## Section I:

In this section, we create a database and enter data. We also introduce you to some dBASE II commands that will be developed and added to throughout the rest of this manual. For a complete definition of a command, check Part II.

## How to CREATE a database

We'll start by creating a database of names for a mailing list system. Each reoord in the database will contain the following information:

NAME: up to 20 characters long
ADDRESS: up to 25 characters long
CITY; up to 20 characters long
STATE: 2 characters long
ZIP CODE: 5 characters

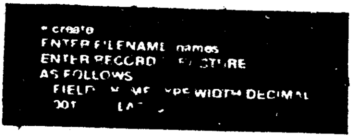
First, type ^**CREATE**^.

dBASE II responds with: **ENTER FILENAME:**.

Enter a filename starting with a letter and up to 8 characters long (limited by CP/M), no colons, no spaces. Since this is a file of names, let's call it something that makes sense to a human being: type ^**Names**^.

When you hit return, dBASE II creates a file called **<NAMES.DBF>**. The part of the name after the period is the CP/M file **name** extension, and is short for <u>d</u>ata<u>b</u>ase <u>f</u>ile (Section V, File Types).

In a **database management** system, each one of the items that we want to enter into a single related grouping is called a <u>field</u> and the grouping is called a <u>record</u> (Section V, Database Basics). In our example, each record will have 5 fields. dBASE needs to know the name of each field, whac type of data it will contain, how long it is and how many decimal places if the data is numeric.



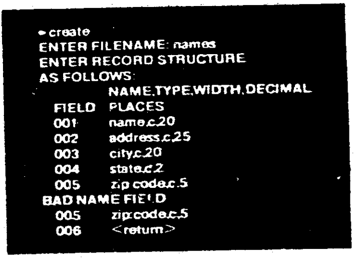
```
* create
ENTER FILENAME names
ENTER RECORD STRUCTURE
AS FOLLOWS
  FIELD NAME TYPE WIDTH DECIMAL
  001       LA
```

Field names can be up to 10 characters long, and may be entered in upper and/or lowercase. The name must start with a letter and cannot contain spaces, but can contain digits and embedded colons. Don't abbreviate any more than you have to: the computer will understand what you mean, but people might not.

The type of data is specified by a single letter: **C** for Character, **N** for Numeric and L for Logical. In this case, all fields contain character data.

Field width can be any length up to 254 characters. If the field is numeric and decimal places are specified, remember that the decimal point also takes one character position.

We know what names we want to give our fields, the type of data that they will contain, and their lengths so type the information in now. Here's what the screen looks like when you're finished:



```
•create
ENTER FILENAME: names
ENTER RECORD STRUCTURE
AS FOLLOWS:
              NAME,TYPE,WIDTH,DECIMAL
   FIELD      PLACES
   001        name,c,20
   002        address,c,25
   003        city,c,20
   004        state,c,2
   005        zip code,c,5
 BAD NAME FIELD
   005        zipcode,c,5
   006        <return>
```

Notice what happened at field 5: we made an error by entering a space in the field name, so dBASE II told us what the error was and gave us a chance to correct it.

Notice also that the data type for the ZIP code was specified as "character", even though we normally think of the digits here as numbers.

This was done because a dBASE II command such as ^TOTAL^ can total all the numeric fields in a record (without you specifically listing them all). Doing so with the ZIP code field would simply be a waste of time. We can still use the relational operators ("greater than", "less than", "equal or not equal to") with the character data, so this will not interfere with any ZIP code sorting we may want to do later.

When dBASE II asked us for the specifications for a sixth field, we hit <enter> to end the data definition. dBASE II saved the data structure, then asked if we wanted to enter data in it.

The <Names.DBF> database is immediately ready for data entry, so type ^y^. On the next page we tell you how tc enter the data.

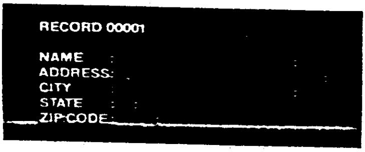
## Entering data into your new database

If you do not have full screen editing on your terminal, the record number and the field names will appear one at a time below whatever has been typed on the screen up until now. The length of each field is shown by two colons, with the cursor positioned for you to start writing. When you fill the field or press <enter>, the next field will appear. After the last field in a record has been filled (or ignored), you start on a new record.

To stop entering data, hit <enter> when the cursor is at the first character position of the first field in a new record.

If you installed dBASE II with full screen editing, the screen will be erased, then the record number and all the fields will be displayed starting in the upper left-hand corner of the screen, with the cursor at the first character position of the first field.

(If you chose one of the standard terminals on the installation list, the field names may be in reverse video or at half-intensity. If you want to change this later, you can disable it by using the "Y - CHANGE/MODIFY" option in the installation procedure).



```
RECORD 00001

NAME    :                    :
ADDRESS:                          :
CITY    :                    :
STATE   : :
ZIP CODE:    :
```

NOTE:: If this doesn't look like your screen, there is a problem with INSTALL. Please re-do the installation.

Field lengths are indicated by two colons. When a field is filled or you hit <enter>, the cursor jumps down to the next field. The cursor can be moved back up to a previous field by holding the control key down and pressing the letter E once: ^control-E^, abbreviated as ^ctl-E^. When you are finished with the last field, dBASE II presents another empty record.

Enter the following names and addresses. We'll be using them soon to show you some of the powerful features of dBASE II.

| | | |
|---|---|---|
| ALAZAR, PAT | 123 Crater Rd., Everett, WA | 98206 |
| BROWN, JOHN | 456 Minnow Pl., Burlington, MA | 01730 |
| CLINKER, DUANE | 789 Charles Dr., Los Angeles, CA | 90036 |
| DESTRY, RALPH | 234 Mahogany St., Deerfield, FL | 33441 |
| EMBRY, ALBERT | 345 Sage Ave., Palo Alto, CA | 94303 |
| FORMAN, ED | 456 Boston St., Dallas, TX | 75220 |
| GREEN, TERRY | 567 Doheny Dr., Hollywood, CA | 90046 |
| HOWSER, PETER | 678 Dusty Rd., Chicago, IL | 60631 |

If you make any mistakes that can't be corrected by backspacing and writing over them, read the next two pages on editing before moving on to the next record. If you accidentally get back to the dBASE dot prompt, type:

**^USE Names^**
**^APPEND^**

and continue with your entries. (This will be explained later in the manual).

To stop entering data, after you've entered the last ZIP code and while you are on the first character of the first field of the next record, hit <enter>. If you have typed in some data or moved the cursor, hold the control key down and press the letter "Q: (^control-Q^).

dBASE leaves the data entry mode and presents its dot prompt (.) to show you that it's ready for your commands.

If you want to stop now, simply type ^QUIT^.

^QUIT^ **must be typed every time you terminate a dBASE II session. This automatically closes all files properly. Unless you do so, you may destroy your database.**

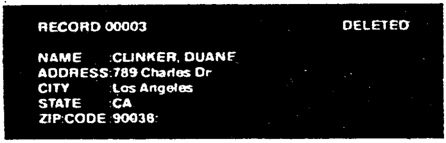
## Modifying data with EDIT

If you made any errors in the entries, you can correct them quickly and easily in the Full Screen Edit mode. Type:

**^USE Names^**
**^EDIT <number>^**

where "number" is the number of one of the records in the database.

dBASE brings up the entire record and you can use the Full Screen Editing commands to modify any or all of the data in the record. To move to the next record, use **^ctl-C^**. To move to the previous record. use **^ctl-R^**. To try it, type **^EDIT 3^**.



```
RECORD 00003                         DELETED

NAME    :CLINKER, DUANE
ADDRESS:789 Charles Dr
CITY    :Los Angeles
STATE   :CA
ZIP:CODE:90036:
```

If you mark a record for deletion by using **^ctl-U^**, "DELETED" appears at the top of the screen. Pressing **^ctl-U^** again removes the word and "un-deletes" the record. If you **^LIST^** (pp.20 and 21) or **^DISPLAY^** (pp. 20 and 23) your database, you will see an asterisk next to all records marked for deletion.

To abort full-screen editing, use **^ctl-Q^**. This does not make the changes that were on the screen when you exited.

To exit gracefully and save the changes made so far, use **^ctl-W^** (**^ctl-O^**--the letter "O"--with Superbrain).

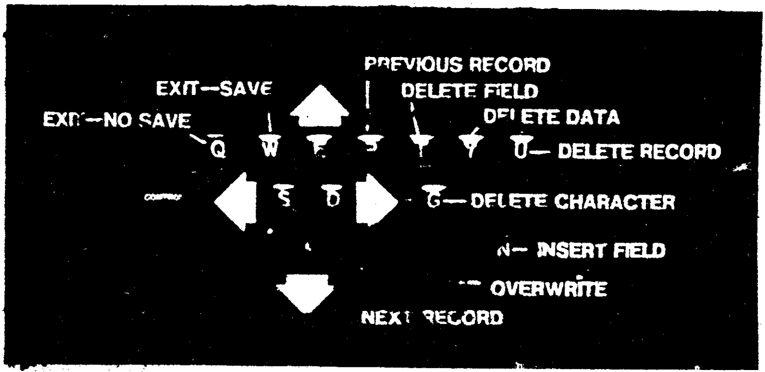
**FULL SCREEN EDITING FEATURES:**

ctl-X moves cursor down to the next field (or ctl-F).
ctl-E moves cursor back to the previous field (or ctl-A).
ctl-D moves cursor ahead one character.
ctl-S moves cursor back one character.
ctl-V toggles between overwrite and insert modes.
ctl-G deletes the character under the cursor.
<Rubout> deletes the character to the left of tne cursor.
ctl-P toggles your printer ON and OFF.
ctl-Q quits and returns to normal dBASE II operation without making changes, even in the MODIFY mode.

**^MODIFY^ functions:**
ctl-T deletes the field where the cursor is and moves all the lower fields up.
ctl-Y clears the current field to blanks, but leaves all fields where they were.
ctl-N moves fields down one position to make room for insertion of a new field at the cursor position.

**^APPEND^ functions:**
ctl-R writes the record to disk and moves to the next record.
<Enter> when cursor is at the initial position of a new record resumes normal dBASE II operation.
ctl-Q erases the record and resumes normal dBASE II operation

**^EDIT^ functions:** (Do not use in ^APPEND^ mode)
ctl-C writes the record to disk and advances to the next record.
ctl-R writes the record to disk and backs up to the previous record.
ctl-U toggles the record deletion mark on and off.
ctl-W saves any changes made and resumes normal dBASE II operation. With Superbrain, use ^ctl-O^ (the letter "O").
ctl-Q aborts any changes in the record you're working on and prints the coordinate prompt. Hit <enter> to resume normal dBASE II operation.



## An introduction to dBASE II commands and the error correction dialog (USE, LIST, DISPLAY)

dBASE II commands are generally verbs. You type them in when you see the dBASE II dot (.) prompt.

When you want to tell dBASE II which database file you want to work with, you type ^USE <filename>^'.

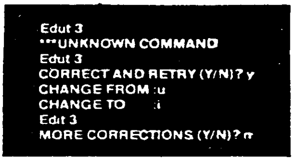
To look at the record you are on, type ^DISPLAY^.

To see all the records in the database, type ^LIST^. (To stop and start the scrolling, use ^control-S^.)

dBASE II commands can be abbreviated to four letters, but if you use more letters they must all be correct (^DISPLAY^, ^DISP^ and ^DISPLA^ are valid commands; DISPRAY is not.)

If you chose the error correction dialog when you installed dBASE II, the command line is scanned and you are prompted with error messages when mistakes are detected. You get a second chance to make corrections without having to retype the entire line.

Type ^EDUT 3^.



```
Edut 3
***UNKNOWN COMMAND
Edut 3
CORRECT AND RETRY (Y/N)? y
CHANGE FROM :u
CHANGE TO      :i
Edit 3
MORE CORRECTIONS (Y/N)? n
```

dBASE II repeats a command it does not know. If you decide to change it, you do not have to retype the entire command.

In response to "CHANGE FROM:" type in enough of the wrong part of the command so that it is unambiguous, then hit <enter>.

In response to "CHANGE TO:" type in the replacement for the material you want changed.

In this example, we changed only a single letter, but you'll find this feature useful when you are testing and debugging long command lines.

Tip: The ^ERASE^ command erases the screen and positions the prompt dot at the upper left-hand corner of the screen so that you can start new commands with a clean slate.

## Expanding commands with expressions and relational operators (LIST)

One of the most powerful features of dBASE II is the ability to expand and "tailor" the commands.

You can add "phrases" and expressions to most commands to further define what the commands will do. Commands can be entered in upper and lowercase letters, and command lines can be up to 254 characters long. To extend the line beyond the width of your display, type in a semicolon (;) as the last character on the line (no space after it). dBASE II will use the next line as part of the command.

Since dBASE II is a relational DBMS, you'll find the **relational operators** useful:

| | |
|---|---|
| < | : less than |
| > | : greater than |
| = | : equal to |
| <= | : less than or equal to |
| >= | : greater than or equal to |

These commands mean exactly what the explanation on the right says. They generate a logical value as a result (True or False). If the expression is True, the command is performed. If the expression is false, the command is not performed.

Earlier, we mentioned that the LIST command will show all the records in the database (to stop and start the scrolling, use ^ctl-S^). The full form of the command is:

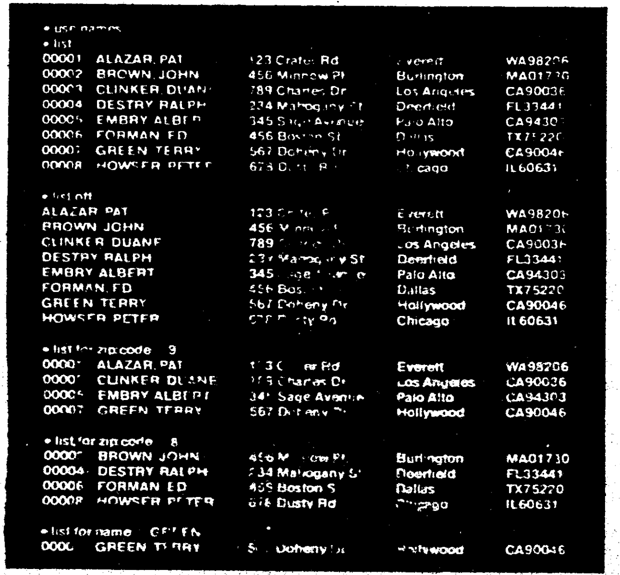
```
^LIST [OFF] [FOR <expression>]^
```

If the optional OFF is used, the record numbers will not be displayed.

If the optional FOR clause is used, dBASE II will list only the records for which the expression is true. Type the following, using single quotes around the character data (more on data types in Section II):

```
^USE Names^
^LIST^
^LIST OFF^
^LIST FOR Zip:Code = '9'^
^LIST OFF FOR Zip:Code < '8'^
^LIST FOR Name='GREEN'^
```

Notice that when you enter only part of the contents of the field, that is all that is compared by dBASE. We did not need Mr. Green's full name, for example, although we might have used it if our database contained several GREEN's.

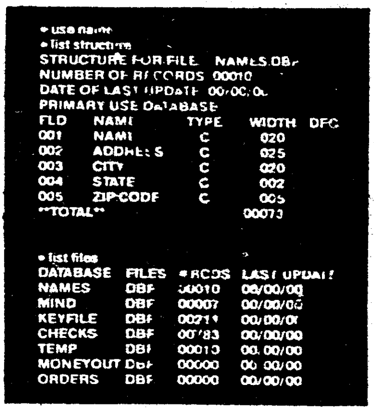


```
• use names
• list
00001  ALAZAR, PAT      123 Crater Rd      Everett       WA98206
00002  BROWN, JOHN      456 Minnow Pl      Burlington    MA01730
00003  CLINKER, DUANE   789 Charles Dr     Los Angeles   CA90036
00004  DESTRY RALPH     234 Mahogany St    Deerfield     FL33441
00005  EMBRY ALBERT     345 Sage Avenue    Palo Alto     CA94303
00006  FORMAN ED        456 Boston St      Dallas        TX75220
00007  GREEN TERRY      567 Doheny Dr      Hollywood     CA90046
00008  HOWSER PETER     678 Dusty Rd       Chicago       IL60631

• list off
ALAZAR PAT              123 Crater Rd      Everett       WA98206
BROWN JOHN              456 Minnow Pl      Burlington    MA01730
CLINKER DUANE           789 Charles Dr     Los Angeles   CA90036
DESTRY RALPH            234 Mahogany St    Deerfield     FL33441
EMBRY ALBERT            345 Sage Avenue    Palo Alto     CA94303
FORMAN, ED              456 Boston St      Dallas        TX75220
GREEN TERRY             567 Doheny Dr      Hollywood     CA90046
HOWSER PETER            678 Dusty Rd       Chicago       IL60631

• list for zipcode   9
00001  ALAZAR, PAT      123 Crater Rd      Everett       WA98206
00003  CLINKER DUANE    789 Charles Dr     Los Angeles   CA90036
00005  EMBRY ALBERT     345 Sage Avenue    Palo Alto     CA94303
00007  GREEN TERRY      567 Doheny Dr      Hollywood     CA90046

• list for zipcode   8
00002  BROWN JOHN       456 Minnow Pl      Burlington    MA01730
00004  DESTRY RALPH     234 Mahogany St    Deerfield     FL33441
00006  FORMAN ED        456 Boston St      Dallas        TX75220
00008  HOWSER PETER     678 Dusty Rd       Chicago       IL60631

• list for name   GREEN
00007  GREEN TERRY      567 Doheny Dr      Hollywood     CA90046
```

In addition to precisely selecting data from your database, the LIST command can be used to provide you with system information.

^LIST STRUCTURE^ shows you the structure of the database in USE.

^LIST FILES^ shows the names of the database (.DBF) files on the logged-in drive. ^LIST FILES ON <drive>^ shows the database files on another drive (do NOT use the usual CP/M colon).



```
• use name
• list structure
STRUCTURE FOR FILE   NAMES.DBF
NUMBER OF RECORDS  00010
DATE OF LAST UPDATE  00/00/00
PRIMARY USE DATABASE
FLD      NAME          TYPE     WIDTH   DEC
001      NAME            C       020
002      ADDRESS         C       025
003      CITY            C       020
004      STATE           C       002
005      ZIP CODE        C       005
**TOTAL**                      00073


• list files
DATABASE   FILES   #RCDS    LAST UPDATE
NAMES      DBF     00010    00/00/00
MIND       DBF     00007    00/00/00
KEYFILE    DBF     00219    00/00/00
CHECKS     DBF     00183    00/00/00
TEMP       DBF     00013    00/00/00
MONEYOUT   DBF     00000    00/00/00
ORDERS     DBF     00000    00/00/00
```

## Looking at data with DISPLAY

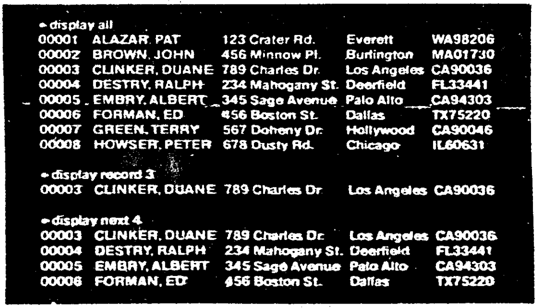
The ^DISPLAY^ command is similar to ^LIST^. Its full form is:

```
        [All      ]
DISPLAY [Record n] [OFF][FOR <expression>]
        [Next n   ]
```

This gives you the option of specifying the scope for the ^DISPLAY^ command (also ^LIST^).

Specifying "Record n" displays only that record; "Next n" displays the next "n" records, including the current record. ^DISPLAY ALL^ is the same as ^LIST^, except that ^LIST^ will scroll all the records in the database up the screen, while ^DISPLAY ALL^ shows you the database in groups of 15 records at a time (pressing any key displays the next 15 records). Type the following:

```
^DISPLAY All^
^DISPLAY Record 3^
^DISPLAY Next 4^
```



```
•display all
00001  ALAZAR, PAT     123 Crater Rd.     Everett      WA98206
00002  BROWN, JOHN     456 Minnow Pl.     Burlington   MA01730
00003  CLINKER, DUANE  789 Charles Dr.    Los Angeles  CA90036
00004  DESTRY, RALPH   234 Mahogany St.   Deerfield    FL33441
00005  EMBRY, ALBERT   345 Sage Avenue    Palo Alto    CA94303
00006  FORMAN, ED      456 Boston St.     Dallas       TX75220
00007  GREEN, TERRY    567 Doheny Dr.     Hollywood    CA90046
00008  HOWSER, PETER   678 Dusty Rd.      Chicago      IL60631

•display record 3
00003  CLINKER, DUANE  789 Charles Dr     Los Angeles  CA90036

•display next 4
00003  CLINKER, DUANE  789 Charles Dr.    Los Angeles  CA90036
00004  DESTRY, RALPH   234 Mahogany St.   Deerfield    FL33441
00005  EMBRY, ALBERT   345 Sage Avenue    Palo Alto    CA94303
00006  FORMAN, ED      456 Boston St.     Dallas       TX75220
```

As with ^LIST^, the optional FOR clause can be used to select specific data by using logical expressions.

The DISPLAY command can also be used like the LIST command for system functions:

```
DISPLAY STRUCTURE = LIST STRUCTURE.
DISPLAY FILES = LIST FILES.
```

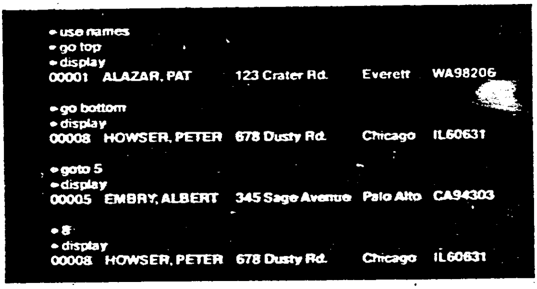
Both ^LIST^ and ^DISPLAY^ can show you specific types of files on a drive using the CP/M "wild cards". ^DISPLAY FILES LIKE *.COM ON B^, for example, would display all the ".COM" files on drive B. If uncertain, check your CP/M manual, then use this form:

```
^DISPLAY FILES LIKE <wild card>^
```

## Positioning commands (GO or GOTO and SKIP)

Once you have your database set up, you can also move from record to record quickly and easily with dBASE II. Type the following:

```
^USE Names^
^GO TOP^
^DISPLAY^
^GO BOTTOM^
^DISPLAY^
^GOTO 5^
^DISPLAY^
^8^
^DISPLAY^
```

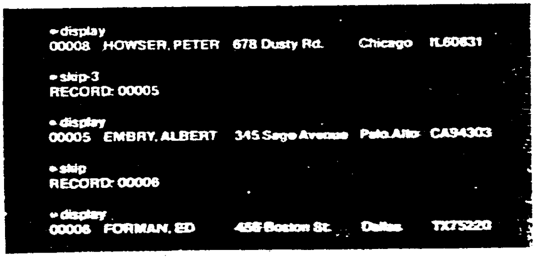


```
• use names
• go top
• display
00001  ALAZAR, PAT        123 Crater Rd.      Everett    WA98206

• go bottom
• display
00008  HOWSER, PETER      678 Dusty Rd.       Chicago    IL60631

• goto 5
• display
00005  EMBRY, ALBERT      345 Sage Avenue     Palo Alto  CA94303

• 8
• display
00008  HOWSER, PETER      678 Dusty Rd.       Chicago    IL60631
```

^GO TOP^ (or ^GOTO TOP^) moves you to the first record in the database. ^GO BOTTOM^ moves you to the last record. You can go to a specific record by using ^GOTO^ <number> (or GO <number>). And you can even eliminate the GO and just specify the record number.

^SKIP^ moves you to the next record. ^SKIP ± n^ moves you forward or backward "n" records. You can also use ~^SKIP ±<variable/expression>, with the number of records skipped determined by the value of the variable or expression (both defined later). Type the following:

```
^DISPLAY^
^SKIP-3^
^DISPLAY^
^SKIP^
^DISPLAY^
```



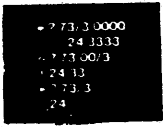
```
• display
00008  HOWSER, PETER      678 Dusty Rd.       Chicago    IL60631

• skip-3
RECORD: 00005

• display
00005  EMBRY, ALBERT      345 Sage Avenue     Palo Alto  CA94303

• skip
RECORD: 00006

• display
00006  FORMAN, ED         456 Boston St.      Dallas     TX75220
```

## The interactive ? command

The ^?^ command allows you to use dBASE II in the calculator mode. Simply type in the question mark and a space followed by the quantity or mathematical function you want evaluated and dBASE II will provide the answer on the next line. Using ?? puts the answer on the same line.

Type the following:

```
^? 73/3.0000^
^? 73.00/3^
^? 73/3^
```

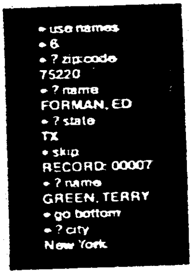


```
• ? 73/3.0000
      24.3333
• ? 73.00/3
 24.33
• ? 73/3
 24
```

The ^?^ command shows the answers to a mathematical operation to the same number of decimal places as the maximum in the numbers entered.

You can also think of ^?^ as meaning: "What is ...", with the dots replaced by an expression, a variable (a field name or a memory variable), a dBASE II function or a list of these separated by commas. Type the following:

```
^USE Names^
^6^
^? Zip:Code^
^? Name^
^SKIP^
^? Name^
^GO BOTTOM^
^? City^
```



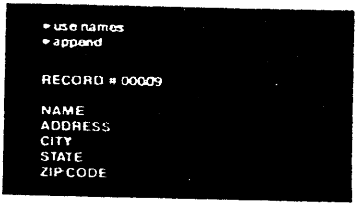
```
• use names
• 6
• ? zip:code
75220
• ? name
FORMAN, ED
• ? state
TX
• skip
RECORD: 00007
• ? name
GREEN, TERRY
• go bottom
• ? city
New York
```

In the section on functions and commands, we'll show you how the ^?^ can be used to access other dBASE II functions, and to display CRT prompts to the operator from a command file.

## Adding more data with the APPEND and INSERT commands

You can add data to any database quickly and easily with a one-word command. First choose the database file into which you want to enter data by typing ^USE <filename>^, then typing in the command ^APPEND :

**^USE Names^**
**^APPEND^**



```
• use names
• append

RECORD # 00009

NAME
ADDRESS
CITY
STATE
ZIP CODE
```

dBASE II responds by displaying the record number that follows the last record in the file and the fields for that database. If you fill in the record, it is added onto the end of the file (appended).

The display includes the names of the fields, with colons showing field lengths. The cursor is at the first position where you can start to enter data. If you fill up the entire field with data, the cursor automatically moves down to the next field. If not, hit <enter>.

If there is no data to be entered in a field, use <enter> to move the cursor to the next field. Character fields will automatically be filled with blanks, numeric fields will show a zero. When entering numeric data, if there are no digits after the decimal, there is no need to type the decimal. dBASE II automatically puts in the decimal point and the necessary number of following zeros.

Records can be inserted into a specific location in a database (to keep them alphabetical, for example) by typing

**^INSERT [BEFORE] [BLANK]^**

Using the word **^INSERT^** alone inserts the record just after the current record. Specifying **BEFORE** will insert the record just before the current record. In either case, you are prompted the same way as with the **^APPEND^** and **^CREATE^** commands. If **BLANK** is specified, an empty record is inserted and there are no prompts.

Add the following names alphabetically to the <Names.DBF> database:

```
EDMUNDS, JIM   392 Vicarious Way, Atlanta, GA   30328
INDERS, PER    321 Sawtelle Blvd., Tucson, AZ   85702
JENKINS, TED   210 Park Avenue, New York, NY    10016
```

The sequence of commands is:

```
^USE Names^
^5^
^INSERT BEFORE^   (enter the data for the first name)
^APPEND^          (enter the data for the last names)
```

In the ^INSERT^ mode, when you fill the last field, dBASE II will return to the command mode (dot prompt).

To exit the ^APPEND^ mode, position the cursor at the start of a new field, then hit <enter> or ^control-Q^.

In either mode, you can exit from inside a record by using ^ctl-W^ (^ctl-O^ with Superbrain). This will save what has been entered up to that point and return you to the command mode.

## ^Cleaning up a database (DELETE, RECALL, PACK)

Deletions can be made directly from dBASE II as well as in the ^EDIT^ mode.

To delete the current record, type ^DELETE^.

To delete more than one record, use the form ^DELETE <scope>^, where the scope is the same as for other dBASE II commands: All, Record n, or Next n.

To make the deletions conditional, expand the command to:

^DELETE [scope] [FOR <expression>]^

where "expression" is a condition or set of conditions that must be met. (This is developed in more detail in Section II).

Type ^DELETE FILE <drive>:<filename>^ to delete a file. But once you've done this, the data is gone forever, so be careful.

Unlike files, records marked for deletion can be recovered. Rather than erasing the data, ^DELETE^ marks each record with an asterisk. You will see the asterisks when you ^LIST^ or ^DISPLAY^ the records. dBASE II then ignores these records, and does not use them in any processing.

To restore the records, use the following command:

^RECALL [scope] [FOR <expression>]^

This operates the same way ^DELETE^ does, with the scope and condition being optional. If a conditional expression is used, it does not have to be the same as was used to mark the records for deletion.

At some point, however, you will want to clean up your files to clarify displays or to make more room for storage. To do this, type:

^PACK^.

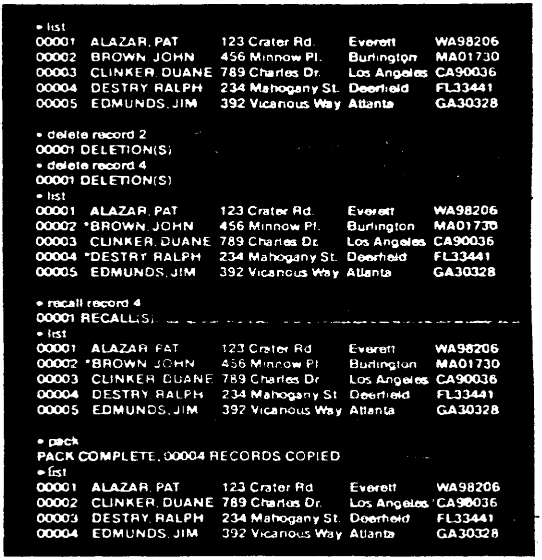
This erases all records marked for deletion, and tells you how many records are in the database.

Note: once you use this command, the records are lost forever.

To see how these commands work, type the following:

```
^USE Names^
^LIST^
^DELETE RECORD 2^
^DELETE RECORD 4^
^LIST^
^RECALL RECORD 4^
^LIST^
^PACK^
^LIST^
```

The screen below shows the first few records in our <Names.DBF> as we perform these commands.



```
• list
00001   ALAZAR, PAT       123 Crater Rd.      Everett      WA98206
00002   BROWN, JOHN       456 Minnow Pl.      Burlington   MA01730
00003   CLINKER, DUANE    789 Charles Dr.     Los Angeles  CA90036
00004   DESTRY, RALPH     234 Mahogany St.    Deerfield    FL33441
00005   EDMUNDS, JIM      392 Vicarious Way   Atlanta      GA30328

• delete record 2
00001 DELETION(S)
• delete record 4
00001 DELETION(S)
• list
00001   ALAZAR, PAT       123 Crater Rd.      Everett      WA98206
00002  *BROWN, JOHN       456 Minnow Pl.      Burlington   MA01730
00003   CLINKER, DUANE    789 Charles Dr.     Los Angeles  CA90036
00004  *DESTRY, RALPH     234 Mahogany St.    Deerfield    FL33441
00005   EDMUNDS, JIM      392 Vicarious Way   Atlanta      GA30328

• recall record 4
00001 RECALL(S)
• list
00001   ALAZAR, PAT       123 Crater Rd.      Everett      WA98206
00002  *BROWN, JOHN       456 Minnow Pl.      Burlington   MA01730
00003   CLINKER, DUANE    789 Charles Dr.     Los Angeles  CA90036
00004   DESTRY, RALPH     234 Mahogany St.    Deerfield    FL33441
00005   EDMUNDS, JIM      392 Vicarious Way   Atlanta      GA30328

• pack
PACK COMPLETE, 00004 RECORDS COPIED
• list
00001   ALAZAR, PAT       123 Crater Rd.      Everett      WA98206
00002   CLINKER, DUANE    789 Charles Dr.     Los Angeles  CA90036
00003   DESTRY, RALPH     234 Mahogany St.    Deerfield    FL33441
00004   EDMUNDS, JIM      392 Vicarious Way   Atlanta      GA30328
```

## Section I Summary

At this point, you have learned about the power over data that a relational database management system like dBASE II can give you.

You can now ^CREATE^ a new database and start entering data in minutes.

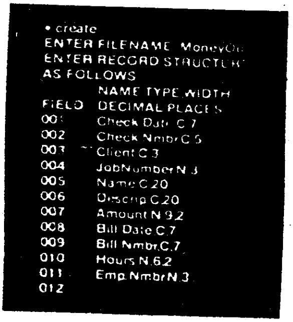
If you want to change the data, this is easily done with ^EDIT^, ^DELETE^, ^RECALL^ and ^PACK^.

You can ^APPEND^ or ^INSERT^ more data as required, and ^LIST^ and ^DISPLAY^ entire files or precisely selected records. You can also ^GOTO^ and ^SKIP^ around within a database quickly and easily.

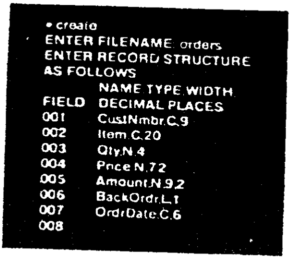
Additionally, dBASE II can be used interactively as a powerful calculator (and more) with the ^?^ command.

We have introduced you to expressions and how they can be used to expand the power of dBASE II commands. In the next section, we will go into this in more detail and show you how to get useful information out of your databases quickly and easily.

Before that, please ^CREATE^ these two files, as we will need them for other examples.



```
• create
ENTER FILENAME MoneyOu
ENTER RECORD STRUCTURE
AS FOLLOWS
          NAME,TYPE,WIDTH
FIELD     DECIMAL PLACES
001       Check Date,C,7
002       Check Nmbr,C,5
003       Client,C,3
004       JobNumber,N,3
005       Name,C,20
006       Descrip,C,20
007       Amount,N,9,2
008       Bill Date,C,7
009       Bill Nmbr,C,7
010       Hours,N,6,2
011       Emp.Nmbr,N,3
012
```



```
• create
ENTER FILENAME: orders
ENTER RECORD STRUCTURE
AS FOLLOWS
          NAME,TYPE,WIDTH
FIELD     DECIMAL PLACES
001       CustNmbr,C,9
002       Item,C,20
003       Qty,N,4
004       Price,N,7,2
005       Amount,N,9,2
006       BackOrdr,L,1
007       OrdrDate,C,6
008
```