IRWIN
ADVANTAGE
SERIES FOR
COMPUTER
EDUCATION

HUTCHINSON
SAWYER
COULTHARD

dBase III PLUS

dBASE III PLUS®

Sarah E. Hutchinson
Stacey C. Sawyer
Glen J. Coulthard

The Irwin Advantage Series
For Computer Education

IRWIN
Burr Ridge, Illinois
Boston, Massachusetts
Sydney, Australia

Printed in the United States of America.

ISBN 0-256-13515-0

dBASE III Plus is a registered trademark of Borland International.

8 9 0 ML 0 9 8 7 6 5

USING THIS GUIDE

This tutorial is one in a series of learning guides that lead you through the most popular microcomputer software programs available. Concepts, skills, and procedures are grouped into session topics and are presented in a logical and structured manner. Commands and procedures are introduced using hands-on examples, and you are encouraged to perform the steps along with the guide. Although you may turn directly to a later session, be aware that some sessions require, or at least assume, that you have completed the previous sessions. For maximum benefit, you should work through the short-answer and hands-on exercises appearing at the end of each session.

The exercises and examples in this guide use several standard conventions to indicate menu instructions, keystroke combinations, and command instructions.

KEYSTROKES AND KEYSTROKE COMBINATIONS

When you must press two keys together, the tutorial's instruction line shows the keys joined with a plus sign (+). For example, the following statement shows how to finish defining a report column heading.

PRESS: Ctrl+End

This instruction tells you to press the Ctrl key first and then hold it down while you press End. Once both keys have been pressed, they are then immediately released.

COMMAND INSTRUCTIONS

This guide indicates with a special typeface data that you are required to type in yourself. For example:

TYPE: `George Washington`

When you are required to enter unique information, such as the current date or your name, the instructions appear in *italics*. The following instruction directs you to type your name in place of the actual words: "your name."

TYPE: *your name*

Instructions that use general directions rather than a specific option or command name appear italicized in the regular typeface.

PRESS: Enter *to highlight the Heading option.*

CONTENTS

SESSION 2

RETRIEVING INFORMATION AND EDITING 35

SESSION 3

RECORDING DATABASE RECORDS 67

SESSION 1

dBASE III PLUS FUNDAMENTALS

Modern database management systems for microcomputers provide the business user with the means of managing and manipulating large amounts of data. They were introduced to the business community for use on microcomputers at about the same time as electronic spreadsheets. Received with great enthusiasm, both types of software are powerful and easy to use, and they are often used together. This session introduces you to the fundamentals of using the popular database management system (DBMS) package named dBASE III Plus.

PREVIEW

When you have completed this session, you will be able to:

Describe what a database management system is

•

Describe DBMS features used to:
Create and add data
Search a database
Reorder a database
Create and print reports

•

Use dBASE III Plus to create a database and add data

•

Save and use a database

SESSION OUTLINE

WHY IS THIS SESSION IMPORTANT?

Picture an office with a row of file cabinets that extends as far as you can see—and you're responsible for them! You use your filing system mainly to track customer-related information, and everything is perfectly organized in alphabetical order by last name. Not even one customer name is out of place! You know exactly where to look to find information on each customer. Great. But what if you need to pull out all folders that contain information on customers who live in a particular area? Your alphabetical organization scheme doesn't help. Your manual filing system has become a problem. You need a microcomputer database management system. A **database management system (DBMS)** is a software tool that makes it easier to create and maintain an information database you can use to produce reports. The term **database** describes a collection of data stored for a variety of business purposes.

Let's look at another example. You are a salesperson whose territory covers Indiana and Ohio. You have over a hundred clients to keep track of and are thinking of using a computerized database management system to keep track of the data. Once you have entered all the data pertaining to each client into your computer, you have a valuable tool at your fingertips that will cost only the time to keep the data current. You're making a trip through northern Ohio? In a few minutes, you can produce a report showing all the customers in that area prioritized by annual sales and the date of the most recent sales call. You can't remember why a client wasn't interested in your product during the last visit? In a few seconds, you can display on your screen any memo text that pertains to that client.

This session explains the features of a microcomputer DBMS and teaches you the fundamentals of using dBASE III Plus to create and add data to a database.

Before proceeding, make sure the following are true:

1. You have access to dBASE III Plus.

2. Your Advantage Diskette is inserted in the drive. You will save your work onto the diskette and retrieve the files that have been created for you. (Note: The Advantage Diskette can be made by copying all the files off your instructor's Master Advantage Diskette onto a formatted diskette.)

DBMS FEATURES

As with any software package, you must be familiar with the concepts and features of a DBMS before you can start to use it. This section introduces you to these concepts and features so that when you begin using dBASE III Plus later in this session, you will understand the whys and whats of DBMS processing.

It is important that you understand a few terms before creating a database and using database software. To define these terms we will use the analogy of a filing cabinet and its contents:

- *File*. Think of a database file as analogous to a filing cabinet. For example, in one filing cabinet you might store information about each of your employees in folders.

- *Record*. Think of a record as a folder in a filing cabinet. Each folder, or record, should contain the same type of data. For example, each folder for an employee would include name, address, phone, and salary data.

- *Field*. Think of a field as analogous to an individual piece of information in a folder, such as a name or address. When creating a database structure, you must define the characteristics of the fields that you want to include in each database record.

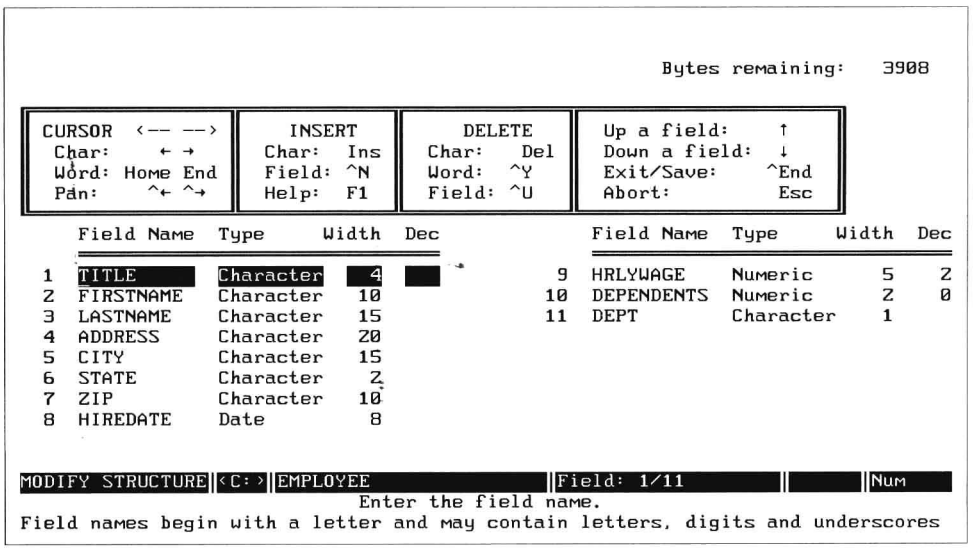
CREATING A DATABASE

After you have loaded dBASE III Plus, you can begin using its commands to create or manage a database. Creating a database is often referred to as creating the **database structure**—defining exactly what you want each field in a database record to look like. Think of it as defining what kind of information each folder should contain. (For example, in a client database, do you want to keep track of the client's name and business address? phone number? names of products purchased? values of products purchased? and so on.)

Figure 1.1 shows a database file structure that was defined using dBASE III Plus. This file structure was saved onto a data disk under the name of EMPLOYEE.

Figure 1.1

Database structure for the EMPLOYEE database



To create a structure using dBASE III Plus, the following items must be defined:

1. *Field name*. You must give a unique name to each field of data you want to store. A **field name** can be no longer than 10 characters. To manipulate the database, you will often be required to refer to one or more elements of data by their field names. Although field names must begin with a letter, you can use letters, numbers, and underscores when naming fields. Blank spaces aren't allowed in field names, nor any other punctuation marks or special characters.

2. *Field type*. DBMS programs require you to define what type of information will be stored in the field; in other words, you must define the **field type**. With dBASE III Plus, data can be one of five types: (a) **character**—data that is **nonnumeric**, such as name and address information; (b) **numeric**—data that will be used in calculations, such as dollar amounts; (c) **date**—data that must be entered in a specific date format so that calculations can be performed on it; (d) **logical**—data that indicates whether a field is true or false (such as data regarding a person's marital status, where true = married and false = single); and (e) **memo**—data in the form of a text paragraph.

3. *Field width*. You must determine what the maximum size of the field will be—its **field width**. For example, if you are defining the structure for a field that is to contain an employee's last name, what is the longest name you will probably have? Will it be 10, 20, or 30 characters long?

You must be sure that enough space is available for all the data you want to enter. You don't have to enter field widths for date (8 positions wide), logical (1 position wide), or memo fields (10 positions wide). The widths for these are set automatically by dBASE III Plus.

4. *Decimal places.* For each numeric field, you must determine the number of **decimal places** you want. For example, you may want to store numbers to two decimal places (such as $10.00) instead of no decimal places (such as $10). The number of decimal places that you assign to the width of a numeric field must be at least two less than the width of the field. The decimal place and decimal point take up space in a field even if the entry is less than one.

You must think carefully about these definitions at the start so you won't have to change your database structure after you've entered a number of records. It is possible to modify a database structure, but depending on the changes you make, you may lose data. (Modifying a database structure is described in more detail in Session 2.) Before the EMPLOYEE database file structure was created, for example, thought was given to having a separate field for TITLE, FIRSTNAME, and LASTNAME because (1) the database can now be sorted into alphabetical order by last name, and (2) the individual name fields can be referenced in a report. You might want to include only TITLE (Mr., Ms., Miss, or Mrs.) and LASTNAME fields in a specific report, for example. When designing the structure for a database, a good rule of thumb is to give every individual piece of data its own field. Otherwise it is sometimes difficult to search for or access the data you want to work with.

ADDING DATA TO A DATABASE

Once you have defined your database structure, you can add data to it. However, because you will often have more than one database file stored on the same disk, first you must tell dBASE III Plus to which file you want to add data. This procedure is similar to opening the drawer of the appropriate filing cabinet before adding another folder. We lead you through adding new records to a database file later in this session. Figure 1.2 pictures a listing of the EMPLOYEE database file (the structure for this database is pictured in Figure 1.1) after 12 records have been entered. The EMPLOYEE database contains 11 fields (the CITY field data is typed in after the ADDRESS field data).

Figure 1.2

EMPLOYEE database file. These records were keyed into the structure shown in Figure 1.1.

Record#	TITLE	FIRSTNAME	LASTNAME	ADDRESS
1	Mr.	Rod	Bannister	7279 Ridge Drive
2	Ms.	Evelyn	Chabot	2613 Henderson Hiway
3	Mr.	Ahmad	Arguello	4 Chestnut Lane
4	Mr.	Michael	Antonucci	4901 101st Place SW
5	Ms.	Rosalie	Gills	1350 Beverly Road
6	Mr.	Bradley	Wachowiak	700 Cumberland Court
7	Ms.	Karen	Shepherd	3107 Peachtree Drive
8	Ms.	Kathleen	Salazar	75 Dorado Terrace
9	Mr.	Arthur	Sotak	1217 Carlisle Road
10	Ms.	Jean	Hurtado	3202 E Dry Creek Rd
11	Mr.	Robert	Keller	416 Whittier Drive
12	Mr.	Herbert	Licon	1220 E Barcelona

Record#	CITY	STATE	ZIP	HIREDATE	HRLYWAGE	DEPENDENTS	DEPT
1	San Mateo	CA	94001	05/06/91	17.00	2	A
2	San Diego	CA	95609	02/14/91	18.00	1	B
3	San Mateo	CA	94001	01/04/91	19.00	0	A
4	San Francisco	CA	94104	01/05/91	11.00	3	B
5	San Francisco	CA	94109	03/16/91	21.00	0	B
6	San Mateo	CA	94001	05/11/91	19.50	2	A
7	San Mateo	CA	94001	02/06/90	21.00	0	A
8	San Francisco	CA	94104	11/13/90	16.50	2	B
9	San Mateo	CA	94001	11/04/90	23.00	0	A
10	San Diego	CA	95609	11/04/90	10.50	1	B
11	San Francisco	CA	94109	01/03/90	17.00	4	A
12	San Francisco	CA	94001	09/01/91	17.00	0	A

SEARCHING A DATABASE

What if you want to see a list—either on your screen or printed out—of only those elements of your database that meet certain criteria? In a manual filing system, depending on how the folders are organized, you might need a long time to perform special searches for requests such as the following: "Pull out the folders in the employee filing cabinet for every employee who makes more than $15.00 per hour." If the folders are organized in alphabetical order by name, this could take hours!

In contrast, performing **searches** with DBMS software and database files stored in computer-usable form is fast and, once you get comfortable with the procedure, easy. As long as you know the field names defined in your database structure (such as TITLE and FIRSTNAME) and are familiar with the types of operations you can perform, you can ask the DBMS for the answer to any number of questions about your database. In Session 2, you will practice searching your database for specific information.

There are three commonly-used categories of database management operations: (1) arithmetic operations, (2) relational operations, and (3) logical operations.

- *Arithmetic operations*. You can perform **arithmetic operations** on the numeric fields in your database. The following are arithmetic operators:

+	Addition
-	Subtraction
*	Multiplication
/	Division
( )	Parentheses used to group operations

- *Relational operations*. **Relational operations** are used to analyze the contents of fields. For example, you would use a relational operator to list the records in your EMPLOYEE database file that have an hourly wage field containing an amount greater than 15 (Figure 1.3), or to list all employees who work in Department A (Figure 1.4). With dBASE III Plus, the following relational operators can be used to perform relational operations:

>	Greater than
<	Less than
=	Equal to
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to

  The result of relational operations is always a true (T) or a false (F) answer. Another operation is usually performed on the basis of the answer, such as listing the record on the screen.

- Logical operations. **Logical operations** allow you to search your database for special information. For example, you might want to list on the screen all the employees who earn more than $15.00 per hour *and* who work in San Diego. Two commonly-used logical operators are described below; each begins and ends with a period (.):

  .AND. allows you to specify that two or more conditions exist before a specific action is taken. For example, using the EMPLOYEE database, test to determine if CITY = San Francisco *and* ZIP = 94104. This command would allow you to screen out all