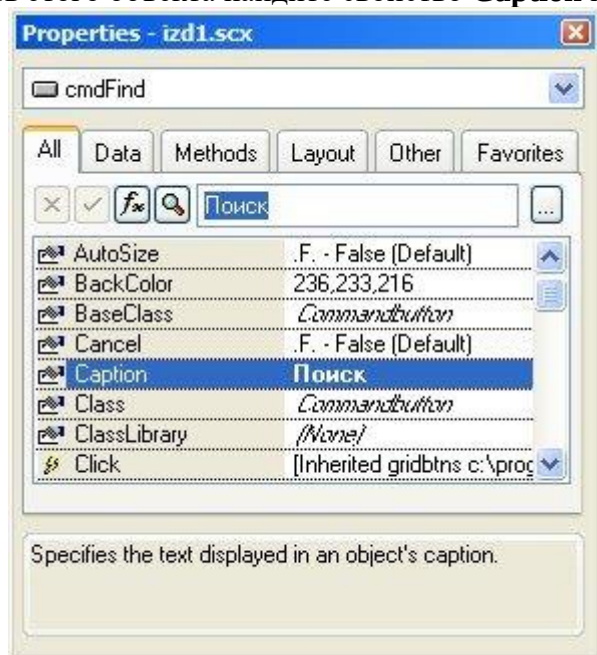
- *изменение расположения и размеров объектов формы:*

Переместите надпись «Данные об издательствах» так, чтобы она располагалась по горизонтали по центру.

Уменьшите размер контейнера, в котором отображаются данные. Запустите форму и проверьте результат внесенных изменений.

- *изменение свойств объектов:*

В окне **Properties** выберите в списке объектов формы объект **cmdFind**. Затем в списке свойств этого объекта найдите свойство **Caption** и введите в строке ввода слово **Поиск**.



В результате в форме эта кнопка будет иметь название «Поиск». Аналогично измените названия остальных кнопок: **cmdPrint** - Печать, **cmdAdd** - Ввод, **cmdDelete** - Удаление, **cmdExit** – Выход.

Для кнопки Выход установите свойство **FontBold** равным **Т.** , свойство **ForeColor** равным **255,0,0** . Обратите внимание, что значения всех свойств, которые мы изменяли в окне **Properties**, изменяют свой цвет на **фиолетовый**.

Запустите форму и проверьте результат внесенных изменений.

Закройте форму и сохраните все изменения.

3. Создание формы на основе двух таблиц.

Задание: создать с помощью мастера форму на основе таблиц **tip** и **knig**.

Запустите мастер форм из проекта. В окне **WizardSelection** выберите пункт **Oneto-ManyFormWizard** для создания формы один-ко-многим:



Выполните шаги мастера:

1 шаг – для родительской таблицы **tip** выберите поле **naim**.

2 шаг – для дочерней таблицы **knig** выберите поля **autor**, **nazvanie**, **cena**, **kol**. 3 шаг – проверьте, правильно ли установлена связь между таблицами – должна быть установлена связь по полю **kodt**.

4 шаг – в качестве оформления формы выберите стиль **Embossed** и тип кнопок **Picturebuttons**. 5 шаг – установите сортировку по полю **Name**.

6 шаг - введите заголовок формы – **Перечень книг**, просмотрите результат с помощью кнопки **Preview**.

После этого нажмите кнопку **Finish**.

В появившемся окне **SaveAs** укажите путь к своей папке и задайте имя новой формы – **tip_knig**.

4. Тестирование формы.

Задание: запустить форму **tip_knig** и проверить ее работу.

Запустите форму из проекта. Проверьте работу всех кнопок. Обратите внимание на то, какие недочеты имеет форма, чтобы в дальнейшем их устранить.

5. Модификация формы.

Задание: внести изменения в форму **tip_knig**, используя конструктор.

Откройте созданную форму **tip_knig** в режиме конструктора. Откройте все необходимые для работы панели инструментов.

-изменение размеров объектов:

Так как данные таблицы не полностью видны в форме, то необходимо изменить размер формы (увеличить по ширине) и размер объекта **Grid1** так чтобы отображались все данные. С помощью мыши измените размеры указанных объектов.

Запустите форму, сохраните ее и проверьте результат внесенных изменений.

-изменение свойств объектов:

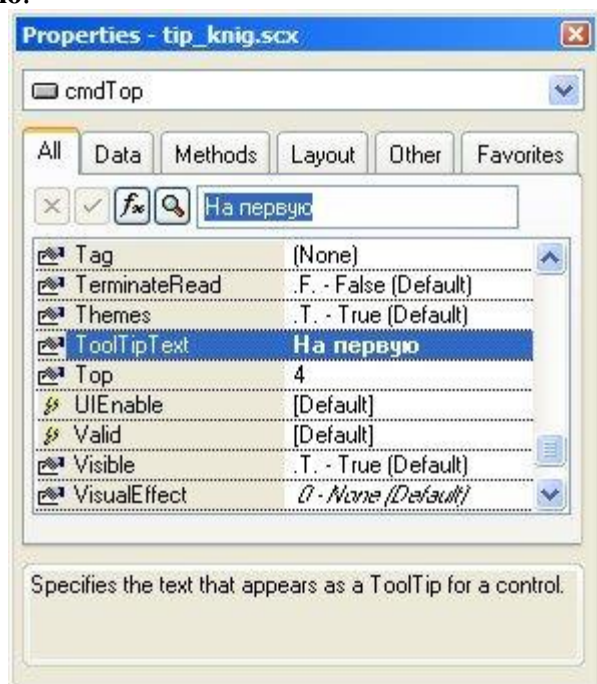
В окне **Properties** выберите в списке объектов формы объект **Form1**. Затем в списке свойств этого объекта найдите свойство **MaxButton** и измените его значение на **.F.** . Аналогично для свойства **MinButton** установите значение **.F.** . Обратите внимание, что в заголовке формы исчезли кнопки максимизации и минимизации окна.

Запустите форму, сохраните ее и проверьте результат внесенных изменений.

Для надписи **Перечень книг** измените цвет текста и расположите ее по горизонтали по центру. Для текстового поля **Name** и подписи к нему измените размер шрифта и цвет текста.

При запуске формы обратите внимание, что для каждой кнопки в форме имеется всплывающая подсказка, но на английском языке. Это не очень удобно.

В окне **Properties** выберите в списке объектов формы объект **cmdTop**. Затем в списке свойств этого объекта найдите свойство **ToolTipText** и введите в строке ввода слово **На первую**.



Запустите форму и проверьте измененную подсказку. Аналогично измените текст подсказки для всех остальных кнопок – **Предыдущая**, **Следующая**, **На последнюю**, **Поиск**, **Печать**, **Ввод/Сохранение**, **Редактирование**, **Удаление**, **Выход**.

Запустите форму, сохраните ее и проверьте результат внесенных изменений.

Завершите работу с формой **tip_knig**.

6. Покажите результат работы преподавателю.

«Конструктор форм. Настройка параметров формы. Определение среды окружения»

1. Создание экранной формы с помощью команды CREATE FORM. Задание: создать форму, настроить основные свойства формы.

В командном окне выполните команду **CREATE FORM f1**. В результате запускается конструктор форм и открывается окно со свойствами формы.

Для работы с конструктором форм необходимо окно свойств **Properties**, панели инструментов **FormDesigner** и

FormC

ontrols.

Панель **FormDesigner**

Панель инструментов **FormDesigner** автоматически отображается вместе с конструктором форм.

Окно свойств

Окно **Properties** открывается со списком свойств и событий выделенного объекта и используется для установления его свойств.

Панель **FormControls**

С помощью панели инструментов **FormControls** можно создавать элементы

управления в формах.

Если окно свойств отсутствует, то нажмите кнопку  на главной панели инструментов.

Если панели инструментов отсутствуют, то для их активации выберите пункт меню **View – Toolbars** и выберите нужную панель.

В окне свойств установите основные свойства формы:

AutoCenter – .t.	центрирование объекта Form
BorderStyle – 1	стиль обрамления объекта
Caption – «Типы книг»	текст названия объекта
MaxButton – .f.	Кнопка максимизации объекта
MinButton – .f.	Кнопка минимизации объекта
Picture	Установка фонового рисунка
WindowState – 0	0 – немодальный, меню активно. 1 - модальный. В этом случае будет активна только данная форма, никакой другой объект Form не может стать активным, меню неактивно.

Запустите форму для проверки установленных свойств.

Для запуска формы используется кнопка  на главной панели инструментов.

2. Определение среды окружения формы и вставка в форму элемента «Сетка».

Задание: подключить к форме таблицу **tip**, добавить в форму элемент **Grid**, настроить необходимые свойства.

Для подключения данных к форме используется диалоговое окно **DataEnvironment**. Оно отображает элементы среды данных, с которой работает экранная форма. Для активации окна **DataEnvironment** используются следующие способы: через главное меню **View**, через контекстное меню, через панель инструментов **FormDesigner** – кнопка .

Используйте любой из предложенных способов – на экране появится окно для выбора нужной таблицы. Выберите таблицу **tip** и закройте окно. Выбранная таблица будет вставлена в окно **DataEnvironment**.

Выделите таблицу и установите для нее свойство **Order** – индекс **t1**. Данное свойство подключает индекс для сортировки таблицы.

Для вставки таблицы в форму достаточно перетащить таблицу за заголовок мышкой на форму. Выполните это. В результате в форме создается элемент управления **Grid**. После этого закройте окно **DataEnvironment**, так как оно нам пока не потребуется.

Рассмотрите в окне свойств **Properties** свойства элемента **Grid**, которые установлены по умолчанию (они выделены фиолетовым цветом).

RecordSource	Задаёт источник данных, к которому привязан элемент управления Grid.
--------------	--

Установите дополнительные свойства:

DeleteMark - .f.	Определяет, будет ли в элементе управления Grid отображаться столбец маркеров удаления.
ReadOnly - .t.	Определяет, может ли пользователь редактировать элемент управления или разрешены ли обновления таблицы или представления, связанных с объектом Cursor. .t. - Пользователь не может редактировать элемент управления.

Для настройки элемента **Grid** можно щелкнуть по этому элементу правой кнопкой мыши использовать команду **Builder...** (построитель) из контекстного меню. Выберите эту команду.

На вкладке **GridItems** указаны выбранные поля.

На вкладке **Style** выберите по желанию стиль оформления элемента **Grid**.

На вкладке **Layout** в разделе Caption введите заголовки столбцов таблицы: 1 столбец – **Код**, 2 столбец – **Наименование**. Здесь же можно изменить ширину столбцов. 1 столбец сделайте поуже, а 2 столбец – пошире, чтобы данные не обрезались.



На вкладке **Relationship** устанавливаются связи между таблицами. Так как у нас в форме будет отображаться одна таблица, то здесь ничего не делаем.

После установленных изменений нажмите кнопку **Ok**.

Измените размеры элемента **Grid** – уменьшите ширину так, чтобы были видны только столбцы таблицы. Разместите таблицу в центре формы.

Запустите форму для просмотра и проверьте внесенные изменения.

3. Вставка в форму элемента «Кнопка».

Задание: создать в форме кнопку «Выход», при нажатии которой форма будет закрываться.

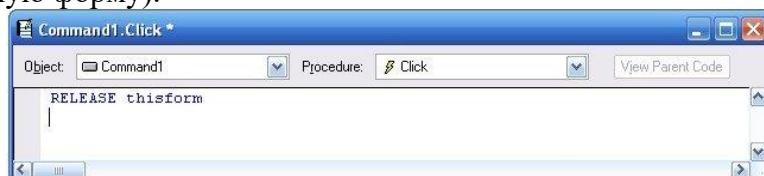
Чтобы вставить элемент Кнопка, щелкните на изображении кнопки  на панели **FormControls** (CommandButton), затем установите указатель мыши в форме и щелкните для вставки элемента управления. В форме появится элемент кнопка с именем **Command1**.

Для элемента Кнопка установите следующие свойства:

Caption – «Выход»	Название кнопки
FontSize – 12	Задаёт размер шрифта для текста, отображаемого вместе с объектом
ForeColor – выбрать красный цвет	Задаёт основной цвет символов, используемые при отображении текста и графики в объекте.
FontBold – .t.	Полужирное начертание букв
Cancel – .t.	Определяет, является ли элемент управления CommandButton кнопкой Cancel; это означает, что если пользователь нажмет клавишу ESC, произойдет событие Click для кнопки Cancel

Для того, чтобы кнопка работала в форме, необходимо обработать событие, которое будет происходить при нажатии кнопки. В кнопках всегда обрабатывается событие **Click** – однократное нажатие левой кнопки мыши.

Щелкните два раза по кнопке **Command1**. Откроется окно **Command1.Click** для ввода обрабатывающей процедуры. Введите одну команду - **releasethisform** (эта команда закрывает активную форму).



Закройте окно **Command1.Click**.

Запустите форму и проверьте работу созданной кнопки.

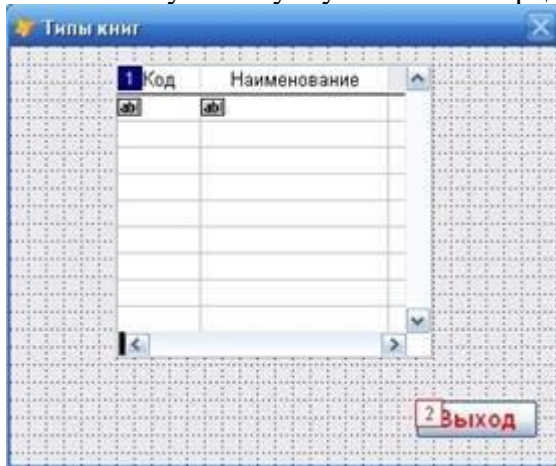
4. Настройка порядка обхода элементов управления в форме.

Задание: проверить и настроить порядок обхода элементов в форме.

Для настройки порядка обхода элементов управления в форме используется кнопка

SetTabOrder  на панели инструментов **FormDesigner**.

Нажмите эту кнопку и установите порядок элементов: сетка – 1, кнопка – 2.



Нажмите кнопку повторно. Порядок обхода сохранится. Запустите форму для просмотра.

Как видите, при запуске формы сразу активна таблица и по ней можно перемещаться с помощью стрелок вверх и вниз. Для выхода из формы достаточно нажать кнопку **Выход**.

5. Добавление в форму второй кнопки.

Задание: добавить в созданную форму кнопку «Ввод новой записи», при нажатии на которую будет загружаться новая форма – для ввода новой записи в таблицу.

Добавьте в форму новую кнопку - **Command2**. Установите нужные свойства:

Caption – «Ввод новой записи».

Двойным щелчком мыши вызовите окно **Command2.Click**. Введите следующие команды:

```
do FORM f2 && запуск новой формы  
thisform.grdTip.SetFocus    && установка фокуса на элемент Сетка  
thisform.Refresh           && обновление формы
```

Здесь после знака && введены комментарии. Закройте окно **Command2.Click**.

Созданная кнопка пока не будет работать, так как у нас не создана форма f2. Если попробовать запустить форму и нажать данную кнопку то выйдет ошибка.

Сохраните форму. Закройте конструктор форм.

Созданную форму необходимо добавить в проект. Перейдите в проекте на вкладку **Documents** и выберите пункт **Forms**. Нажмите кнопку **Add** и добавьте созданную форму в свой проект.

6. Создание экранной формы из проекта.

Задание: создать форму, установить основные свойства, подключить данные, создать коды обработки событий.

В проекте перейдите на вкладку **Documents** и выберите пункт **Forms**. Нажмите кнопку **New - NewForm**.

Задайте основные свойства формы: AutoCenter – .t.

BorderStyle – 1

Caption – название формы «Ввод новой записи» MaxButton – .f.

MinButton – .f.

WindowType – 1

Подключите к форме таблицу **tip** с помощью окна **DataEnvironment**. Выделите таблицу и установите для нее свойство **Order** – индекс **t1**. Перетащите в форму из окна **DataEnvironment** из таблицы **tip** поле **name**. В результате в форму будут вставлены два элемента – **Label** (текст) и **TextBox** (поле для ввода).

Необходимо отредактировать данные элементы.

Выделите элемент **lblName** и измените для него свойство **Caption** – Наименование.

Передвиньте элемент так, чтобы он не закрывал поле для ввода.

Выделите элемент **txtName** и измените для него свойство **ControlSource** – m.name.

Это свойство задает источник данных, к которому привязывается объект. В нашем случае осуществляется привязка к переменной памяти, которая будет создана командой SCATTER MEMVAR BLANK.

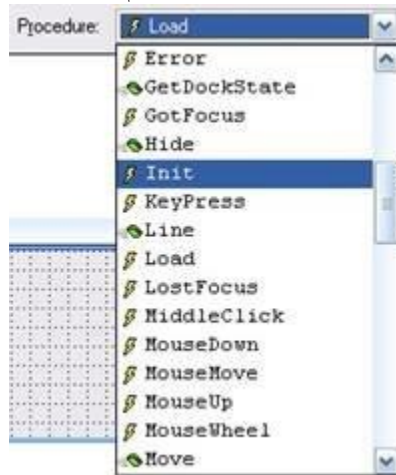
Справка:

Команда SCATTER MEMVAR - копирует содержимое текущей записи таблицы в набор переменных памяти. При ссылке на переменную памяти, чтобы отличать ее имя от имени соответствующего поля в таблице, ставьте перед именем переменной префикс 'm.'.

Команда SCATTER MEMVAR BLANK - Создает набор пустых переменных памяти.

Для данной формы необходимо создать код для обработки события загрузки формы – это событие **Init**. Чтобы мы могли ввести новую запись в таблицу **tip**, необходимо создать пустой набор переменных памяти при загрузке формы.

Выполните двойной щелчок мыши по любому свободному месту формы. Откроется окно для обработки кодов формы – **Form1.Load**. Выберите в разделе **Procedure** из разворачивающегося списка событие **Init**:



Обратите внимание, что у окна поменяется название на **Form1. Init**. Введите команду SCATTER MEMVAR blank и закройте окно. **Сохраните форму с именем f2.**

7. Вставка в форму кнопок.

Задание: добавить в форму кнопки «Выход» и «Запись».

Добавьте в форму кнопку «**Выход**» аналогично первой форме.

Добавьте в форму кнопку «**Запись**», при нажатии которой будет осуществляться запись новых данных в таблицу. Поэтому для данной кнопки нужно ввести код для события **Click**:

```
APPEND BLANK
```

```
REPLACE naim WITH thisform.TxtName.Value
```

```
RELEASE thisform
```

Настройте порядок обхода элементов управления в данной форме: элемент **txtName** – 1, кнопка **Запись** – 2, кнопка **выход** - 3.

Сохраните форму.

Для проверки работы созданных форм запустите форму **f1** (кнопка Run проекта), в ней нажмите кнопку «Ввод новой записи». В появившемся окне введите произвольную новую запись.

8. Покажите результат работы преподавателю.

9. Ответьте на контрольные вопросы.

Контрольные вопросы:

1. Для создания формы используется команда ...
2. Окно Properties используется для ...
3. Для создания среды окружения формы используется...
4. Панель инструментов FormControls используется для ...

5. Событие Click означает - ...
6. Событие Init означает - ...
7. Элемент Grid используется для ...
8. Элемент Label используется для ...
9. Элемент TextBox используется для ...

Отчет

Сохраните работу.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа - 10 баллов	Точность выполнения работы - 10 баллов	Защита работы аудиторно - 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы - 10 баллов	Защита своей работы персонально - 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы - 10 баллов	Оформление своей работы несколькими способами - 20 баллов	

Лабораторное занятие №13

Тема:Задание значений и ограничений поля. Проверка введенного в поле значения. Отображение данных числового типа и типа дата. (2 часа)

Цель занятия:

1. Преобразование реляционной БД.

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: омпьютер, операционная система Windows, FoxPro

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:

- номер, наименование практической работы и тему; • ответы на контрольные вопросы;
- выводы.

4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

Сортировка базы данных Задание:

1. Отсортировать базу данных

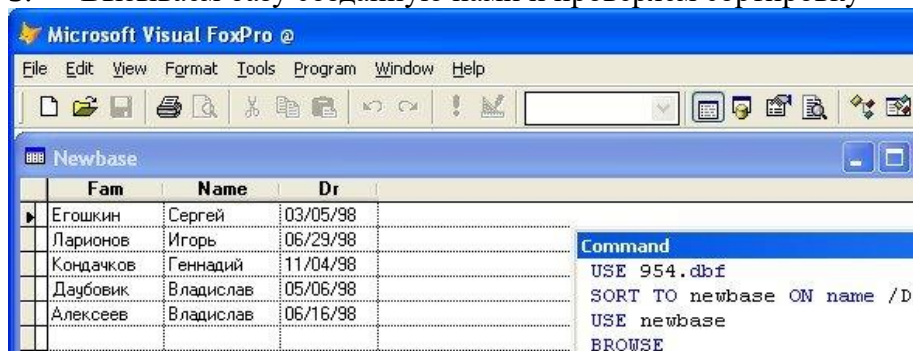
1. Вызываем базу которую мы будем сортировать

```
Command
USE 954.dbf
```

2. Сортируем базу 954.dbf по параметру “name” на убывание при этом создаём новую базу под названием “newbase”

```
Command
USE 954.dbf
SORT TO newbase ON name /D
```

3. Вызываем базу созданную нами и проверяем сортировку



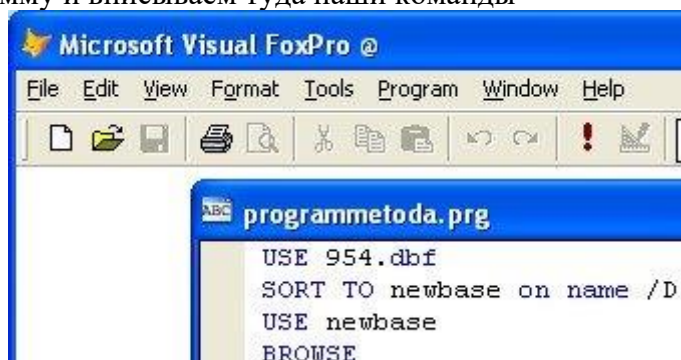
The screenshot shows the Microsoft Visual FoxPro interface. The main window displays a table named 'Newbase' with the following data:

Fam	Name	Dr
Егошкин	Сергей	03/05/98
Ларионов	Игорь	06/29/98
Кондачков	Геннадий	11/04/98
Даубовик	Владислав	05/06/98
Алексеев	Владислав	06/16/98

To the right of the table, a 'Command' window is open, showing the following commands:

```
Command
USE 954.dbf
SORT TO newbase ON name /D
USE newbase
BROWSE
```

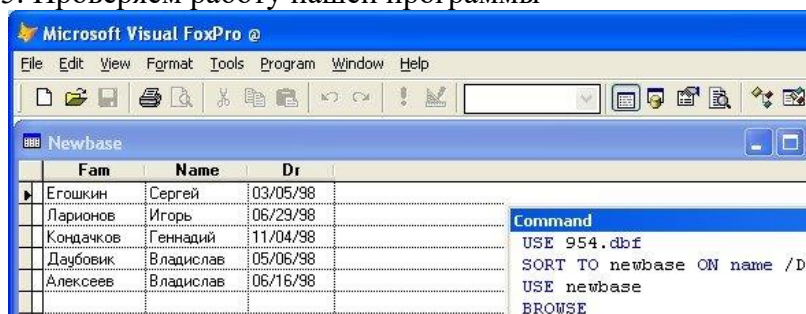
Как мы видим поле “name” прошло сортировку в порядке убывания 4. Теперь создаём программу и вписываем туда наши команды



The screenshot shows the Microsoft Visual FoxPro interface with a program named 'programmetoda.prg' open. The program contains the following commands:

```
programmetoda.prg
USE 954.dbf
SORT TO newbase on name /D
USE newbase
BROWSE
```

5. Проверяем работу нашей программы



The screenshot shows the Microsoft Visual FoxPro interface. The main window displays a table named 'Newbase' with the following data:

Fam	Name	Dr
Егошкин	Сергей	03/05/98
Ларионов	Игорь	06/29/98
Кондачков	Геннадий	11/04/98
Даубовик	Владислав	05/06/98
Алексеев	Владислав	06/16/98

To the right of the table, a 'Command' window is open, showing the following commands:

```
Command
USE 954.dbf
SORT TO newbase ON name /D
USE newbase
BROWSE
```

Чтобы проверить работоспособность программы нужно открыть пункт меню Program в нём выбрать Do... далее выбираем нашу программу и пишем ACTIVATE MENU <Название меню>

Создание поиска

Запускаем программу меню:

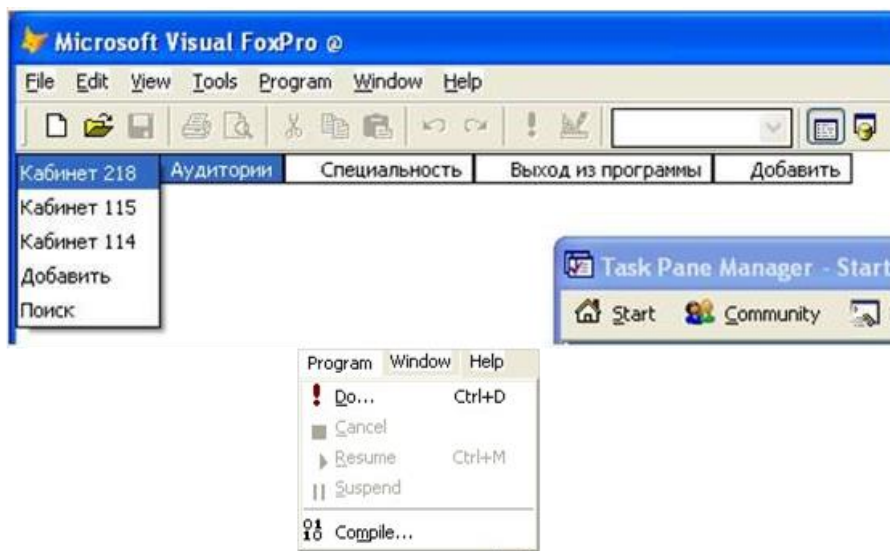


Рис.1(Наше меню.)

Далее нажимаем на кнопку «Поиск» дабы нам необходимо запустить нашу программу:

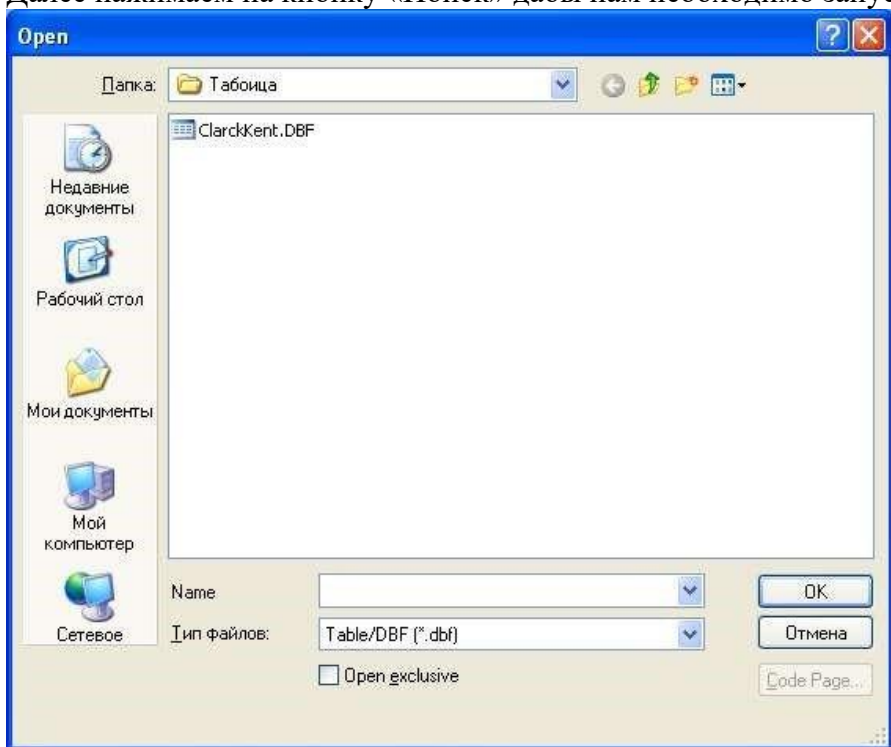


Рис.2(Открывается поиск нашей таблицы.) Выбираем.

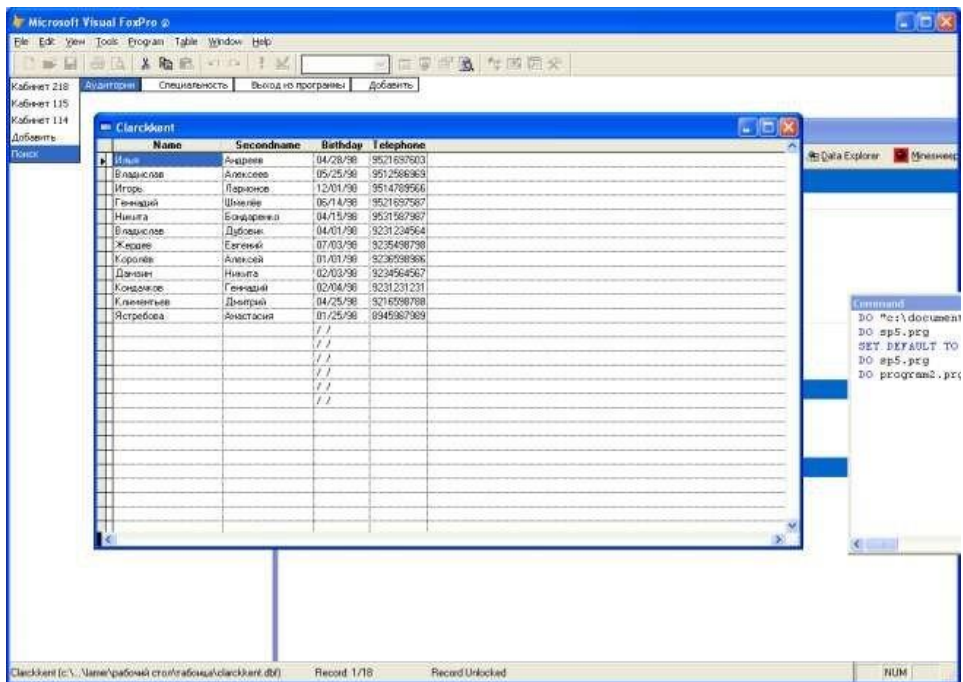


Рис.3(Открылась таблица.)
Заходим в пункт меню «Поиск»

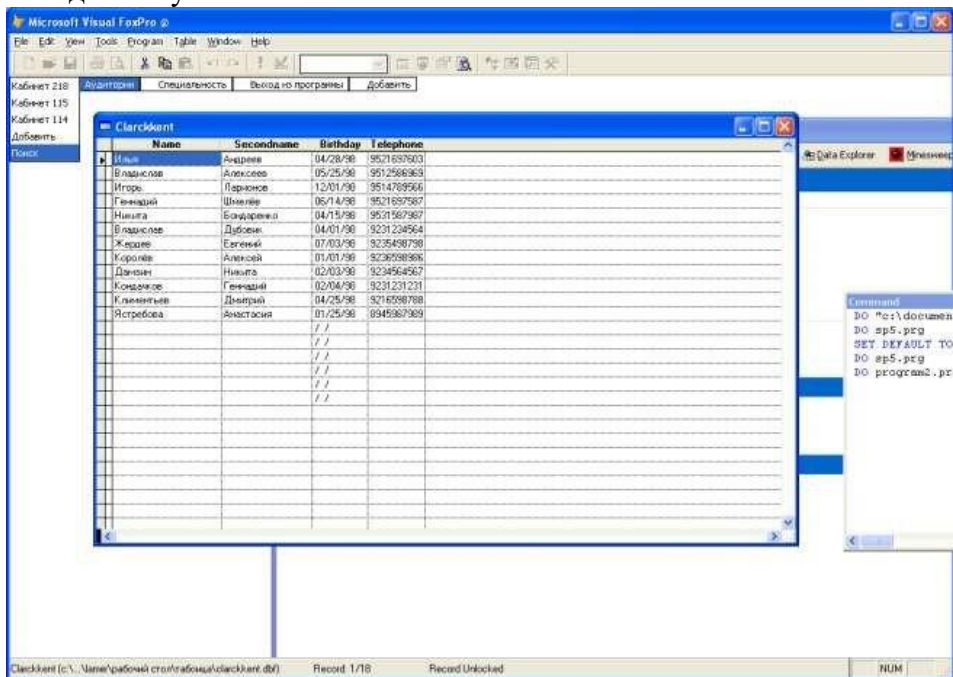


Рис.4(Ну и соответственно видим нашу выполненную работу.) И немного расскажем, как мы это сделали:

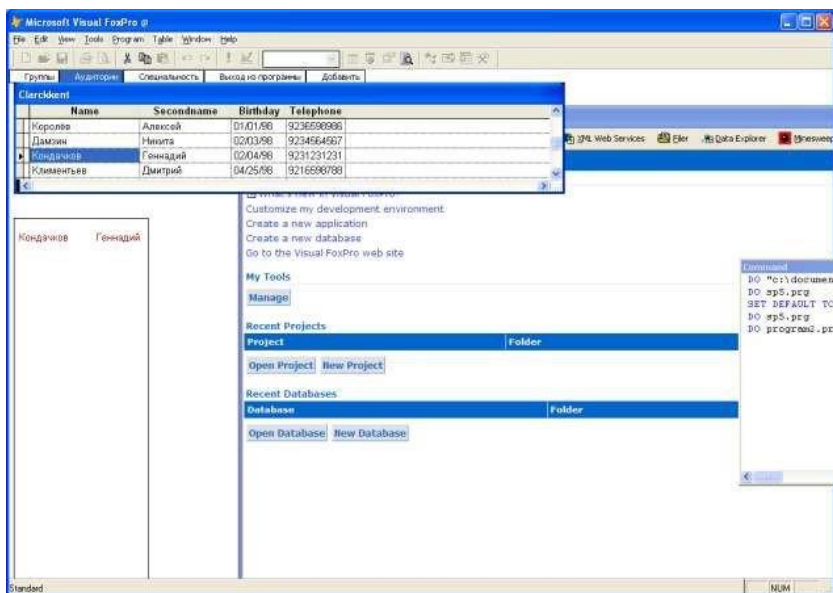


Рис.5(Добавление пункта)

Прописываем в нашем меню пункт «Поиск» чтобы он обращался к нашей программе, которая будет выполнять поиск.

Ну, и конечно же, наша сама программа выполняющая поиск:

```
define porup B
define bar 1 of B prompt "кабинет 218"
define bar 2 of B prompt "кабинет 115"
define bar 3 of B prompt "кабинет 114"
define bar 4 of B prompt "добавить"
define bar 5 of B prompt "Поиск"
ON SELECTION PAD f2 OF Andreev activate porup B
on selection bar 1 of B do sp1
on selection bar 2 of B do sp1
on selection bar 3 of B do sp2
on selection bar 4 of B do sp4
on selection bar 5 of B do sp5
```

Рис.6(Программа)

Контрольные вопросы:

1. Для чего предназначена сортировка?
2. Как осуществляется сортировка в FoxPro? 3. Как осуществить поиск в БД?

Отчет

Сохраните работу.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10 баллов	Защита своей работы персонально – 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №14

Тема: Создание и модификация таблиц БД. Выборка данных из БД. Модификация содержимого БД. (2 часа)

Цель занятия:

1. Получить некоторые практические навыки по редактированию базы данных в программе FoxPro

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: компьютер, операционная система Windows 7, FoxPro

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Краткие теоретические сведения:

Команды необходимые для выполнения практической работы

Use<ИМЯ> — открыть подготвить

Select<ИМЯ> — выбрать

Browse — открыть визуализировать

Copy – копирование БД

Rename – переименование БД

Appendblank– добавить запись

DeleterecordN– пометить на удаление запись N

Pack – удаление помеченных записей

Replace<поле>with<значение> – присвоить значение указанному полю

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:

- номер, наименование практической работы и тему;
 - ответы на контрольные вопросы;
 - выводы.
4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

Редактирование базы данных

1. Скопировать БД с 5 практической работы;
2. Переименовать БД;
3. Удалить свою запись;
4. Добавить 2-х Ивановых (Петра, Ивана);
5. Добавить поля доход и налог;
6. Заполнить поле доход (произвольно), поле налог 12% от дохода.

Открываем созданную на 5 практической работе нашу БД командой “**Browse**” и выбираем путь, если ваш файл лежит «Не в папке по умолчанию» (Рисунок 1)

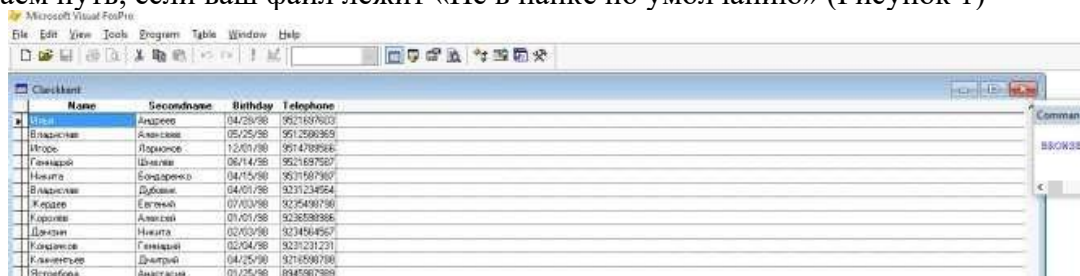


Рисунок 1

Задание 1,2

С помощью команды скопируем созданную ранее БД
 СОРУ(ИМЯ БД) ТО (ПЕРЕИМЕННОВАННОЕ ИМЯ БД)
 «Писать без скобок» (Рисунок 2)

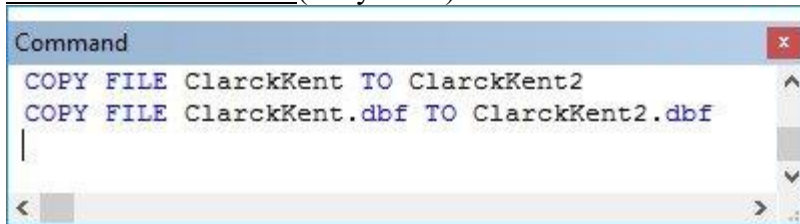


Рисунок 2 (Команда для выполнения копирования вашей БД и автоматическое переименование Новой БД, скопированной)

ОБЯЗАТЕЛЬНО! Сначала запустите программу БЕЗ ЗАПУСКА ВАШЕЙ БД,
 иначе будет выдавать ошибку, что «Файл уже используется»
 После выполнения команды, в папке вы увидите изменения:

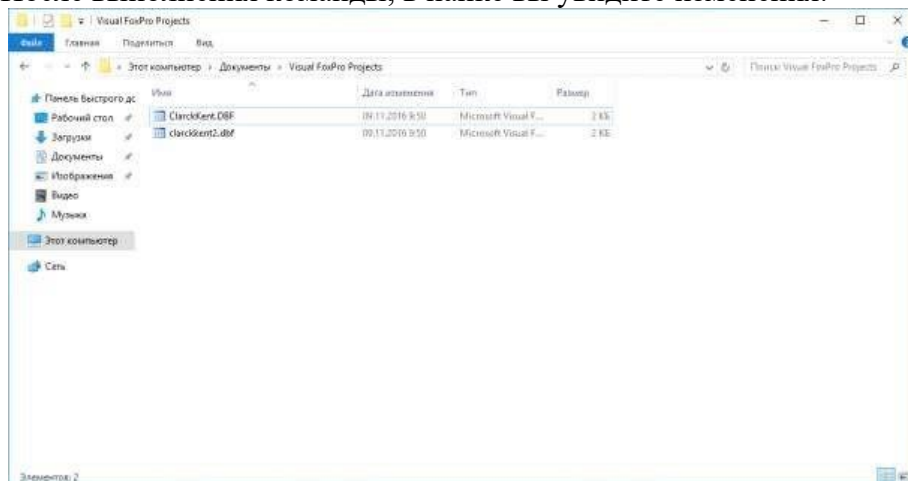


Рисунок 3 (Результат выполнения команды)

Задание 3

Для того чтобы удалить свою запись нам потребуется команды: *Delete record N* – пометить на удаление запись N, *Pack* – удаление помеченных записей.

Чтобы пометить необходимую вам строку, просто впишите вместо буквы N цифру строки, которую вам необходимо пометить (Рисунок 4)

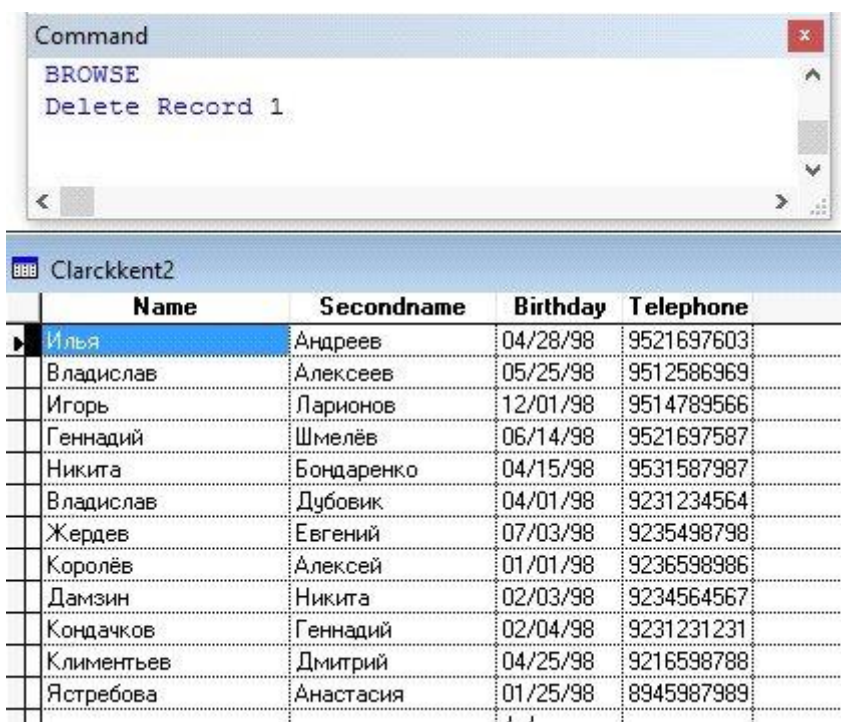


Рисунок 4 (Результат выполнения команды)

И так, тут мы видим, что 1 строчка в нашей БД пометилась «Черным прямоугольником» Ее нам и надо удалить.

В этом нам поможет команда «Pack-Удаление помеченных записей» (Рисунок 5)

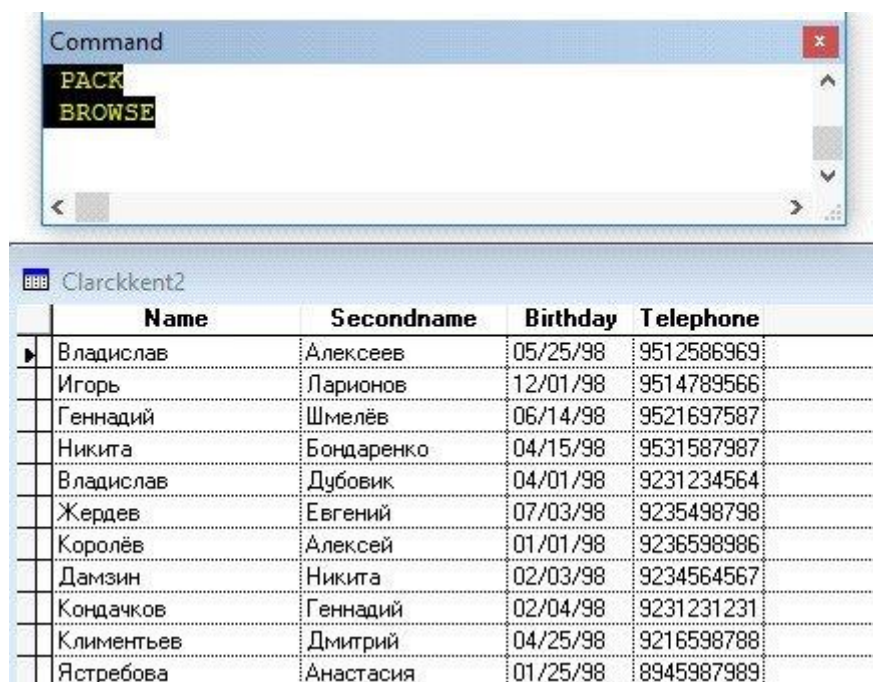


Рисунок 5 (Результат выполнения команды)

Как вы видите, что после выполнения команды Pack, строка пропала и после её выполнения, ваша БД закроется, для этого воспользуйтесь командой “Browse” она откроет вам вашу БД.

Задание 4

Добавим в нашу БД двух Ивановых (Петра и Ивана), с помощью команды *REPLACE name WITH* (Рисунок 6)

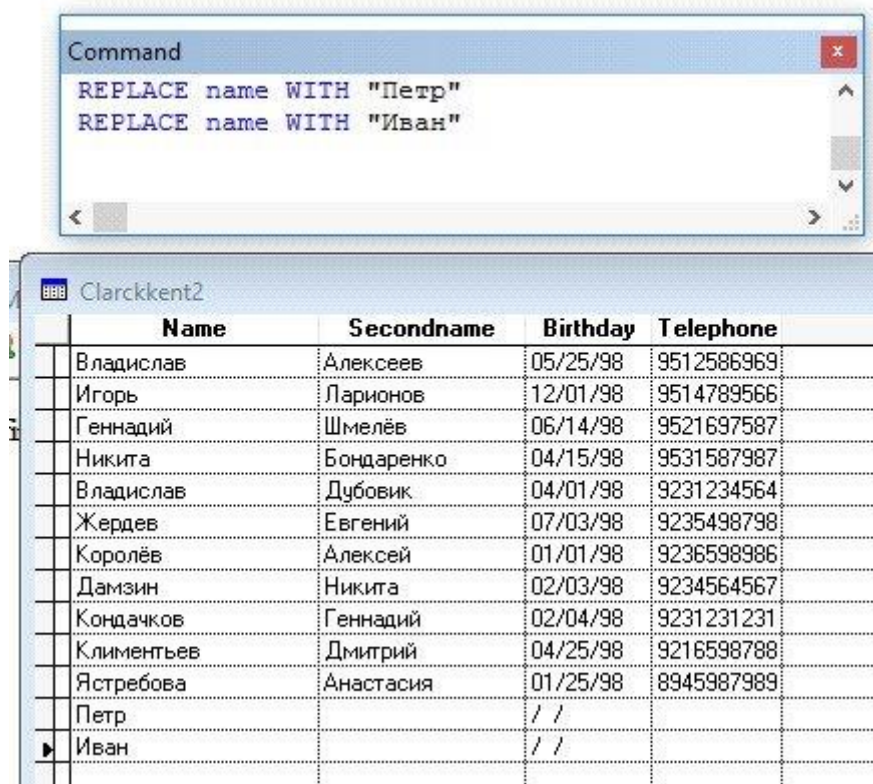
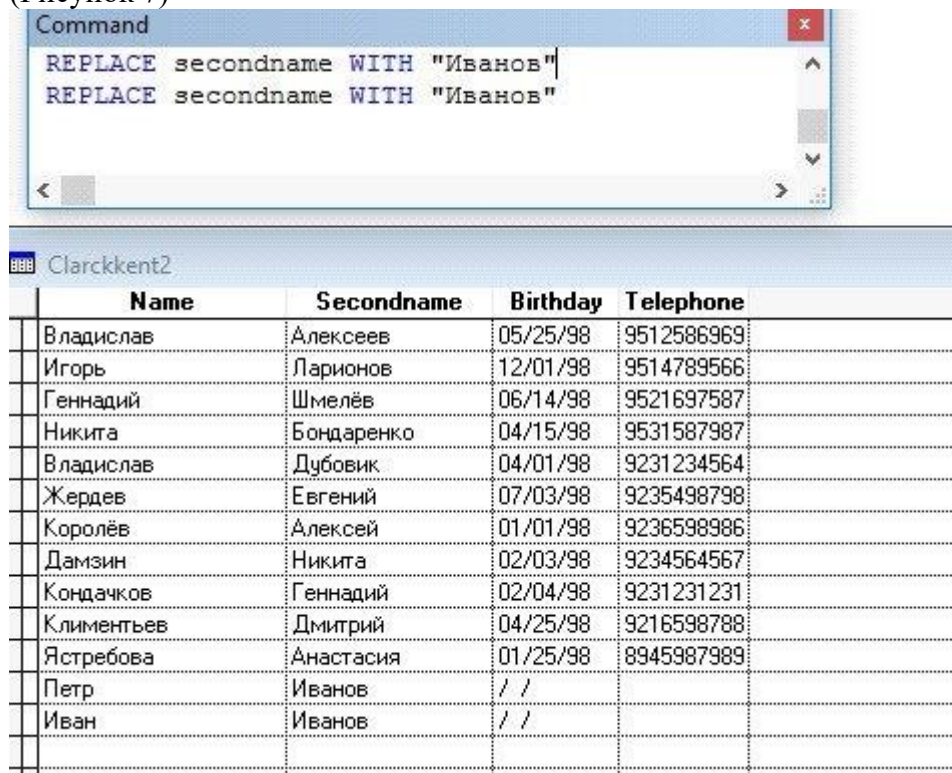


Рисунок 6 (Команда для добавления Имени)

Теперь добавим фамилии, при помощи команды *REPLACE secondname WITH* (Рисунок 7)



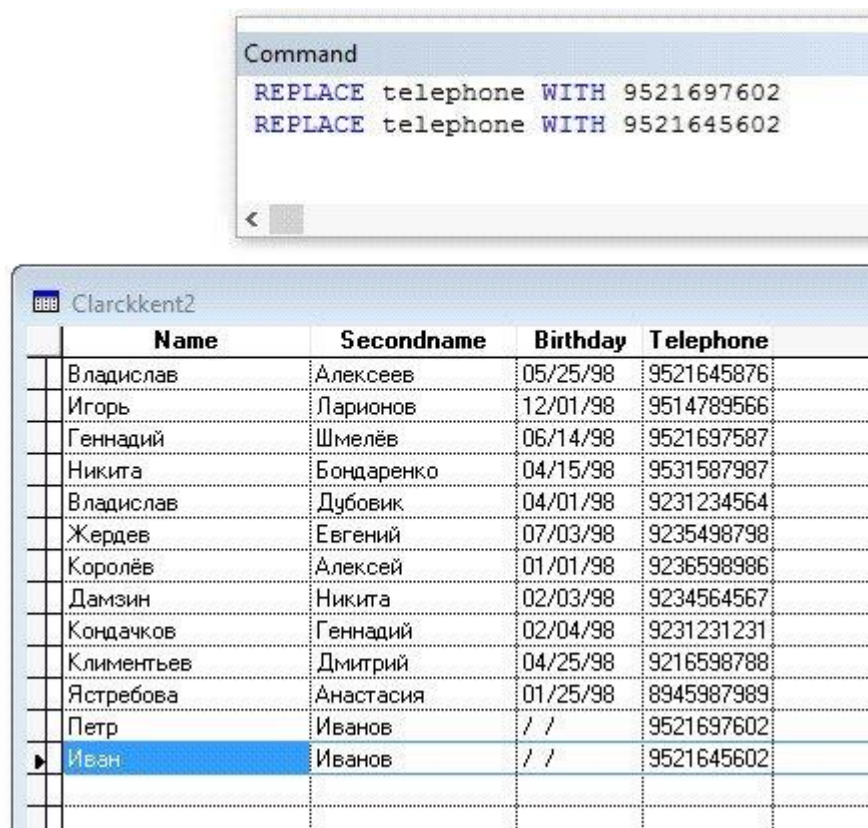
Command

```
REPLACE secondname WITH "Иванов"
REPLACE secondname WITH "Иванов"
```

Name	Secondname	Birthday	Telephone
Владислав	Алексеев	05/25/98	9512586969
Игорь	Ларионов	12/01/98	9514789566
Геннадий	Шмелёв	06/14/98	9521697587
Никита	Бондаренко	04/15/98	9531587987
Владислав	Дубовик	04/01/98	9231234564
Жердев	Евгений	07/03/98	9235498798
Королёв	Алексей	01/01/98	9236598986
Дамзин	Никита	02/03/98	9234564567
Кондачков	Геннадий	02/04/98	9231231231
Климентьев	Дмитрий	04/25/98	9216598788
Ястребова	Анастасия	01/25/98	8945987989
Петр	Иванов	/ /	
Иван	Иванов	/ /	

Рисунок 7 (Результат выполнения команды)

Также добавим номер телефона, командой *REPLACE secondname WITH* (Рисунок 8)



Command

```
REPLACE telephone WITH 9521697602
REPLACE telephone WITH 9521645602
```

Name	Secondname	Birthday	Telephone
Владислав	Алексеев	05/25/98	9521645876
Игорь	Ларионов	12/01/98	9514789566
Геннадий	Шмелёв	06/14/98	9521697587
Никита	Бондаренко	04/15/98	9531587987
Владислав	Дубовик	04/01/98	9231234564
Жердев	Евгений	07/03/98	9235498798
Королёв	Алексей	01/01/98	9236598986
Дамзин	Никита	02/03/98	9234564567
Кондачков	Геннадий	02/04/98	9231231231
Климентьев	Дмитрий	04/25/98	9216598788
Ястребова	Анастасия	01/25/98	8945987989
Петр	Иванов	/ /	9521697602
Иван	Иванов	/ /	9521645602

Рисунок 8 (Результат выполнения команды)

Добавляем дату рождения командой *REPLACE birthday WITH* (Рисунок 9)

Command				
REPLACE birthday WITH {^1998-01-02}				
REPLACE birthday WITH {^1998-05-02}				
<				

Clarckent2				
	Name	Secondname	Birthday	Telephone
	Владислав	Алексеев	05/25/98	9521645876
	Игорь	Ларионов	12/01/98	9514789566
	Геннадий	Шмелёв	06/14/98	9521697587
	Никита	Бондаренко	04/15/98	9531587987
	Владислав	Дубовик	04/01/98	9231234564
	Жердев	Евгений	07/03/98	9235498798
	Королёв	Алексей	01/01/98	9236598986
	Дамзин	Никита	02/03/98	9234564567
	Кондачков	Геннадий	02/04/98	9231231231
	Климентьев	Дмитрий	01/02/98	9216598788
	Ястребова	Анастасия	01/25/98	8945987989
	Петр	Иванов	01/02/98	9521697602
	Иван	Иванов	05/02/98	9521645602

Рисунок 9 (Результат выполнения команды)

Задание 5

Добавляем поле доход и заполняем его (произвольно) (Рисунок 10)

Фамилия	Имя	Др	Телефон	Доход
Алексеев	Владислав	03/14/98	9300-910-96-31	50000
Андреев	Илья	12/09/98	9321-21-23-900	12000
Бондаренко	Никита	11/12/98	949-77-68-95-4	12500
Дамзин	Никита	03/14/98	9349-444-99996	4100
Дубовик	Владислав	03/15/98	9699-996-95-44	12400
Евгений	Евгений	06/10/98	9699-777-77-00	23000
Евгений	Евгений	03/15/98	9495-232-55-66	130000
Климентьев	Дмитрий	07/25/98	9444-555-57-91	43000
Кондачков	Геннадий	03/21/98	9254-994-11-00	10000000
Королёв	Алексей	06/15/98	9399-999-98-57	12500
Ларионов	Игорь	05/11/98	9599-123-41-06	120
Шмелёв	Геннадий	01/01/98	9479-342-60-11	90000
Ястребова	Анастасия	09/05/98	9300-999-95-95	2200

Command	
REPLACE доход WITH 50000	
REPLACE доход WITH 12000	
REPLACE доход WITH 12500	
REPLACE доход WITH 2200	
REPLACE доход WITH 35500	
REPLACE доход WITH 4100	
REPLACE доход WITH 12400	
REPLACE доход WITH 23000	
REPLACE доход WITH 100000	
REPLACE доход WITH 100	
REPLACE доход WITH 10000000	
REPLACE доход WITH 43000	
REPLACE доход WITH 10000000	
REPLACE доход WITH 12500	
REPLACE доход WITH 120	
REPLACE доход WITH 12500	
REPLACE доход WITH 120	
REPLACE доход WITH 90000	
REPLACE доход WITH 2200	

Рисунок 10 (Результат выполнения команды)

Задание 6

Добавляем поле налог, поле налог 12% от дохода (Рисунок 11)

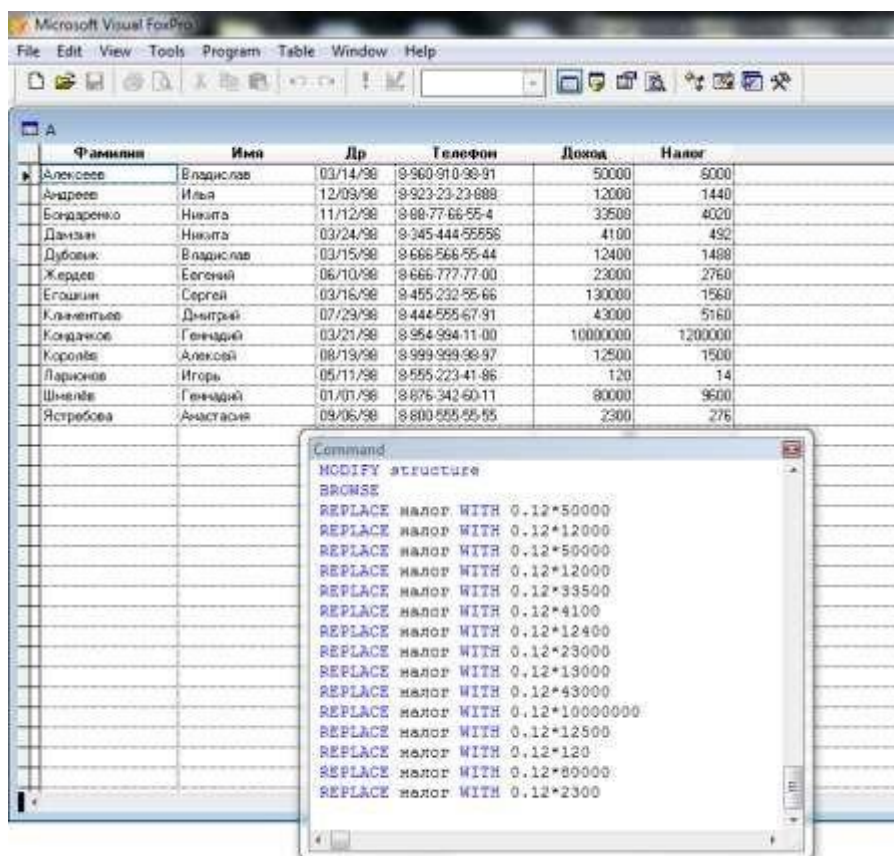


Рисунок 11 (Результат выполнения команды)

Контрольные вопросы:

1. К какому виду СУБД относится FoxPro?
2. С какими командами вы научились сегодня работать?
3. Как добавить новое поле в БД?

Отчет

Сохраните работу.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа- 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10 баллов	Защита своей работы персонально – 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Лабораторное занятие №15

Тема:Обработка транзакций. Использование функций защиты для БД. (2 часа)

Цель занятия:

1. Преобразование реляционной БД.

ИНСТРУКЦИОННАЯ КАРТА

Материально - техническое оснащение

Оборудование: компьютер, операционная система Windows 7, FoxPro

Характер выполнения работы: обучающиеся выполняют работу индивидуально.

Практические задания

Краткие теоретические сведения: Команды для создания меню:

DefineMenu<имя меню>color r/w – Создание меню

DEFINE PAD <имя пункта> OF <имя меню>PROMPT «название» – Создание горизонтального пункта

ON PAD <имя пункта> OF <имя меню> ACTIVATE POPUP <имя меню раскрывающегося> – Активация горизонтального пункта

DEFINE POPUP <имя меню раскрывающегося> – Создание раскрывающегося пункта

DEFINE BAR 1 OF <имя меню раскрывающегося> PROMPT «название» –Создание вертикального пункта

ON SELECTION BAR 1 OF <имя меню раскрывающегося>do<имя программы>– Активация вертикального пункта

Порядок выполнения практической работы:

1. Изучить теоретический материал.
2. Выполнить предлагаемые задания.
3. Ответить на контрольные вопросы и предоставить в тетради в виде отчета. Отчет должен включать:

- номер, наименование практической работы и тему; • ответы на контрольные вопросы; • выводы.

4. Выполненную работу и отчет по проделанной работе предъявить преподавателю.

Задания для выполнения практической работы:

Создание меню

1. Создать меню;
2. Создать в меню 3 PAD;
3. Создать в каждом PAD POPUP;
4. Создать в каждом POPUP 3 BAR;
5. 1 PAD и 2 PAD, а также в 3 PAD 1 и 2 BAR могут называться как угодно;
6. В 3 PAD 3 BAR должен обязательно называться.
7. Написать программу для BAR «Выход»; 1. Для начала создадим меню.

```
DEFINE MENU Дамзин_Меню COLOR r/w
```

Рисунок – 1 Создание меню.

2. После создания меню нужно добавить горизонтальные пункты.

```
DEFINE PAD p1 OF Дамзин_Меню PROMPT "Вид"
```

```
DEFINE PAD p2 OF Дамзин_Меню PROMPT "файл"
```

```
DEFINE PAD p3 OF Дамзин_Меню PROMPT "Настройки"
```

Рисунок – 2 Создание горизонтальных пунктов.

3. Создадим раскрывающийся пункт.

```
DEFINE POPUP pop1
```

```
DEFINE POPUP pop2
```

```
DEFINE POPUP pop3
```

Рисунок – 3 Создание раскрывающихся пунктов.

4. Теперь нам нужно чтобы при нажатии на горизонтальный пункт появлялся раскрывающийся пункт.

```
ON PAD p1 OF Дамзин_Меню ACTIVATE POPUP pop1
```

```
ON PAD p2 OF Дамзин_Меню ACTIVATE POPUP pop2
```

```
ON PAD p3 OF Дамзин_Меню ACTIVATE POPUP pop3
```

Рисунок – 4 Активация раскрывающихся пунктов.

5. Далее примемся за создание вертикальных пунктов в каждом раскрывающемся пункте.

```
DEFINE BAR 1 OF pop1 prompt "Окно"
DEFINE BAR 2 OF pop1 prompt "Окно (без рамки) "
DEFINE BAR 3 OF pop1 prompt "Полный экран"
DEFINE BAR 1 OF pop2 prompt "Экспорт"
DEFINE BAR 2 OF pop2 prompt "Импорт"
DEFINE BAR 3 OF pop2 prompt "Открыть"
DEFINE BAR 1 OF pop3 prompt "Инструменты"
DEFINE BAR 2 OF pop3 prompt "Информация"
DEFINE BAR 3 OF pop3 prompt "Выход"
```

Рисунок – 5 Создание вертикальных пунктов.

6. Напишем программу для закрытия нашего меню.

```
ON SELECTION BAR 3 OF pop3 do exitprog
PROCEDURE exitprog
* activate window wihod
CLOSE ALL
in_stat=.f.
DEACTIVATE POPUP m_menu
quit |
```

Рисунок – 6 Алгоритм закрытия меню.

7. Теперь нам нужно все это сохранить в файл *.prg (По умолчанию сохраняется в C:\Users\Имя пользователя\Documents\VisualFoxProProjects НА WIN 8.1!!!).

```
SAVE TO ProgramMetoda.prg
DO programmetoda.prg
```

Рисунок – 7 Создание файла *.prg.

8. Через пункт меню file выбираем пункт open и открываем наш файл *.prg и в него уже копируем все из командной строки и сохраняем комбинацией клавиш Ctrl+S.

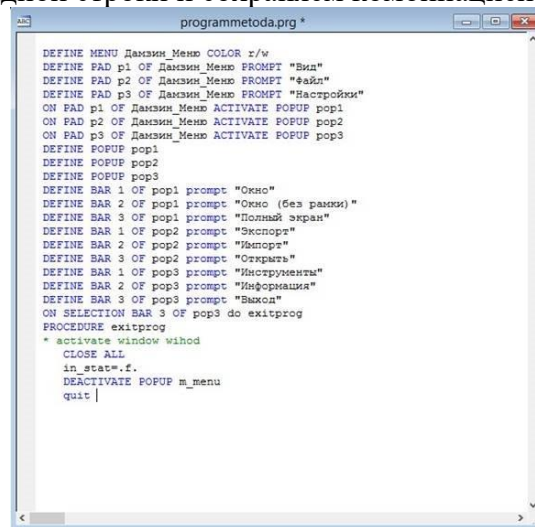


Рисунок – 8 Общий результат работы.

Чтобы проверить работоспособность программы нужно открыть пункт меню Program в нём выбрать Do... далее выбираем нашу программу и пишем ACTIVATE MENU <Название меню>.

Контрольные вопросы:

1. Что такое меню?
2. Перечислите этапы разработки меню в FoxPro?

Отчет

Сохраните в своей папке под названием Лабораторная работа 9.

Критерии оценивания отчета:

Своевременность и успешность составления отчета	Качество усвоенного материала	Творческая составляющая	Перевод баллов в оценку
Своевременно выполнена работа – 10 баллов	Точность выполнения работы – 10 баллов	Защита работы аудиторно – 20 баллов	70 б – оценка «5» 40-60 б – оценка «4» 30 баллов – оценка «3» Менее 20 баллов – оценка «2»
Досрочно выполненная работа – 20 баллов	Эстетичность выполнения работы – 10 баллов	Защита своей работы персонально – 10 баллов	
Не своевременно выполнена работа – 0 баллов	Полное содержание выполненной работы – 10 баллов	Оформление своей работы несколькими способами – 20 баллов	

Информационные источники

Печатные издания:

1. Иванченко А.Н. Теоретические основы разработки и реализации языков программирования. (Бакалавриат). Учебное пособие Кнорус, 2020.
2. Федорова Г. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности. Изд. Инфра-М-2019

Электронные издания (электронные ресурсы):

1. ЭБС «Научная электронная библиотека eLIBRARY» [Электронный ресурс]. – Режим доступа: <http://elibrary.ru/defaultx.asp>
2. ЭБС «Издательства Лань» [Электронный ресурс]. – Режим доступа: <http://e.lanbook.com/>
3. ЭБС «Университетская библиотека online» [Электронный ресурс]. – Режим доступа: www.biblioclub.ru
4. ЭБС «Библиокомплектатор» [Электронный ресурс]. – Режим доступа: (<http://www.bibliocomplectator.ru/>)
5. ЭБС «ЮРАЙТ» [Электронный ресурс]. – Режим доступа: (<http://biblio-online.ru>)
6. Свободный каталог периодики библиотек России [Электронный ресурс]. – Режим доступа: <http://ucpr.arbicon.ru/>

Дополнительные источники:

1. Батаев А.В. Операционные системы и среды. Учебник Изд. Академия. 2020г.