Министерство образования Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра экономики

Ф.А. Красина

БАЗЫ ДАННЫХ

Учебно-методическое пособие

2002

Корректор: Красовская Е.Н.

Красина Ф.А.
Базы данных: Учебно-методическое пособие. – Томск: Томский межвузовский центр дистанционного образования, 2002. – 44 с.

СОДЕРЖАНИЕ

КОНТРОЛЬНАЯ РАБОТА ПО ДИСЦИПЛИНЕ «БАЗЫ ДАННЫХ»

Контрольная работа состоит из четырех заданий (задания 1-4). Для выполнения контрольной работы необходимо изучить учебно-методическое пособие. Контрольная работа выполняется с использованием СУБД FOX PRO. 2.6.

1. СОЗДАНИЕ И ПЕРВИЧНАЯ ОБРАБОТКА DBF-ФАЙЛОВ

Создание файла базы данных включает два этапа: создание структуры файла и его заполнение данными.

Создание структуры файла

Действие команды рассмотрим на примере. Создадим простой файл базы данных, который содержит сведения о кадровом составе предприятия, включающие следующие данные (названия полей указаны в скобках):

1. Фамилия и инициалы работника (FAM);
2. Дата рождения (DTR);
3. Табельный номер (TAB);
4. Количество детей (DET);
5. Пол (POL);
6. Семейное положение (SEM);
7. Средняя зарплата (SZAR);
8. Подразделение, место работы - отдел, цех (PODR);
9. Сведения о перемещениях по службе (PER).

В поле ПЕРЕМЕЩЕНИЯ (PER) будем заносить информацию о местах работы и должностях сотрудника внутри предприятия. Назовем файл базы данных KADR.DBF. Выберем для его полей типы и размеры:

1. FAM - символьный тип (Character) длиной 25 символов.
2. DTR - тип дата (Date) со стандартной длиной 8.
3. TAB - числовой (Numeric) тип длиной 3 разряда целых.
4. DET - числовой тип длиной 1 разряд целых.
5. POL - символьный тип длиной 1 символ (М или Ж).
6. SEM - символьный тип длиной 1 символ (допускаются значения: Б - в браке, Х - холост, Р - разведен).

7. SZAR - числовой тип общей длиной 7 разрядов целых (максимальная зарплата 9 999 999 руб.).

8. PODR - символьный тип длиной 15.

9. PER - ввиду непредсказуемости длины этой информации возьмем для нее поле примечаний (тип Memo-поле). Структура файла базы данных типа DBF создается командой

CREATE <имя файла>

В нашем случае для создания файла KADR.DBF это команда

CREATE kadr

Расширение имени файла DBF указывать необязательно, так как оно добавляется автоматически.

Если файл создается не на активном в данный момент диске и/или директории, нужно указывать и дисковод, и путь доступа, например D:\KADR

CREATE d:\kadr\kadr

В ответ СУБД представит окно-форму для ввода данных о структуре создаваемого файла базы данных, а именно для каждого вводимого поля его имя, тип, длину и для числового поля – точность (число дробных позиций). В нашем случае для файла KADR.DBF заполним предлагаемую форму, как показано на рис.1.1.

Тип поля выбирается в момент, когда курсор находится в колонке TYPE, нажатием первой буквы имени типа (C, N, F,. D, L, M, G) или клавиши Space/Enter (Пробел/Ввод). В ответ появляется меню выбора (рис.1.2).

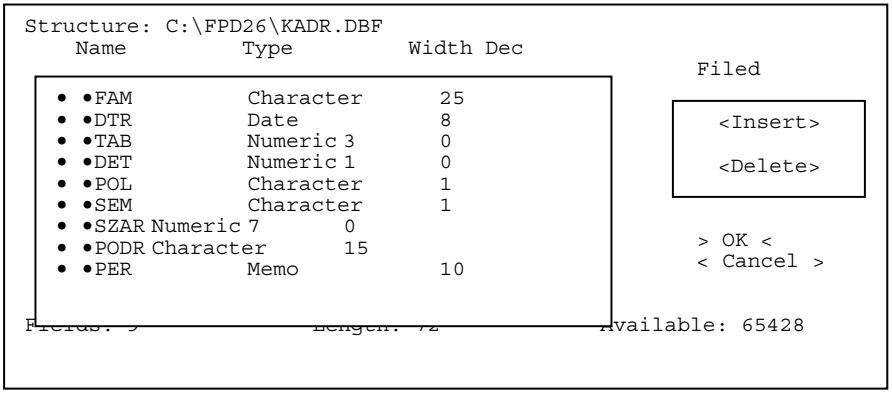


Рис.1.1

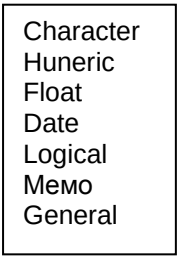


Рис. 1.2

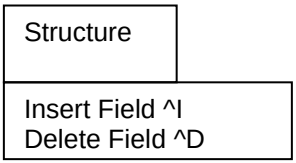
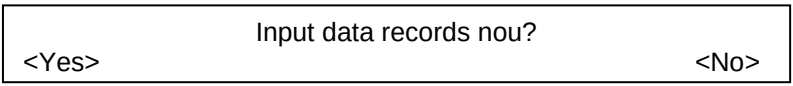


Рис. 1.3

После окончания формирования структуры файла она должна быть запомнена на диске. Это осуществляется одновременным нажатием клавиш Ctrl-End или Ctrl-W. Нажатие клавиши Escape вызовет отказ от сохранения структуры. Тот же результат может быть достигнут нажатием Enter после перемещения курсора в позицию >OK< или <Cancel> соответственно. Все действия по управлению созданием базы могут быть реализованы и мышью.

Затем FoxPro запросит ввод данных:



Ответим No.

Если в дальнейшем обнаружится, что структура базы данных нас не удовлетворяет, ее можно изменить командой модификации структуры

- MODIFY STRUCTURE.

При этом мы попадем в меню, идентичное меню команды CREATE, где и увидим структуру модифицируемого файла. Здесь можно удалять, переименовывать и дополнять поля в базе данных, а также изменять их параметры. Если в файле базы данных к этому моменту имелись данные, они будут (если это возможно) сохранены. Модифицируемый файл должен быть предварительно открыт. При модификации базы данных старые структуры сохраняются на диске с расширениями имен BAK для DBF-файлов и TBK для FPT-файлов.

Прежде чем перейти к следующему материалу командой SET STATUS ON установите на экране статус-строку (для удобства ориентирования в среде FoxPro). В этой строке будет со-

держаться полезная для пользователя, а иногда и для программиста информация: имя выполняемой программы (если есть), активный диск, имя открытой базы, номер текущей записи, общее число записей в базе данных (эти параметры отображаются в виде дроби как числитель и знаменатель), признак пометки текущей записи к удалению (слово Del), положение клавиш Num Lock и Caps Lock (Num и Caps). Удаляется статус-строка командой SET STATUS OFF (эта форма действует по умолчанию). Статус-строка полезна при обучении и отладке, но в готовых программах она, конечно, не нужна.

Далее при работе в командном окне увидите, что результаты выполнения многих команд автоматически отображаются на экране. Такое качество FoxPro определяется командой SET TALK ON/OFF.

По умолчанию ON. Программисту эта особенность обычно только мешает, и она должна быть подавлена (SET TALK OFF).

Открытие файла базы данных

Файл после создания структуры остается открытым, т.е. доступным для команд ввода, просмотра и изменения. Однако, если СУБД только загружена в память, должно быть выполнено открытие нужного файла базы данных командой открытия

USE [<DBF-файл>].

Команда USE без имени файла закрывает базу данных. Более полный синтаксис команды мы рассмотрим позже.

Закрытие всех файлов баз данных и связанных с ними файлов других типов индексов, форматов, отчетов и т.д. во всех рабочих областях с переходом в первую рабочую область осуществляется командой

CLOSE DATABASE.

Закрытие вообще всех файлов выполняется командой

CLOSE ALL.

Дополнение базы данных

Дополнение файла новыми записями осуществляется командой

- APPEND [BLANK],

которая предъявляет окно ввода данных со всеми пустыми полями создаваемой записи. Необязательная фраза BLANK озна-

чает, что новая запись останется пустой и не будет отражена на экране.

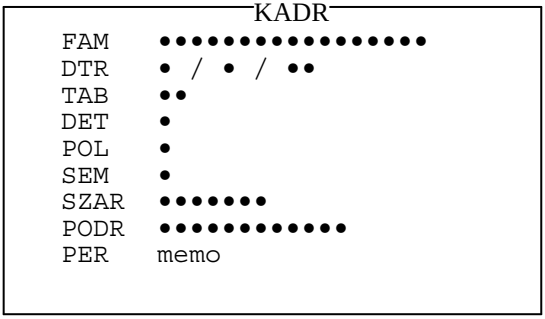


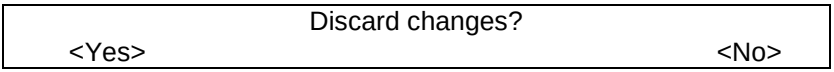
Рис 1.4

В нашем случае команды

```
USE kadr
APPEND
```

обеспечат доступ к окну редактирования данных (рис.1.4) к самой первой записи незаполненного пока файла.

Все поля базы выделяются контрастным цветом в соответствии, с их длиной. Мемо-поле указывается только словом «memo».После ввода текущей записи автоматически происходит доступ к следующей записи и т.д. Для того чтобы войти **в мемо-поле,** необходимо в него поместить курсор и нажать **Ctrl-Home** или дважды кнопку мыши. **Выход** из мемо-поля с сохранением сделанных в нем изменений на диске осуществляется клавишами **Ctrl-W** или **Ctrl-End**, без сохранения **- Escape**, При этом, если изменения в мемо-поле были сделаны, FoxPro выдает вопрос-предупреждение «Не сохранять изменения?»:



Ответ Yes завершит работу в окне, причем все внесенные вами изменения будут проигнорированы. No - вернет нас обратно в окно ввода.Выход из окна APPEND осуществляется аналогичным образом. При нажатии Escape не сохраняется только новая информация в поле, в котором находится курсор.В окне редактирования возможны возврат к предыдущим записям для

их изменения и вообще листание записей (клавиши PgUp/PgDn). Допускается также пометка записей к удалению нажатием Ctrl-T. Более подробно клавиши управления редактированием будут рассмотрены позже.

По умолчанию в FoxPro принят американский (AMERICAN) формат даты - две цифры месяца, дня и года, отделенные косой чертой, т.е. ММ/ДД/ГГ. Имеется возможность установить иные формы даты командой

SET DATE <тип даты>.

Приведем ее важнейшие типы и форматы:

SET DATE AMERICAN - ММ/ДД/ГГ;
SET DATE GERMAN - ДД.ММ.ГГ.

При отладке и/или работе прикладной системы может возникнуть необходимость обращаться в несколько директорий. Эту возможность предоставляет команда

SET DEFAULT TO [<путь>],

которая устанавливает диск и/или директорию в качестве используемого по умолчанию. По этой команде выполняется команда CD (смена директорий) операционной системы. Далее розыск имеющихся файлов и создание новых файлов будут выполняться именно здесь. В начале директорией по умолчанию считается стартовая директория - директория, откуда был выполнен вызов FoxPro.

Пример. Задание активного диска/директории D:\APLAN:

```
SET DEFAULT TO D:\PLAN
```

Окно редактирования

При выдаче команд APPEND, INSERT, EDIT, CHANGE, BROWSE и наличии открытой базы данных FoxPro развертывает для пользователя окно редактирования. Вся информация в окне доступна для изменения. Кроме того, возможны дополнение базы и удаление записей.

Стандартное окно редактирования имеет две формы. Для первых четырех команд оно будет выглядеть одинаково - все поля базы данных располагаются вертикально (см. выше экран к команде APPEND). На экране видно столько записей и полей, сколько удается их здесь разместить. Назовем форму такого окна CHANGE-ОКНОМ.

Другую форму предъявления данных осуществляет команда BROWSE (BROWSE-ОКНО). Здесь все поля каждой записи располагаются горизонтально - колонками. Если какие-то поля

записи не умещаются в строке, с помощью клавиш управления курсором и мыши возможно перемещение (скролинг) изображения вправо или влево.

Клавиши управления действуют практически одинаково в обеих формах окон. Пометка записей к удалению выполняется клавишами Ctrl-T. Признаком пометки является появление специального значка (точки) в самой левой колонке записи. Это же действие может быть реализовано с помощью мыши, для чего ее маркер должен быть установлен в самую левую колонку и нажата кнопка мыши.

Дополнение базы новой записью осуществляется нажатием клавиш Ctrl-N (кроме команды APPEND, в которой это происходит автоматически).

Перемещение внутри базы данных осуществляется с помощью клавиш перемещения курсора или мыши. Кроме того, могут использоваться следующие клавиши:

Ctrl-право/лево	- переход к следующему/предыдущему слову;
Home/End	- переход к началу/концу поля;
Tab/Shift-Tab	- переход к следующему /предыдущему полю;
PgUp/PgDn	- переход назад/вперед на один экран;
Enter	- переход к следующему полю;
Ctri-Y	- очистка поля;
Ctrl-Home, Ctrl-PgUp, Ctrl-PgDn или двойное нажатие кнопки мыши	- вход в мемо-поле;
Ctrl-W/End	- выход с сохранением измененных данных;
Ctrl-Q, Escape	- выход без сохранения.

В последнем случае не сохраняются только изменения, сделанные в текущем поле, если курсор не был из него выведен. Изменения, сделанные в других полях, сохраняются.

Для предъявления CHANGE-окна введите команду CHANGE. При этом вы увидите уже знакомую по команде APPEND форму отображения данных.

После ввода команды BROWSE (при открытой базе KADR.DBF) появляется изображение данных. Форма окна приведена с некоторыми упрощениями.

В мемо-поле стоит слово memo/Memo. Если оно начинается со строчной буквы, то поле пустое, если с прописной - в

нем имеются данные. Чтобы войти в мемо-поле, необходимо переместить в него курсор и нажать Ctrl-Home или Ctrl-PgDn/PgUp, или дважды кнопку мыши.

По умолчанию окно редактирования может быть видно не целиком. Раскрывается окно во весь экран нажатием клавиш Ctrl-F 10. Повторное нажатие клавиш возвращает окно к исходной форме.

KADR								
Fam	Dtr	Tab	Pol	Sem	Det	Pdr	Szar	Per
Сидоро П.С.	12.10.56	13	М	Х	1	ОГМ	635000	memo
Потапов Д.П.	04.09.60	98	М	Б	3	ОГМ	268000	Memo
Кулакова М.И.	15.04.49	6	Ж	Б	2	ОГМ	290000	memo
Попов А.А.	25.03.46	234	М	Р		ХБ	550000	Memo
Романова М.С.	09.10.66	890	Ж	Х		ОХ	395000	Memo
Миронов Р.И.	09.09.70	460	М	Х		ОГМ	542000	memo
Яковлев А.И.	10.12.30	54	М	Б	2	ВОХР	418000	memo

Рис 1.5

BROWSE-окно

Команда BROWSE - один из наиболее мощных и удобных инструментов доступа пользователя к данным в FoxPro. По существу это не просто команда - это целая среда доступа и управления данными. Допускается создавать так называемые вычисляемые поля. Эти поля фактически не являются полями базы данных, но могут быть их функциями и отображаются на экране наравне с настоящими полями, что дает возможность пользователю, например, оценивать свои данные по заданному критерию. Такой режим соответствует работе с электронной таблицей.

Формат команды (основные опции перечислены по алфавиту):

- BROWSE [FIELDS <поля>] [FOP<условие1] [LAST] [NOAPPEND] [NOEDIT/NOMODIFY] [TITLE <выр$C2>]

Управление доступом к полям базы

FIELDS <список полей> - перечень предъявляемых полей. По умолчанию отображаются все поля базы данных. Имя каждого поля может сопровождаться ключами, определяющими режим доступа к нему:

В <список полей> могут включаться вычисляемые поля. Эти поля являются функциями других полей, переменных и т.д.

Такие поля не могут редактироваться и не запоминаются в базе данных. Вычисляемые поля сами могут содержать пользовательские функции, что делает их важным средством отображения и управления данными. Например, в команде BROWSE для базы KADR.DBF введем вычисляемое поле POM для определения материальной помощи. Считаем, что помощь устанавливается на одного ребенка в размере 70% средней зарплаты, но не более 90000 руб. и только тем, у кого средняя зарплата не превышает 300000 руб. Таким образом POM=IIF(szar>300000, 0, MIN(0.7*szar*det, 90000)).

Здесь, забегая вперед, мы использовали две новые функции: MIN() и IF(). Функция MIN() возвращает минимальное значение из 0.7*szar*det и 90000. Функция IF() возвращает 0, если szar>300000, или MIN(...) в противном случае. Подробнее эти функции разбираются в гл. 16.

Видимый размер вычисляемого поля POM будет определяться принятыми умолчаниями на размер числовых выводов. Этим процессом можно (и лучше) управлять с помощью параметра ограничения длины поля :<выpN>. Еще удобнее применение шаблонов (ключ :P), которыми может быть установлена не только длина, но и формат выдачи.

FOR <условие1> - устанавливает фильтр записей для базы. В BROWSE-окне предъявляются только записи, удовлетворяющие заданному <условию>.

Пример:

```
USE kadr
BROWSE FOR szar>=530000.AND.szar<=750000
```

Здесь команда BROWSE предъявляет только те записи базы KADR.DBF, в которых значения поля SZAR (средняя зарплата) от 530000 до 750 000 руб.

LAST- конфигурация окна сохраняется в системном файле для использования в следующем сеансе работы.

NOAPPEND – дополнение базы данных с помощью клавиш Ctrl-N невозможно.

NOEDIT/NOMODIFY – редактирование невозможно, разрешена пометка к удалению.

TITLE <выpC2>] – задает заголовок окна.

Пример. Настроим окно редактирования таким образом, чтобы оно имело собственные заголовки колонок FAM, DTR, POL, DET, SEM, SZAR, заголовок окна - КАДРЫ, указания на

возможные действия в окне (Ctrl-T, Ctrl-N, Ctrl-W), ограничение на верхнюю границу средней зарплаты в 9000000 руб. и вычисляемое поле POM (Помощь).

```
USE kadr
SET DATE GERMAN
BROWSE ;
TITLE     '^T-удал., ^N-доп. КАДРЫ ^W-выход';
FIELDS    fam      :H='Фамилия' :12, ;
          dtr      :H='Родился', ;
          pol      :H='Пол', ;
          det      :H='Детей', ;
          sem      :H='Сем. пол.', ;
          szar :H='Ср. зар.' :B=,9000000 , ;
          pom=IIF(szar>300000,0,MIN(0.7*azar*det, 90000)), ;
          :H='Помощь'  :P='# # # # #' LEDIT
```

Поскольку из-за заголовков колонок ширина большинства полей увеличилась, видимая часть поля FAM здесь уменьшена до 12 колонок с возможностью прокрутки этого поля с помощью клавиш управления курсором. Кроме того, для вычисляемого поля POM принят шаблон «# # # # #» (пять разрядов целых). Иначе результат в этом поле будет отображаться разрядностью по умолчанию. Вид нашего BROWSE-окна приведен на рис.1.6.

^T-удал., ^N-доп. КАДРЫ ^W-выход'

Фамилия	Родился	Пол	Детей	Сем.пол.	Ср.зар.	Помощь
Сидоро П.С.	12.10.56	М	1	Х	635000	0
Потапов Д.П.	04.09.60	М	3	Б	268000	90000

Рис 1.6

Замечание к нотации примеров. Все тексты примеров изображаются так, как будто они являются фрагментами программ, и при этом длинные команды разбиваются на части приемлемого размера с помощью знака «:». Поскольку еще не рассматривались средства создания и исполнения командных файлов, эти команды следует вводить в командном окне, можно в одну строку, опуская «;». Кроме того, не следует повторно вводить команды, если нужный результат уже ранее достигнут. Так, нет необходимости снова открывать файл базы данных, если он уже открыт. Убедиться в этом можно, например, по виду статус-строки.

CHANGE /ADIT-окно

По функциям и возможностям команда CHANGE аналогична команде BROWSE, но предъявляет все поля всех записей в одну колонку.

- CHANGE [WHILE <условие>][опции]

Опции - набор режимов по составу и действию полностью аналогичных опциям команды BROWSE (отсутствуют только опции NOLGRID/NORGRID).

Пример. Выдать на редактирование поля FAM (Фамилия) И SEM (Семейное положение) всех записей файла K.ADR.DBF, для которых значение поля DET>1 (детей больше одного).

```
USE kadr
CHANGE FIELDS fam,sem FOR det>1
```

Перемещения в базе данных

При работе с базой данных необходимы средства перемещения внутри нее. Запись, на которой находится указатель записей, является текущей, и только к ней в данный момент возможен непосредственный доступ.

Имеется несколько разновидностей команд, изменяющих положение указателя записей:

- GO TOP [IN <область>] - переход к самой первой записи файла;
- GO BOTTOM [IN <область>} - переход к самой последней записи;
- GO <выpN> [IN <область>] - переход к записи с указанным в <выpN> номером;
- SKIP <выpN> [IN <область>] - переход к записи, отстоящей от текущей на указанное в <выpN> число записей.

В последней команде <выpN> может быть и отрицательным, что означает движение указателя назад. SKIP без параметра идентичен SKIP 1 (переход на следующую запись). Параметр IN <область> указывает над базой из какой рабочей области должна выполниться команда. Если он опущен, имеется в виду текущая рабочая область.

Для контроля положения указателя и наличия записей в файле предусмотрены функции:

RECNO([<область>]) - указывает номер текущей записи;

RECCOUNT([<область>]) - выдает общее число записей в файле базы данных, включая записи, помеченные к удалению;
EOF([<область>) - функция конца файла. Она истинна (.T.), если конец достигнут, и ложна (.F.) в противном случае;
BOF([<область>]) - то же, но для начала файла.

Необязательный параметр <область> указывает, для какой рабочей области запрашивается значение функции. По умолчанию текущая область.

Функции в FoxPro имеют характерный синтаксис - скобки, даже если никакого аргумента нет и они остаются пустыми.

Просмотр данных

Просмотр данных в FoxPro осуществляется очень близкими по смыслу и синтаксису командами LIST и DISPLAY:

DISPLAY [<границы>] [<поля>[WHILE <условие>]
[FOR <условие>][OFF][TO PRINT/TO FILE <файл>]
Здесь:
OFF - указание на то, что номера записей не выводятся;
TO PRINT - результат команды выдается на принтер;
TO FILE <файл> - результат выдается в <файл>. Если не указать расширение имени, то оно будет TXT.

В качестве заголовка вывода командой выдаются имена Полей базы данных. При заполнении экрана выполнение команды приостанавливается с индикацией указания «Нажмите любую клавишу». При нажатии просмотр может быть продолжен. После выполнения команды указатель записи перемещается на последнюю показанную запись. Записи, помеченные к удалению, команда выдает со звездочкой (если SET DELETED OFF). Выдачу имен полей можно подавить командой

SET HEADING OFF

(восстановление - ON). Чтобы выдать мемо-поля, их имена нужно явно указать в списке FIELDS, иначе они выводятся просто столбцом «memo». Команда

SET MEMOWIDTH <выpN>

определяет фактическую ширину строки для выводимого мемополя. Команда DISPLAY без параметров осуществляет выдачу всех полей базы данных только одной текущей записи.

Примеры:

```
USE kadr
DISPLAY ALL TO FILE kadr                    - выдача всех записей в файл
                                              KADR. TXT
DISPLAY fam WHILE tab#'890'- выдача полей FAM всех записей до тех
                                              пор, пока не встретится за-
                                              пись с TAB='890'

GO 4                                        - переход к четвертой записи
DISPLAY REST FOR pol='Ж' .AND.sem='Б'
          - выдача всех записей файла, начиная с четвертой, для всех
              женщин, состоящих а браке
GO TOP    - переход в начало файла
SET MEMOWIDTH TO 40
          -     установление ширины вывода мемо-поля в 40 символов
DISPLAY fam, per OFF
          - вывод иэ текущей (первой) записи поля FAM и мемо-поля
              PER. Номер записи не выводится
```

Команда с похожими функциями LIST не делает периодических остановок при выдаче данных, и по умолчанию область ее действия не текущая запись, а весь файл. Ввиду этого команда более пригодна для выдачи данных на принтер/файл.

Удаление данных

В FoxPro имеется несколько команд удаления данных:

ERASE <файл> – удаление любого не открытого в данный момент файла. Расширение имени обязательно. Совершенно аналогичные функции выполняет команда DELETE FILE <файл>.

ZAP - удаление всех записей в активном файле базы данных с сохранением его структуры.

DELETE [<границы>] [WHILE <условие>] [FOR <условие>] -пометка к удалению записей в указанных границах и/или отвечающих указанным условиям. DELETE без параметров помечает только одну текущую запись.

PACK [MEMO][DBF] - физическое удаление помеченных ранее записей и сжатие файла. После выполнения команды указатель записей устанавливается в начало базы. Если имеются открытые индексы, они перестраиваются. По умолчанию упаковывается как файл данных (DBF), так и файл мемо-полей (FPT). Если указан параметр MEMO, то упаковывается только FPT-файл, если DBF - то только DBF-файл.

RECALL [<границы>] [WHILE <условие>] [FOR <условие>] – снятие пометок к удалению. RECALL без параметров действует только на текущую запись.

Удаление записей в базе данных выполняется в два этапа: сначала пометка записей на удаление (она возможна и в окнах редактирования нажатием клавиш Ctrl-T) командой DELETE, а затем их физическое уничтожение командой PACK. Если упаковка файла еще не произведена, можно спасти нужные записи командой RECALL.

Команду PACK имеет смысл применять не только для удаления записей, но и для сжатия мемо-полей (PACK MEMO). Дело в том, что даже при видимом уменьшении размера мемополя (например, в результате редактирования) уменьшается только доступная пользователю часть, но размер самого FPT-файла никогда не сокращается. Таким образом, при интенсивном обновлении данных с участием мемо-полей вполне возможно недопустимое «разбухание» файла примечаний.

Примеры:

```
USE kadr
?RECNO(),RECCOUNT() - выдача номера текущей записи и общего их числа
            1           7
GO 5          — переход к пятой записи
SKIP –3                         — возврат ко второй записи
DELETE NEXT 3                   - пометка к удалению Записей 2,3,4
RECALL RECORD 4     - снятие пометки о Записи 4
PACK                            - сжатие файла с Возвратом в начало
?RECNO(),RECCOUNT()
            1           5       - осталось 5 записей
```

Для иллюстрации работы FoxPro применена команда, выдающая на экран перечисленные в ней выражения. Более подробно команда будет рассмотрена позже.

Команда сжатия PACK в базе реальных размеров выполняется медленно, и поэтому лучше ее делать один раз в день или даже раз в несколько дней. Чтобы помеченные к удалению записи до их уничтожения не участвовали далее в обработке данных, можно использовать команду

SET DELETED ON

При этом такие записи делаются как бы невидимыми для FoxPro и пользователя, за исключением случаев прямого на них указания. Например, команда GO 20 установит указатель записей на двадцатую запись независимо от того, была или нет она помечена. По умолчанию принято значение OFF.

Изменение данных

В FoxPro имеется возможность не только вручную редактировать данные, но и изменять их путем присвоений или вычислений.

REPLACE [<границы>] [WHILE <условие>] [FOR <условие>] <поле1> WITH <выражение> [,<поле2> WITH <выражение> ...]
[ADDITIVE][NOOPTIMIZE]

Эта команда осуществляет множественное изменение полей базы данных в соответствии с заданными выражениями, в установленных границах и при заданных условиях. Если отсутствует параметр <границы> или <условия>, изменена будет только текущая запись. Параметр ADDITIVE действует для мемо-полей и означает, что заданное <выражение> будет дописываться в конец поля. Если этот параметр опущен, то старое значение мемо-поля будет замещено <выражением>.

Для иллюстрации применения этой команды создадим еще один файл базы данных по учету месячной выработки (зарплаты без вычетов) всех членов бригады номер 1. Назовем его BRIG1.DBF. Файл будет иметь два поля: TAB - табельный номер и VIR - выработка-зарплата. Командой CREATE brig установим структуру файла (рис.1.7).

Name	Type	Uldth	Dec
TAB	NAMERIC	3	0
VIR	NAMERIC	7	0

Рис 1.7

Пример. Предположим, что бригаде 1 (файл BRIG1.DBF) выделена премия в размере 20% зарплаты, а бригадиру (табельный номер 98) - еще 500000 руб. Премия не начисляется тем рабочим, которые в данном месяце отработали менее одного дня (выработка менее 10000 руб.). С учетом сказанного в файле необходимо увеличить значения всех полей VIR в 1.2 раза, если VIR> 10000, а в записи с TAB=98 прибавить еще 100000.

```
 USE brig1
REPLACE vir WITH vir*1.2    FOR vir>10000
REPLACE vir WITH vir+100000 FOR tab=98
```

Начальное содержимое базы данных представлено на рис.1.8, а новое - на рис.1.9.

	TAB	VIR
1	98	250000
2	6	14000
3	13	400
4	68	201000

Рис. 1.8

	TAB	VIR
1	98	400000
2	6	168000
3	13	4000
4	68	241200

Рис. 1.9

Команда REPLACE эквивалентна знаку равенства в операции присвоения для переменных. Буквально фраза <поле> WITH <выражение> соответствует присвоению <поле>=<выражение>.

Очистку полей базы данных выполняет команда

BLANK [<границы>] [FIELDS <поля>]
[WHILE <условие>] [FOR <условие>] [NOOPTIMIZE

Если отсутствуют <границы> или <условия>, очищена будет только текущая запись. BLANK без параметра FIELDS очищает все поля базы.

Локализация и поиск данных

Фильтрация данных

Хотя в команде BROWSE, например, имеется возможность осуществить отбор записей из базы с помощью опций FOR и KEY, в других командах это может оказаться невозможным или неудобным. В силу этого в FoxPro предусмотрена специальная команда вида

SET FILTER TO [<условие>],

которая позволяет установить FOR-условие для всех без исключения команд обработки данных. Здесь <условие> указывает на то, какие именно записи могут быть доступны для обработки. Например, команда

SET FILTER TO fam='ИВ'

сделает доступными для обработки только записи, в которых фамилия сотрудника начинается с букв «ИВ». Команда SET FILTER действует исключительно на ту базу, которая открыта и активна в данный момент.

Выяснить выражение фильтра для текущей или указанной рабочей области можно с помощью функции

FILTER([<область>])

В FoxPro имеются разнообразные команды поиска записей, которые реализуют как последовательный, так и ускоренный алгоритмы. При последовательном поиске выполняется сплошной перебор записей файла базы данных до установки указателя записей на искомую запись.

Начальный поиск данных

Следующая команда осуществляет последовательный поиск одной самой первой записи в базе данных, удовлетворяющей заданному FOR-условию, среди записей, находящихся в заданных границах, и до тех пор, пока соблюдается WHILE-условие (если есть).

- LOCATE [<границы>] FOR <условие> [WHILE <условие>]

В случае, если границы и WHILE-условие отсутствуют, поиск ведется во всем файле, начиная с первой записи. При успешном поиске указатель записей устанавливается на найденную запись, функция RECNO() равна номеру этой записи, а функция FOUND(), оценивающая результат поиска, возвращает значение «Истина» (.T.). При неудачном поиске функция RECNO() равна числу записей в базе плюс 1, FOUND()=.F., а функция достижения конца файла EOF() возвращает .T.

Продолжение поиска

Команда, которая продолжает поиск записей, начатый ранее командой LOCATE, приведена ниже:

- CONTINUE,

которая эквивалентна команде LOCATE REST FOR <условие> [WHILE <условие>]. Если командой LOCATE или CONTINUE не было найдено нужных записей, указатель записей устанавливается на нижнюю границу поиска (если она введена в команде) или на конец файла (EOF()=.T.).

Результатом применения команд (если SET TALK ON) являются сообщения о номере каждой найденной записи или/и достижении границы поиска.

Пример. В файле KADR (его содержимое представлено при описании команды BROWSE) необходимо найти все записи о женщинах, т.е. те записи, в которых POL='Ж'. Вводимые команды и реакции системы изображены ниже (найдены записи с номерами 3 и 5).

```
SET TALK ON
USE kadr
LOCATE FOR pol='Ж'
Record = 3                    (Запись 3)
CONTINUE
Record = 5                    (Запись 5)
CONTINUE
End of Locate scope.      (Конец границы поиска.)
```

Кроме рассмотренных команд имеется полезная функция поиска

- LOOKUP(<поле1>,<выр>,<поле2>)

Функция ищет первое вхождение искомого <выражения> в указанном <поле 2> активной базы данных и возвращает значение <поля 1> из той же базы. Если файл индексирован и индекс открыт, поиск ведется ускоренным методом, если нет - последовал тельным (подобно команде LOCATE). Если поиск оказался безуспешным, функция возвращает пустую строку, а указатель записей становится ниже последней записи базы (EOF()=.T.).

Пример. В базе KADR.DBF с помощью функции LOOKUP() ищется в поле FAM первая фамилия, начинающаяся на букву 'П', и выводится ее табельный номер. Кроме того, для проверки в команде ? выводится и сама фамилия, и номер записи (FAM и RECNO()). Результаты работы команды приведены справа после знаков &&. Видим, что сначала выводится табельный номер 98 (ПОТАПОВ Д.П. - запись номер 2). После открытия индексного файла KADRFAM.IDX (индексирование сделано по полю FAM) выводятся уже данные для ПОПОВА А.А. (запись номер 4), поскольку по алфавиту он стоит выше ПОТАПОВА Д.П.

```
USE kadr
? LOOKUP(tab,'П',fam),fam,RECNO()
&& 98 ПОТАПОВ Д.П. 2
SET INDEX TO kadrfam
? LOOKUP(tab,'П',fam),fam,RECNO()
&& 98 234 ПОПОВ А.А.  4
```

Математическая обработка базы данных

COUNT [<границы>] [WHILE <условие>] [FOR <условие>] [TO <переменная>]

По команде COUNT подсчитывается число записей в заданных границах, удовлетворяющих условиям, которое заносится в указанную <переменную>.

SUM [<границы>] [WHILE <условие>][FOR <условие>] <список выражений>[TO <переменные>/TO ARRAY <массив>]

По команде SUM суммируются значения перечисленных числовых полей в указанные <переменные> или <массив

AVERAGE [<границы>] [WHILE <условие>] [FOR <условие>] <список выражений> [TO <переменные>/TO ARRAY <массив>]

По команде AVERAGE подсчитывается среднее арифметическое при тех же допущениях, что и для предыдущей команды.

CALCULATE [<границы>] [WHILE <условие>
[FOR <условие>] <список выражений>
[TO <переменные>/TO ARRAY <массив>]

Команда CALCULATE позволяет вести математические расчеты в базе данных. <Список выражений> может содержать любую комбинацию следующих внутренних для данной команды функций:

-AVG(<выpN>) - среднее арифметическое для выражения с полем, -CNT() - число записей в базе данных;

-MAX(<выр>) - максимальное значение <выр> от поля любого типа;

-MIN(<выр>) - минимальное значение <выр> от поля любого типа.

Задание 1

1. Создать файлы со следующей структурой:

Имя файла : GROUP

Имя поля	Тип поля	Содержание
NSTUD	Числовое	Идентификатор
FAMILY	символьное	Фамилия
NAME	Символьное	Имя
DATE	Дата	Дата рождения
SEX	Символьное	Пол студента
POLICE	Номер полиса	
PLASE	Memo- поле	Адрес

Имя файла : STUDY

NSUD	Числовое	Идентификатор
SUBJECT	символьное	Название предмета
BALL	Числовое	оценка

2. Заполнить файлы значениями данных (не менее 10 в файле GROUP и не менее 30 в файле STUDY.
3. Вывести для просмотра на экран содержимое файлов с соответствующими заголовками полей на русском языке.
4. Увеличить все значения в поле NSTUD на 10 для файла STUDY (для выполенеия этого задания создать файл STUDY1, в который предварительно копируется вся информация из исходного файла) .
5. Пометить для удаления все записи файла STUDY1, относящиеся к математике.
6. Вывети в текстовый файл всю информацию из файла GROUP для студентов, родившихся до 1.01.1985 г.
7. Вывести в текстовые файлы поля NSTUD, FAMILY, NAME, DATA отдельно для мужчин и женщин.

Для проверки выполнения лабораторной работы студент должен представить файлы-DBF : GROUP, STUDY,STUDY1 и текстовые файлы с результатами выполнения заданий 6, 7.

2. ИНДЕКСИРОВАНИЕ БАЗ ДАННЫХ

Важнейшим элементом любой системы управления базами данных является наличие средств ускоренного поиска данных, поскольку поиск - самая распространенная операция в системах обработки данных. Этот механизм обычно реализуется введением так называемых индексных файлов (индексов). Они имеют расширение имени IDX или CDX.

Если файл проиндексирован, команды DISPLAY, EDIT, BROWSE, SKIP, REPLACE и все другие команды, связанные с движением в базе данных, перемещают указатель записей в соответствии с индексом, а не с физическим порядком расположения записей. Так, команды GO TOP и GO BOTTOM устанавливают указатель записей не на первую (номер 1) и последнюю физические записи, а на начальную и конечную записи индекса соответственно.

Один файл базы данных может быть проиндексирован по нескольким полям и иметь любое число индексов (индексных файлов), которое ограничено только дисковой памятью компьютера. Индексные файлы содержат информацию о расположении записей файла базы данных в алфавитном, хронологическом или числовом порядке для того поля/полей, по которому выполнено индексирование. Допускается индексирование и по логическим полям. Расширение индексного файла – IDX.

Размер индексного файла сравним с объемом дискового пространства, занимаемого полем базы данных, по которому было произведено индексирование. Таким образом, например, если база проиндексирована по всем полям, суммарный размер всех индексов будет близок (или больше) к размеру всей базы данных. Кроме того, замедляются операции ввода/редактирования данных в базе, поскольку при дополнении ее новой записью индексный файл должен быть автоматически перестроен в соответствии с новыми или измененными данными.

Индексирование выполняется следующей командой:

- INDEX ON <выр>TO <IDX-файл>

[FOR <условие>] [COMPACT] [DESCENDING]
[UNIQUE] [ADDITIVE]

Опции команды:

<выр> - индексный ключ-выражение. Его длина может достигать 100 символов для IDX-файлов и 240 для CDX-файлов. Чаще всего ключ это имя поля, по которому нужно упорядочить файл. Однако, ключ может быть и составным - из нескольких полей. Он может быть и функцией полей и переменных.

TO <IDX-файл> - дает имя одноиндексному файлу.

FOR <условие> - устанавливает режим отбора в индекс только тех записей базы данных, которые отвечают заданному <условию>. При наличии такого, действующего, как фильтр, индекса доступ к нужным данным осуществляется исключительно быстро.

COMPACT - с этой опцией будет создан компактный IDX-файл. Целесообразно использовать только компактные индексы

Для IDX-файлов индексирование всегда осуществляется по возрастанию, однако параметр DESCENDING можно включить в команды открытия индексов любого типа, независимо от того, какой закон был указан в команде индексирования. По умолчанию индексирование выполняется по возрастанию.

UNIQUE - означает, что, если в базе данных встречаются записи с одинаковым значением ключа, все такие записи, кроме первой, игнорируются (не включаются в индекс). Этим процессом можно управлять также с помощью команды SET UNIQUE TO.

ADDITIVE - вновь создаваемые индексные файлы не закроют уже открытые к этому моменту. По умолчанию вновь создаваемые индексы закрывают все ранее открытые индексы для текущей базы данных.

Пример. Пусть мы хотим упорядочить базу KADR в порядке возрастания табельных номеров. Тогда необходимо создать индексный файл по полю TAB. Назовем его KADRTAB.IDX.

```
USE kadr
INDEX ON tab TO kadrtab
LIST tab,fam
Record    #          TAB          FAM
          3          6            КУЛАКОВА М.И.
          1          13           СИДОРОВ П.С.
          7          54           ЯКОВЛЕВ А.И.
          2          98           ПОТАПОВ Д.П.
          4          234          ПОПОВ А.А.
          6          468          МИРОНОВ Р.И.
          5          890          РОМАНОВА М.С.
```

Как видно, команда LIST показала записи именно в желаемом, а не в фактическом порядке. Файл базы KADR.DBF никакому изменению не подвергся, но всеми перемещениями указателя записей управляет теперь индексный файл KADRTAB.IDX.

Каково содержимое индексного файла? Хотя в FoxPro не предусмотрена какая-либо возможность непосредственного доступа к индексу, мы можем просмотреть его во внутреннем текстовом редакторе FoxPro (с помощью команды MODIFY FILE KADTRAB.IDX). Тогда мы увидим, что кроме технической информации в заголовке файл KADRTAB.IDX будет содержать значения ключевого поля TAB индексируемой базы данных, расположенные в порядке возрастания, и (в закодированной форме) фактические номера этих записей в базе данных. В нашем случае

3-6, 1-13, 7-54, 2-98, 4-234 и т.д.

Пример. Рассмотрим включение FOR-условия в команду индексирования. Пусть требуется создать индексный файл KADRPOL.IDX, содержащий ссылки только на те записи базы KADR.DBF, которые соответствуют всем мужчинам (pol='M'), причем упорядоченные в алфавитном порядке фамилий.

```
USE kadr
INDEX ON faro TO kadrpol FOR pol='M' COMPACT
LIST FAM, POL
Record        #       FAM                 POL
              6       МИРОНОВ Р.И.  M
              4       ПОПОВ А.А.    M
              2       ПОТАПОВ Д.П.  M
              1       СИДОРОВ П.С.  M
              7       ЯКОВЛЕВ А.И.  M
```

Видим, что содержание и порядок предъявления записей из базы KADR.DBF отвечают желаемым.

Если порядок предъявления фамилий безразличен и по ним не предполагается поиск, ключевое поле (в данном случае FAM) можно опустить, заменив его пробелом в команде индексирования

```
INDEX ON ' 'TO kadrpol FOR pol='M' COMPACT
```

При этом индексный файл будет меньших размеров.

Если индексный файл был уже создан, его нужно открыть при внесении новых записей или редактировании старых (если затрагиваются индексные поля) и, конечно, если предполагается

индексный поиск. Индексные файлы могут быть открыты совместно с открытием своей базы данных (<DBF-файл>) командой:

USE <DBF-файл> [INDEX <список индексных файлов>

Команды SET INDEX открывает только индексы (база должна быть уже открыта):

SET INDEX TO [<список индексных файлов>

Команда SET INDEX TO без параметра закрывает все индексные файлы, кроме структурного, для текущей базы. Такое же действие осуществляет команда

CLOSE INDEX

Ускоренный поиск

Индексный файл не только упорядочивает базу данных для просмотра, но и ускоряет поиск в ней по ключу, заданному в индексе, если пользоваться командой

SEEK. <выражение>

Команда применяет специальный алгоритм ускоренного поиска, в котором база просматривается не сплошь, а в соответствии с информацией, содержащейся в индексе.

При наличии индекса сначала именно в нем, а не в самой базе ведется поиск номера записи с указанным в команде SEEK знанием выражения в индексном поле. При этом поиск в индексе выполняется не последовательно, а скачками (так называемый двоичный поиск), что позволяет быстро локализовать номер нужной записи. Только после этого указатель записей устанавливается на искомую запись в основной базе данных. Команда SEEK разыскивает только одну первую запись, в которой в индексном поле наблюдается <выражение>, т.е. когда <поле>=<выражение>, и устанавливает на нее указатель записей.

Пример. Проведем в базе KADR.DBF поиск записи с табельным номером 234.

```
USE kadr INDEX kadrtab
SEEK 234
DISPLAY tab,fam
Record          #       TAB             FAM
                4       234             ПОПОВ А.А.
```

Функции RECNO(), FOUND(), EOF() реагируют на результаты поиска командой SEEK точно гак же, как и командами LOCATE и CONTINUE. Если поиск удачный, RECNOQ равно но-

меру найденной записи, FOUND()=.T., EOF()=.F.; если нет, RECNO() равно числу записей в базе данных плюс единица, FOUND()=.F., EOF()=.T.. Все это относится и к индексам с FOR-условием.

Для индексированных баз существует модификация функции указания номера записи с аргументом нуль - RECNO(O), которая в случае неудачного поиска возвращает номер записи, имеющей самое близкое следующее значение к ключу поиска, заданному в команде SEEK. Использовав этот номер, можно затем перейти в указанную запись. Однако если действует команда

SET NEAR ON,

то в случае неудачного поиска указатель записей сразу установится не на конец файла, а на эту близкую запись. По умолчанию SET NEAR OFF.

Это предоставляет инструмент ускоренного поиска по ключу, заданному приблизительно или даже частично неправильно. Например, задана фамилия с неверными инициалами или окончанием. Часто такая ситуация встречается при поиске в числовых полях. Пусть в базе KADR.DBF нужно найти запись, где средняя зарплата равна 600000 руб. Ввиду того что, возможно, никто не получает именно такую зарплату, поиск окажется неудачным, хотя и есть зарплаты, близкие к этой цифре. Если же мы имеем возможность позиционировать указатель на записи с ближайшим значением, то, вызвав затем какое-нибудь средство просмотра данных (например, команду BROWSE) и пролистав данные в базе вблизи найденного места, мы получим возможно' все-таки найти и отобрать подходящие записи.

```
USE kadr
INDEX ON szar TO kadrzar COMPACT
SET NEAR ON
SEEK 600000
BROWSE
```

При этом мы не будем стеснены в возможности перемещаться в базе данных. Такой механизм называется «мягким» или приблизительным поиском в отличие от обычного, точного поиска.

Пример. Пусть нужно указать фамилии родителей, у которых трое детей. Если их не окажется, выводить ничего не нужно.

```
USE kadr
INDEX ON det TO kadrdet COMPACT
IF SEEK(3)
LIST fam, det WHILE det=3
```

```
ENDIF
```

Управление индексами

Один и тот же файл DBF может иметь любое число индексов, и все они могут быть одновременно открыты командами SET INDEX или USE-INDEX. В нашем случае команда

- `USE kadr INDEX kadrtab, polfam`

открывает два индекса KADRTAB.IDX и POLFAM.IDX.

При вводе, удалении, редактировании записей все открытые индексные файлы будут соответствующим образом изменяться. Однако главным управляющим индексом, т.е. таким, в соответствии с которым при необходимости будет перемещаться указатель записей, может быть, конечно, только один. Им является индексный файл, открытый самым первым в команде (здесь KADRTAB.IDX), или назначенный таковым опцией ORDER.

В случае, если необходимо сделать главным другой индекс, используется команда

- SET ORDER TO <IDX-файл>

Команда объявляет главный индекс среди открытых индексных файлов в текущей или указанной рабочей <области>. Опции команды совпадают с описанными выше для команды SET INDEX TO. Например, следующие команды сделают главным индекс POLFAM.IDX

SET ORDER TO polfam или SET ORDER TO 2

Команда SET ORDER TO 0 или просто SET ORDER TO без параметра отключает все индексы от управления перемещением указателя записей. Теперь уже не будет главного индекса. Однако сами индексы остаются открытыми и чувствительными к изменениям в базе данных.

В случае изменения большого объема данных необходимо отключить индексы командой SET INDEX TO с последующим восстановлением командой REINDEX.

Индексирование не только логически упорядочивает записи базы данных, но позволяет легко выполнить и физическую сортировку данных. Для этого только надо скопировать проиндексированную базу в новый файл командой COPY TO <новый

файл>. В таком <новом файле> записи будет уже расположены заданным образом.

Задание 2

1. Упорядочить файл GROUP по датам рождения отдельно для мужчин и женщин.
2. Упорядочить файл STUDY по названиям предмета.
3. Упорядочить файл GROUP по номеру полиса отдельно для мужчин и женщин.
4. Упорядочить файл GROUP по фамилиям отдельно для мужчин и женщин.

Для проверки выполнения задания студент должен выслать в текстовом виде команды FOX PRO, выполняющие задания 1-4.

3. УСТАНОВЛЕНИЕ СВЯЗЕЙ МЕЖДУ ФАЙЛАМИ

Понятие о рабочих областях

В FoxPro можно обрабатывать сразу несколько файлов баз данных. Каждый такой файл типа DBF и все вспомогательные файлы (например, индексные) открываются в своей отдельной рабочей области. Переход из области в область осуществляется командой

- SELECT <рабочая область/псевдоним>

Первые десять рабочих областей идентифицируются номерами 1-10 или буквами A - J. Области с 11-й по 25-ю обозначаются номерами или буквенно-цифровыми именами W11 - W25. Область, в которой мы находимся в данный момент, называется активной рабочей областью и в ней можно работать с находящейся здесь базой данных, используя все допустимые команды системы. Одновременно даже в одной команде можно иметь доступ (с некоторыми ограничениями) к полям других баз. В этом случае имя поля из неактивной области – составное. Имени поля тогда предшествует имя рабочей области, разделенные знаками «-» и «>« или (что более удобно) точкой:

<рабочая область > - > <имя поля>

или

<рабочая область>.<имя поля>

Например, следующие составные имена для поля FAM из базы KADR.DBF идентичны, если эта база открыта в области A или 1:

KADR.FAM, A.FAM, KADR->FAM, A->FAM.

При входе в СУБД активизируется область 1 (или A), и, если вы работаете только с одной базой, заботиться об открытии областей не нужно.

Пример. В файле KADR среди прочих содержатся сведения о фамилиях и табельных номерах работников, а в файле BRIG1 - о табельных номерах TAB и выработке VIR. Необходимо по фамилии (например, МИРОНОВ) из файла KADR найти его выработку из файла BRIG1.

```
CLOSE DATA
HSE kadr                          - открытие файла KADR.DBF
USE brigl IN 0                    -открытие BRIG1.DBF в свободной области
LOCATE FOR fam='МИРОНОВ'          - поиск записи с фамилией МИРОНОВ
SELECT brigl                      - переход в область BRIG1
LOCATE FOR tab=b.tab              - поиск записи в файле BRIG1 с табельным номером МИРОНОВА
? a.fam,    tab, vir      - выдача данных из обеих баз
МИРОНОВ Р.И. 468 204000
```

Замечание. Как уже говорилось, в команде USE можно одновременно указывать и область, в которой открывается база данных. Однако переход в указанную область при этом не происходит. Так, после выполнения команд USE brig1 IN 0 мы все равно остаемся в текущей области A.

Установление связей между файлами

В FoxPro допускается работа сразу с несколькими базами данных в при этом возможно установление связей между ними. Указатели записей в таких связанных базах будут двигаться синхронно. База, в которой указатель движется, произвольно, считается старшей, а база/базы, в которой указатель следует за указателем старшей базы, - младшей. В старшей и младших базах должны быть поля, несущие какой-то общий признак, иначе, хотя связь и возможна, она будет бессмысленна. Допускается сцепление одной базы с несколькими другими. Младшие базы, в свою очередь, могут быть связаны с базами следующего уровня и т.д.

Возможно установление двух типов связей между записями двух сцепленных баз данных. Связь типа одна-запись-к-одной перемещает указатель в младшей базе таким образом, что он всегда устанавливается в младшей на первую встреченную им запись с совпадающим признаком. Остальные такие записи (если есть) остаются «не замеченными». Эта связь устанавливается просто командой SET RELATION. Связь типа одна-запись-ко-многим позволяет обратиться ко всем записям младшей базы с совпадающим признаком (команды SET RELATION и SET SKIP TO).

Оба типа связей могут быть распространены на несколько баз сразу.

Связь вида одна-запись-с-одной

Команда
SET RELATION TO <ключ> INTO <область>
[,<ключ> INTO <область>...] [ADDITIVE]
связывает указатель записей в активной рабочей области с указателями записей из других рабочих областей, имена которых указаны после слова INTO, по заданному общему полю (ключу). Единственное условие - файл, с которым устанавливается связь, должен быть проиндексирован по этому полю.

Пример. Связать файлы KADR.DBF и BRIG1.DBF по полю TAB. Вывести для каждого табельного номера файла BRIG1.DBF соответствующую фамилию и выработку.

```
USE brigl IN a
USE kadr INDEX kadrftab IN b
SET RELATION TO tab INTO Ь
LIST tab,vir,b.fam
```

Record	#	TAB	VIR	B.FAM
	1	98	446000	ПОТАПОВ Д.П.
	2	6	480072	КУЛАКОВА М.И.
	3	13	120000	СИДОРОВ П.С.
	4	468	204000	МИРОНОВ Р.И.

Здесь выведены записи файла BRIG1.DBF, в которые включено поле соответствующих им фамилий KADR.DBF (B.FAM).

В FoxPro имеется возможность устанавливать связи с несколькими базами одновременно. Если со старшим файлом, который уже связан с другим, необходимо связать некоторый третий (четвертый и т.д.), следует во все последующие команды SET RELATION включить слово ADDITIVE, которое обеспечит сохранение связей, установленных ранее.

Связь между всеми файлами разрывается командой SET RELATION TO без параметров. Связь с отдельным файлом в заданной <области> - командой

SET RELATION OFF INTO <область>

Для сохранения созданной связи используют команду

CREATE VIEW < имя файла связи>

Открыть файл связи можно при помощи команды

SET VIEW TO < имя файла связи>

Связь вида одна-запись-со-многими

Следующая команда устанавливает связь такого типа между двумя или несколькими базами данных:

- SET SKIP TO [<область1> [,<область2>]...]

При этом с каждой записью из старшей базы могут быть сцеплены несколько записей из младшей базы. Связь может быть установлена сразу с несколькими младшими базами, находящимися в указанных <областях>.

Прежде чем использовать команду SET SKIP TO, необходимо выполнить начальное сцепление вида одна-запись-с-одной командой SET RELATION.

Удаление связи одна-запись-со-многими осуществляется командой SET SKIP TO без параметров.

Пример. Пусть в бригадных файле BRIG3.DBF некоторые табельные номера могут встречаться несколько раз (например, если фиксируются выработки каждого работника по отдельным нарядам). Допускаем также, что рабочие могут работать сразу в нескольких бригадах. Требуется для каждой фамилии и табельного номера из базы KADR.DBF предъявить все выработки данного работника из баз BRIG3.DBF и BRIG5;DBF, которые проиндексированы по полю TAB (индексы BRIG3.IDX и BRIG5.IDX). BROWSE-ОКНО показано на рис. 1., решение приведено ниже:

```
SELECT a
USE kadr IN a                          && Открытие старшей базы области A
USE brig3 IN b INDEX brig3             && Открытие младшей базы в области B
USE brig5 IN c INDEX brig5             && Открытие другой младшей базы в C
* Установление связи одна-с-одной базы KADR.DBF с
* базами BRIG3.DBF, BROG5.DBF
SET RELATION TO tab INTO b, tab INTO c
* Установление связи одна-со-многими базы KADR.DBF с
SET SKIP TO b,c                        && базами BRIG3.DBF и BRIG5.DBF
BROWSE FIELDS a.fam :H='Фамилия' .a.tab :H='Табель',;
              b.tab :H='Таб/Бриг3',b.vir :H='Выр/Бриг3', ;
              c,tab :H='Таб/Бриг5',c.vir :H='Выр/Бриг5'
ЗET RELATION TO                  && Отмена всех связей
```

Фамилия	Табель	Таб/Бриг3	Выр/Бриг3	Таб/Бриг5	Выр/Бриг5
Ефимов А.П.	446	446	280050	446	50000
		446	130000	446	80065
		446		446	10020
Ларионов Т.С.	321	321	25070	321	650000
		321	60000	321	

Рис. 1

Для наглядности здесь выведены табельные номера работников из всех бригад. Видим, что ЕФИМОВ А.П. с табельным номером 446 в файле BRIG3.DBF имеет две записи, а в файле BRIG5.DBF - три, ЛАРИОНОВ Т.С. в бригаде номер 3 - две записи, а в бригаде номер 5 -одну. Здесь повторяющиеся поля из старшей базы отображены символом заполнения. Для быстрого перемещения от записи к записи в старшей базе можно использовать клавиши Ctrl-<вниз>/<вверх>.

Хотя клавиши дополнения и удаления (Ctrl-N, Ctrl-T) здесь доступны, они действуют только на старшую базу. Если при этом нужно, чтобы что-то происходило и с младшими базами, следует их перепрограммировать.

Команды DISPLAY/LIST предъявляют записи похожим образом, но для каждой сцепленной записи из младших баз значения полей из старшей базы будут повторяться. Ниже приведен результат выполнения команды

```
LIST a.fam,a.tab, b.tab,b.vir,c,tab,c.vir OFF :

A.FAM                A.TAB    B.TAB    B.VIR          C.TAB
   C.VIR
ЕФИМОВ А.П.          446      446      280050         446
   50000
ЕФИМОВ А.П.          446      446      130000         446
   80065
ЕФИМОВ А.П.          446      446      0              446
   100020
ЛАРИОНОВ Т.С.        321      321      25070          321
   650000
ЛАРИОНОВ Т.С.        321      321      60000          321      0
```

Рассмотренный пример соответствует сцеплению одной базы с одной. При этом реализовано два уровня данных. База KADR.DBF образует старший, первый уровень, а база BRIG3.DBF второй уровень (рис.2,).

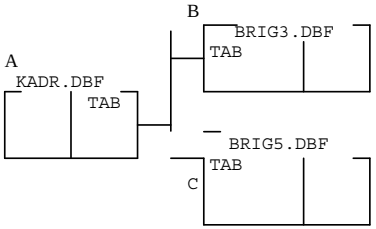


Рис. 2

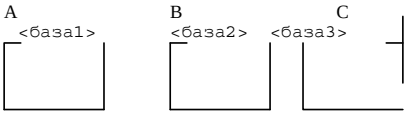


Рис.3

Сцепление баз можно распространить и на большее число уровней. Например, пусть имеется три базы данных (<база1>, <база2>, <база3>), в которых содержатся данные о предприятиях, цехах для всех предприятий и участках для всех цехов всех предприятий (рис..3). Необходимо связать эти базы таким образом, чтобы для каждой текущей записи из базы предприятий указатель записей базы цехов становился на первую запись, где находятся сведения о цехе, принадлежащем данному предприятию, а указатель в базе участков - на запись с информацией об участке этого цеха. Естественно, что все базы должны иметь поля, позволяющие осуществить их содержательное сцепление: база цехов должна иметь поле с названиями предприятий, а база участков - поле с названиями предприятий и поле с номерами цехов. Именно по этим полям должно быть сделано индексирование.

В отличие от предыдущего примера, где сцепление базы KADR.DBF выполнялось со всеми подчиненными базами сразу, здесь команды SET RELATION TO вводятся по мере перемещения из области в область. Схема организации связей приведена ниже:

```
USE <бава1> IN a
USE <база2> IN b INDEX <ивдекс2>
USE <бава3> IN c INDEX <индекс3>
SELECT a

SET RELATION TO <ключ базы1> INTO Ь          &&Сцепление Базы1 с Базой2

SELECT Ь
SET RELATION TO <ключ базы2> INTO c          &&Сцепление Баэы2 с Базой3
```

Связь между базами может быть установлена не только по некоторому ключевому полю, как показано в примерах, но и по номеру записи, если в качестве <ключа> в команде SET

RELATION использовать функцию RECNOQ. Это позволяет, при необходимости соединить две «параллельные» базы. Такая ситуация может возникнуть, если в базе должно быть более 255 предельно допустимых полей. В этом случае можно организовать вторую базу, являющуюся продолжением первой, и связать их по номеру записи как показано ниже.

```
USE bazal
USE baza2 IN O
SET RELATION TO RECNO ('bazal') INTO baza2
                                                    && связывание
SROWSE FIELDS bazal.<поле>,baza2.<поле>
                                                    && просмотр
```

Аппарат сцепления баз командой SET RELATION является мощным средством доступа к «родственным» данным. Однако поскольку такое сцепление влечет синхронное перемещение указателей записей во всех подчиненных базах вслед за перемещением указателя в главной базе, это может отнимать много времени в случае, если доступ к младшей базе в данный момент не нужен. Поэтому часто бывает целесообразным временное разъединение баз. Во многих случаях вообще лучше прибегнуть к поиску нужной записи командой/функцией SEEK, нежели установлению постоянной связи.

Задание 3

1. Установить связь между файлами GROUP и STUDY (создать файл GR-ST.VEIW).
2. Вывести в текстовый файл фамилии, названия предметов и оценки только для женщин.

Для проверки представить файлы DBF :GROUP STUDY , файл GR-ST.VEIW и текстовый файл, созданный в вопросе 2.

4. ФОРМИРОВАНИЕ ЗАПРОСОВ ИЗ БАЗЫ ДАННЫХ

Команда SELECT

Команда является мощным средством обработки запросов. С ее помощью из базы-источника выделяются нужные данные и пересылаются на экран или в файл-приемник. Данные могут быть извлечены из разных баз, а также сгруппированы и упорядочены желаемым образом.

Команда имеет массу опций-возможностей. Ввиду этого сначала приведем ее предварительный синтаксис, который позволит затем лучше осознать детали.

SELECT <что выводится>
FROM <откуда (источник)> INTO <куда (получатель)>
WHERE <каким условиям должно отвечать >
GROUP BY <колонки, по которым выполняется группирование>
HAVING <условие группирования записей в одну строку>
ORDER BY <в каком порядке выводить данные>

Команда SELECT и вообще команды SQL сами открывают нужные им базы данных и индексные файлы. Если необходимых для выполнения команды индексов нет, они будут созданы, а по завершении команды уничтожены. Открытие соответствующей базы данных (например, с мощью команды SELECT) открывает и индекс.

Команда SELECT допускает включение в себя других внутренних команд SELECT (формирование подзапросов).

Все примеры использования команды SELECT сгруппированы в конце раздела. Сейчас рассмотрим опции команды.

Указание результатов выборки и источников данных

SELECT [DISTINCT] [<псевдоним>.]<выражение> [AS <колонка>]
FROM <БД> [<псевдоним>] [,<БД> [<псевдоним>]...]

Здесь указывается, что и откуда берется в выборку. Перед словом FROM перечисляются отбираемые <выражения>, а по-

сле – имена баз, из которых берутся данные.Если необходимо построить выборку из всех полей базы, вместо их перечня можно указать символ «*».

В <выражении> могут быть использованы любые функции FoxPro. Кроме того, здесь есть еще собственные специальные арифметические функции, действующие «по вертикали». Это функции вычисления среднего, минимального и максимального значений суммирования, а также количества записей:

AVG(<выр>), MIN(<выр>), MAX(<выр>), SUM(<выр>), COUNT(<выр>).

Последняя функция может иметь в качестве аргумента звездочку (COUNT(*)), что означает подсчет всех записей, попавших в выборку.

Включение опции DISTINCT исключает возможность вывода одинаковых строк в выборке.

Указание объекта, куда пересылается выборка

Данные из запроса могут передоваться в другую базу данных, массив, текстовый файл, экран и принтер.Кроме того, информация может быть переслана в так называемый Курсор. Курсор - это временный набор данных, который может быть областью памяти или временным файлом FoxPro и имеет режим «Только чтение». Данные Курсора могут быть, например, предъявлены в команде BROWSE, напечатаны, из них может быть образовано меню и т.д. Курсор может быть обработан другой командой SELECT. К колонкам Курсора надо обращаться по имени, этих колонок, возможно, с префиксом - именем Курсора (через точку).

Итак:

INTO <получатель >

<Получатель> может быть одного из следующих типов:

ARRAY <массив> - задается вновь создаваемый двумерный <массив>.

CURSOR <курсор> - задается имя Курсора.

DBF/TABLE <БД> - новая база данных с указанным именем. Слова DBF и TABLE здесь являются синонимами.

Кроме того, данные можно переслать в файл или на принтер.

TO FILE <файл> [ADDITIVEJ/TO PRINTER - выборка посылается в текстовый <файл> или на принтер. Если используется слово ADDITIVE, то выборка будет добавлена в конец существующего файла без его перезаписи.

Следующие опции имеют смысл только при выдаче на экран (команда используется без слова INTO):

NOCONSOLE - выборка не выдается на экран.

PLAIN - заголовки колонок не выдаются.

NOWAIT - не делаются паузы при заполнении экрана.

Критерий отбора данных

WHERE <условие связи> [AND <условие связи> ...]

[AND/OR <условие отбора> [AND/OR <условие отбора>…]]

Здесь:

<Условие связи> - применяется в случае, если выборка делается более, чем из одной базы данных, и указывает критерий, которому должны отвечать поля из разных баз. В условии связи указываются поля из разных баз. Здесь разрешается использовать знаки отношения =, #, ==, >, >=, <, <=. Допускается задание нескольких критериев, соединенных знаком AND.

<Условие отбора> - строится аналогично, но из выражений только для одной базы, и допускается использование логических операторов OR и NOT.

Условия, кроме любых функций FoxPro, могут содержать следующие операторы SQL:

LIKE - позволяет построить условие сравнения по шаблону, где символ «_» указывает единичный неопределенный символ в троке, «%»- любое их количество. Эти символы аналогичны символам маски «?» и «*» в MS DOS. Формат оператора:

<выражение> LIKE <шаблон>

BETWEEN - проверяет, находится ли выражение в указанном диапазоне. Формат оператора: <выражение> BETWEEN <нижнее знач.> AND <верхнее знач.>

IN - проверяет, находится ли выражение, стоящее слева от слова IN, среди перечисленных справа от него (аналогично функции INLIST). Формат оператора:

<выражение> IN (<выражение>,<выражение>,...)

Все указанные операторы можно комбинировать с помощью связок OR, AND, NOT и скобок. Операторы LIKE и

BETWEEN не следует путать с одноименными функциями FoxPro, которые, впрочем, тоже можно использовать.

Группированные данных

GROUP BY <колонка>[,<колонка>...] – задаются колонки, по которым производится группирование выходных данных. Все записи базы, для которых значения колонок совпадают, отображаются в выборке единственной строкой. Группирование удобно для получения некоторых сводных характеристик (сумм, количеств) группы.

HAVING <условие отбора> - опция задает критерий отбора данных в каждую сформированную в процессе выборки группу.

Сортировка

ORDER BY <колонка> [ASC/DESC][,<колонка> [ASC/DESC]...] -опция задает упорядочение по заданной колонке/колонкам. По умолчанию сортировка выполняется по возрастанию (ASC), но, может быть задана и по убыванию (DESC).

Примеры запросов

1. Выборка всех полей из базы KADR.DBF. Все колонки выборки будут иметь имена полей базы данных.

```
SELECT * FROM kadr
```

2. Вывод минимального, максимального и среднего значений поля SZAR (средняя зарплата). Колонки получат имена MIN_SZAR, MAX_SZAR и AVG_SZAR.

```
SELECT MIN(szar),MAX(szar),AVG(szar) FROM kadr
```

3. Вывод фамилий работников, получающих от 300000 до 800000 рублей.

```
SELECT fam FROM kadr;
        WHERE szar BETWEEN 300000 AND 800000
```

4. Вывод фамилий всех сотрудников, кроме работающих в подразделениях ОГМ и КБ.

```
SELECT fam FROM kadr WHERE podr NOT IN ('ОГМ', 'КБ')
```

5. Выборка названий всех подразделений (поле PODR) предприятия из базы KADR.DBF. Опция DISINST предотвращает

повторный вывод одних и тех же названий, если они повторяются.

```
SELECT DISTINCT podr FROM kadr
```

6. Выборка фамилий (PAM) всех мужчин из KADR.DBF.

```
SELECT fam FROM kadr WHERE pol='M'
```

7. Выборка всех фамилий и табельных номеров из KADR.DBF сцепленных с выработками из базы BRIG1.DBF для записей, у которых совпадают табельные номера.

```
SELECT s.fam,s.tab,t.tab,t.vir;
        FROM kadr a,brig1 t WHERE s.tab=t.tab
```

Здесь для сокращения записи команды базам KADR.DBF и BRIG1. DBF заданы новые временные имена S и T. Они никакие связаны с рабочими областями. Базы будут открыты в свободных областях системы. Сами колонки выборки получат имена FAM, TAB_A, TAB_B,VIR.

8. Если мы хотим задать собственные имена колонкам, а не использовать умолчания, нужно воспользоваться опцией AS.

Пусть нужно вывести фамилии и табельные номера (поля FAM и TAB) по алфавиту и с другими именами колонок FAMILII и TABEL.

```
SELECT fam AS familii, tab AS label;
        FROM kadr ORDER BY fam
```

9. Выборка фамилий всех родившихся в текущем месяце с указанием дня (числа) рождения, количества лет и премии по этому поводу - 50 % от значения средней зарплаты.

```
SELECT fam, DAY (dtr),'число',YEAR (DATE ()) -YEAR(dtr),;
        'лет', 'премия',0.5*szar FROM kadr;
WHERE MONTH(dtr)=MONTH(DATE())
```

Колонки получат имена FAM и от EXP_2 по EXP_7.

10. Вывод полей FAM и TAB, отсортированных по полям POL (главное поле) и FAM (подчиненное поле) в базу FAMTAB.DBF, которая затем открывается в текущей области.

```
SELECT fam, tab FROM kadr;
  ORDER BY pol,fam INTO TABL famtab
```

Чтобы увидеть содержимое этой базы, можно, например, сразу ввести команду BROWSE.

11. Вывод для каждого табельного номера из базы KADR.DBF выработок из баз BRIG1.DBF и BRIG3.DBF, а также суммарной выработки работника в обеих бригадах.

```
SELECT kadr.tab,brigl.vir,brig3.vir,brigl.vir+brig3.vir;
    FROM kadr,brigl,brig3;
    WHERE kadr.tab=brigl.tab AND kadr.tab=brig3.tab
```

Имеется в виду, что работник может работать в нескольких бригадах в течение месяца, но в каждом бригадном файле он может встретится только раз.

12. Вывод фамилий всех работников, работавших ранее в конструкторском бюро (КБ). Поиск ведется в мемо-поле PER базы KADR.DBF.

```
SELECT fam FROM kadr WHERE per LIKE «%КБ%»
```

13. Вывод табельных номеров и суммарной выработки каждого работника в бригаде номер 1. Вывод осуществляется в порядке увеличения табельных номеров.

```
SELECT tab,SUM(vir) FROM brigl;
       GROUP BY tab ORDER BY tab
```

Задача имеет смысл, если один и тот же человек может встречаться несколько раз в одной бригаде, например, если в бригадном файле фиксируются не итоговые выработки, а все наряды.

14. Вывод названий всех подразделений, количества сотрудников Значений суммарной заработной платы (фонда оплаты). Информация выводится только для подразделений, где количество сотрудников больше пяти.

```
SELECT podr,COUNT(*),SUM(szar);
  FROM kadr GROUP BY podr HAVING COUNT(*)>5
```

Задание 4

Создать запросы и результаты запросов сохранить в файлах DBF/TXT.

1. Вывести сведения о мужчинах, родившихся с 1 января по 1 апреля заданного года.
2. Вывести фамилии, названия предметов и оценки отдельно для мужчин и женщин.

3. Вычислить количество студентов, имеющих оценки 2,3, 4, 5 (запрос с группировкой по полю BALL).
4. Вывести фамилии, оценки и названия предметов, сгруппировав сведения по предметам.
5. Вывести фамилию студента, имеющего максимальный номер полиса.

Для проверки представляются результаты запросов.