1884

dBASE II®

A Comprehensive User's Manual

KERMAN D. BHARUCHA

```
FIELD 01   :EMP:NUM     C  006  000 :
FIELD 02   :EMP:NAME    C  015  000 :
FIELD 03   :TOWN        C  012  000 :
FIELD 04   :ORG         C  003  000 :
FIELD 05   :YR:OF:HIRE  N  004  000 :
FIELD 06   :SALARY      N  008  002 :
FIELD 07   :DUMMY       C  001  000 :
FIELD 08   :                        :
FIELD 09   :                        :
FIELD 10   :                        :
OK
```

dBASE II®

A Comprehensive User's Manual

KERMAN D. BHARUCHA

TAB BOOKS Inc.
Blue Ridge Summit, PA 17214

FIRST EDITION

FIRST PRINTING

Printed in the United States of America

Library of Congress Cataloging in Publication Data

Bharucha, Kerman D.
dBase II—a comprehensive user's manual.

Includes index.
1. dBase II (Computer program) 2. Business—Data
processing. I. Title. II. Title: dBASE two—a
comprehensive user's manual. III. Title: dBASE 2—a
comprehensive user's manual.
HF5548.4.D22B43 1985 001.64′25 84-23959
ISBN 0-8306-0884-2
ISBN 0-8306-1884-8 (pbk.)

Cover photograph by the Ziegler Photography Studio of Waynesboro, PA.

dBASE II®

A Comprehensive User's Manual

Introduction

dBASE-II is the name of a software package marketed by Ashton Tate, Inc., and is a powerful development tool for microcomputer business applications. If you utilize a microcomputer for personal or business needs but have never had the pleasure of working with dBASE-II, or if attempts at working with dBASE-II through the technical manual have proven frustrating, then the solution to both situations is contained in this book.

"dBASE-II—A Comprehensive User's Manual" discusses version 2.41, dated February 1, 1984, and has been written for the person who wants to get started *overnight* on serious development of business applications without going through the hassle and frustration of reading the technical manual on dBASE-II. You begin with the study of some fundamental computer concepts, including a general discussion on data-bases, before beginning your study of dBASE-II. Within the dBASE environment, you will learn how to create a data-base, edit/modify the data-base so as to guarantee the integrity of both data and structure, sort/index the data to rearrange it for reporting, pull off reports from the data, and write your own computer programs in dBASE to increase your power and flexibility with dBASE-II.

Obviously, no attempt has been made at any time to replace the manual. The full use of this book will take you a very long way towards effectively utilizing this powerful software, but the most complete repertory of commands is still the dBASE-II technical manual. The study of this book will guarantee your ability to understand, without further external assistance, any specific commands from the manual not explicitly covered here.

As each command is discussed, several "What-If" types of possibilities are explored for that command, so that you are not left in any doubt about multiple choices of action available, or their outcomes. Specific instructions are covered in one area of the book so you will not have to search through the whole book for answers. The index will direct you to the explanation for the various commands.

Literally zero data processing knowledge is presumed, and the approach taken has been that of guiding a novice through the paces of running and effectively utilizing dBASE-II. To that extent, the book has been formatted in three sections. Section 1 discusses some fundamental computer concepts and data-base concepts. Section 2 starts off with the study of dBASE-II Basic Features, and you will learn how to create, edit, modify, sort, index and report from any data-base. Section 3 is the Advanced Features section. In this section, you learn to write your own programs in dBASE-II, starting with the simplest of programs and working your way up to a high degree of programming sophistication.

The appendices contain the report generation program and all of the menu programs. They have been provided for your use without all the comments and explanations that were supplied in the text.

If you feel that you already know some dBASE-II, then follow through the table of contents down to the logical point you wish to pick up from. However, reading all the preliminary material will virtually guarantee even the experienced dBASE-II user of absorbing additional, useful information on this subject.

Contents

Introduction xi

PART 1 FUNDAMENTALS 1

Chapter 1 Data Processing Fundamentals 3

Chapter 2 Operating System Fundamentals 7

Chapter 3 dBASE-II Fundamentals 9

What Is a Data-Base?
Who Needs This Book?
Which Version Is Discussed?
Relational Data-Base
A Word on Documentation
Starting Up with dBASE-II
Installation—Making a Backup Copy—Loading—Date Check—Dot Prompt—Help Feature
General Explanations and Conventions
Syntax—<CR> Symbol—Enter or Return—Control Keys—Specific Computer—Disk-Drive Names—Files and Data-Bases

PART 2 BASIC FEATURES 19

Chapter 4 Creation Process **21**

Creating a Data-Base Structure

Directory Listings—Software and Data Handling—Setting the Fields—Observations on CREATE—Displaying and Documenting the Structure—Logical Fields—Modifying Structures—Cursor Controls—The Extra Field

APPEND Command

Append Common Information—Observations on APPEND

DISPLAY Command

Moving the Record Pointer—The Scope Parameter—The Field List/Expression-List Parameter—The FOR Condition—Effect of Sequencing—The FOR Condition, Continued—Using Logical Fields with DISPLAY—Complex/Multiple Conditions—The Off Parameter

LIST Command

Special Functions

Substring Function—String Function—Value Function—Date () Function—Uppercase Function—Record Number Function—Type Function—The + Function—The – Function—Trim Function

LOCATE Command

Simultaneous Display of Data from Two Data-Bases

Chapter 5 Editing Process **65**

EDIT Command

BROWSE Command

Scrolling—Fields Subparameter

DELETE Command

List Deleted Records—List Active Records

RECALL Command

PACK Command

INSERT Command

APPEND Command

From<file> Parameter—For Condition—SDF Parameter—Delimited Parameter—Appending a Blank Record—Data Movement Outcomes

COPY Command

To <file> Parameter—Next Parameter—Field Parameter—The For Condition—SDF Parameter—Structure Parameter—Delimited [with <delimiter>] Parameter

REPLACE Command

Save Deleted Records in Another Data-Base

Using the Dummy Field—Without the Use of the Dummy Field

MODIFY Command

Adding/Deleting Fields in the Structure or Changing Field Lengths—Changing Field-Names in the Structure—Changing Field-Names and Field Lengths in the Structure—Recovering Deleted Records after Structure Modification—Deleting All Records Instantly

JOIN Command

Default Output File Structure—Issuing the JOIN Command—Specifying Output File Structure

Summary

Chapter 6 Sequencing Process **109**

Physical Sorting

Sort on Any Character Field—Sort Character Field in Descending Sequence—Sort on Any Numeric Field—Sort Numeric Field in Descending Sequence—Sort on More Than One Field—Sort on Three Fields—Sort on a Combination Ascending/Descending—Conserving Space—Disadvantages of Sorting

Logical Indexing

Indexing on a Character Field—Indexing on a Numeric Field—Indexing without Regard to Upper/Lowercase—Removing Index File Influence—Restoring Index File Influence—Indexing on a Substring—Indexing on Multiple Fields—Indexing on Multiple Substrings—Hierarchical Indexing on a Combination of Field-Types—Indexing on a Sum—Advantage of Indexing—General Approach to Indexing—Indexing Numeric Fields in Descending Sequence—Hierarchical Indexing with Descending Sequence—Combination of a Descending Sum and Hierarchy—Status on Active Indexes—Caution on Duplicate Keys—Disadvantage to Indexing—Mismatching Files and Indexes

Combination of Sort and Index

Sorting vs Indexing

FIND Command
Record Out of Range Problem
Summary

Chapter 7 Reporting Process 139

Building the Report Format File
Report-Format File—Creating Format Files—Observations on Report Formats—Dolling Up Column Headings—Dolling Up the Main Heading—Multiple Main Heading Lines—Displaying Record Numbers—Obtaining a Good Spread—Expressions in Report Formats—132 Column Spread—Vertical Formats—Obtaining a Ratio—Simultaneous Reporting from Two Data-Bases—Taking Totals—Record Counts of Selected Records—Expressions in a Report Column—Totals in Vertical Formats—Sub-totals in Reports—Grouping Records—Sub-totals at More Than One Level of Control
The While Parameter for Report-Formats
Making Changes to Existing Report Format Files
Summary

PART 3 ADVANCED FEATURES 173

Chapter 8 Programming in dBASE-II 175

Overview of dBASE Programming
A Sequential Command File—General Comments on dBASE Programs—A Decision Command File—A Repetition Command File—Observations
A Practical Program
Extending the Logic with More Variables—Accepting Numeric Input—Providing a Record Cosnt—
Providing Averages
Called Programs
Nested F Statements
The Case Approach

Chapter 9 Writing Your Own Report Program 195

Report Program Explanation
Accepts Program—Header Program
Positioning and the Use of Masks
Changes to Existing Report Format Files

Chapter 10 Writing a Menu Driven System 201

Menu Shell
General Approach—Checking the Results of Operator Action—Text Approach
Program A
Program B
Program C
Generating Screen Formats
Program D
Advantages of Using a Format File—Edit Check Enhancement
Understanding Macros
Using Macros in the Menu System—Program A Revisited
Program E
Editing Existing Data
Program F
An Important Caution When Editing Data in an Indexed File
Program G
Program H
Program I
Program J
Program K
Program L
Program M
Program N

Program O
Concluding Observations
Summary

Chapter 11 Additional Tips and Techniques 249
Combining Multiple Files
Enhanced Screen Layouts
Table Lookup
The Update Command

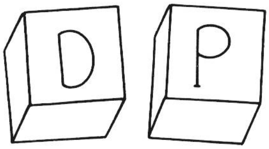
Chapter 12 Shattering the Myth of the 32-Field Structure Limitation 259
Creating
Sequencing
Reporting
Editing
Summary

Appendix A The Report Program 271

Appendix B The Menu Programs 275

Index 303

Part 1
Fundamentals



1. Data Processing Fundamentals

In this chapter, I will introduce some definitions, explain some procedures, and answer some questions. If you are generally familiar with microcomputers and if you already use dBASE-II, you may prefer to skip all preliminary information and scan the table of contents for specific topics of interest.

Data Processing. *Data processing* is the systematic *collecting, analyzing, summarizing* and *reporting* of data. The function of data collection is, by far, the costliest of these four functions, and the most time-consuming, since it requires substantial human input into the process. Subsequent functions, of course, are purely mechanized by way of the computer.

Computer Program. A *computer program* is nothing more than a series of instructions to a computer. These instructions are, for the most part, *sequential* in nature. If you can manage to write a few instructions, in any computer language, and if you can manage to store these instructions in a module on an external device (say, a floppy diskette), then you have created a computer program. Computer programs can range from simple, to supersophisticated, depending on the output requirements of the program.

Hardware. The term *hardware* refers to the physical computer itself. Anything you can see and touch, the electronics of the machine, the various peripheral devices such as CRT (screen), keyboard, disk-drives, printer, and modem are all encompassed in the term hardware.

Software. *Software* refers to the computer program or programs that control a computer system at any point in time. The microcomputer usually runs only one program at a time, but in the case of main-frame computers, several hundred program modules could be executing simultaneously.

CPU. The basis of any computer system is the piece of hardware known as the *central processing unit* (CPU). This piece of hardware contains all the electronic circuitry required to perform the functions of *arithmetic, logic,* and *control.* It is the brains of the microcomputer system.

At any point in time, the CPU performs one of the following three functions:

1. It *obtains* the *next sequential instruction.*
2. It *interprets* that instruction.
3. It *executes* that instruction.

The cycle repeats, beginning again at step 1, for each instruction.

Memory. Another piece of hardware in a computer system is that known as the *main memory structure* of the system. For now, it is useful to visualize memory as being comprised of individual cells, like post office boxes, with each box being capable of containing one character.

The term *character* refers to a digit (0 through 9), a letter (a through z, or A through Z) or a special character (such as the @ # $ % & *, etc). The CPU has direct access to any memory-cell at random, and hence the main-memory structure is known as *Random(ly) Access(ible) Memory* or *RAM.* Since the CPU goes through life obtaining, interpreting and executing instructions, it stands to reason that the computer system must have a way of providing these instructions to the CPU. This is the function of main memory. In the case of a microcomputer system, the entire program that is currently executing needs to be in main memory, so that the program instructions can be accessed by the CPU. Also, main memory contains that portion of the data that the program is currently working on.

Peripherals. Surrounding the CPU and RAM are the external devices for permitting information to flow into and out of the computer system. The *keyboard* is the basic input device for placing information directly into a reserved area of memory. The *screen* is the basic output device for visual display of a reserved area of memory. The *disk drives* are used for permitting access to floppy diskettes on which information can be magnetically stored, or from which information can be retrieved. The *printer,* of course, is the basic output device for hard-copy displays.

Characters. If you want to store your name in a computer system, you will have to provide the system with the letters comprising your name. These are the individual *characters* of your name.

Suppose you also want to store your address in the computer system. You will have to supply to the system all the individual characters (letters and numbers) comprising your full address. So also, for your employee number, or organization, or salary, or any other *piece of information* you want to maintain for yourself.

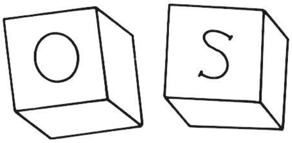
Field. Each piece of information so created through the use of characters is called a *field* of information. Thus, you may have created your name-field, your organization-field, and your salary-field with the use of the appropriate characters.

Record. Now, if you have stored all the information you may want about yourself, in the form of fields of information, you have managed to create one *record* of information, about yourself.

File. As you may have guessed, if you can do the same for some of your colleagues working in the same department, you will have managed to create a *file* of information comprising several records of information.

These last four definitions can be summarized as follows:

A *file* is made up of individual records of information.
A *record* is made up of individual fields of information.
A *field* is made up of individual characters of information.
A *character* is any number, letter, or special character.



2. Operating System Fundamentals

An operating system is a collection of program modules that provide, among other things, an interface between an executing program's logical requests for input/output operations, and the physical aspects of carrying out the input or output request.

For example, your program has started execution, and at some point in time, it issues a logical read type of request. The operating system's module takes over, since it has to figure out where on the face of the disk the next block of data is located. Having obtained that information, it has to complete the task of loading that block into computer memory and then hand control back to your program.

Your program continues churning away, until it provides a request to write out to disk. Once again, the operating system takes over. It has to figure out where on the face of the disk there is space for the next block of information to be stored, and having figured that out, it has to complete the output operation and write the block out from memory to disk, and hand control back to your program.

Each time your program requests any kind of input or output operation to be performed, the modules of the operating system take over, and perform that task. This is how the operating system *interfaces* between your program's logical input/output requests and the physical aspects of carrying out the input or output request. Apart from providing this area of commonality for input/output for all programs, the operating system also handles disk-file maintenance and access procedures, and the physical loading and execution of programs. This, of course, is a very simplified explanation of some of the workings of an operating system, and depending on the computer system you are using, the operating system for that computer could be anything from basic to super complex.