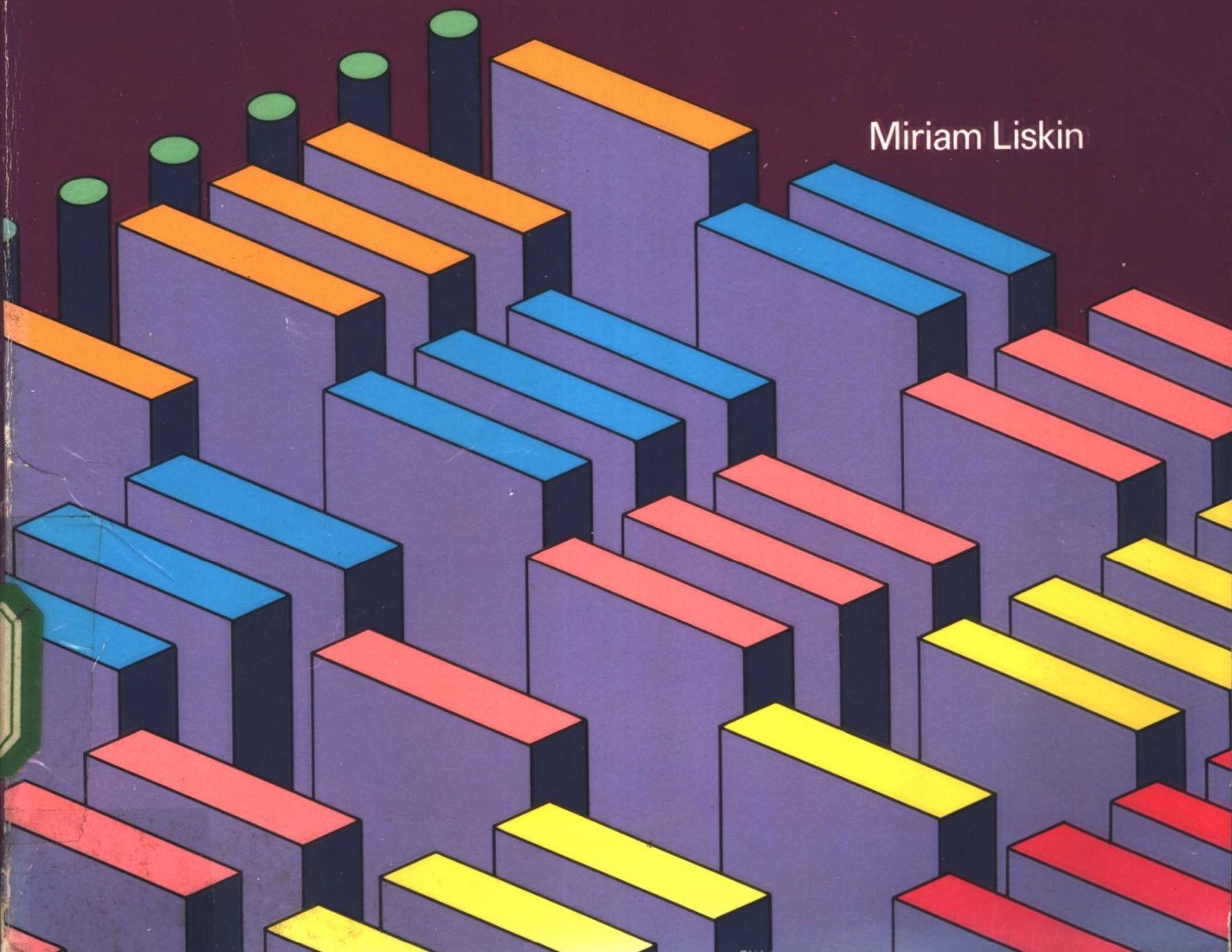


OsborneMcGraw-Hill

A D V A N C E D dBASE® III

Programming and Techniques


Miriam Liskin



ADVANCED dBASE® III: PROGRAMMING & TECHNIQUES

Miriam Liskin

Osborne McGraw-Hill
Berkeley, California



Osborne McGraw-Hill
2600 Tenth Street
Berkeley, California 94710
U.S.A.

For information on translations and book distributors outside of the U.S.A., please write to **Osborne McGraw-Hill** at the above address.

A complete list of trademarks appears on page 646.

The information contained in Appendixes E and F is Copyright © Ashton-Tate 1985. All rights reserved. dBASE III and Ashton-Tate are registered trademarks of Ashton-Tate. Used by permission.

ASHTON-TATE
20101 Hamilton Avenue
Torrance, CA 90502
(213) 329-8000

ADVANCED dBASE® III: PROGRAMMING & TECHNIQUES

Copyright © 1986 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1975, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

234567890 DODO 89876

ISBN 0-07-881196-1

Cynthia Hudson, Acquisitions Editor
Denise Penrose, Technical Editor
Kevin Shafer, Senior Editor
Lynn Heimbucher, Editorial Assistant
Fran Haselsteiner, Project Editor
Nancy Leahong, Text Design
Yashi Okita, Cover Design

ACKNOWLEDGMENTS

I would like to thank some of the people who contributed directly or indirectly to the publication of this book by providing exactly the kind of help I needed at some of the crucial turning points in my life:

My parents, Philip Liskin and Florence Mindell, my first and best teachers, who led me to believe early in life that I could be and do anything I set my mind to.

The "Kirsch Lab Encounter Group," for all the good times. And thanks, Becky, for suggesting that there might be a computer career in my future.

Cliff Krouse, who helped me get my foot in the door by giving me my first job in the microcomputer industry, "taught me everything he knew," and introduced me to the joys of data base management.

Stan Politi, editor and publisher of *Bay Area Computer Currents*, who launched my writing career. And thanks to the entire staff of the best little computer newspaper around—Lynne Verbeek, Anne Skjelbred, Metece Riccio, Pam Ivie, Eve Levine, Bill Bevis, Kris Warrenburg, Carol Warner, and Ken Margolis—for a congenial place to work, outstanding gourmet lunches, remembering my birthday, and for inspiring, appreciating, and debugging all those dBASE programs.

Linda Allen, literary agent, who had faith and worked hard for me back in the days when there was no objective evidence that anything would ever come of it.

Denise Penrose, who, in the process of editing the manuscript, taught me a great deal about writing a book and kept me on track with marginal notes like "How did I get this far and suddenly feel so lost?"

My coworkers, students, and clients of the past five years, who consistently challenged me to come up with better programming solutions and clearer explanations, and for whom I should have written this book a long time ago.

INTRODUCTION

This is a comprehensive guide to the design and implementation of a dBASE III data base application. dBASE III may be used in two distinctly different ways: by typing commands at the dot prompt (or, to a more limited extent, through the built-in ASSIST menus) or by writing programs in the dBASE III language. This book will help you make the most of both modes of operation. It addresses the needs of four categories of dBASE III users:

- People who want to learn to utilize more of the power and sophistication of the built-in dBASE III commands used at the dot prompt.
- People who are comfortable working at the dot prompt and want to learn the dBASE III language as a first programming language.
- People who have written programs in other higher-level languages (or dBASE III) but have little or no experience with system design.
- People who have designed systems and written programs and who want to learn to best utilize dBASE III's built-in commands and the strong points of its programming language.

No previous programming experience, either with dBASE III or any other language, is necessary. This book does assume a basic familiarity with using dBASE III at command level (at the dot prompt) to work with simple filing systems involving one data base, although many of the more difficult concepts and advanced commands will be reviewed. To use this book effectively, you should already understand the following:

- How to create a data base file with character, numeric, logical, date, and memo fields.
- How to modify the structure of an existing data base.
- How to add, view, edit, and delete individual records and groups of records.
- How to use the full-screen cursor movement and editing commands.
- How to build an index for a data base.
- How to use indexes to control the order in which records are displayed or printed and to rapidly retrieve individual records.
- How to design and print columnar reports with subtotals.
- How to design and print labels.
- How to specify selection criteria for records to be displayed on the screen or printed on reports or labels.

Unlike many textbooks on programming, this book is not structured around specific commands, programming structures, or algorithms. Rather, its organization follows the process of designing, implementing, testing, and refining a typical business application. The dBASE III language is used primarily as an application development tool. Most people buy dBASE III because they need software that is highly customized to the requirements of a particular type of business or perhaps to the unique processing needs of one organization. In some cases, no comparable packaged software is available; in others, it is very expensive, inflexible, or both.

Many smaller data base applications are handled adequately by using dBASE III's built-in commands at the dot prompt. For more complex needs or less knowledgeable operators, you can use the dBASE III programming language to create a specialized turn-key system. Because the dBASE III language can take advantage of all the same file management capabilities accessible from the dot prompt, the development process generally takes far less time than writing the same programs in another higher-level language like Pascal or COBOL. Yet learning to analyze a manual system and design a computer-based replacement is much more difficult than simply learning the syntax of a new programming language, not only for novices, but also for more experienced programmers who have never had sole responsibility for the design of a complete system.

In many ways learning to use dBASE III to its fullest capacity parallels the development of an application: from working at command level, you will progress to writing simple batch-type programs made up of exactly the same commands and gradually to writing longer and more complex programs. For readers who have no experience with other higher-level languages, this approach eases the transition to learning dBASE III programming. If you have programmed in

other languages, you will recognize the basic programming structures and concepts in the dBASE III language. This book presents the best ways to make use of the unique features offered by dBASE III: the built-in file management capabilities that enable you to concentrate on the functional requirements of an application, freeing you from the burden of trying to write a faster sort or index search routine or the tedium of writing procedures to dissect a data base record into its component fields.

This book uses as a case study a sample accounts receivable system for an imaginary company called National Widgets. You may want to build this system on your own computer as you read. If you are currently working on a dBASE III system, you may prefer to parallel the development process with your own application. You will find that you are the most relaxed and receptive to new ideas if you have a specific project to which you can relate the concepts in this book but are not under pressure to complete the project to meet a deadline.

The best way to learn any programming language is by experimentation. You will get the most out of this book if you take an active, exploratory approach to the learning process. Whenever you find yourself wondering "Why did she do it this way?" or "What would happen if I tried something slightly different?" go ahead and try it. There is almost always more than one correct way to accomplish any goal. If, in certain contexts, one method has strong advantages, they will be pointed out, but you should feel free to vary the techniques presented or substitute others that seem more appropriate to your situation.

The programs in this book attain a level of sophistication applicable to real-world data base problems. As the development of the National Widgets system progresses, it may seem that the programs are being tailored increasingly to the requirements of this imaginary business. Try not to lose sight of the basic principles illustrated by the specific examples—in particular, the way the data base files are related to one another, the programming strategies, and the way the programs interact with the user. For example, Chapter 18 presents a program to print the National Widgets Customer Reference List complete with financial transactions. The same program structure could also be used to print a donor list with gift transactions for a nonprofit organization, a patient list with a history of all office visits for a doctor's office, or an inventory report listing all stocked items with shipping and receiving transactions. The National Widgets programs are presented as models that may be adapted and expanded to suit the unique requirements of your own data base systems.

A disk containing the programs and sample data bases in this book is available for \$10 by writing to

Miriam Liskin
2000 Center Street #1229
Berkeley, CA 94704

I

PLANNING THE SYSTEM

Part I covers the design and prototyping phase of application development. After introducing the imaginary company, National Widgets, Inc., which is used as a detailed case study throughout the book, this part will present a systematic method (which will be applied to this company) for analyzing data entry, storage, and reporting requirements and setting up the necessary data base structures and indexes. A general strategy will be outlined and specific methods described for entering realistic test data and working out many command level procedures.

If you have no prior experience with system design, you should study carefully the way that the users' description of their needs is translated into the file structures and command level procedures. Readers who understand the principles of systems analysis but who have not used dBASE III extensively should focus on the way the files are structured and related to each other and on the process of modeling the application at command level.

If your applications are simple, Part I will help you use dBASE III at the dot prompt to its fullest capacity and push the built-in report and label generators to their limits. But this command level testing phase is an excellent way to begin the development of any dBASE III system, even when you are sure at the outset that additional programming will be required. Working at the dot prompt is a fast and efficient way to clarify the relationships among the data files, ensure that the file structures are correct and complete, and define the weekly, monthly, and annual processing cycles.

CONTENTS

Part I	Planning the System	
Chapter 1	Defining Your Needs	1
Chapter 2	Establishing the Data Files	19
Chapter 3	Testing With Sample Data	35
Chapter 4	Developing Command Level Procedures	43
Chapter 5	Printing Labels and Reports at Command Level	71
Chapter 6	Evaluating the New System	85
Part II	Automating the System	
Chapter 7	Writing Simple Programs to Automate Procedures	99
Chapter 8	Creating Custom Data Entry Screens	121
Chapter 9	Programming Structures	143
Chapter 10	Programming Strategy	165
Chapter 11	Using Memory Variables	177
Chapter 12	A Menu Program for the System	195
Chapter 13	Evaluating the System	221
Part III	Refining the System	
Chapter 14	Testing and Debugging	235
Chapter 15	A More Efficient and Flexible System	255
Chapter 16	Accessing Multiple Files	273
Chapter 17	Programs to Print Reports	303
Chapter 18	Printing Reports From Multiple Files	347

Part IV	Optimizing the System	
Chapter 19	Data Base Update and Inquiry Programs	383
Chapter 20	Data Validation	409
Chapter 21	Giving the User More Control	437
Chapter 22	Using Procedure Files	471
Chapter 23	Providing On-Line Help and Access to DOS	507
Chapter 24	Documenting the System	537
Appendix A	Command Syntax Notation	561
Appendix B	dBASE III Command Syntax Summary	565
Appendix C	dBASE III Functions	599
Appendix D	The CONFIG.DB File	611
Appendix E	dBASE III Anomalies	615
Appendix F	Error Messages	631
	Index	647

DEFINING YOUR NEEDS

The first step in setting up a dBASE III application is to define what the system must accomplish. If the computer will replace an existing “paper” system, a detailed description of the manual procedures, accompanied by a set of samples of the various paper forms and reports, is the logical starting point for outlining the functions of the new software. The users will almost always want to add new capabilities not included in the current manual system; this “wish list” will also become an important part of the system specifications. In other cases, no manual system exists, and the dBASE III application must assume responsibility for a new set of information management tasks. This situation may arise when a business branches out in a new direction, an organization decides to begin in-house processing of information formerly maintained by an outside service bureau, or the managers hope to use the computer to produce analytical reports that were too difficult or time-consuming to produce by hand.

To define your needs, specify in as much detail as possible:

- *The input to the system* — the information that will be entered into the computer.
- *The system’s processing functions* — the calculations the computer must carry out and the ways information will be transferred among the data base files.
- *The output of the system* — the ways in which the data will be displayed on the screen and printed on paper.

This chapter will outline a systematic method for gathering the information you need to design a dBASE III data base system. If you have already designed and implemented applications in another higher-level language, many of the

concepts presented here will already be familiar to you. If you have worked with dBASE III at *command level* (using dBASE III built-in commands to perform functions only—not to program), you will have some understanding of how the built-in commands will allow you to accomplish the three fundamental processes just summarized. If you have never written a program before, or if you have no formal training in systems analysis, you should study carefully the methods described in this chapter, but don't feel that you are starting off at a disadvantage. You will find that common sense and the ability to think logically are greater assets during the planning stages than prior programming experience.

This chapter introduces the imaginary company that we will follow as a case study throughout this book. Readers who are new to programming and system design may want to follow this example closely as a tutorial. If you are already involved in designing a dBASE III application for your organization, you might instead choose to apply the general principles observed in the case study to your own application.

STUDY THE EXISTING MANUAL SYSTEM

Before you can begin building a dBASE III application, you must make a thorough study of the manual system that the programs will augment or replace. If no manual system exists, you will have no choice but to proceed directly to a description of the proposed new software. You should recognize, however, that without the concrete examples provided by a paper system, you will need to spend more time and energy to formulate the overall goals for the system and then to define the particulars.

In the course of your investigation, you will have to interview everyone in the organization who is or will be involved in the normal cycle of processing. Each person will have a set of personal aims and priorities, as well as a unique perspective on the flow of information within the organization, and each one can provide valuable insight into how the new system might improve on existing procedures.

If you, the prospective programmer, are affiliated with the company, you will have the advantage of familiarity, but you must also learn to take a step back and adopt the vantage point of an outsider. Depending on your position in the organizational hierarchy, you may be accustomed to dealing only with the big picture and long-range plan, or you may be more intimately familiar with the day-to-day procedures. Although you will have to ask fewer questions than an outside consultant, it may seem awkward to be interrogating your co-workers. An in-house programmer may also have more difficulty spotting those instances

in which the transition to a computerized system would be made easier by altering the work flow or redefining job responsibilities.

An independent consultant will naturally bring a more objective perspective to the project and will be less likely to regard existing procedures as sacrosanct, but an outsider must spend more time becoming familiar with the organization, its structure and personnel, and the specialized vocabulary of the business or profession. You may also have to work to gain the confidence of the staff. Any feelings of anxiety or hostility surrounding the acquisition of the computer will to a certain degree extend to you, the programmer. In this atmosphere, it can be difficult to foster the spirit of active cooperation and participation that will help you to elicit the information you need.

Describe the Work Flow And Schedule

During the first phase of the system development, you must describe in detail the present and projected work flow and processing schedule. This analysis need not—in fact, should not—be couched in technical jargon. Whether or not you are an experienced programmer, at this stage you should think solely in terms of how information is entered, updated, transformed, and printed in the course of a normal processing cycle. You will also find that conducting your preliminary discussions in ordinary conversational English will help you to communicate clearly with users and clarify your own understanding of the existing procedures.

For each operation or task in the manual system, your conversations with the staff must provide the answers to these crucial questions:

- What specific items of information are involved?
- How big is each item?
- How many of each item are there?
- What mathematical calculations are carried out?
- What transfers of information are involved?
- How often is the operation performed?
- How long does it take to carry out each task?

These general queries may take the form of specific questions, such as:

- What information do you keep on your customers besides name and address?
- What is the highest price you charge for any item?
- How many customers do you have now?

- How many orders per month do you anticipate two years from now?
- How is the sales tax calculated?
- How are the order totals posted to the customer ledger cards?
- How often do you type invoices?
- How much time does it take to type a set of mailing labels?

The Work Flow at National Widgets We will use as a case study a dBASE III system which keeps track of customer information and accounts receivable data for an imaginary company called National Widgets, Inc. This firm, located in Berkeley, California, is a mail-order supplier of accessory products for microcomputers. National Widgets has been in business for three years and had about \$600,000 in gross sales last year. The company is managed by the owner and employs four other people full-time: a bookkeeper, two clerk-typists who also answer the telephones, and one person who handles shipping and receiving.

National Widgets has a microcomputer which is currently used for word processing, spreadsheet analysis, and managing prospect mailing lists with dBASE III. In the past year the number of customers and the volume of orders has grown to the point that the manual accounts receivable and order processing system is becoming unwieldy, and the owner now hopes to begin using the computer for accounting and inventory management as well. Because the owner could not find an inexpensive, off-the-shelf accounting package that could produce the desired inventory reports and integrate mailing-list management functions with the accounts receivable system, the company hired a consultant to customize a system with dBASE III. Accounts receivable was chosen as the first application to be implemented for two reasons. First, the staff is already comfortable using dBASE III to work with name-and-address lists, so the customer data base will seem relatively familiar and unintimidating. Second, the owner and bookkeeper have both placed a high priority on improving the company's cash flow by exercising a tighter control over the accounts receivable.

The consultant spent a full day with the employees of National Widgets, interviewing them, watching them at work, and collecting a set of samples from the manual system. These observations yielded the following description of the present operation of the manual accounts receivable system.

National Widgets has about 350 regular customers who account for about one third of its monthly volume of 500 orders. At the current rate of growth, both the number of customers and the number of orders are expected to double in the next two years. The bookkeeper maintains the customer records on 8 1/2-by-11-inch *ledger cards*, which are stored in alphabetical order in a vertical file. Because many first-time customers never place another order, a ledger card is

prepared for a new customer only if the company representative who calls mentions that the company has ordered from National Widgets in the past or asks for a credit application.

Almost all of the orders from regular customers are placed by telephone and are transcribed onto *standard order forms* by the clerk who answers the phone. A typical order contains from three to eight different items, totaling about \$100. Of the orders from first-time or occasional customers, more than half come in by phone and are shipped C.O.D. The remainder are sent in with a check attached to an order form clipped from one of National Widgets' ads in local newspapers and magazines. Some of these people never order again, while others eventually become regular customers. In the current manual system, there is no easy way for the clerk to determine whether someone has previously placed an order, unless the customer specifically mentions it, asks for credit terms, or is already familiar to the clerk.

Every afternoon, the day's orders are typed on *three-part invoices* by the bookkeeper, who looks up the prices in the current price list and calculates any applicable volume discounts. For prepaid orders the invoice is marked PAID, and the customer's check is paperclipped to the office copy of the invoice. Currently, there is no formal procedure for processing back orders. If an item is temporarily out of stock, the entire order is held for a period of time, usually no more than a week. If the item still has not come in, the original invoice is amended by hand and the remainder of the order is shipped to the customer, who is contacted by telephone to see whether or not a new order should be written for the missing item. The bookkeeper calculates the shipping charges and, for customers in California, the sales tax (at 6 1/2 percent for customers in Bay Area counties, or 6 percent for the rest of the state). Payment terms are net 30 days for all regular customers. The bookkeeper sends one copy of the invoice back to the warehouse, where the clerk uses it to pull the items ordered and then sends it along with the shipment as a packing slip.

The bookkeeper updates the *customer ledger cards* at least once a week to reflect the latest batch of invoices and payments. Ledger cards are made up for new customers and any changes in address, staff contact, or telephone number for regular customers are noted. In the last six months, the bookkeeper has noticed a number of ledger cards for customers who have had no activity for over a year, but no formal guidelines govern when an inactive customer should be removed from the file.

After the ledger cards are updated, the invoice totals are transcribed onto a worksheet (formerly on paper, but currently maintained using a spreadsheet program) in the form of a *monthly invoice register*, with columns for invoice number, date, subtotal, discount, sales tax, shipping charges, and net invoice amount. The checks are separated from the invoices and the amounts entered

onto *bank deposit slips*. Finally, a clerk mails the second copy of the invoice to the customer and files the third copy in a filing cabinet in numerical sequence.

At the end of the month, the bookkeeper goes through the ledger cards and types *monthly statements* for all customers who have one or more past-due invoices (usually about 75 to 100 customers). The monthly statement includes all outstanding invoices and any payments which have been applied to these invoices, as well as all activity for the current month.

National Widgets sends two *mailings* per year to all of its regular customers: a new price list in the spring and, in the fall, an announcement of Christmas specials and year-end sale items. For these mailings a clerk types the customers' addresses on pressure-sensitive mailing labels. These are transferred to folded and stapled flyers, which are sorted by hand into ZIP code order. Early in December, Christmas cards are sent to all regular customers. For this mailing, which is not sorted by ZIP, the addresses are read from the ledger cards and typed directly onto envelopes.

At various times during the year, selective mailings (typically consisting of new product announcements or special sale notices) are sent to specific groups of customers based on the type of computer equipment they own or the products they order frequently. Since the customer ledger cards do not contain any record of what types of products a customer orders, the clerk who types the labels for these mailings is guided by her personal recollections of each customer. If no one remembers a particular customer, the clerk must pull some invoices from the file to decide whether or not the company should be included in a mailing.

Every three months the bookkeeper prepares the *quarterly sales tax report* from the monthly invoice registers, and at the end of the year, she compiles an *accounting summary* to be used by the accountant to prepare the annual financial statements. Total sales, discounts, tax, and payments are taken from the invoice registers, total receivables are added up from the customer ledger cards, and cost of goods is calculated from the inventory control records.

Study Samples From the Manual System

As you question each staff member about his or her job, collect a set of the documents that comprise the manual system. A study of these paper forms will enable you to extract a detailed list of all the items of information that must go into the data bases if the dBASE III system is to duplicate the existing procedures. The samples you collect should contain matching data—for example, an invoice, a statement for the same customer, and a copy of the ledger card as it looked at the time the statement was typed—so that you can trace the flow of information through the system.

The Samples From National Widgets The set of samples from the National Widgets manual system is relatively small: a customer ledger card (illustrated in Figure 1-1), an order form, a price list, an invoice (Figure 1-2), a statement, mailing labels, a monthly invoice register (Figure 1-3), a bank deposit slip, a quarterly sales tax report, and an annual accounting summary. The elements of the manual system and their functions are summarized in Figure 1-4.

DESCRIBE NEW OPERATIONS AND REPORTS

Once you have a clear picture of the existing systems and procedures, the next step is to obtain a list of all of the desired new functions for the system in order of priority, including information not currently being tracked, additional calculations, and, most importantly, new reports and printed forms. In practice, it is often difficult to separate this step from the study of the existing system. Especially if the staff members express a high level of optimism and enthusiasm for the computer, their descriptions of their current job responsibilities will often be interspersed with items from their own personal "wish lists" for the new data base system. In the early planning stages, encouraging the users to mention anything that comes to mind will broaden your perspective, although low-priority operations which are difficult to program or slow to execute may be dropped from consideration later.

As you scrutinize the manual system to determine the details of how the company does its work, remember the reasons for computerizing these functions

Lewis and Associates 408 Grand Avenue Oakland, CA 94610		889-5014	Accts. Payable: Joan Mills	
Date	Invoices	Payments	Balance	
12/1/83 Invoice # 2158	102.15		102.15	
1/10/84 Check #321		102.15	0.00	
4/12/84 Invoice 4012	79.62		79.62	
5/13/84 Invoice 4468	65.20		139.82	
5/18 Check #408		79.62	65.20	
6/18 Check #441		65.20	0.00	
9/20 Invoice 6736	129.89		129.89	
11/1 Check #563		129.89	0.00	
2/10/85 Invoice 7958	138.50		138.50	

Figure 1-1. A customer ledger card