

КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени В.И.УЛЬЯНОВА-ЛЕНИНА

Факультет географии и экологии

Е.А.Костерина

РАБОТА В СУБД FOXPRO
Вводный курс

Учебно-методическое пособие

Издательство
Казанского государственного университета
2007

Печатается по решению учебно-методической комиссии
факультета географии и экологии
Казанского государственного университета

Составитель: кандидат физико-математических наук,
старший преподаватель Е.А.Костерина

Костерина Е.А.

Работа в СУБД FOXPRO. Вводный курс / Е.А.Костерина. – Казань: Изд-во
Казанск. гос. ун-та, 2007. – 46 с.

Учебно-методическое пособие к курсу «Информатика» для студентов 1 курса дневной формы обучения факультета географии и экологии, специальности «Экология», «Природопользование», «Геоэкология». Учебно-методическое пособие составлено с целью развить у студентов навыки работы в СУБД FoxPro и помочь им освоить основные команды языка программирования FoxPro на простых примерах. Рассматривается версия Microsoft Visual FoxPro 9.0.

Содержание

Стр.

| | | |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1. | Первое знакомство с СУБД FoxPro. Команды <i>set default to, quit</i> | 4 |
| 2. | Создание новой таблицы. Команда <i>create</i> | 7 |
| 3. | Использование ранее созданной таблицы. Рабочая область. Завершение сеанса работы в FoxPro. Команды <i>use, select, close database</i> | 9 |
| 4. | Заполнение таблицы. Изменение содержания таблицы: добавление новых записей, редактирование существующих записей, удаление записей. Команды <i>browse, pack, set talk on/off, clear</i> | 11 |
| 5. | Отображение календарных дат. Встроенные функции FoxPro. Команда ? вывода данных. Команды <i>set date to, set century on/off</i> . Функция <i>date()</i> | 12 |
| 6. | Работа с мемо-полями. Ввод и вывод информации из мемо-полей. Команды <i>append memo, copy memo</i> | 14 |
| 7. | Изменение структуры таблицы. Команда <i>modify structure</i> | 15 |
| 8. | Упражнение на закрепление пройденного материала | 16 |
| 9. | Указатель записи. Перемещение указателя записи. Команды <i>go top, go bottom, go, skip</i> . Встроенные функции, связанные с положением указателя записи: <i>recno(), reccount(), eof(), bof()</i> | 16 |
| 10. | Программа. Составление простой программы. Окно редактирования программы. Команды <i>modify command, do, return, wait</i> | 20 |
| 11. | Временные переменные. Ввод и вывод значений переменных. Команды <i>store, @...say/get, read</i> . Вывод значений полей. Функции <i>str(), dtoc()</i> . Начальные значения полей и переменных | 25 |
| 12. | Ввод значений полей с помощью команды <i>@...say/get</i> | 30 |
| 13. | Управление доступом к полям таблицы. Команда <i>browse</i> . Функция <i>IFF()</i> | 31 |
| 14. | Изменение содержания таблицы. Команды <i>append blank, replace, delete, recall, set deleted on/off</i> | 34 |
| 15. | Команды <i>blank, zap, erase</i> | 35 |
| 16. | Сортировка данных. Индексные файлы. Команды <i>sort, index on, set index to, set order to, reindex</i> | 36 |
| 17. | Поиск данных. Команды <i>locate, continue, seek, set near on</i> . Функция <i>found()</i> | 40 |
| 18. | Математическая обработка данных. Команда <i>calculate</i> и функции <i>avg(), cnt(), max(), min(), sum(), std(), var()</i> | 44 |
| | Литература | 45 |

1. Первое знакомство с СУБД FoxPro. Команды *set default to, quit*

FoxPro – это система управления базами данных, предоставляющая пользователю много различных возможностей. Под FoxPro можно понимать, в зависимости от контекста, язык программирования баз данных, основную выполняемую программу всей системы FoxPro, среду разработчика или пользователя баз данных (интерфейс) или всю систему в целом.

Основная программа СУБД FoxPro называется `vfp9.exe`. Она находится в папке, которая была выбрана при установке FoxPro на Ваш рабочий компьютер. Для удобства работы рекомендуется поместить на рабочий стол ярлык этой программы.

После запуска СУБД FoxPro вступают в действие ее настройки, такие как имя рабочей папки, наличие статусной строки в главном окне среды пользователя, формат отображения календарных дат, участие помеченных к удалению записей в дальнейшей обработке данных и другие. Эти настройки называются установленными по умолчанию. Их можно изменить временно, или можно задать новые настройки по умолчанию. Временные изменения будут иметь силу до их следующего изменения либо до окончания сеанса работы в FoxPro. При повторном запуске СУБД FoxPro опять действуют настройки по умолчанию.

Папка, куда помещаются файлы пользователя во время сеанса работы в FoxPro и где разыскиваются файлы пользователя, называется рабочей папкой. После запуска FoxPro важно сразу же правильно задать имя рабочей папки. Для этого предназначена команда **set default to** (установить путь по умолчанию).

Выполните следующие действия.

1. Поместите на рабочий стол ярлык программы `vfp9.exe`, если он отсутствует.
2. Укажите мышкой на ярлык, нажмите правую кнопку мыши и в контекстном меню выберите пункт «Свойства». В появившемся окне указано расположение файла `vfp9.exe` и имя рабочей папки, устанавливаемой по умолчанию. Убедитесь в этом. Закройте окно.
3. С помощью программы «Мой компьютер» создайте свою рабочую папку (обычно на диске D). В этой рабочей папке будут храниться все таблицы, программы и прочие файлы, создаваемые Вами во время работы в FoxPro.
4. Запустите СУБД FoxPro двойным щелчком мыши на ее ярлыке.

После запуска FoxPro перед Вами появляется главное окно среды пользователя. Оно содержит (при стандартной настройке) строку основного меню, панель инструментов «Стандартная», строку состояния (статусную строку), командное окно. Каждому действию по работе с таблицами, которое пользователь может выполнить, соответствует команда FoxPro. Команды записываются в командное

окно по одной. Каждая команда располагается в одной строке. Команда выполняется после нажатия клавиши Enter.

Команды, записанные в командном окне, сохраняются в нем в течение сеанса работы. Сохраняется все, что пользователь писал в окне – и правильные, и неправильные команды. После завершения сеанса работы в FoxPro текст, который был написан в командном окне, теряется (в младших версиях FoxPro) и сохраняется в версии 9.0. Чтобы очистить командное окно, нужно привести указатель мыши на поле окна, нажать правую кнопку мыши и в контекстном меню выбрать пункт Clear (Очистить).

Чтобы выполнить команду, нужно поставить курсор в любое место строки, содержащей команду, и нажать клавишу Enter. Таким образом, можно повторно использовать команды, набранные ранее и уже присутствующие в окне команд.

FoxPro не различает заглавные и строчные буквы.

Командное окно может отсутствовать на экране в двух случаях: выполняется какая-либо программа FoxPro или окно нечаянно было закрыто. Чтобы получить доступ к командному окну, в первом случае надо прервать выполнение программы, а во втором – воспользоваться главным меню FoxPro (Window – Command Window, то есть Окно – Окно команд).

Выполните следующие действия:

1. Закройте командное окно и откройте его снова через главное меню FoxPro.
2. Задайте свою рабочую папку. Для этого поместите курсор в командное окно, напишите в нем

set default to полный_путь_к_рабочей_папке
(Например, *set default to D:\2007\261\Petrov*)

и нажмите Enter. Если имя папки содержит пробелы, то его нужно взять в кавычки, например, *set default to "D:\2007\261 a\Petrov Ivan"*.

Можно указывать имя только уже созданной папки.

Если команда была записана синтаксически правильно и указанная папка существует, то FoxPro не выдаст никаких сообщений.

Если допущена синтаксическая ошибка (ошибка в написании фразы *set default to*), то появится сообщение «*Unrecognized command verb*» (см. пункт 10, стр.20). В этом случае следует нажать кнопку «Ok» в окне сообщения или клавишу Esc, поместить курсор в командное окно в то место команды, где допущена ошибка, исправить ошибку, нажать Enter.

Если неверно указаны имя папки или путь к ней, то может появиться сообщение о том, что папка не найдена («*Invalid path or file name*»). Для исправления ошибки следует поступить, как описано выше.

Обратим внимание еще раз на то, что если путь к папке или ее имя содержат пробелы, то полный путь должен быть взят в кавычки, например, *set default to "D:\2007\Петрова Вера"*, иначе команда может быть выполнена неверно, но сообщение об ошибке не появится.

Чтобы убедиться, что рабочая папка установлена верно, можно зайти в пункт основного меню File-Open (Файл-Открыть). Если в диалоговом окне в строке «Папка» указано верное имя, значит, все выполнено правильно. Закрываем диалоговое окно кнопкой «Отмена».

Чтобы было меньше ошибок, можно зайти в свою рабочую папку в программе «Мой компьютер». Тогда (при соответствующих настройках окна программы «Мой компьютер») в строке «Адрес» будет отражаться полный путь к папке. Его можно копировать (нажав комбинацию клавиш Ctrl-C или мышкой) и вставить в нужное место команды *set default to* (нажав комбинацию клавиш Ctrl-V или мышкой).

Можно писать по четыре буквы от каждого ключевого слова в команде (Например, *set defa to*). Однако полное написание помогает запомнить смысл команды.

Выполните следующие действия:

1. Выйдите из среды FoxPro. Для этого предназначена команда **Quit**. Можно просто закрыть главное окно FoxPro обычным способом с помощью мышки.

2. Создание новой таблицы. Команда *create*

Для создания новой таблицы предназначена команда **create**
create имя_таблицы (без указания расширения dbf имени файла)

Создадим таблицу следующего вида (по материалам [1], стр. 105).

Концентрация и общее количество газов в атмосфере

| Вещество | Обозначение | Общее кол-во в атмосфере (10 ⁹ т) | Дата поступления информации | Концентрация в чистом сухом воздухе на уровне море |
|----------------|------------------|----------------------------------------------|-----------------------------|----------------------------------------------------|
| Азот | N ₂ | 3 900 000 | 10 июня 1995 г. | 78.084 % по объему |
| Кислород | O ₂ | 1 200 000 | 12 мая 1995 г. | 20.9476 % |
| Аргон | Ar | 67 000 | 15 октября 1996 г. | 0.934 % |
| Водяной пар | H ₂ O | 14 000 | 10 июня 1995 г. | не учитывается |
| Оксид углерода | CO ₂ | 2 600 | 30 мая 1994 г. | 346 млн ⁻¹ |
| Неон | Ne | 65 | 25 февраля 1996 г. | 18.18 млн ⁻¹ |
| Оксид азота | NO ₂ | 0.013 | 2 марта 1994 г. | <0.02 млн ⁻¹ |
| Оксид азота | NO | 0.005 | 2 марта 1994 г. | следы |

Итак, следует создать новую базу данных. Она будет состоять из одной таблицы.

Во-первых, этой таблице следует дать имя. Желательно, чтобы это было одно слово из латинских букв и цифр, хотя допускаются названия, состоящие из нескольких слов и содержащие кириллические знаки. Пусть названием таблицы будет *Gazy_v_Atmosfere* (несколько слов соединены в одно за счет использования знака подчеркивания).

Во-вторых, чтобы создать таблицу, необходимо

- дать каждому столбцу название из одного слова (желательно из латинских букв и цифр, хотя допускается использование русского алфавита);
- определить тип данных в каждом столбце;
- определить, сколько знаков может занимать информация в каждом столбце;

- выбрать, какой столбец может однозначно определять (обозначать, выделять, идентифицировать) строку таблицы. Если такого столбца нет, его следует добавить. Такой столбец будет называться **первичным ключом**.

Вся перечисленная информация называется **структурой таблицы**.

Строка в таблице базы называется **записью**, а столбец – **полем**.

Охарактеризуем столбцы таблицы следующим образом.

| Название столбца в таблице | Содержание столбца | Название поля в таблице базы данных | Тип данных | Ширина поля |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------|-------------------------------------|------------------------|--------------------------------------------------------------------------------------------|
| Вещество | текст максимум из 14 символов | Veshestvo | Character (Символьный) | 15 символов |
| Обозначение | текст максимум из 3 символов | ChimObozn | Character | 5 символов |
| Общее кол-во в атмосфере | число, целая часть которого занимает максимум 7 позиций, а после точки встречается максимум 3 значащих цифры | Kolich | Numeric (Числовой) | ширина = 7+3+1 = 11 знаков с учетом десятичной точки, из них 3 знака после точки (Decimal) |
| Дата поступления информации | календарная дата | DataInfo | Date (Дата) | ширина задается FoxPro |
| Концентрация в чистом сухом воздухе на уровне море | информация произвольного типа и размера | KonzNaMore | Мемо (Мемо) | ширина задается FoxPro |

Нет столбца, однозначно выделяющего запись в таблице, поэтому создадим дополнительный столбец ID (от слова «идентификатор») типа Numeric (Числовой) размером 3 знака (целое число, следовательно, 0 знаков после десятичной точки).

| | | | | |
|----------------|-------------|----|--------------------|-------------------------------|
| Первичный ключ | целое число | ID | Numeric (Числовой) | 3 знака, из них 0 после точки |
|----------------|-------------|----|--------------------|-------------------------------|

Выполните следующие действия.

1. Запустите FoxPro.
2. Задайте свою рабочую папку.
3. С помощью команды **create Gazy_v_Atmosfere** создайте таблицу. После нажатия клавиши Enter появится окно создания структуры таблицы. В него следует ввести всю информацию о структуре, указанную выше, и нажать кнопку «Ok». После этого в младших версиях FoxPro появляется вопрос о том, будете ли Вы вводить сами данные («*Input data records now?*»). Следует ответить «Нет», поскольку форма ввода данных, которая могла бы быть предложена после ответа «Да», не самая удобная.
4. Обратите внимание, что теперь в статусной строке (внизу экрана) указано имя созданной таблицы (так как сейчас Вы с ней работаете и она активна), полный путь к ней, а также указано, что записей в ней нет (Record: None) и что она открыта эксклюзивно (Exclusive), то есть с ней работает единственный пользователь и он имеет право вносить любые изменения.
5. Не закрывая FoxPro, с помощью программы «Мой компьютер» убедитесь, что в Вашей рабочей папке появились файлы Gazy_v_Atmosfere.dbf (таблица) и Gazy_v_Atmosfere.fpt (содержание мемо-полей).
6. Вернитесь в FoxPro и закройте таблицу (отключите доступ к ней) командой **use**. Файл Gazy_v_Atmosfere.fpt также закроется.
7. Выйдите из среды FoxPro.

3. Использование ранее созданной таблицы. Рабочая область. Завершение сеанса работы в FoxPro. Команды *use, select, close database*

Создание таблицы – это создание файла с расширением DBF, в котором хранится структура таблицы и содержание таблицы (т.е. сами данные). Для создания таблицы, как уже известно, предназначена команда *create*.

При выполнении предыдущего упражнения Вами создана таблица Gazy_v_Atmosfere.dbf. Поскольку она не была заполнена, т.е. данные не были введены, в этом файле пока хранится только структура таблицы.

Если таблица создана, ее можно открыть, т.е. сделать доступной для работы в FoxPro. Для этого предназначена команда **use**:

use имя_таблицы (без указания расширения dbf имени файла)

Например, *use Gazy_v_Atmosfere*

Одновременно с таблицей автоматически открываются относящийся к ней файл с содержанием мемо-полей (файл с расширением fpt) и относящийся к ней мультииндексный структурный файл (файл с расширением sdx), если он создан. Вы будете создавать такой файл позже.

Можно одновременно работать с несколькими таблицами, т.е. открыть одновременно несколько таблиц. Каждая таблица при открытии помещается в

область памяти, называемую **рабочей областью**. В одной рабочей области может находиться одна и только одна таблица. Число рабочих областей велико, но ограничено, а их имена соответствуют буквам латинского алфавита. В каждый момент времени одна из рабочих областей является активной. Информация о находящейся в ней таблице отображается в статусной строке. Переход из одной рабочей области в другую осуществляется командой **select**:

select имя_рабочей_области_куда_перейти

При запуске СУБД FoxPro активной рабочей областью автоматически становится область *a*. Если к моменту открытия какой-то таблицы активная рабочая область уже занята другой таблицей, то расположенная в области таблица будет закрыта, и вновь открываемая таблица будет помещена в эту область.

Итак, можно одновременно работать с несколькими таблицами. В каждый момент времени будут доступны все эти таблицы, но активна будет только одна из них, а именно, та, которая открыта в активной рабочей области. Имя активной таблицы и ее расположение (путь к ней) отображаются в статусной строке.

По окончании работы с таблицей ее следует закрыть (т.е. отключить доступ FoxPro к ней). При этом сохраняется последняя версия структуры и содержания таблицы, а также соответствующие таблице *fmt*-файл и *cdx*-файл.

Для закрытия активной таблицы предназначена команда **use**. Активная таблица в каждый момент времени одна, поэтому имя таблицы не указывается. Для закрытия всех открытых таблиц одновременно предназначена команда **close database** (кроме того, существуют команды *close tables* и *close databases*).

Для завершения сеанса работы в FoxPro следует:

1. остановить работу всех выполняющихся программ пользователя;
2. закрыть все открытые окна (кроме командного окна);
3. закрыть все открытые таблицы;
4. выйти из FoxPro.

Выполните следующие действия.

1. Запустите FoxPro.
2. Задайте свою рабочую папку.
3. Проверьте, что действие выполнено верно, и в диалоговом окне, соответствующем пункту меню File-Open, в строке «Папка» отображается имя именно Вашей папки. Закройте это окно.
4. Откройте созданную ранее таблицу командой

use Gazy_v_Atmosfera

Таблица помещается в рабочую область *a*. Обратите внимание, что имя таблицы и путь к ней появились в статусной строке. Таблица доступна для работы и активна.

5. Закройте таблицу. Для этого напишите в командном окне слово *use* и нажмите Enter. Имя таблицы исчезнет из статусной строки.
6. Откройте таблицу снова.

4. Заполнение таблицы. Изменение содержания таблицы: добавление новых записей, редактирование существующих записей, удаление записей. Команды *browse, pack, set talk on/off, clear*

Существуют разные способы внесения данных в таблицу. Самый наглядный из них – заполнение таблицы, когда она представлена на экране именно как таблица. Получить такое представление можно с помощью команды **browse**.

Команда **browse** предназначена для просмотра и редактирования содержания активной открытой таблицы.

Чтобы добавить в таблицу новую запись (т.е. строку), следует нажать Ctrl-Y. Далее следует поставить курсор в то поле записи (т.е. в ту клетку таблицы), куда Вы хотите записать данные. Можно писать на русском и английском языках, переключая раскладку клавиатуры, как обычно.

Чтобы изменить имеющиеся данные, нужно ставить курсор в соответствующие поля и вносить изменения. При этом старые данные полностью стираются.

Для удаления какой-либо записи (строки) необходимо сделать следующее.

1. Поставить курсор в любое поле стираемой записи.
2. Отметить эту запись на удаление, нажав Ctrl-T, или щелкнув мышкой в прямоугольнике слева от записи. В обоих случаях у отмеченной записи этот прямоугольник станет черным.
3. Закройте browse-окно. Это окно можно закрыть, нажав на крестик в его правом верхнем углу либо нажав клавишу Esc. Кроме того, можно закрыть окно, нажав Ctrl-W, где W означает write (записать). В последнем случае гарантировано, что Вы сохраняете все изменения, которые Вы сделали в browse-окне.
4. Выполнить команду **pack**. Эта команда предназначена для физического удаления отмеченных записей.

Итак, после внесения всех необходимых изменений, следует закрыть browse-окно. Повторим, что это окно можно закрыть, нажав на крестик в его правом верхнем углу, либо нажав клавишу Esc. Кроме того, можно закрыть окно, нажав Ctrl-W, где W означает write (записать). В последнем случае гарантировано, что Вы сохраняете все изменения, которые Вы сделали в browse-окне.

Выполните следующие действия.

1. Убедитесь, что таблица *Gazy_v_Atmosfere.dbf* открыта. Если она не открыта, откройте ее. Откройте окно редактирования таблицы, т.е. выполните команду *browse*. Эта команда всегда применяется к открытой и активной в данный момент таблице, поэтому имя таблицы не указывается.
2. Заполните столбцы ID (пусть идентификатором записи будет ее порядковый номер при вводе), *Veshstvo*, *ChimObozn* (нижние и верхние индексы набрать нельзя, поэтому вместо N₂ пишите N2) и *Kolich*. Обратите внимание, что при каждом нажатии *Ctrl-Y* меняется информация в статусной строке. Например, *Record: 3/3* означает «Сейчас указатель записи (условно говоря, курсор) стоит в 3-ей записи из 3-х существующих в таблице». Если вы переставите курсор в другую строку, Вы увидите в статусной строке новую информацию *Record: номер строки с курсором/общее число строк в таблице*.
3. Закройте *browse*-окно, нажав *Ctrl-W*.
4. Удалите одну любую запись в таблице (как описано выше). Убедитесь, что она действительно удалена: во-первых, в статусной строке общее число записей стало на 1 меньше, во-вторых, в *browse*-окне можно видеть отсутствие ранее выделенной записи.
5. Добавьте эту запись вновь.
6. Закройте таблицу.

Возможно, что при выполнении команды *pack* на белом поле главного окна *FoxPro* появилось сообщение о выполнении этой команды. Такие сообщения не всегда желательны. Чтобы отключить их вывод на экран, следует использовать команду ***set talk off***. После ее выполнения никаких видимых изменений на экране не произойдет, но в дальнейшем во время текущего сеанса работы в *FoxPro* системные сообщения о выполнении команд на экране появляться не будут. Чтобы разрешить вывод таких сообщений, следует использовать команду ***set talk on***.

Для очистки поля главного окна *FoxPro* предназначена команда ***clear***.

5. Отображение календарных дат. Встроенные функции FoxPro. Команда ? вывода данных. Команды *set date to*, *set century on/off*. Функция *date()*

Календарные даты могут быть представлены в *FoxPro* в различном формате. По умолчанию формат дат имеет вид:

месяц/день/год_последние_два_знака

Такой формат называется американским (*American*). Для изменения формата отображения календарных дат предназначена команда ***set date to***.

В СУБД *FoxPro* имеется большое количество встроенных функций: математических, статистических, преобразования данных и др. Признаком функции являются круглые скобки, в которых указываются аргументы

функции. Даже если аргумент не нужно указывать явно, следует поставить пустые круглые скобки как обозначение использования функции. В частности, значением функции **date()** является текущая календарная дата.

Выполните следующие действия.

1. В командном окне наберите *?date()* и нажмите Enter. На белом поле главного окна FoxPro появится сегодняшнее число в описанном выше формате.

Знак *?* означает, что значение функции *date()* следует вывести на экран. Знак *?* – это одна из команд вывода данных.

Выводимые данные – это те данные, которые пользователь получает из памяти компьютера. Вывод – это передача данных от компьютера пользователю (печать на экране, печать на бумаге, запись в файл для хранения и др.).

Вводимые данные – это те данные, которые компьютер получает от пользователя в процессе работы. Ввод – это передача данных от пользователя в память компьютера (например, ввод с клавиатуры или чтение данных из указанного пользователем файла).

Выполните следующие действия.

1. Откройте таблицу *Gazy_v_Atmosfere.dbf*.
2. Откройте browse-окно.
3. Введите дату в первой строке таблицы. Согласно текущему формату отображения дат, следует ввести 06/10/95.
4. Закройте browse-окно, нажав Ctrl-W.

Можно установить более привычный и удобный формат отображения дат:

дата.месяц.год_последние_два_знака

Такой формат называется немецким (German). Как говорилось выше, для изменения формата отображения календарных дат предназначена команда **set date to**.

Выполните следующие действия.

1. Выполните команду *set date to german*.
2. Выполните команду *?date()*. Вы видите, что сегодняшнее число выглядит на экране так, как принято в нашей стране.

Чтобы год отображался четырьмя знаками (иногда это важно), следует выполнить команду **set century on**. Чтобы год отображался двумя знаками, следует выполнить команду **set century off**. По умолчанию обычно установлено *set century off*.

Выполните следующие действия.

1. Выполните команду *set century on*.

2. Выполните команду *?date()*. Убедитесь, что изменение формата произошло.

Итак, более привычный для России формат дат установлен. Заполните поле *DataInfo* таблицы *Gazy_v_Atmosfere.dbf*.

Выполните следующие действия.

1. Откройте browse-окно.
2. Заполните столбец *DataInfo*.
3. Закройте browse-окно, нажав Ctrl-W.

6. Работа с мемо-полями. Ввод и вывод информации из мемо-полей. Команды *append memo*, *copy memo*

Для того чтобы заполнить мемо-поле какой-либо записи, нужно двойным щелчком мыши открыть окно редактирования мемо-поля, ввести туда информацию и закрыть окно редактирования.

Заполните поле *KonzNaMore* таблицы *Gazy_v_Atmosfere.dbf*.

Выполните следующие действия.

1. Откройте browse-окно.
2. Заполните столбец *KonzNaMore*.
3. Закройте browse-окно, нажав Ctrl-W.

Содержание мемо-поля может быть добавлено в таблицу из файла или сохранено в файле.

Команда **append memo** предназначена для добавления данных из файла в мемо-поле (при этом к существующему содержанию мемо-поля добавляется содержимое файла):

append memo имя_мемо_поля *from* имя_файла_с_расширением

При использовании параметра *overwrite* существующие данные в мемо-поле будут удалены, и в поле будет помещено содержимое файла:

append memo имя_мемо_поля *from* имя_файла_с_расширением *overwrite*

Команда **copy memo** предназначена для копирования содержания мемо-поля во вновь создаваемый текстовый файл. Если не указано другое, файл получит расширение *txt*. Если задан режим *additive*, мемо-поле будет добавлено в конец существующего текстового файла.

copy memo имя_мемо_поля *to* имя_файла

или

copy memo имя_мемо_поля *to* имя_файла *additive*

Выполните следующие действия.

1. Создайте в текстовом редакторе «Блокнот» файл, содержащий слово «примечание». Сохраните файл в своей рабочей папке и назовите его *TestMemo.txt*. Закройте редактор.

2. Поставьте указатель записи на первую запись открытой таблицы *Gazy_v_Atmosfere.dbf*. Далее Вы будете работать с мемо-полем первой записи. Точно так же можно было бы работать с мемо-полем любой другой записи. Добавьте содержание созданного текстового файла к содержанию мемо-поля командой

append memo KonzNaMore from TestMemo.txt

3. Убедитесь, что содержание мемо-поля изменилось. Удалите слово «примечание» из мемо-поля.

4. Сохраните содержание мемо-поля в файле *TestCopyMemo.txt* командой

copy memo KonzNaMore to TestCopyMemo.txt

5. С помощью программ «Мой компьютер» и «Блокнот» убедитесь, что в Вашей рабочей папке появился новый файл с указанным именем и его содержание соответствует содержанию мемо-поля таблицы (первой записи).

6. Закройте browse-окно, если оно открыто.

7. Изменение структуры таблицы. Команда *modify structure*

Для изменения структуры таблицы, т.е. изменения названий, типа, ширины полей (столбцов), добавления новых полей или удаления лишних полей, предназначена команда **modify structure**. Она всегда применяется к открытой и активной в данный момент таблице, поэтому имя таблицы не указывается.

Дадим полю *KonzNaMore* новое имя *Komment*.

Выполните следующие действия.

1. Выполните команду *modify structure*.

2. В диалоговом окне внесите изменения.

3. Нажмите кнопку «ОК». Появится вопрос «*Make structure changes permanent?*» (Сохранить изменения?) Ответьте на вопрос.

4. Убедитесь с помощью команды *browse*, что изменения произошли.

5. Закройте таблицу.

Если в окне редактирования структуры сделаны случайные ошибочные изменения, их следует отменить, нажав кнопку «Отмена» или клавишу Esc. Появится вопрос «*Discard changes?*» (Отменить изменения?) Если изменения были ошибочны, следует ответить «Да».

8. Упражнение на закрепление пройденного материала

Создайте и заполните новую таблицу, указанную ниже.

Среднее содержание металлов в пылеобразных частицах дыма мусоросжигательной печи ([1], стр.159)

| Состав пыли | Концентрация мг/м ³ |
|-------------|-----------------------------------|
| Алюминий | 12.056 |
| Цинк | 3.080 |
| Свинец | 1.760 |
| Медь | 0.185 |
| Олово | 0.167 |
| Кадмий | 0.071 |
| Хром | 0.044 |
| Ртуть | 0.001 |

9. Указатель записи. Перемещение указателя записи. Команды *go top*, *go bottom*, *go*, *skip*. Встроенные функции, связанные с положением указателя записи: *recno()*, *reccount()*, *eof()*, *bof()*

Каждой таблице при открытии ставится в соответствие указатель записи. Он выделяет среди всех записей таблицы ту, к которой в данный момент возможен непосредственный доступ. Номер этой записи отображается в статусной строке FoxPro. Например, Record: 3/8 означает, что в таблице, открытой в активной в данный момент рабочей области, всего 8 записей, а доступна третья запись (указатель записи стоит на записи, которая является третьей согласно порядку занесения записей в таблицу).

Каждая запись в таблице имеет номер, соответствующей порядковому номеру записи при добавлении (вводе) ее в таблицу. Например, запись, которая была добавлена в таблицу пятой, всегда будет иметь меньший порядковый номер, чем запись, добавленная десятой. Если, например, шестая и седьмая записи будут удалены из таблицы, оставшиеся записи будут перенумерованы по порядку и пятая запись останется пятой, а бывшая десятая станет восьмой.

Кроме того, в процессе работы с таблицей записи могут быть предъявлены в порядке, отличающемся от порядка добавления записей. Это связано с понятием «индексирование» и будет подробно обсуждаться позже. В таком случае каждая запись имеет номер согласно временно установленному порядку, а также номер согласно порядку ввода.

Важно помнить, в каких случаях или при выполнении каких команд указатель подчиняется порядку ввода записей, а в каких – установленному рабочему порядку.

Порядок записей и перемещение указателя

| Команда / действие | Движение указателя: указатель записи перемещается | |
|--------------------------------------------------|-----------------------------------------------------------------------|------------------------------------------|
| Открытие таблицы | на первую запись | согласно установленному рабочему порядку |
| go top | на первую запись | согласно установленному рабочему порядку |
| go bottom | на последнюю запись | согласно установленному рабочему порядку |
| go номер | на запись с указанным номером | согласно порядку ввода записей |
| skip число | на указанное число записей вверх или вниз относительно текущей записи | согласно установленному рабочему порядку |
| Выполнение цикла (операторы for, do while, scan) | | согласно установленному рабочему порядку |
| Поиск данных (команды locate, seek) | | согласно установленному рабочему порядку |

Выполните следующие действия.

1. Запустите FoxPro и задайте свою рабочую папку, либо закройте все открытые таблицы и перейдите в рабочую область *a*.
2. Откройте таблицу *Gazy_v_Atmosfere.dbf*. Поскольку индексирование не используется, порядок записей соответствует порядку их ввода в таблицу. В статусной строке указано Record: 1/8, то есть указатель записи стоит на первой записи из восьми.
3. Выполните команду *browse* и убедитесь, что при открытии browse-окна выделенной является первая запись. В browse-окне всегда выделенной является та запись, на которой стоит указатель записи. Переместите курсор на несколько строк вниз. Убедитесь, что информация в статусной строке соответствует новому положению указателя. Закройте browse-окно.

4. Выполните команду *go top*, проследите за новым положением указателя по информации в статусной строке и по содержанию browse-окна. Закройте browse-окно.
5. Выполните команду *go bottom*, проследите за новым положением указателя по информации в статусной строке и по содержанию browse-окна.
6. Выполните команду *go 5*, проследите за новым положением указателя по информации в статусной строке и по содержанию browse-окна. Указатель записи переместился на пятую запись согласно порядку ввода записей.
7. Выполните команду *skip*. Указатель записи переместится на одну запись вниз.
8. Выполните команду *skip 2*. Указатель записи переместится еще на две записи вниз и будет указывать на восьмую (последнюю) запись.
9. Выполните команду *skip -3*. Указатель записи переместится на три записи вверх и будет указывать на пятую запись.
10. Выполните команду

index on Veshstvo to Veshstvo.idx

В результате будет создан индексный файл *Veshstvo.idx*, в котором будет храниться информация о том, как следует переставить записи таблицы, чтобы записи в ней были упорядочены по названию вещества по алфавиту. Записи перестроятся в этом порядке, и указатель записи будет перемещаться теперь согласно этому новому порядку. (Подробнее об индексировании Вы узнаете позже.)

11. Выполните команду *browse* и убедитесь, что записи располагаются теперь в новом порядке. Указатель записи стоит на первой записи согласно новому их порядку.

12. Выполните команду *go bottom*. Указатель переместится на запись, последнюю по алфавиту и расположенную сейчас в восьмой строке таблицы. Обратите внимание, что номер записи в статусной строке не 8, а 5, то есть это номер записи согласно порядку ввода записей.

13. Выполните команду *go 3*. Номер текущей записи согласно информации в статусной строке равен 3, но в browse-окне эта запись не третья по порядку. Таким образом, команда *go* перемещает указатель всегда согласно порядку добавления записей в таблицу.

В FoxPro есть встроенные функции, значения которых связаны с количеством записей в таблице и с положением указателя записи таблицы, открытой в активной рабочей области.

| Имя функции | Тип значения функции | Результат |
|--------------------------------|-----------------------|-------------------------------------------------------------------------------------|
| recno() «record's number» | Numeric (Числовой) | номер записи, на которую указывает указатель записи, согласно порядку ввода записей |
| reccount() «records' count» | Numeric (Числовой) | число записей в таблице, включая помеченные к удалению |

| | | |
|--------------------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eof() «end of file» | Logical (Логический) | true (.T.), если достигнут конец файла, то есть указатель записи стоит ниже последней записи (формально, номер записи равен числу записей таблицы+1) false (.F.), если указатель записи находится внутри таблицы |
| bof() «begin of file» | Logical (Логический) | true (.T.), если достигнуто начало файла, то есть указатель записи стоит выше первой записи (формально, номер записи равен 1) false (.F.), если указатель записи находится внутри таблицы |

Выполните следующие действия.

1. Отмените упорядоченное по алфавиту расположение записей командой
set order to
2. Убедитесь, что порядок предъявления записей соответствует порядку их ввода в таблицу.
3. Поставьте указатель записи на запись номер 5. Убедитесь, что информация в статусной строке соответствует положению указателя Record: 5 / 8.
4. Выполните команду *?recno()*, что означает «вывести на экран номер текущей записи». На экране FoxPro вы увидите значение функции *recno()*.
5. Выполните команду *?eof()*, что означает «вывести на экран ответ на вопрос о том, вышел ли указатель за нижнюю границу таблицы». На экране FoxPro вы увидите значение функции *eof()*.
6. Выполните команду *?reccount()*, что означает «вывести на экран количество записей в таблице». На экране FoxPro вы увидите значение функции *reccount()*.
7. Выполните команду *go top*. Убедитесь, что указатель стоит на самом верху таблицы. Выполните команду *skip -1*, то есть передвиньте указатель на 1 запись вверх (за пределы таблицы). Убедитесь, что значение функции *bof()* равно *true*, а значение функции *eof()* равно *false*, поскольку указатель вышел за верхнюю границу таблицы. Убедитесь, что номер текущей записи равен 1, хотя на самом деле указатель не указывает ни на одну запись.
8. Выполните команду *go bottom*. Убедитесь, что указатель стоит на нижней записи таблицы. Выполните команду *skip*, то есть передвиньте указатель на 1 запись вниз (за пределы таблицы). Убедитесь, что значение функции *bof()* равно *false*, а значение функции *eof()* равно *true*, поскольку указатель вышел за нижнюю границу таблицы. Убедитесь, что номер текущей записи равен 9, то есть на единицу больше реального числа записей.
9. Закройте таблицу.
10. Выйдите из FoxPro.

10. Программа. Составление простой программы. Окно редактирования программы. Команды *modify command*, *do*, *return*, *wait*

Программа – это последовательность команд, которые выполняются как единое целое.

Текст программы, написанной в среде FoxPro, хранится в файле с расширением prg.

Для создания текста программы предназначен внутренний встроенный текстовый редактор FoxPro. Он вызывается командой **modify command**:

modify command имя_программы (без указания расширения prg имени файла)

В результате выполнения этой команды открывается окно текстового редактора FoxPro. Если программный файл с указанным именем существует в текущей рабочей папке, то его текст отображается в окне редактора, а имя файла отображается в строке заголовка окна редактора. Если программный файл с указанным именем не существует в текущей рабочей папке, такой файл будет в ней создан, имя файла появится в строке заголовка окна редактора, само окно будет пустым.

Если нужно выйти из редактора с гарантированным сохранением текста программы, следует нажать Ctrl-W. При этом текст программы сохраняется в prg-файле, окно редактора закрывается.

После того, как текст программы составлен, программу можно выполнить. Для запуска программы на выполнение предназначена команда **do**, которая пишется в командном окне:

do имя_программы (без указания расширения prg имени файла)

Если текст программы вновь создан или в существующем тексте были сделаны изменения, то при запуске программы FoxPro спросит, сохранить ли текущий текст программы («*Do you want to save changes to* имя_Вашей_программы?»). Следует ответить «Да».

Предыдущий вариант текста программы будет сохранен в той же папке, в файле с тем же именем и с расширением bak. Откомпилированный программный файл будет храниться там же и иметь расширение fhr.

При выполнении программы окно редактора не закрывается, если только его не закрыть специально.

Пока программа выполняется, командное окно недоступно. После успешного выполнения программы управление передается системе FoxPro (активным

может стать командное окно или окно редактора, курсор будет стоять в активном окне).

Признаком конца программы и последней командой в ней является команда **return**. Именно она возвращает управление системе FoxPro.

Если в программе есть ошибка, то выполнение будет прервано и на экране появится окно с сообщением об ошибке. Чтобы продолжить работу в FoxPro, это окно нужно закрыть.

Может появиться одно из перечисленных ниже сообщений об ошибке. В любом случае рекомендуется в качестве ответа на это сообщение нажимать клавишу Esc. В этом случае выполнение программы будет прервано, и курсор встанет в ту строку текста программы, где обнаружена ошибка. Следует исправить ошибку и запустить программу на выполнение повторно.

Список сообщений об ошибках

| Сообщение | Смысл сообщения |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unrecognized command verb | Команда не распознана |
| Command contains unrecognized phrase / keyword | Команда содержит нераспознанное ключевое слово |
| Syntax error | Синтаксическая ошибка |
| Missing comma | Отсутствует запятая |
| File does not exist | Не найдена таблица, процедура или программа с указанным именем |
| Operator / operand type mismatch | Действие производится над данными такого типа, для которого это действие невозможно произвести (т.е. действие не определено) |
| Data type mismatch | Несоответствие типов данных (в частности, см. в предыдущей строке) |
| Nesting error | Ошибка вложения (относится к структурам if...endif, do while...enddo, scan...endscan и т.п.). Например, у оператора if отсутствует заключительное слово endif, или оно расположено неверно, или оно неверно написано. |
| Variable is not found | Не найдено поле или переменная с указанным именем |

Создайте программу, которая открывает таблицу *Gazy_v_Atmosfere.dbf*, показывает ее содержимое, позволяет его редактировать и удалять некоторые строки (по Вашему желанию), затем закрывает таблицу.

Выполните следующие действия.

1. Запустите FoxPro.

2. Задайте свою рабочую папку.

3. Вызовите редактор текста программы, дав программе имя *Atmosfera_example1.prg*. Для этого в командном окне напишите

modify command Atmosfera_example1

и нажмите клавишу Enter. Появится окно редактора, а в его заголовке будет написано *Atmosfera_example1.prg*.

4. В окне редактора наберите нижеследующий текст программы (как Вы обычно набираете в «Блокноте» или в редакторе Word).

use Gazy_v_Atmosfere

browse

pack

use

return

5. Запустите эту программу, поместив курсор в командное окно, написав там

do Atmosfera_example1

и нажав Enter. При этом FoxPro спросит, сохранить ли текст программы. Ответьте «Да».

Команды, составляющие программу выполняются последовательно сверху вниз.

- В результате выполнения команды *use Gazy_v_Atmosfere* в статусной строке появится имя таблицы и путь к ней.

- В результате выполнения команды *browse* на экране появится browse-окно. В нем можно производить любые действия по изменению содержания таблицы и отметить записи на удаление (если нужно).

- Если были отмеченные на удаление записи, команда *pack* удалит их. Если не было записей, отмеченных на удаление, то команда *pack* выполнится, удалив пустое множество записей, т.е. на самом деле ничего не произойдет. В любом случае в статусной строке на некоторое время появится сообщение «скопировано, т.е. сохранено в таблице, столько-то записей», например, «8 records copied».

- Команда *use* закроет таблицу, имя таблицы исчезнет из статусной строки.

- Команда *return* вернет управление системе FoxPro, видимым выражением этого будет курсор в командном окне или в окне редактора.

С помощью программы «Мой компьютер» убедитесь, что в Вашей рабочей папке присутствуют файлы *Atmosfera_example1.prg* (текст программы) и *Atmosfera_example1.fxp* (результат компиляции программы).

6. Запустите программу еще раз (так же, как и в первый раз). Поскольку текст программы не менялся, вопроса о сохранении текста не будет.

7. Добавьте после команды *pack* команду *browse*, чтобы видеть содержание таблицы после удаления записей. Запустите программу.

С помощью программы «Мой компьютер» убедитесь, что в Вашей рабочей папке появился файл *Atmosfera_example1.bak* (это предыдущий текст программы с одним оператором *browse*).

8. Закройте окно с текстом программы, поместив туда курсор и нажав *Ctrl-W*.

Запустить программу на выполнение можно еще тремя способами, которые в неявной форме содержат в себе то же самое выполнение команды *do*:

- В основном меню FoxPro выбрать Program – Do имя_программы (Программа – Do имя_программы).
- Находясь в окне с текстом программы (в любом месте), нажать *Ctrl-E*.
- На панели инструментов нажать на кнопку ! (восклицательный знак). Эта кнопка соответствует запуску команды *do* для той программы, которая находится в активном в настоящий момент окне редактирования программы.

Поскольку команды выполняются быстро, не всегда можно заметить результат работы некоторых из них. Например, при выполнении написанной выше программы моментально выскакивает *browse*-окно, после закрытия которого курсор сразу оказывается в командном окне или окне редактора.

Принудительно прервать работу программы можно в любой момент, нажав на клавишу *Esc*. Появится сообщение «*Interrupted*» («Выполнение прервано»).

Чтобы временно приостановить работу программы после какой-либо команды, существует команда ***wait***. Эта команда выдает на экран сообщение «*Press any key to continue*» («Для продолжения нажмите любую клавишу») и ожидает нажатия клавиши. После нажатия любой клавиши выполнение программы продолжается.

Команда *wait* (как и многие другие команды, выдающие сообщения) выдает сообщение на основном экране FoxPro, начиная с левого верхнего угла. Обычно эта часть экрана закрыта окном редактора текста программы, поэтому перед запуском любой программы, осуществляющей печать сообщений на экране, целесообразно отодвинуть окно с текстом программы направо вниз.

Выполните следующие действия.

1. Откройте написанную ранее программу и создайте ее копию под названием *Atmosfera_example2.prg*. Для этого воспользуйтесь пунктом основного меню FoxPro «*File-Save as*» (Файл – Сохранить как)
2. Между каждыми двумя командами вставьте команду *wait*.
3. Запустите программу и проследите результат выполнения каждой команды, составляющей программу:

- *use Gazy_v_Atmosfera* - в статусной строке появляются имя таблицы и путь к ней
 - *wait* – на экране появляется сообщение, следует нажать любую клавишу
 - *browse* – на экране появляется browse-окно
 - *wait* – после закрытия browse-окна на экране появляется сообщение, следует нажать любую клавишу
 - *pack* – если были отмеченные на удаление записи, то после их удаления изменится общее число записей в таблице, что отображается в статусной строке (в любом случае появляется сообщение о копировании записей)
 - *wait* – на экране появляется сообщение, следует нажать любую клавишу
 - *browse* - на экране появляется browse-окно
 - *wait* – после закрытия browse-окна на экране появляется сообщение, следует нажать любую клавишу
 - *use* - имя таблицы исчезает из статусной строки
 - *wait* – на экране появляется сообщение, следует нажать любую клавишу
 - *return* – курсор появляется в командном окне или в окне редактора
4. Закройте окно с текстом программы.

В программе полезно писать комментарии о том, что делает та или иная часть программы, и прочее. Комментарий – это произвольный текст на любом языке, он никак не влияет на выполнение программы. Кроме того, если при отладке программы нужно временно исключить какую-то ее часть, то эту часть программы лучше закомментировать, а не стирать.

Если комментарий помещается в отдельной строке, эта строка должна начинаться со знака *. Если комментарий располагается в той же строке, что и команда, то комментарий предваряют два знака &&. Ниже помещены пример программы с комментариями (слева) и ее исполняемая часть (та же программа без комментариев, справа)

| | |
|---------------------------------------------|-----------------------------|
| * Открываем таблицу <i>Gazy_v_Atmosfera</i> | |
| <i>use Gazy_v_Atmosfera</i> | <i>use Gazy_v_Atmosfera</i> |
| <i>browse &&</i> редактирование | <i>browse</i> |
| <i>pack &&</i> удаление | <i>pack</i> |
| * Закрываем таблицу | |
| <i>use</i> | <i>use</i> |
| <i>return &&</i> конец программы | <i>return</i> |

Выполните следующие действия.

1. Создайте копию программы *Atmosfera_example1.prg* под названием *Atmosfera_example3.prg*
2. Напишите комментарии к программе.
3. Запустите программу.
4. Закройте окно с текстом программы.

До сих пор Вам встречались команды, которые удобно записывать в одну строку. Команда, вообще говоря, может быть очень длинной. Такие команды при составлении программы можно записывать в несколько строк. Для переноса части команды на следующую строку в тексте программы в конце текущей строки ставится знак «;». В командном окне каждая команда должна быть записана в одну строку.

Умение записывать команду в несколько строк понадобится при изучении темы следующего пункта.

Выполните следующие действия.

1. Самостоятельно напишите программу, показывающую окно редактирования структуры таблицы «Среднее содержание металлов в пылеобразных частицах дыма мусоросжигательной печи» (см. пункт 8, стр. 15) и позволяющую редактировать содержание этой таблицы.

11. Временные переменные. Ввод и вывод значений переменных. Команды *store*, *@...say/get*, *read*. Вывод значений полей. Функции *str()*, *dtoc()*. Начальные значения полей и переменных

Переменная – это именованная область памяти. Таким образом, у переменной есть имя. Кроме того, у переменной есть тип. Имя и тип переменной определяются в момент появления переменной во время сеанса работы в FoxPro. Имена даются переменным по тем же правилам, что и полям. Переменные могут иметь те же типы, что и поля. Тип переменной определяется типом последнего присваиваемого ей выражения. По завершении работы программы (если переменная создана внутри программы) или сеанса работы (если переменная создана в командном окне) СУБД FoxPro «забывает» всю информацию о переменных. Поэтому они и называются временными (в отличие от данных, хранящихся в таблицах). При необходимости временные переменные могут быть сохранены в файле с расширением **mem** и использованы в разных сеансах работы в FoxPro (об этом можно прочитать в [2], глава 13, стр.115).

Значение может быть присвоено переменной двумя командами:

переменная = выражение

или

store выражение *to* имена переменных

Если несколько переменных должны принять одинаковое значение, то предпочтителен второй вариант.

Выполните следующие действия.

1. Закройте все открытые окна FoxPro, кроме командного, и все открытые таблицы.

2. В командном окне запишите команду

$$a=2.5$$

и нажмите Enter. Создана переменная a числового типа, хранящая число 2.5.

3. Выведите на экран значение переменной a командой $?a$.

4. Числовые переменные могут иметь представление как с фиксированной, так и с плавающей точкой. Представление с плавающей точкой применяется в основном для изображения очень больших или очень маленьких чисел. Например, $0.005=5*10^{-3}$ =(по правилам языка FoxPro)=5E-3.

Выполните команду $a=5E-3$ и выведите на экран значение переменной a .

Выполните команду $a=-1.34E4$ и выведите на экран значение переменной a .

5. Выполните команду $a='2'$. Переменная a поменяла тип и приняла значение символьного типа. Признаком константы символьного типа являются обрамляющие апострофы. Выведите на экран значение переменной a . Обратите внимание на то, что хотя на экран выведено символьное выражение, обрамляющие апострофы не выводятся.

6. Выполните команду $stroka='Привет!'$ и выведите значение переменной $stroka$ на экран.

7. Присвойте переменной dat значение типа Date, а именно, 30 мая 1904. Для того чтобы формат представления дат был удобным, выполните команды

set date to german

set century on

Установленный формат представления дат будет действовать при выводе календарных дат (т.е. значений типа Date) и, обычно, при вводе констант типа Date. Если при задании значения переменной типа Date в установленном формате все же произойдет ошибка, следует ввести это значение в формате {^год-месяц-день}. Следует помнить, что константы типа Date обязательно заключаются в фигурные скобки. Итак, выполните команду

$dat=\{30.05.1904\}$

или

$dat=\{^1904-05-30\}$

Выведите значение переменной dat на экран. Обратите внимание, что формат вывода даты соответствует установленному: {день.месяц.год}.

8. Константы логического типа записываются как .T. (true, истина) или .F. (false, ложь). Наличие обрамляющих точек обязательно. Присвойте переменной $flag$ значение «истина» и выведите значение переменной на экран.

9. Очистите экран командой *clear*.

Белый экран (поле основного окна FoxPro) разделен на строки и столбцы. Нумерация строк идет сверху вниз, а нумерация столбцов слева направо. Заданием номеров строки и столбца задается позиция на экране.

Команда **@...say/get** позволяет выводить данные на экран, начиная с любой заданной позиции и в установленном пользователем формате, а также позволяет вводить данные. Позиция на экране задается после знака **@**. Вывод описывается частью команды после слова *say*, а ввод – частью команды после слова *get*. У команды **@...say/get** очень много возможностей. Краткие ее варианты таковы:

@ номер строки, номер столбца *say* что_вывести *get* что_ввести
read
или

@ номер строки, номер столбца *say* что_вывести
или

@ номер строки, номер столбца *get* что_ввести
read

Команда *read* позволяет изменять значения в области ввода на экране и помещает вводимые данные в память компьютера после нажатия клавиши Enter. Команда *read* обязательно должна присутствовать после команды, содержащей *get*-часть, либо после группы таких команд.

Выполните следующие действия.

1. Выведите на экран слово «Привет» командой

@ 2,1 say 'Привет!'

Курсор помещается во вторую строку, первый столбец экрана и, начиная с этой позиции, печатается слово «Привет». После ключевого слова *say* должно следовать символьное выражение. Слово «Привет» является константой символьного типа, признак которого – обрамляющие апострофы.

2. Введите с экрана любое слово, состоящее из 10 или менее букв. Сохраните это слово в переменной *slovo*.

- Создайте новую программу *sayget1.prg* командой

modify command sayget1

- В окне редактирования программы наберите

clear

@ 3,1 say 'Введите любое слово из 10 или менее букв ';

get slovo default ' ' size 1,10

read

return

Как видно, команда **@...say/get** записана в две строки (в месте перехода на новую строку знак «;»), поэтому необходимо создание программы.

- Переместите окно редактора направо вниз, так чтобы был виден левый верхний угол поля основного окна FoxPro.

- Запустите программу на выполнение.

Курсор будет поставлен в позицию 3,1 на экране и, начиная с этой позиции, будет выведен текст «Введите любое слово из 10 или менее букв ». Обратите внимание, что после слова «букв» стоит пробел. Это сделано для того, чтобы

текст не сливался с полем ввода значения переменной *slovo*. Часть команды *default* ‘ ‘ означает, что начальным значением переменной *slovo* является символьное выражение, состоящее из символа «пробел». Часть команды *size 1,10* говорит, что поле ввода значения переменной *slovo* занимает одну строку, где можно набрать максимум 10 символов.

Итак, необходимо было создать новую переменную символьного типа и присвоить ей значение, набранное пользователем на экране. Имя переменной указывается после ключевого слова *get*. Тип переменной и начальное значение задаются опцией *default*. За поле ввода, то есть за максимальную длину слова, отвечает параметр *size*.

3. Выведите на экран значение переменной *slovo*. Для этого добавьте в программу команду *@ 4,1 say slovo* перед командой *return* и выполните программу.

4. Закройте окно редактирования текста программы.

5. Заведите (создайте) новую переменную *num* числового типа и задайте с экрана ее значение в предположении, что число должно быть отрицательным и может не быть целым. Для этого создайте и выполните программу следующего содержания.

```
@ 5,1 say 'Введите число ' get num;  
default 0 picture '###.###' valid (num<0)  
read  
return
```

Опция *default* определяет тип и начальное значение переменной. Опция *picture* задает формат числа. В данном случае, целая часть числа (включая знак) занимает максимум 3 позиции, и после десятичной точки – максимум 3 позиции. Параметр *valid* содержит условие на допустимое вводимое значение.

6. Выведите на экран значение переменной *num*. Для этого воспользуйтесь командой

```
@ 6,1 say 'num = '+str(num,7,3)
```

После ключевого слова *say* должно стоять символьное выражение. В данном случае оно составлено из пояснения и символьного представления значения числовой переменной *num*. Функция *str* преобразует числовое выражение в символьную строку. Эта функция имеет следующие параметры:

```
str(параметр1, параметр2, параметр3),
```

где параметр1 – числовое выражение;

параметр2 – общее число знаков (включая десятичную точку);

параметр3 – число знаков после десятичной точки.

7. Закройте окно редактирования текста программы.

8. Создайте новую переменную *datap* типа *Date* и присвойте ей значение. Пусть дата должна быть не ранее 1995 года.

```
@ 7,1 say 'Введите дату ' get datap default {};  
valid (datap>={^1995-01-01})  
read  
return
```

Следует учесть текущий формат представления дат. Начальным значением переменной *datap* будет пустая дата. Фигурные скобки являются признаком типа Date. Поскольку дата пустая, между скобками нет никаких знаков.

9. Выведите на экран значение переменной *datap*.

```
@ 8,1 say 'Значение переменной datap: '+dtoc(datap)
```

Функция *dtoc* преобразует свой аргумент (выражение типа Date) в символьное выражение.

10. Закройте окно редактирования текста программы.

11. Создайте новую переменную *z* логического типа и задайте ее значение. Эта переменная может принимать только два значения: .T. или .F.

```
@ 9,1 say 'Введите Т (истина) или F (ложь)' get z default .T.
```

```
read
```

```
return
```

12. Не существует функции, преобразующей константу логического типа в символьное выражение, поэтому для вывода значения переменной *z* с помощью команды *@...say/get* необходимо воспользоваться оператором *if...endif* ([2], глава 14, стр.117). Вывести значение переменной *z* можно командой *?z*. Сделайте это. Результат будет отображен внизу экрана FoxPro. Закройте окно редактирования текста программы.

Если требуется создать новую переменную и задать ее значение, то наличие опции *default* обязательно. Рекомендуется указать в ней наиболее вероятное значение создаваемой переменной, либо «никакое» значение (как, например, пустая дата).

Если требуется ввести с экрана значение уже существующей переменной, то опцию *default* использовать не нужно, так как переменная уже имеет определенный тип и какое-то значение.

Начальное значение переменной – это значение, присваиваемое ей в момент создания. Когда в таблицу добавляется новая запись, ее поля тоже принимают свои начальные значения. Это те значения, которые поля имеют до того, как пользователь присвоит им какие-либо значения.

Команда *@...say/get* позволяет вводить и выводить значения полей той записи таблицы, на которой стоит указатель.

Выполните следующие действия.

1. Очистите экран командой *clear*.

2. Откройте таблицу *Gazy_v_Atmosfere.dbf*.

3. Выполните команду *browse* и поставьте указатель записи на вторую запись. Запомните значения полей во второй записи и закройте *browse*-окно.

4. Выведите значения полей *Veshestvo*, *Kolich* и *DataInfo* с помощью команды *@...say/get*. Они выводятся, как значения переменных. Все это можно сделать в одной программе.

5. Выполните команду *browse* и добавьте в таблицу новую запись. Не заполняйте запись и выведите значения ее полей на экран. Это будут начальные значения полей вновь добавленной записи. Убедитесь, что они соответствуют таблице ниже.

| Тип поля | Начальное значение |
|------------------------|----------------------------------------------------------------------|
| Numeric (Числовой) | 0 |
| Character (Символьный) | ' ' (два апострофа – пустая строка, не содержащая ни одного символа) |
| Date (Дата) | { } (пустая дата) |

6. Удалите пустую строку таблицы.

7. Закройте таблицу.

Подробнее о команде *@...say/get* читайте в [2] (глава 12, стр. 96).

12. Ввод значений полей с помощью команды *@...say/get*

Для редактирования таблицы, кроме команды *browse*, существуют и другие возможности. В частности, можно заполнять поля записи с помощью команды *@...say/get*. Для этого нужно поставить указатель записи на запись, поля которой необходимо заполнить. После этого можно вводить значения полей, как значения переменных, с тем отличием, что опция *default* в этом случае не используется.

Выполните следующие действия.

1. Очистите экран командой *clear*.

2. Откройте таблицу *Gazy_v_Atmosfere.dbf*.

3. Выполните команду *browse* и добавьте новую запись в таблицу. Убедитесь, что указатель записи указывает на новую запись. Не меняйте его положение, чтобы в последующем вводить данные именно в эту запись. Закройте *browse*-окно.

4. С помощью команды *@...say/get* внесите в новую строку приведенные ниже данные. Для этого составьте и выполните программу. Предусмотрите, что количество газа в атмосфере – число положительное, и учтите число позиций до и после десятичной точки. Для заполнения мемо-поля используйте команду *append memo* (см. пункт 6, стр. 14).

| Вещество | Обозначение | Общее кол-во в атмосфере (10 ⁹ т) | Дата поступления информации | Концентрация в чистом сухом воздухе на уровне море |
|-------------|-------------|----------------------------------------------|-----------------------------|----------------------------------------------------|
| Оксид азота | NO | 0.005 | 2 марта 1995 г. | следы |

5. Выполните команду *browse* и убедитесь, что новая строка заполнена верно.

6. Удалите созданную в этом упражнении строку.
7. Закройте таблицу.

13. Управление доступом к полям таблицы. Команда *browse*. Функция *IIF()*

Команда *browse* предназначена для просмотра и редактирования таблицы и имеет большое число дополнительных возможностей.

Можно предъявлять таблицу на экране и при этом

- давать browse-окну желаемый заголовок (по умолчанию заголовком окна является название таблицы, отображенной в окне) (параметр *title*);
- разрешать любые действия, кроме добавления новых записей (параметр *noappend*);
- разрешать любые действия, кроме редактирования существующих записей (параметр *noedit*);
- разрешать любые действия, кроме пометки записей к удалению (параметр *nodelete*);
- показывать только записи, удовлетворяющие определенному условию (параметр *for*);
- показывать не все поля, а только перечисленные в списке (параметр *fields*);
- показывать поля, которых нет в таблице, но значения которых вычисляются по значениям других полей (вычисляемые поля);
- ограничивать доступ к некоторым полям и проверять значения, имеющиеся либо вводимые в поля;
- и другое (см. [2])

Выполните следующие действия.

1. Откройте таблицу *Gazy_v_Atmosfere.dbf*.
2. Предъявите эту таблицу для просмотра, т.е. без возможности вносить какие-либо изменения, с заголовком «Просмотр» командой

browse title 'Просмотр' noappend noedit nodelete

Убедитесь, что изменить, добавить или отметить записи на удаление невозможно. Убедитесь, что в заголовке окна вместо названия таблицы написано слово «Просмотр». Закройте browse-окно.

3. Предъявите только те записи таблицы, где количество газа в атмосфере больше 1. Дайте окну заголовок «Выбор записей по условию». Для этого выполните команду

browse title 'Выбор записей по условию' for Kolich>1

Закройте browse-окно.

4. Предъявите только те записи таблицы, где дата поступления информации позже 1994 года.
5. Предъявите только те записи таблицы, где количество газа в атмосфере больше 1 или дата поступления информации позже 1994 года. Соедините два

условия в одно логическое выражение с помощью логической операции *.or.* (обрамляющие точки обязательны) (см. пункт 17, стр.40).

6. Предъявите все поля таблицы, кроме мемо-поля, и добавьте к ней столбец, в котором поставьте знак «+», если количество газа в атмосфере в текущей записи меньше 10000. Новый столбец будет вычисляемым полем. Назовите его *Flag*. Воспользуйтесь командой

browse fields Id, Veshestvo, ChimObozn, Kolich,DataInfo,
Flag=IFF(Kolich<10000,'+', ' ')

Параметр *fields* команды *browse* используется, если требуется показать на экране не все поля таблицы, а лишь некоторые. Поля, которые нужно показать, перечисляются после слова *fields* через запятую.

Функция ИФ совпадает по смыслу с функцией «Если» в табличном процессоре Excel. Она имеет три параметра:

- а) условие;
- б) значение функции, если условие выполняется;
- в) значение функции, если условие не выполняется.

В приведенном выше примере значение функции будет присвоено полю *Flag*. В каждой строке таблицы условие *Kolich<10000* будет проверено, и поле *Flag* будет заполнено.

Значение вычисляемого поля может быть получено по любой формуле, связывающей значения полей текущей записи и, возможно, значения каких-либо переменных и констант. Использование функции *IFF* или другой функции не является обязательным.

Значением вычисляемого поля нельзя воспользоваться при выполнении каких-либо действий, в отличие от значений тех полей таблицы, которые входят в ее структуру. Создание и предъявление вычисляемого поля на экране – это форма вывода информации.

Для создания и предъявления вычисляемого поля необходимо использование в команде *browse* параметра *fields*. Описание вычисляемого поля при перечислении полей в команде *browse* имеет вид *имя_поля=формула_для_вычисления_значения_поля*. В примере выше это описание имеет вид *Flag=IFF(Kolich<10000,'+', ' ')*.

Выполните следующие действия.

1. Откройте окно редактора FoxPro командой *modify command browse_prog* (будет создана программа *browse_prog.prg*) и запишите следующие команды *set century on*

```
browse fields ID :H='Идентификатор' :R, ;  
          Veshestvo :H='Вещество' :10, ;  
          Kolichestvo :H='Количество газа' :B=0.001,10000000, ;  
          DataInfo :H='Дата поступления информации' ;  
                  :W=(DataInfo<{^1995-01-01}) ;  
                  :V=(DataInfo<{^2000-01-01})
```

Это означает, что предъявить в browse-окне нужно не все, а только перечисленные поля таблицы, причем каждому полю в browse-окне будет дано название, указанное с помощью ключа *H*, а также

- поле *ID* можно будет просматривать, но не редактировать ключ (:R);
- видимая ширина поля *Veshestvo* на экране будет 10 символов, хотя доступны для работы будут все 15 символов (ключ : число);
- в поле *Kolichestvo* можно будет вводить только числа от 0.001 до 10000000 (ключ :B);
- поле *DataInfo* можно будет редактировать, только если в поле стоит дата раньше 1995 года (ключ :W), а ввести можно будет только дату не позднее 1999 года (ключ :V).

Ключи :*B* и :*V* задают условия на значения, вводимые в поле. Они могут применяться к полям разного типа. Если указанным условиям должны удовлетворять не только вновь вводимые, но и уже введенные данные, то для проверки существующих данных следует после ключей с условиями добавить :*F*. В таком случае, если курсор попадет в поле, значение которого не соответствует условиям, выйти из поля можно будет только после исправления значения поля.

Обратите внимание на расстановку знаков препинания, а именно, внутри описания одного поля нет запятых, а при переходе от описания ключей одного поля к описанию другого поля ставится запятая. Точка с запятой ставится при переходе на новую строку внутри одной команды независимо от того, в каком месте команды осуществляется разрыв строки.

Выполните следующие действия.

1. Запустите составленную программу на выполнение, дав ей подходящее по смыслу имя, добавьте в таблицу новую запись и проверьте действие ключей.
2. Удалите добавленную запись.
3. Закройте окно редактирования текста программы.
4. Закройте таблицу.

14. Изменение содержания таблицы. Команды *append blank*, *replace*, *delete*, *recall*, *set deleted on/off*

Как известно, команда *browse* позволяет добавлять записи в таблицу, изменять значения полей и помечать записи на удаление. Все эти действия можно выполнить иначе, без использования команды *browse*.

Команда **append blank** предназначена для добавления в таблицу новой пустой записи.

Команда **replace** предназначена для присвоения значения полю той записи, на которую указывает указатель записи:

replace поле *with* значение

Значение может вычисляться по формуле, или оно может быть задано константой.

Пометить запись на удаление можно с помощью команды **delete**. Снять пометку записи на удаление можно с помощью команды **recall**.

Обычно помеченные к удалению записи участвуют в обработке данных до тех пор, пока не будут удалены командой *pack*, то есть обычно установлено **set deleted off**. Чтобы исключить помеченные к удалению записи из дальнейшей работы до выполнения команды *pack*, нужно использовать команду **set deleted on**.

Выполните следующие действия.

1. Откройте таблицу *Gazy_v_Atmosfere.dbf*. Обратите внимание на то, сколько записей содержит таблица (это указано в статусной строке).
2. Добавьте в таблицу новую запись командой *append blank*. Обратите внимание, что информация в статусной строке изменилась. Кроме того, указатель записи переместился на новую запись.
3. Внесите в добавленную запись данные, приведенные ниже.

| Вещество | Обозначение | Общее кол-во в атмосфере (10 ⁹ т) | Дата поступления информации | Концентрация в чистом сухом воздухе на уровне море |
|-------------|-------------|----------------------------------------------|-----------------------------|----------------------------------------------------|
| Оксид азота | NO | 0.005 | 2 марта 1995 г. | следы |

Например, для заполнения поля *Veshestvo* выполните команду
replace Veshestvo with 'Оксид азота'

4. Выполните команду *browse* и убедитесь, что запись заполнена верно. Закройте *browse*-окно.

5. Пометьте третью запись таблицы к удалению. Для этого поставьте указатель записи на третью запись (команда *go*, пункт 9, стр. 16) и выполните команду *delete*. С помощью команды *browse* убедитесь, что нужная запись помечена, и закройте *browse*-окно.
6. Выполните команду *set deleted on*. Убедитесь, что помеченная к удалению запись стала невидимой.
7. Выполните команду *go 3*. Хотя запись номер 3 не участвует в обработке данных, она существует в таблице. Указатель будет установлен на нее. Об этом свидетельствует информация в статусной строке. Команда *?Veshstvo* выведет на экран название вещества «Аргон» из третьей записи. Однако, если выполнить команду *browse*, указатель автоматически перейдет на следующую по порядку запись. Убедитесь в этом. Закройте *browse*_окно. Очистите экран.
8. Снимите с третьей записи пометку к удалению. Для этого поставьте указатель записи на третью запись и выполните команду *recall*. Убедитесь, что хотя установлено *set deleted on*, третья запись стала видимой.
9. Выполните команду *set deleted off*.
10. Закройте таблицу.

15. Команды *blank*, *zap*, *erase*

Возникают ситуации, когда необходимо не удалить запись, а очистить одно или несколько ее полей. Для этого предназначена команда **blank**.

Команда **zap** позволяет удалить все записи в таблице, открытой в активной рабочей области. При этом структура таблицы сохраняется.

Команда **erase** позволяет полностью удалить любой закрытый файл – таблицу, программу или что-то еще:

erase имя_файла_с_расширением

Выполните следующие действия.

1. Откройте таблицу *Gazy_v_Atmosfere.dbf*.
2. Установите указатель на последнюю запись. Очистите в ней поля *Kolich* и *DataInfo* командой

blank fields Kolich, DataInfo
3. Выполните команду *browse* и убедитесь, что поля очищены. Закройте *browse*-окно.
4. Очистите текущую запись всю целиком. Для этого выполните команду *blank*. Убедитесь, что ожидаемый результат достигнут.
5. Закройте таблицу.
6. Создайте новую таблицу с именем *zap_erase.dbf* и с произвольной структурой. С помощью программы «Мой компьютер» убедитесь, что таблица расположена в Вашей рабочей папке. Внесите в эту таблицу несколько записей с помощью команды *browse*. Закройте *browse*-окно.

7. Поскольку перед созданием таблицы активной рабочей областью была область *a*, то новая таблица открыта и находится в области *a*. Таким образом, таблица *zap_erase.dbf* находится в активной рабочей области. Об этом можно судить и по тому, что информация о таблице отображается в статусной строке. Удалите все записи таблицы, выполнив команду *zap*. С помощью команды *browse* убедитесь, что ожидаемый результат достигнут. Закройте таблицу.

8. Удалите файл, содержащий таблицу, командой

erase zap_erase.dbf

С помощью программы «Мой компьютер» убедитесь, что таблица удалена из Вашей рабочей папки.

16. Сортировка данных. Индексные файлы. Команды *sort, index on, set index to, set order to, reindex*

Сортировка таблицы – физическое упорядочение файла (создается новый файл, в котором записи таблицы идут в новом порядке).

Для сортировки открытой и активной таблицы предназначена команда **sort**.

sort to имя_новой_таблицы (без расширения dbf) *on* поле /A

sort to имя_новой_таблицы *on* поле /A *fields* список_полей

sort to имя_новой_таблицы *on* поле1 /A, поле2 /A *fields* список_полей

В результате этой команды создается новый файл с указанным именем. Эта новая таблица не открывается автоматически. Для ее просмотра следует открыть ее в свободной рабочей области и выполнить команду *browse*.

Опция /A означает «упорядочить записи по возрастанию (ascending) значения в указанном поле». Вместо этого можно написать /D – по убыванию – descending. Кроме того, можно указать /C – игнорировать регистр букв (строчные/заглавные). Можно указать для одного поля два ключа, например, /DC.

Можно упорядочивать по нескольким полям (двум и более). В этом случае сначала произойдет упорядочивание по значению поля 1, а внутри одинаковых значений поля 1 – по значению поля 2, и т.д.

В новую сортированную таблицу можно копировать не все поля исходной таблицы. В этом случае после слова *fields* следует указать (через запятую) имена копируемых полей.

Выполните следующие действия.

1. Откройте таблицу *Gazy_v_Atmosfere.dbf*. Она будет открыта в рабочей области *a*.

2. Отсортируйте ее по полю *Veshestvo* по алфавиту командой

sort to Gazy_Veshestvo on Veshestvo /A

С помощью программы «Мой компьютер» убедитесь, что возник новый файл *Gazy_Veshestvo.dbf*.

3. Вернитесь в FoxPro и просмотрите содержание новой таблицы:

- перейдите в рабочую область *b* командой *select b*
- откройте таблицу *Gazy_Veshestvo.dbf*
- выполните команду *browse*. Обратите внимание, что строки с одинаковым названием «Оксид азота» идут в том же порядке, как и в исходной таблице.

4. Закройте таблицу *Gazy_Veshestvo.dbf* и вернитесь в область *a* командой *select a*. В области *a* открыта таблица *Gazy_v_Atmosfere.dbf*. Убедиться в этом можно по информации в статусной строке.

5. Отсортируйте таблицу по полю *Veshestvo* по алфавиту, а затем одновременно по полю *Kolich* по возрастанию

sort to Gazy_Veshestvo_Kol on Veshestvo /A, Kolich /A

6. Посмотрите результат (как в пункте 3). Обратите внимание, строки с «Оксидом азота» поменялись местами.

7. Отсортируйте таблицу *Gazy_v_Atmosfere.dbf* по полю *DataInfo* по возрастанию. Посмотрите на результат.

8. Отсортируйте таблицу *Gazy_v_Atmosfere.dbf* по полю *DataInfo* по убыванию. Посмотрите на результат.

9. Отсортируйте таблицу *Gazy_v_Atmosfere.dbf* по полю *DataInfo* по убыванию и скопируйте в новую таблицу только поля *Veshestvo* и *Kolich*. Посмотрите на результат.

10. Закройте все открытые объекты и закончите работу.

Индексирование таблицы – создание индексного файла, использование которого позволяет предъявлять записи таблицы в желаемом порядке и перемещаться по таблице согласно этому порядку (при неизменности файла, хранящего таблицу).

Индексный ключ – поле или комбинация полей, по которым проводится индексирование (упорядочивание).

Индексные файлы бывают двух типов.

- Обычный индексный файл. Он имеет расширение *idx* и содержит один индексный ключ (результат одного упорядочивания).
- Мультииндексный файл. Он имеет расширение *cdx* и может хранить результаты нескольких действий по упорядочиванию. Результат каждого упорядочивания хранится в отдельном теге.

Мультииндексные файлы бывают двух видов.

- Обычный мультииндексный файл (то, что описано выше).
- Структурный мультииндексный файл – мультииндексный файл, название которого совпадает с названием упорядочиваемой таблицы и который открывается, изменяется и закрывается вместе с этой таблицей.

Индексный файл создается при выполнении команды **index on**. Чтобы пользоваться результатом индексирования, индексный файл нужно открыть (команда **set index to**).

Обычный индексный файл или обычный мультииндексный файл следует открывать после открытия соответствующей таблицы и закрывать перед закрытием таблицы.

Если в таблице были сделаны изменения, то открытые индексные файлы автоматически обновятся. Если же какой-то индексный файл не был открыт в момент внесения изменений в таблицу, то после открытия индексного файла его следует обновить командой **reindex**.

Структурный мультииндексный файл открывается, перестраивается и закрывается вместе с таблицей, которой он соответствует. Информацию о том, что содержит структурный мультииндексный файл, можно видеть в окне редактирования структуры таблицы на вкладке «Indexes» (это окно вызывается командой *modify structure*).

Чтобы воспользоваться результатом индексирования, необходимо не только открыть индексный файл, но и назначить главный индекс (команда **set order to**). Переназначая главный индекс, можно «переупорядочивать» таблицу в зависимости от того, что необходимо сделать.

Забегая вперед, следует подчеркнуть, что основное назначение индексирования не в том, чтобы предъявить таблицу на экране в удобной форме, а в том, чтобы обеспечить работу некоторых команд, с которыми Вы познакомитесь позже.

Цель нижеследующих заданий – научить создавать, открывать, переназначать индексы.

Выполните следующие действия.

1. Откройте таблицу *Gazy_v_Atmosfere.dbf*. Выполните команду *browse* и запомните порядок записей в таблице.
2. Ранее, в пункте 9 (стр. 16), был создан индексный файл *Veshestvo.idx*. Откройте его командой

set index to Veshestvo.idx

3. Обновите файл, выполнив команду *reindex*.
4. Сделайте открытый индекс главным с помощью команды

set order to Veshestvo

Здесь слово *Veshestvo* соответствует имени индексного файла, а не названию поля, по которому произведено индексирование.

5. С помощью команды *browse* убедитесь, что записи таблицы предъявляются в новом порядке, так что названия веществ расположены по алфавиту. Закройте *browse*-окно.

6. Отмените назначение главного индекса с помощью команды

set order to

С помощью команды *browse* убедитесь, что записи таблицы расположены теперь в первоначальном порядке. Закройте browse-окно.

7. Закройте индексный файл командой

set index to

Убедитесь, что теперь команда *set order to Veshestvo* не работает, и появляется сообщение об ошибке «*Table has no index order set*» (т.е. «У таблицы нет индекса»).

8. Определите количество записей в таблице с помощью функции *reccount()*.

9. Проиндексируйте таблицу по полю *Kolich*

index on Kolich to K.idx

В статусной строке появится сообщение о том, что все записи таблицы проиндексированы. С помощью программы «Мой компьютер» убедитесь, что в Вашей рабочей папке появился файл *K.idx*. Сразу после создания индексный файл открыт, индекс является главным. С помощью команды *browse* убедитесь, что записи таблицы предъясвляются в новом порядке, так что количество вещества упорядочено по возрастанию. Закройте browse-окно.

10. Индексный файл *K.idx* сейчас открыт, соответствующий индекс главный. Откройте также файл *Veshestvo.idx*. Открытие второго индексного файла привело к тому, что главный индекс сменился, и записи расположились по названию вещества по алфавиту. Убедитесь в этом с помощью команды *browse*.

11. Назначьте главным индекс по количеству вещества. Проверьте результат с помощью команды *browse*.

12. Отмените назначение главного индекса с помощью команды *set order to*. С помощью команды *browse* убедитесь, что записи таблицы расположены теперь в первоначальном порядке.

13. Закройте индексные файлы командой *set index to*. Убедитесь, что теперь команда *set order to Veshestvo* не работает, и появляется сообщение об ошибке «*Table has no index order set*» (т.е. «У таблицы нет индекса»). Команда *set order to K* также не работает, и появляется сообщение об ошибке «*Variable 'K' is not found*» (т.е. «Переменная 'К' не найдена»). Ошибки возникают, поскольку файлы *Veshestvo.idx* и *K.idx* закрыты. Сообщения об ошибках разные, поскольку в таблице есть поле с названием *Veshestvo* и нет поля с названием *K*.

В индексном файле с расширением *idx* можно хранить только результаты упорядочивания по возрастанию. Если требуется упорядочивать записи по убыванию, то следует работать с мультииндексными файлами.

Выполните следующие действия.

1. Откройте таблицу *Gazy_v_Atmosfere.dbf*, если она закрыта.

2. Проиндексируйте ее по полю *Veshestvo* по возрастанию и поместите результат индексирования в тег *VeshAsc* файла *Gazy_Mult.cdx*:

index on Veshestvo tag VeshAsc of Gazy_Mult.cdx

С помощью программы «Мой компьютер» убедитесь, что в Вашей рабочей папке появился файл *Gazy_Mult.cdx*.

3. Проиндексируйте таблицу по полю *Veshestvo* по убыванию и поместите результат индексирования в тег *VeshDesc* файла *Gazy_Mult.cdx*:

index on Veshestvo tag VeshDesc of Gazy_Mult.cdx descending

Поскольку файл *Gazy_Mult.cdx* был создан ранее (неважно, был он открыт к моменту выполнения очередной команды *index on* или нет), новый файл не создается, индексирование проходит успешно.

4. Расположите записи таблицы по названию вещества по алфавиту:

set order to VeshAsc

Проверьте результат с помощью команды *browse*.

5. Расположите записи таблицы по названию вещества от Я до А:

set order to VeshDesc

Проверьте результат с помощью команды *browse*.

6. Отключите использование индексов:

set order to

Проверьте результат с помощью команды *browse*.

7. Закройте файл *Gazy_Mult.cdx* командой

set index to

8. Проиндексируйте таблицу по полю *DataInfo* по возрастанию и поместите результат индексирования в тег *DataAsc* структурного мультииндексного файла:

index on DataInfo tag DataAsc

С помощью программы «Мой компьютер» убедитесь, что в Вашей рабочей папке появился файл *Gazy_v_Atmosfere.cdx*.

9. Закройте таблицу. Структурный мультииндексный файл закроется вместе с ней.

10. Откройте таблицу *Gazy_v_Atmosfere.dbf*. Структурный мультииндексный файл откроется вместе с ней. Назначьте главным индекс по полю *DataInfo*:

set order to DataAsc

Проверьте результат с помощью команды *browse*.

11. Закройте таблицу.

В заключение заметим, что имя тега может совпадать с именем поля.

17. Поиск данных. Команды *locate*, *continue*, *seek*, *set near on*. Функция *found()*

Поиск данных, удовлетворяющих простому условию, относящемуся к одному полю одной таблицы в базе данных (а именно, проверка на равенство или неравенство без использования операций «И» и «ИЛИ») может осуществляться двумя способами:

- с использованием индекса по полю, к которому относится условие (команда ***seek***);
- без использования индексов (команда ***locate***).

В остальных случаях проще пользоваться командой *locate*.

Команда **locate** имеет вид

locate for условие

Здесь «условие» – это логическое выражение, которое может объединять в единое целое несколько простых условий. При составлении логического выражения могут быть использованы логические операции. Обратите внимание, что знаки логических операций (кроме !) окаймляются точками или пробелами.

| Логические операции | | |
|---------------------|----------|-----------------|
| Приоритет операции | Операция | Запись операции |
| 1 | НЕ | .not. или ! |
| 2 | И | .and. |
| 3 | ИЛИ | .or. |

Команда **seek** имеет вид

seek значение

Здесь «значение» – это искомое значение того поля таблицы, индекс по которому назначен главным.

Преимуществом использования команды *seek* по сравнению с использованием команды *locate* является высокая скорость поиска, что особенно важно при работе с базами данных большого объема.

Рассмотрим подробнее команду *locate*.

Команда *locate* осуществляет поиск нужной записи методом последовательного перебора записей и их проверки, начиная с первой (согласно текущему рабочему порядку) записи и до тех пор, пока не будет найдена подходящая запись. Указатель записи будет указывать на первую встреченную запись, удовлетворяющую условиям поиска, если она существует, или на конец файла, если нужной записи нет. (Подробнее см. в [2], глава 9, стр.58)

Функция **found()** позволяет узнать, был ли поиск успешен. Она принимает значение .T. (истина), если запись, удовлетворяющая условиям поиска, существует, и принимает значение .F. (ложь), если такой записи нет.

Может оказаться, что в таблице есть несколько записей, удовлетворяющих условиям поиска. В таком случае в результате работы команды *locate* указатель записи остановится на первой из них. Чтобы найти следующую запись, удовлетворяющую тем же условиям, необходимо воспользоваться командой **continue**.

Выполните следующие действия.

1. Откройте таблицу *Gazy_v_Atmosfere.dbf*.
2. Найдите записи, соответствующие оксиду азота. Для этого выполните команду

locate for Veshestvo='Оксид азота'

Указатель записи переместится на седьмую запись. Следовательно, поиск был успешен.

3. Убедитесь еще раз с помощью функции *found()*, что нужная запись найдена, выполнив команду

? found()

4. Проверьте, что название вещества в найденной записи – это «Оксид азота» и что указатель записи указывает именно на первую такую запись. Для этого воспользуйтесь командой *browse*. Закройте *browse*-окно.
5. Найдите следующую запись, соответствующую оксиду азота, с помощью команды *continue*. Убедитесь, что такая запись существует.
6. Попробуйте найти следующую такую запись. Убедитесь по информации в статусной строке, что такой записи нет. Убедитесь в этом еще раз, проверив значение функции *found()*.

Если необходимо найти запись, удовлетворяющую нескольким условиям или одному из нескольких условий, то для объединения этих условий в одно логическое выражение используются логические операции, перечисленные в таблице выше. Желательно каждое условие заключать в скобки. Порядок выполнения логических операций определяется их приоритетами, то есть при прочих равных условиях выполняются все операции «НЕ», потом все операции «И» и наконец все операции «ИЛИ». Изменить это можно, заключив в скобки те части логического выражения, которые должны быть вычислены первыми.

Выполните следующие действия.

1. Найдите запись, где в обозначении вещества есть буква N, а информация поступила позднее 1995 года:

locate for ChimObozn='N'.and. DataInfo>{^1995-12-31}

2. Найдите запись, где общее количество газа находится в пределах от 100 до 10 000 единиц измерения:

locate for Kolich>100 .and. Kolich<10000

3. Найдите запись, где обозначение вещества содержит символ O или общее количество газа находится в пределах от 100 до 10 000 единиц измерения:

locate for (ChimObozn='O') .or. ((Kolich>100) .and. (Kolich<10000))

Первой подходящей записью будет та, что содержит информацию о количестве кислорода.

4. Найдите запись, где обозначение вещества содержит символ A и при этом общее количество газа меньше 100 либо не менее 10 000 единиц измерения:

locate for (ChimObozn='A') .and. ((Kolich<100) .or. (Kolich>=10000))

5. Закройте таблицу.

Обратите внимание еще раз, что когда условие поиска записи состоит из нескольких условий, скобки показывают порядок выполнения логических операций. Наличие скобок в таких случаях всегда желательно, а в случае задания 4 – строго обязательно.

Рассмотрим теперь подробнее команду *seek*.

Команда **seek** осуществляет поиск заданного значения методом половинного деления в том поле, индекс по которому является главным. Если в поле поиска указанное значение найдено, то поиск успешен (указатель записи указывает на нужную запись), иначе поиск не успешен (указатель записи указывает на конец файла). Функция *found()* принимает соответствующие значения.

Если требуется найти записи, значение поля поиска которых меньше или больше заданного значения, нужно воспользоваться тем фактом, что индекс по полю главный и в момент поиска значения в поле упорядочены (обычно по возрастанию), так что записи, где значения поля меньше искомого, будут располагаться строго выше найденной записи, а записи, где значения поля больше искомого, – строго ниже найденной записи.

Если нужно искать записи, где значение поля меньше или больше величины, которой в поле нет, необходимо воспользоваться приближенным поиском. Указатель записи будет установлен на первую по порядку запись, где значение поля больше заданного значения. При приближенном поиске функция *found()* имеет значение .F. всегда, кроме случая, когда достигнуто точное совпадение значения поля и искомой величины. В этом последнем случае поиск хотя и был приближенным, результат его такой же, как при точном поиске.

Приближенный поиск включается командой **set near on** и отключается командой **set near off**. По умолчанию приближенный поиск выключен. Если его включить, он будет действовать для всех последующих команд *seek*, пока приближенный поиск не будет отключен.

Выполните следующие действия.

1. Откройте таблицу *Gazy_v_Atmosfere.dbf*.
2. Найдите вещество, общее количество которого в атмосфере составляет $2\,600 \times 10^9$ т. Для этого назначьте главным индекс по полю *Kolich*, используя результаты пункта 16 (стр. 36).

set index to K.idx
seek 2600

3. Убедитесь с помощью функции *found()*, что поиск был успешен. Проверьте, что найдено вещество «Оксид углерода».
4. Найдите вещество, общее количество которого более $10\,000 \times 10^9$ т. Первое такое вещество «Водяной пар». Для выполнения этого задания назначать главный индекс не надо, потому что главный индекс по полю поиска сделан

главным в задании выше и это назначение не было отменено. Поскольку значения 10 000 в поле *Kolich* нет, следует воспользоваться приближенным поиском.

set near on
seek 10000

5. Убедитесь по информации в статусной строке и с помощью команды *browse*, что вещество «Водяной пар» найдено. Проверьте значение функции *found()*.
6. Отключите приближенный поиск командой *set near off*.
7. Закройте индексный файл командой *set index to*.
8. Закройте таблицу.

18. Математическая обработка данных. Команда *calculate* и функции *avg()*, *cnt()*, *max()*, *min()*, *sum()*, *std()*, *var()*

Команда **calculate** имеет вид

calculate функция *for* условие_отбора_строк_для_обработки *to*
переменная_ответ

и позволяет обрабатывать значения полей таблицы с помощью следующих функций:

| Функция | Комментарий |
|----------------------------|-----------------------------------------------------|
| <i>avg</i> (матем. выр-ие) | среднее арифметическое для выражения с полем |
| <i>cnt</i> () | число записей |
| <i>max</i> (выражение) | максимальное значение выражения от поля любого типа |
| <i>min</i> (выражение) | минимальное значение выражения от поля любого типа |
| <i>sum</i> (матем. выр-ие) | суммирование значений поля или выражения с полем |
| <i>std</i> (матем. выр-ие) | среднеквадратическое отклонение |
| <i>var</i> (матем. выр-ие) | дисперсия |

Полученное в результате работы команды *calculate* значение сохраняется в переменной, имя которой было задано при записи команды. Если результатом выполнения команды является число, при сохранении ответа в переменной это число округляется до двух знаков после запятой.

Выполните следующие действия.

1. Откройте таблицу *Gazy_v_Atmosfere.dbf*.
2. Вычислите сумму по количеству газа в атмосфере. Сохраните результат в переменной *TotalKolich*. Для этого выполните команду
calculate sum(Kolich) to TotalKolich
3. Результат выполнения команды, то есть вычисленное значение переменной *TotalKolich*, появится на экране, если в СУБД FoxPro действует установка *set*

talk on. В противном случае следует вывести на экран значение переменной *TotalKolic* командой

?TotalKolic

4. Выполните команду *set talk on*. Теперь результат каждого выполнения команды *calculate* будет сразу появляться на экране.

5. Найдите самую раннюю дату поступления информации, содержащейся в таблице, и сохраните результат в переменной *MinData*. Чтобы ответ был представлен на экране в привычном формате, выполните команду *set date to german*. Затем получите искомую дату:

calculate min(DataInfo) to MinData

6. Найдите наибольшее общее количество газа в атмосфере среди тех веществ, общее количество которых не превышает 1000×10^9 т, и сохраните результат в переменной *MaxKolic*:

calculate max(Kolic) for Kolic <= 1000 to MaxKolic

7. Найдите количество записей таблицы, в которых указана дата поступления информации позже 1994 года, и сохраните результат в переменной *RecordsAfter1994*:

calculate cnt() for DataInfo > {^1994-12-31} to RecordsAfter1994

8. Очистите экран.

9. Закройте таблицу.

10. Выйдите из среды FoxPro.

Литература

1. Экологическая химия: Пер. с нем. / Под ред. Ф. Корте. – М.: Мир, 1997. – 396.
2. Попов А.А. Создание приложений для FoxPro 2.5 / 2.6 в DOS и Windows. – М.: ДЕСС, 1999. – 672.