1

Introduction to Solve it! 2.9 Millennium

Welcome to the challenge of *Solve it!*

Solve it! For Windows 2.9 Millennium is a new learning system which teaches management problem solving skills through the use of PC spreadsheet and database software. *Solve it!* is designed for use by university students in schools of business and management.

The management cases in this *Solve it!* package will help you learn how to solve management problems using contemporary spreadsheet and database software on micro computers. The cases are all derived from actual business applications.

When you successfully complete these cases, you will have mastered the basics of the software and learned how to apply it in realistic settings.

Learn Through Discovery

Solve it! uses real world case studies to present students with problems and challenges. The cases range from small businesses on Main Street, to banks and brokerage firms of Wall Street, to government agencies. It's up to you to discover the answer.

For each case there is a corresponding data file on the diskette. You will be asked to enter data, formulas, data fields, and programs in response to problems posed in the cases.

In general, we have sought to reduce the amount of elementary data entry to a bare minimum and to emphasize conceptual tasks. The cases do not require advanced financial analysis or accounting skills. Explanations are provided for all formulas and analytic tasks.

Industry Standard

Solve it! is widely used in Fortune 1000 training programs. It is designed to bring students and working professionals up to a common intermediate level of proficiency in Microsoft

Windows, Excel, and Access. From this level, industry training programs and seminars are advised for more advanced training.

Many universities and businesses use one of several powerful alternatives to Excel and Access. *Solve it!* works well with Access workalikes such as Foxbase+, and Paradox, and spreadsheet alternatives like Lotus 1-2-3 and Quattro Pro.

You may also use the data files included with *Solve it!* on Macintosh computers using software such as Microsoft Excel, and Foxbase+/Mac. You can use the *Solve it!* data diskette with any Macintosh software that can read .XLS, .DBF, and/or ASCII files. Ask your instructor for more information.

You may wish to keep both the book and the completed, graded exercises to show potential employers precisely what skills you have learned. You should also place this information on your resume.

Classic Business Problems

Solve it! uses cases which illustrate classic business problems typically encountered in the real world.

Problems like net present value analysis, payroll accounting, inventory management, break-even analysis, accounts receivable aging reports, pro forma financial statements, quality assurance, production planning, marketing database management, sales management systems, and personnel tracking.

When you complete the cases in *Solve it!* you will be well prepared to work effectively in a contemporary business environment.

Web Exercises and Case

Solve it! now contains a new chapter of exercises and case studies which explore the use of the World Wide Web in business problem solving. While the Web is still evolving, it has already proved itself a valuable research tool for business. On the Web you can discover industry trends, statistics, growth patterns, markets, and resources. In short, all the ingredients needed to make a well-informed business plan. The Web cases in Chapter 6 will show you how to use the Web to build a business plan, plus help you explore other features of the Web.

Documentation Included

Solve it! contains all the documentation you will need on how to use spreadsheet software like Excel and database software like Access. The documentation is provided in the form of hands-on tutorials which show you how to use the software skills required by each of the cases.

The cases and documentation were written using Microsoft Excel and Access for Windows. You will find the data files in *Solve it!* are compatible with all versions of the software.

In general, the documentation instructions work equally well for clone software with only minor changes.

Students may wish to consult the original documentation for the software being used, or any one of several large reference manuals. These are generally available in your PC Lab, corporate or college library.

Solve it! assumes the student has a basic familiarity with Windows 98 and Windows 2000. If this is not the case, you should read through an introductory text on Windows or the Windows Users Guide.

The Skills Matrix

In order to select and develop cases, we created a Skills Matrix to identify both the PC software and management skills we sought to teach.

The Skills Matrix for spreadsheet software is shown on the following page (see Figure 1-1). A similar matrix is used for database problems.

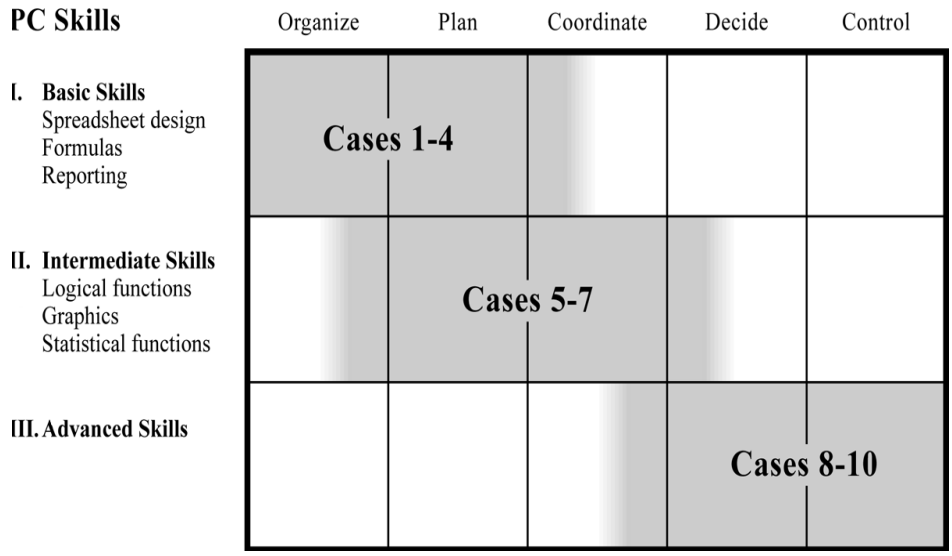
The management skills are to organize, plan, coordinate, decide, and control. The PC software skills are basic (set-up and editing), intermediate (data analysis and organization), and advanced (database management, programming and interfaces).

Each *Solve it!* package contains a mix of skill levels and skill areas. About one third of the problems involve basic PC skills and elementary management skills of organization and planning. The remaining cases develop intermediate and advanced PC skills along with more advanced management skills.

As you proceed from beginning to more advanced cases, the problems become less structured and more analytic. More advanced cases require a written summary.

Each *Solve it!* case identifies in the beginning the specific skills involved in the case. In addition, the approximate completion times for persons at different skill levels are also included. These expected times are based on our experience in university classrooms and industry settings.

Figure 1-1



How to Use Solve it!

There are ten spreadsheet, ten database, and three World Wide Web cases in each *Solve it!* package. The cases are graduated in difficulty, both in terms of software skills and management skills. The cases are short enough to be answered in one computer session lasting no more than 2-3 hours for a novice.

Each case has an estimated completion time. Students are strongly advised not to skip early problems. If you skip early problems you will not learn the software skills required in later cases. This will, in turn, lengthen the time required to answer later cases by several hours.

Each case is followed by a Tutorial Documentation section which carefully describes the software skills needed to solve the case. The software skills are demonstrated using sample spreadsheets and database files.

You should first read the case to understand the nature of the problem. Then you should study the Tutorial carefully to be sure you understand how the software works. Last, you should begin work on the case itself.

How to Cope With Ambiguity in Cases

Because *Solve it!* cases derive from real world events and circumstances, they often contain ambiguities--just like the real world. In advanced cases you will typically find more than

one way to solve a problem, and you will find that certain assumptions and value judgments must be made in order to arrive at any solution.

You should first identify clearly the nature of the ambiguity. Then consider the alternative solutions. Choose the solution you prefer and clearly state the assumptions and value judgments you are making. Be prepared to defend these assumptions, as well as to learn from others who made different assumptions.

System Requirements

Solve it! assumes that you have some knowledge of IBM or IBM compatible computers, including the operating system. *Solve it!* provides specific instructions on how to start up the software in special **Getting Started** sections of Chapters 2 and 4.

Solve it! assumes you are using an IBM or IBM compatible PC computer with at least 32 megabytes of RAM and sufficient hard disk capacity to run the application software. You will also require a printer with graphics capabilities.

General Instructions

1. Make a copy of the *Solve it!* data diskette. Store the original diskette in a safe place. This is always a wise procedure to prevent damage to the original diskette. The original can also be used to restore the shipping data sets should you accidentally change or alter them.

   Label the copy diskette *"Solve it! Copy"* and use this diskette for all your work.

2. Read the appropriate case study in the case book quickly to identify the specific data file used in the case. This will also provide an overview of the basic issues of the case.

3. If you are working with a computer which has a hard disk, place the *Solve it!* data diskette in drive A and copy all the relevant files to the same directory in which your software program is loaded. Configure your software to read this directory when it searches for files.

You are now ready to begin. Chapter 2 introduces you to spreadsheet software in general, and specifically guides you through the basics of Microsoft Excel. This is followed by ten spreadsheet business cases in Chapter 3. Chapter 4 introduces you to database concepts, and Microsoft Access. This is followed by Chapter 5 which contains ten database business cases. Chapter 6 describes the World Wide Web and provides several exercises and cases for you to solve.

2

Introduction to Spreadsheet Software

An *electronic spreadsheet* is the computerized version of traditional financial modeling tools -- the accountant's columnar pad, pencil and calculator. It is organized into a grid of rows and columns. The power of the electronic spreadsheet is that when you change a value or values, all other related values on the spreadsheet will be automatically recomputed.

Spreadsheets are especially well suited to business applications that involve numerous calculations with pieces of data that are interrelated with each other. The power of electronic spreadsheet software is the ease with which modeling and "what if" analysis can be performed: after a set of mathematical relationships has been constructed, the spreadsheet can be recalculated immediately using a different set of assumptions. As models become more complex, this capability for instant "what-if" analysis becomes even more valuable.

The term *spreadsheet* should be distinguished from the term *worksheet*. A spreadsheet is a set of program instructions (such as Excel), whereas a worksheet is a model or representation for a specific application that is created using the spreadsheet software package. (Excel uses the term workbooks. Individual worksheets are considered part of workbooks similar to the pages in a spiral notebook.)

We now teach you how to use Excel, the leading spreadsheet software package. Each spreadsheet case is followed by tutorials showing you the Excel software skills you will need to solve the case..

Introduction to Microsoft Excel

An Excel spreadsheet is divided into rows and columns, with each row and each column uniquely labeled. A sample spreadsheet is illustrated in Figure 2-1.

Rows are identified numerically, with values ranging from 1 to a maximum of 65536 in the latest version of Excel. Columns are identified alphabetically, with letters ranging from A to Z and then from AA to AZ, BA to BZ and so on to IV. Excel can accommodate a maximum of 256 columns.

Cells

The intersection of every column and row is called a *cell*. Each cell represents a unique location on the spreadsheet for storing a piece of data. Cells are identified by their column and row coordinates. For example, the cell located at the intersection of column B and row 8 is called B8. The maximum number of cells on the spreadsheet is equal to the number of rows times the number of columns. In recent versions of Excel, this amounts to nearly 17,000,000 cells.

Ranges

A rectangular block of cells is termed a *range*. A *range* can be a single cell, a row, a column, or several rows and columns. Many Excel commands are based on ranges. Ranges are identified by naming the cells that bound their diagonally opposite corners, usually those on the upper left and lower right cells. Thus the range occupied by the cells containing NAME, QUIZ, MIDTERM, and FINAL in the following illustration could be identified as either A1:D1 or as D1:A1. Range naming conventions require that you separate the cell addresses that specify the boundaries of the range by a colon or by two periods (e.g., A1:C4 or A1..C4).

Another way to specify a range is to use a *range name*. Naming ranges can be easier than the other ways of specifying a range if you need to specify the same range frequently for different tasks or if the range is very large. For instance, we could name the range D2..D5, which contains the final grades, FINAL. You will learn how to name ranges later on in this book.

When you work with your spreadsheet your cursor will always be positioned on one of its cells. The cell where the cursor is presently located is termed the *current cell*. The current cell can be identified by the *cell pointer*, a rectangular highlight which appears on the cell. You can move the cursor from one cell to another by clicking the mouse pointer on the other cell or by pressing the arrow keys on your PC keyboard and by pressing other keys defined for movement by your spreadsheet software. Pressing the <*CTRL-Home*> keys together will move the cursor to the upper left most corner of the spreadsheet, cell A1.

The Excel Screen

At the top of the Excel screen there is a title bar followed by the main menu bar, one or two toolbars (which contains many useful "shortcut" buttons), and a formula bar (or the edit line, which contains the address and contents of the active cell). At the bottom of the screen is the status bar which displays error and status messages or indicators.

The title bar of the Excel window is the line that has the program name ("Microsoft Excel"), the maximize/minimize window buttons and the file name on it. The main menu contains the commands that you use with Excel in the current context. These include commands relating to file handling, editing, formatting, using specialized tools, and using help.

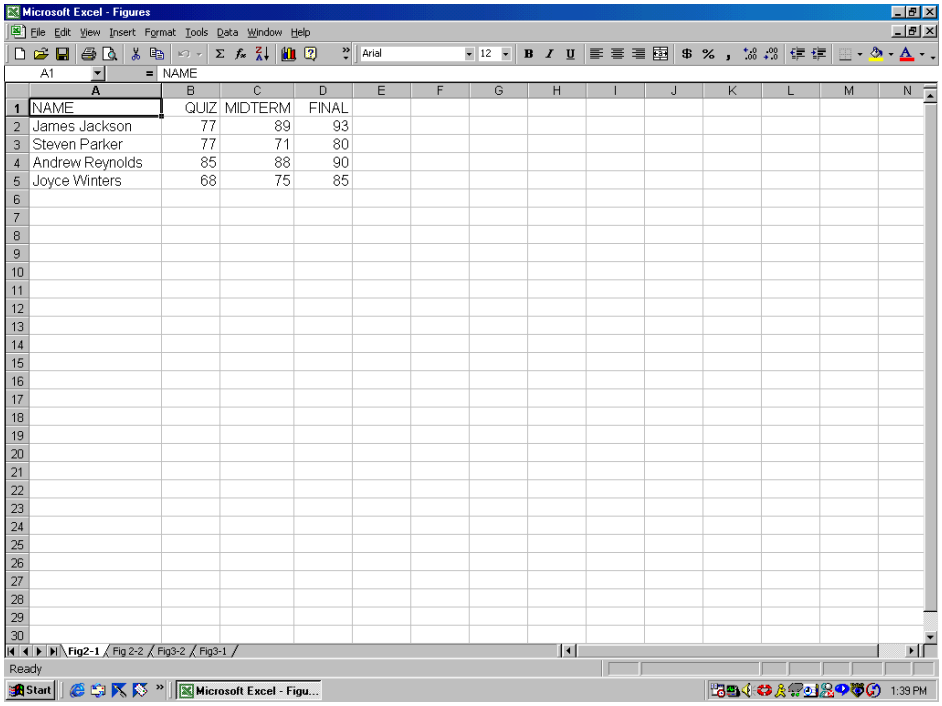
Below the main menu bar are a set of buttons that are shortcuts for many Excel tasks. For example, you can make an entry bold by clicking a button here instead of choosing the **Format-Cells-Font** command and selecting the bold option in the dialog box. These buttons (or icons) can be moved, hidden or customized. When you point to one of the buttons on the toolbars, a ScreenTip appears to explain the function of the button. Familiarize yourself with the Undo and

Redo buttons which can be used to undo (or redo) your last action. (For example, if you deleted a range of cells by mistake and you want to undo your action you could click on the Undo button).

The first item in the formula line is the selection indicator, which displays the name or address of the current selection. The last item on the formula bar is the contents box. When you enter data, such as formulas, numbers, labels and functions, Excel displays the data in the contents box.

Figure 2-1 shows an Excel worksheet screen for a course roster with data entered in range A1..D5.

Figure 2-1



The status bar is located at the very bottom of your Excel window. This gives you information about the current selection and tells you what Excel is doing in addition to allowing you to perform certain simple tasks with the mouse.

Also, just before the Status bar is the "sheet-tabs" line where you can see small icons for the different worksheets within the file. An Excel 2000 or Excel 97 file opens with 3 worksheets, and allows you to include many more worksheets per workbook. (Excel 97 allows up to 255 worksheets per workbook and Excel 2000 allows as many worksheets as your computer memory can accomodate.) Multiple worksheets allow you great flexibility when working with large amounts of data.

The Status bar displays the status and mode indicators that contain useful information on the current context in which you are operating the spreadsheet. The mode indicator at the lower left portion of your screen indicates the state or condition under which the spreadsheet software is

currently operating. Some important mode indicators are READY, indicating Excel is in ready mode, and EDIT, indicating that a cell's contents are being edited.

The status indicator, at the lower right portion of the screen, describes a particular program or key condition. Some important status indicators in Excel are: NUM, indicating the Num Lock is on, SCRL, indicating the Scroll lock is on, and END, indicating that the End key has been pressed and is active.

Moving Around the Spreadsheet

Usually spreadsheets are too large to be viewed on the screen at once. To view other parts of the spreadsheet you must *scroll* the cell pointer up and down the worksheet or across it using the mouse, the arrow keys and other cursor movement keys. When the cell pointer reaches the edge of the current screen, the screen will shift to follow the cell pointer in the direction it is moving.

Clicking the mouse pointer on any cell makes that the current active cell. At the right and at the bottom of the Excel worksheet window are the vertical scroll bar and the horizontal scroll bar. Clicking on the arrow buttons on these scroll bars moves your window one line up/down or one tab right/left. You can also click on any point inside these scroll bars to scroll faster. For instance, clicking in the middle of the vertical bar (and holding the mouse button down) will move you to the middle of your file.

You can also use the keyboard to navigate around the spreadsheet. When Excel is in READY mode, various keys will behave as follows:

Excel Keyboard Pointer-Movement Keys

Key	**Function**
LEFT ARROW	Moves left 1 cell
RIGHT ARROW	Moves right 1 cell
UP ARROW	Moves up 1 cell
DOWN ARROW	Moves down 1 cell
SHIFT-TAB	Moves left 1 cell
TAB	Moves right 1 cell
PAGE UP	Moves up 1 screen
PAGE DOWN	Moves down 1 screen
CTRL-HOME	Moves to upper left corner
END HOME	Moves to lower right corner of the active area

Using the Mouse

You can use the mouse, like the keyboard, to choose commands, highlight ranges, resize windows and many other tasks. There are some actions that you can do only with a mouse such as using the "shortcut buttons" on the toolbars. Whenever a selection is to be made with the mouse, use the left mouse button (unless you specified left-handed use of the mouse in the Windows Control Panel).

Spreadsheet Commands

Commands are tools provided by spreadsheet software to manipulate the spreadsheet in various ways. For example, there are commands for copying data, formatting your worksheet, or printing your worksheet. Some commands affect the entire worksheet, but others only affect certain cells or groups of cells.

Commands appear in the Main Menu, directly below the title bar. You choose commands from the menu to perform actions in Excel. The commands in the main menu change depending on your current selection. When you open a worksheet the initial commands that appear in the main menu are

File Edit View Insert Format Tools Data Window Help

When you choose a command in the main menu a pull-down menu appears listing additional commands you can choose. For example, if you choose "File" from the menu, you see the File pull-down menu from which you can choose "New" to open a new file.

You can choose a command from the menu by clicking with the mouse on the command or by using the arrow keys to select the command and pressing Enter. You can also hit ALT-F to bring up the File pull-down menu. Every command in the menu has a character that is underlined. For example the Format command has "o" underlined. Pressing ALT-O would bring up the Format pull-down menu.

When you choose a command not followed by an "..." (ellipsis) or an arrowhead, Excel performs that command immediately. For example, if you choose "New" under the File pull-down menu Excel opens a new worksheet. However, when you choose "Save As..." under the File pull-down menu , the command is not performed immediately since you need to specify the new file name in the dialog box that comes up.

Pressing ESC at any time you are working with menu commands will return you to the previous command. Pressing ESC as often as necessary will return you to whatever point you wish in the command menus and even bring you out of MENU mode altogether.

Setting Up a Worksheet

Consult with your technical support specialist about how to install and configure Excel for your particular computer system and whether you are allowed to make a backup copy of the program.

To illustrate how spreadsheet software works, we will be developing a simple spreadsheet with which you are very familiar. Let us start by creating a course roster with information on student names and grades for a quiz, midterm, and final exam. The final product will look like Figure 2-1.

Assuming Excel has been properly installed, you can start using it by guiding the mouse pointer over the Start button, clicking on Start, and moving the pointer up the Start menu to Programs. Highlight Programs and then highlight Microsoft Excel. Click once to open the application. (If Excel is not on the Program menu, try looking under Microsoft Office.) Excel will open up with a blank worksheet. You can now start entering data into your worksheet.

How to Enter Data into a Worksheet

The student roster illustrated in Figure 2-1 is a very simple list. You need only enter data about the student's name and grades. No calculations are required. You can start at cell A1 and enter your column headings. Enter NAME in cell A1, QUIZ in B1, MIDTERM in C1, and FINAL in D1.

You enter data into a cell by moving the cell pointer to the cell, typing the entry, and pressing ENTER. The entry appears in the cell as you type.

Each character you type appears in the contents box of the edit line (below the main menu). The cursor indicates where the next character you type will appear. In addition to pressing ENTER, you can complete a cell entry by clicking on another cell or by pressing one of the pointer-movement keys. This will complete the entry and move the cell pointer to another cell as directed by the mouse or the pointer movement key.

You can complete your roster by entering the student names in cells A2 through A5, the quiz grades in cells B2 through B5, the midterm grades in cells C2 through C5, and the final grades in cells D2 through D5.

How to Change Column Widths

You will notice that after you enter the quiz grades in Column B, the student names in Column A become truncated. This occurs because the column widths are at the default width of 8.43 characters, and the names on our list are longer. Since the cells in Column B contain the quiz grades, Excel cuts off the entries in Column A at the right edge of the cells. (However, if the cells to the right of Column A were empty, the entries in Column A would extend into the blank cells in Column B.)

Sometimes pound signs (########) will appear in a cell containing a numeric value. This happens when the column width is too narrow to accommodate the number of places in the numeric value plus additional punctuation for decimal points, commas, dollar signs, and so forth.

You can make column widths smaller or larger by using the **Format - Column** and then **Width** command. Let us reset the column for student names in our example to 20 positions.

First select the range for which we would like to adjust the column width. You can select a range by first clicking at the center of the cell, and then holding the mouse button down while dragging the mouse pointer to the opposite corner of the range. To select more than one range at a time keep the Control key pressed while selecting the next range. In our example we need to select an entire column which can be done in an easier manner. Every column in the worksheet has column headings just above the first row of the worksheet (the column headings range from A to IV. Clicking on a column heading marks (selects) the entire column. Click on Column A and you will find the whole column highlighted.

Then choose **Format** in the main menu by clicking on it and the Format pull-down menu will appear. Point to **Column** and a sub menu will appear. You have five choices here. You could adjust the column width for the selected range to a certain number of columns. To do this, click Width and then enter the desired width in the **Width** box. You could also select the **Auto Fit Selection** option to adjust the column width so that the widest entry in the column fits exactly. You could hide or unhide the column. You could also change the default width of all the columns in the worksheet. In our example set the width of Column A to 20 characters and click on OK.

Values and Labels

Excel, like 1-2-3, has two types of cell entries: *values* and *labels*. Values are either numbers, functions or formulae. Labels are used for text entries within your spreadsheet. Labels can't be used in calculations.

As long as there is a letter in the cell Excel will consider the entry as a label, even if the entry begins with a numerical character. The entry will always be treated as a value if the beginning character is one of the following:

= + - $

Position the cell pointer on the student name Steven Parker. Then position it on the column heading NAME. You will notice that the contents box in the edit line displays Steven Parker or NAME. Any cell containing a letter is considered a label. The entries could be aligned left, right or center, but labels are left-aligned by default.

Justifying Labels

You can right-justify the labels above the grades by selecting them and clicking the align right button or center justify them by clicking the center button. Alternatively, you can use the **Format - Cells** command to align a selected range of labels. In the dialog box that appears you can specify the alignment and the range.

Numbers in Excel

In Excel a number can contain a maximum of 240 characters, cannot contain spaces and is limited to only one decimal point. (You can change the cell formats for displaying a number, which will be discussed later.) A number can be entered in scientific notation, or can end with % to indicate percentage. When a number ends with a percent sign, Excel will divide the number that precedes the sign by 100.

Let us review the student roster you've entered. At this point, the labels will all be right-justified except for NAME and the numbers will be right-justified as well. The width of column A will be 20 characters and that of the rest of the columns will remain at the default value of 8.43 characters. The worksheet will look like Figure 2-1.

How to Edit Data

You can edit a cell entry in either of two ways:

1. You can enter data into a cell that already contains information using the same procedure for entering data into an empty cell. This will cause the new data to replace the earlier entry.

2. You can also use the F2 (Edit) key to edit data in a cell. Position the cursor on the cell you wish to edit and press the F2 key. This will switch Excel into EDIT mode. The easier way to switch into the EDIT mode is to click the mouse button inside the Contents Box on the edit line (where the contents of an active cell appear).

Erasing Worksheet Data

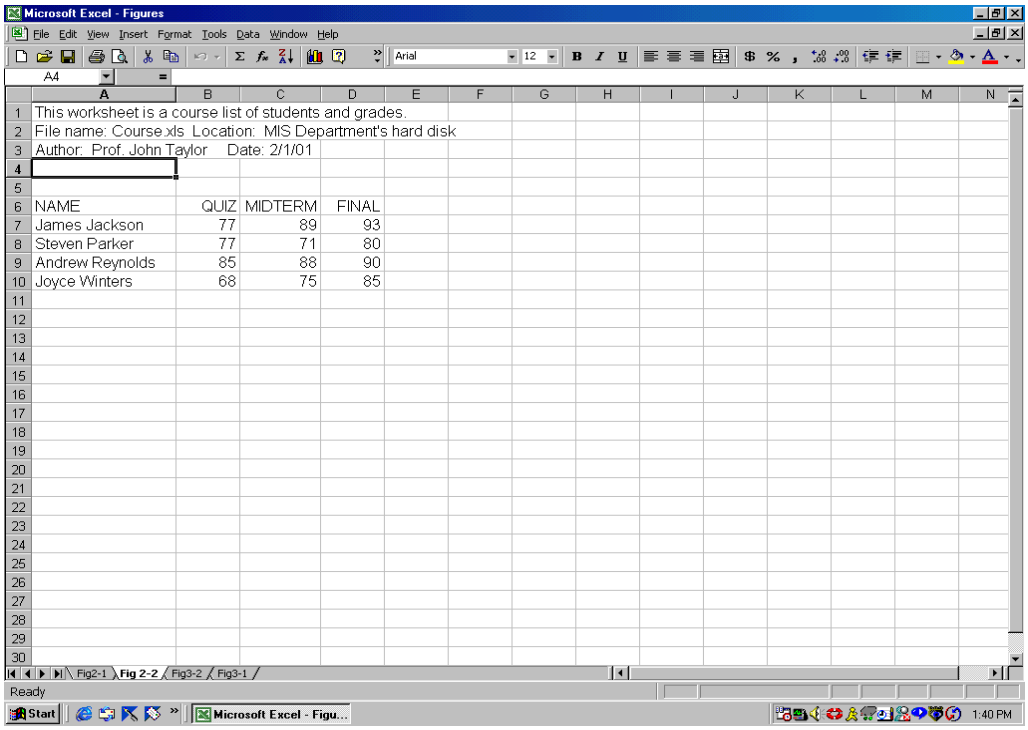
The **Cut** button (below the main menu), symbolized by a pair of scissors, is a useful shortcut (for **Edit-Cut**) to delete data. If you want to delete data in a particular cell, click the mouse in the cell (in order to make that cell the "active cell") and either click on the Cut button or hit the Delete key.

If you want to delete an entire range of cells you would first have to mark (or select) the range that you want to delete. If the range you want to delete is A1..B2 (4 cells), click inside A1, hold the mouse button down and drag the mouse pointer to B2. This procedure highlights the range selected. Once the range is selected you can hit the Delete key or click on the Cut button to erase the contents of the cells in the marked range.

Moving Data

It is a good idea to document each worksheet you create. Figure 2-2 illustrates how the worksheet we just created could be documented. The cell A1 explains the purpose of the worksheet. Cell A2 provides the name and location of the worksheet file. Cell A3 identifies the author of the worksheet and the date it was created.

Figure 2-2



We need to make room for this documentation at the top of the worksheet by moving the worksheet down five rows. To do this, first highlight the range A1..D5 using the mouse. Choose **Edit-Cut** from the main menu. This will delete the selected range (but save the contents in the Clipboard so that it can be pasted elsewhere using the Paste command). Then click in cell A6 (which is where you want the upper-left corner of your selected range to be) and choose **Edit-Paste**.

You could equivalently use the shortcut buttons to make the task simpler. After highlighting the range A1..D5 click on the "Cut" button. Then click inside the cell A6 and click on the "Paste" button.

Another method of inserting rows is to use the main menu. First highlight the number of rows you would like to insert and then from the menu bar choose **Insert - Rows**. For instance, if you want to insert three rows at the beginning of your worksheet you could select the first three rows using the mouse and choose **Insert -Rows**. Similarly if columns needed to be inserted, the same procedure could be repeated using the **Columns** command from the **Insert** menu.

Once you have moved the worksheet down, you can enter the documentation in cells A1..A3 so that your worksheet looks like Figure 2-2. Note that the contents of these cells seem to extend into the blank cells in Columns B through F because the cells to the right of A1..A3 are empty.

How to Save Files

You will learn more about formatting and organizing the worksheet in the case problems. Let's save the practice worksheet for future use. To save a file, you must use the **File-Save** command. This command will make an exact copy of your worksheet on disk, including any special formats and settings you have specified.

First, click your mouse button on **File** in the main command menu. A pull-down menu will appear, on which you click on **Save**. If this is the first time that you are working on this file and saving it, Excel will provide a default filename (book1.xls) in the **File name** box. You can click inside this box, delete the default name and type in a name (such as "Course") that you choose for the worksheet. The "Save in" box indicates the drive and folder in which Excel will store the file. (The drive and folder are established during configuration but they can be changed.) You can change the folder in which you want to save the file by clicking the "Save in" list arrow.

Enter the name you have assigned to your worksheet file in the **File name** box and click on **Save**. Let's call the worksheet Course. Windows 98 supports file names of up to 255 characters. The file name can contain uppercase or lowercase letters, numbers, and most symbols. When you a file name, Excel will automatically assign a three character extension, depending on the type of file. (Recent versions of Excel use the extension .xls for an Excel workbook.) You could choose to save the file in other formats (including Web page format) using the **File - Save As - Save as type** commands.

Be sure to save Course.xls. after you use it for a tutorial. You can save it under the same name after each tutorial session. Most of the changes you make to your worksheet during a tutorial will be required by subsequent sessions. Follow tutorial instructions to determine what changes to your worksheet must be saved or erased.

Ending Your Excel Session

To exit Excel, choose **FILE-EXIT** in the main command menu. It is a good practice to save all your files and Close them before quitting. If you had not saved a file, Excel brings up a dialog box that asks you whether you want to save the file. Click on **YES** to save the file or on **NO** to lose the changes and exit. Clicking on **CANCEL** takes you back into Excel in the READY mode.

Spreadsheet Design Principles

Like any helpful tool, Excel worksheets can be abused and misused, especially if worksheets are carelessly built, poorly documented, and based on false assumptions. These problems can be minimized by following a few basic principles of spreadsheet design that have emerged over the last decade.

A vertical five-section structure can produce more accurate and more easily understood spreadsheets. The five section are: documentation, assumptions, input, calculations, and macros. These sections are illustrated in Figure 2-5.

The first section contains *documentation*--a complete description of the name, author, and purpose of the worksheet. In general, you should try to keep the description simple, and to the point. The first line of documentation section shows the purpose of the worksheet: to display three-year sales projections for EBZ Toy Company.

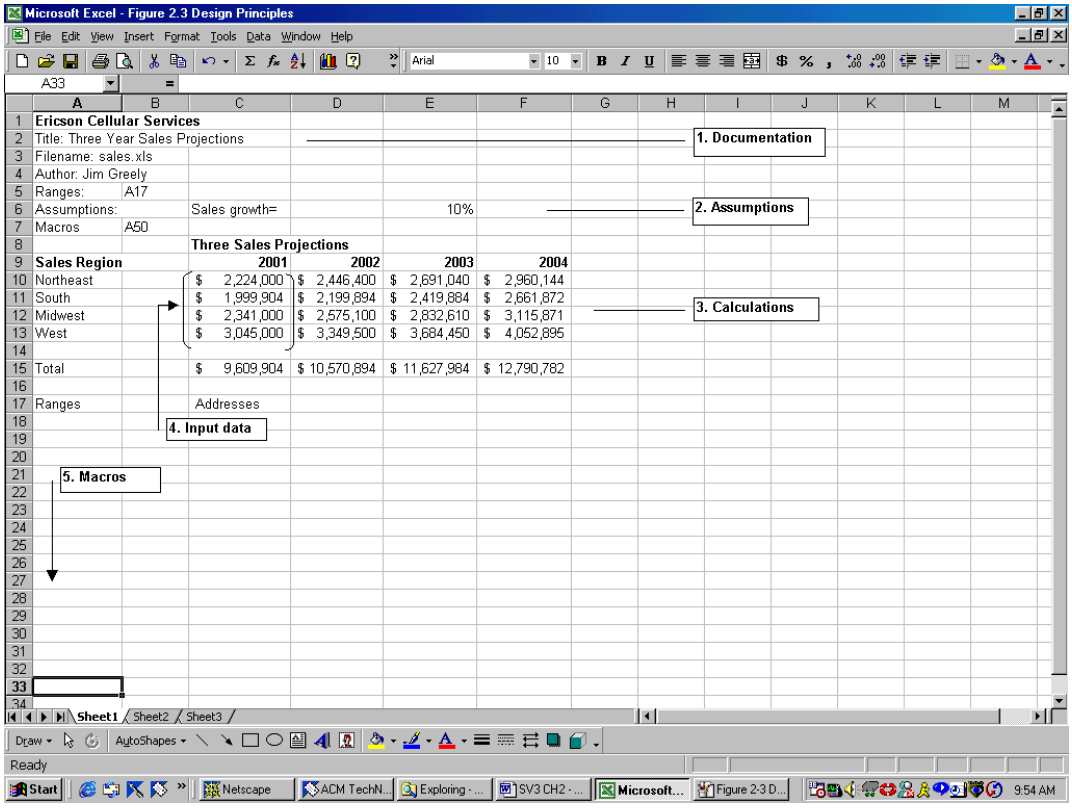
The second line of the documentation section shows the worksheet file name and identifies where the worksheet is located in the firm's computer systems. Generally, spreadsheets will either be stored on a network or the hard disk of a specific machine.

The third line of documentation identifies the author of the worksheet and the date the worksheet was created.

The fourth line of the documentation shows where any table of range (block) names and any strings of commands called "macros" are located. A macro is a set of instructions for automating spreadsheet tasks and consists of a series of keystrokes. The documentation shows that the upper left corner of the macro area for this worksheet starts in cell F28.

The second section in a spreadsheet is reserved for *assumptions*. Assumptions are variable factors that may change in a worksheet. It is important the basic assumptions used to create the output are clearly identified. In our sample application, we are assuming an annual sales growth figure of 10%. This could of course change and assumptions are frequently changed in spreadsheets to test various "what-if" conditions. For instance, the results of this worksheet will be different if we change the assumption for annual sales growth to 5% or to 15%. By isolating the assumptions in a specific section, it is very easy to change the spreadsheet as conditions change.

Figure 2-5



	A	B	C	D	E	F
1	**Ericson Cellular Services**					
2	Title: Three Year Sales Projections					
3	Filename: sales.xls					
4	Author: Jim Greely					
5	Ranges:	A17				
6	Assumptions:		Sales growth=		10%	
7	Macros	A50				
8			**Three Sales Projections**			
9	**Sales Region**		**2001**	**2002**	**2003**	**2004**
10	Northeast		$ 2,224,000	$ 2,446,400	$ 2,691,040	$ 2,960,144
11	South		$ 1,999,904	$ 2,199,894	$ 2,419,884	$ 2,661,872
12	Midwest		$ 2,341,000	$ 2,575,100	$ 2,832,610	$ 3,115,871
13	West		$ 3,045,000	$ 3,349,500	$ 3,684,450	$ 4,052,895
14						
15	Total		$ 9,609,904	$ 10,570,894	$ 11,627,984	$ 12,790,782
16						
17	Ranges		Addresses			

The third section is the *input* section. The input data are the raw data which must be supplied to the worksheet so that they can be manipulated to produce the required results. In the spreadsheet application illustrated here, the data consist of the names of the sales regions and actual sales figures for 2001.

The fourth section is the *calculations* section. The calculations section presents the final output of the spreadsheet, the result of calculations performed on the raw input data. The calculations are the projected sales for 1992-1993 and the total sales for each year. Calculations are usually placed to the right of or below the input data.

The fifth section is the *macro* section. The macro section is the area of the worksheet for storing macros and their documentation. Macros should be located below and to the left of the current worksheet in a portion of the worksheet that will not be affected by changes made to the rest to the worksheet. Usually Macros are placed on the left margin of the spreadsheet. That way, macros are unlikely to be affected by having data copied or moved or by having columns inserted or deleted. The macros in the worksheet illustrated here print and save the worksheet automatically.

The worksheet files on your data disk are designed to encourage you to use these spreadsheet design principles. The first four lines of each worksheet file are reserved for documentation. The *Solve it!* worksheet files provide much of the data for the input section of the worksheet. You will then complete the input and calculation sections of the worksheet to develop the solution for the spreadsheet case. If required by the problem, you can add an assumptions section by inserting rows above the input section or by moving the input section down several rows. Spreadsheet Case 10 will teach you how to set up a worksheet with a macro section.

3

Spreadsheet Management Software Cases

Spreadsheet Case 1

TDC Office Environments

Problem:	Develop a receiving log
Management skills:	Organization Controlling
PC skills:	Data input Worksheet formatting Printing
File:	TDC_Q.XLS

TDC Office Environments is a small firm that manufactures office furniture and equipment, such as computer and printer stands, desks, and chairs. It receives its components from various suppliers in the United States and Europe. These materials are delivered to TDC's Receiving Department. The Receiving Department uses a receiving log to record each shipment delivered.

Receiving Department staff complain that they have to revise the log constantly as new shipments arrive during the day. Revising the logs is becoming very time-consuming because they have to write each new shipment down in a notebook. There are too many entries to keep track of manually and corrections are difficult to make. Theresa Connolly, head of the Receiving Department wants to find a way to create a receiving log that can be immediately revised, rearranged, and updated.

From your diskette load the worksheet file TDC_Q.XLS. This worksheet contains a list of shipments for February 1, 1999. Use your spreadsheet software to create a receiving log schedule that Theresa and her staff can refer to on-line or print out. The log should list the date, shipper's name, purchase order number, and the number of packages for each shipment received. The log must be organized so that shippers are listed in alphabetical order by date.

The look and shape of spreadsheets are extremely important if the information they contain is to be utilized effectively. Professional looking spreadsheets are formatted in special

ways so that information can be located and digested quickly. The format should allow for easy changes and updating. This problem shows you how to develop professional looking, maintainable spreadsheets.

Tasks

There are 6 tasks to this problem:

1. Create appropriate column headings to capture the required information. There should be column headings for Date, Shipper Name, Purchase Order, and Number of Packages. The column headings for Date and Shipper Name have already been provided in the worksheet and can be used as a model.

2. Create appropriate widths for each column and decide whether to left justify, right justify or center the column labels. Some of the worksheet columns have already been widened for you.

3. Complete the log by entering the remaining shipment information. The following table will be helpful:

Date	**Shipper**	**Purchase Order**	**No. Packages**
1/28/99	Bernolli Inc.	A2203	6
1/28/99	RH Jensen	A3405	3
1/31/99	Barton Plastics	A4903	9
1/31/99	CDC Frames	A2216	7
1/31/99	Warren Lumber	B0219	12
2/1/99	Ace Upholstery	B1101	9
2/1/99	Greenwood Fittings	B3102	4

Note: Be sure to enter the date as a label.

4. The Receiving Department just found out that a shipment of 2 packages was received from Zeitler's Finishings on January 31, 1999 and another shipment of 5 packages was received from Barton Plastics on February 1, 1999. Add the information about these shipments to the log by inserting rows to make sure the log remains in sequence by shipper name in alphabetical order by date.

5. Complete the aesthetic re-make of the log by placing dotted lines under column labels.

6. Print out the log. You should be able to print this spreadsheet on a single 8 and one-half by 11 inch page. Go to maximum margins and page length.

Time Estimates

Expert: 30 minutes
Intermediate: 45 minutes
Novice: 1.5 hours

Excel Tutorial For Spreadsheet Case 1

This case draws upon the data entry skills you have already acquired in developing COURSE.XLS, plus new skills for formatting and printing spreadsheets. You will need to use COURSE.XLS again for this tutorial.

How to Retrieve a Data File

Begin by accessing Excel again. When the spreadsheet screen appears, your first step will be to load the data file COURSE.XLS. This can be done by either selecting the Open File button on the toolbar or accessing the Open command under the pull-down File menu. The Open dialog box will appear.

The settings within the dialog box need to be amended in order to load the file. The four items that have to be set are the File Name, the File Type, the Folder (or Directory for Windows 3.1) and the Disk Drive. A list of files of the type specified on the Disk Drive specified, in the Folder or Directory specified appears in the File List window. To change the settings to retrieve COURSE.XLS, firstly ensure **Microsoft Excel Files** appears as the File Type. If it does not immediately, it can be changed by pressing the downward pointing arrow next to the File Type window with the mouse arrow, revealing a list of File Types Excel can import. To move up and down the list, select the arrows on the scroll bars next to the list. Select the required File Type with the mouse pointer.

For Windows 95, use the Look in list arrow to select the Folder and Disk Drive where the Solve it! files are located. (Users of Windows 3.1 should select the Folder or Directory where the *Solve it!* files are located.) The COURSE.XLS worksheet should also be stored here and displayed in the File Name list. The files are listed alphabetically so if the desired file is not visible, you can move down the list using the scroll bars. Alternatively, you can simply type COURSE.XLS under the File Name. Typing the three letter extension to specify your file is optional When the settings are correct, select the OPEN (or the OK) button or press the ENTER key.

How to Insert or Delete Columns and Rows

Suppose you wanted to add lines under the column headings in your student roster. You can insert columns and rows in a worksheet in either of two ways in Excel. Firstly, through the menu command **Insert/Rows**; and, secondly, using the shortcut menu displayed by activating the right-hand button of the mouse, and selecting **Insert**, which presents four options:

Shift Cells Right
Shift Cells Down
Entire Row
Entire Column

Selecting **Entire Row** from the options presented inserts a blank row.

A third method is to select an entire row and then select **Insert** from the shortcut (mouse) menu, or by selecting **Insert/Cells** or **Insert/Rows** from the menu. To select an entire row, you move the mouse cursor over the row number at the left of the worksheet and press the left mouse button. To select multiple rows, keep the left button depressed and drag up or down.

Now to insert the blank line in your student roster. Move the worksheet current cell to the row below the column headings, that is move to row 7; it doesn't matter to which column. Now select **Insert/Row** from the menu and you will notice a blank row is inserted in row 7 and the data previously in rows 7 through 10 will be moved to rows 8 through 11. Your worksheet screen now looks like Figure 3-1.

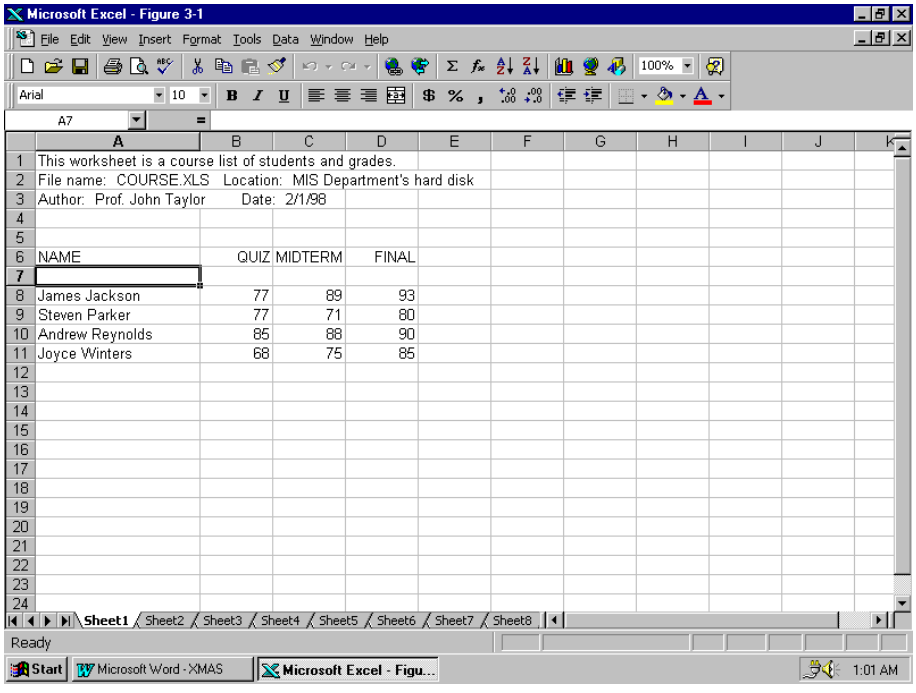
You can delete one or multiple rows and columns using the same principles to inserting rows and columns. There are two ways of accessing the **Delete** command in Excel: firstly, through the pull-down menus, located at **Edit/Delete**; and secondly through the shortcut menu activated by the right-hand mouse button. If entire rows or columns are selected when these commands are selected, the effects will be immediate: the row(s) or column(s) will disappear straight away.

If entire rows or columns are *not* selected when these commands are selected, the following options will appear:

Shift Cells Left
Shift Cells Up
Entire Row
Entire Column

Selecting **Entire Row** or **Entire Column** from the options presented inserts a blank row or column.

Figure 3-1



To make the worksheet documentation section conform to the spreadsheet design principles introduced in Chapter 2, let's add a fourth line to the documentation section. In cell A4 enter "Ranges: none Macros: none". (We will add macros and range names in later tutorials.) Then insert a row so that there are two rows between the documentation section and the course list itself.

Moving and Copying the Contents of Cells

After using the **Insert/Rows** command again, you will have a blank Row 8. You can now add separator lines in this row to further set off the column headings from the data on the list.

This is a convenient time to explain some extremely commonly used and useful operations: Moving and Copying. A Move is referred as a Cut in Excel and is simply relocating the contents of one or more cells. A Copy reproduces the contents of the cells. Both Cuts and Copies have to be accompanied by a Paste operation. The Cut or Copy designates *from where* the cells are cut or copied whilst the Paste designates *to where* they are being placed.

As with most operations in Excel, there are several ways to achieve cutting and copying. There are five different ways to achieve cuts and copies:

1. Using the **Cut**, **Copy** and **Paste** functions from the **Edit** menu
2. Using the **Cut**, **Copy** and **Paste** functions from the **Shortcut** menu
3. Using the buttons on the Toolbar for **Cut**  , **Copy**  and **Paste** 
4. Using the keystrokes for **Cut** (Ctrl-X), **Copy** (Ctrl-C) and **Paste** (Ctrl-V)
5. Using the mouse cursor by dragging and copying.

Each of these is worth exploring at least once and you can decide which you find the most convenient. Generally speaking, most users use two of the methods listed. For example, some users cut and copy within a small area in a spreadsheet using the mouse dragging and copying, and cut and copy between larger areas and between spreadsheets using one of the other techniques. Once you become familiar with each method you will see they are very similar, and they all require you specify a source range of cells, an operation (cut or copy), a destination or target range and the paste function.

It is worth explaining the mouse method of cutting and copying since it is extremely useful. This method applies to a single cell or a multiple selection. When you move your mouse cursor to the border of a selection on the worksheet, it transforms from a white cross to an arrow. This is the signal that you can now perform the drag or copy.

To drag (or cut) the contents of the selection, depress the left mouse button and move the mouse, keeping the button down. As you move the mouse across the worksheet a shadow of the selection will also be moved until the mouse button is released to designate the destination of the drag. This operation may require some practice but will become second nature very quickly and an important editing function.

To perform the mouse copy, the actions are nearly identical to the drag (cut). The mouse cursor is moved to the border of the selection and the cross till turn into an arrow. At this time, press the Control key and keep it down. A small tell-tale cross will appear next to the arrow, indicating that we are performing a copy rather than a move. Now drag the selection as we did for the move and when the mouse button is released you will see the copy operation has worked.

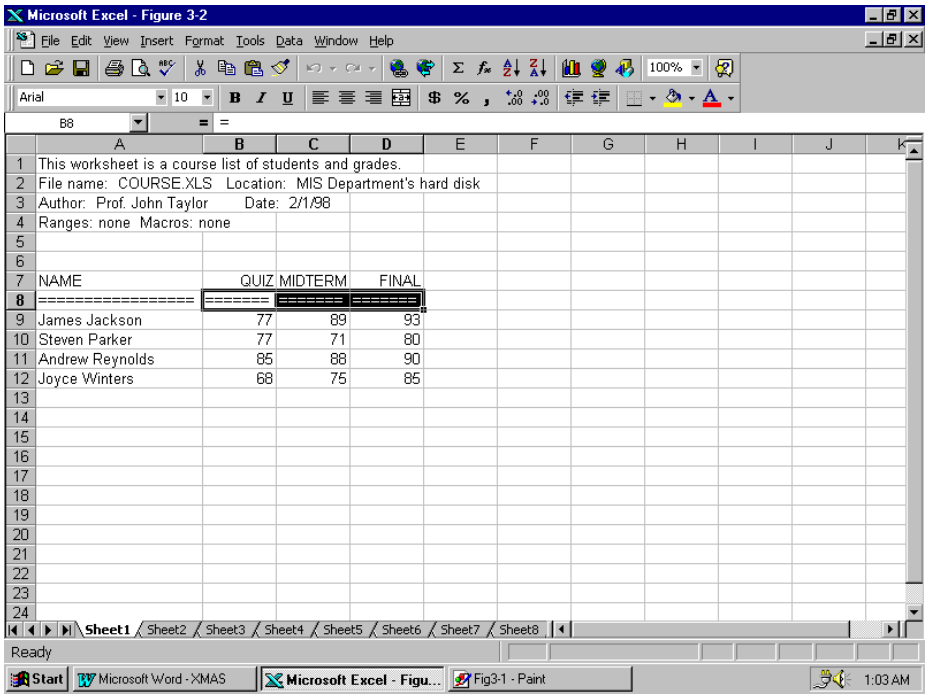
After practice you will realize that each of the different cut and copy methods result in identical results.

Returning to COURSE.XLS, we now want to insert some decorative separators in the cells in Row 8 to distinguish the headings from the body of the table. To do this, we will enter "=" (without the quotes) and select the **Alignment Fill** command to fill the cell width with this symbol.

First, press "=" and ENTER in cell A8. Now select **Format/Cells** for the Formatting box to appear. This box has six formatting tabs: Number, Alignment, Font, Border, Patterns, Protection. Select Alignment by clicking on the tab label at the top of the box. In the Horizontal Alignment section, select Fill and then select the OK Button or press ENTER. The cell will be filled with the "=" symbol.

Once you have set up the first cell, you can replicate it in all the cells in Row 8 below column heads. Do this by using the **Edit/Copy** command. First, select the **Copy** command from the **Edit** menu. Notice the moving border around cell A8, indicating a copy or cut source. Next, select cells B8, C8 and D8, referred to as range B8:D8. Do this by moving the mouse cursor to cell B8 and depress the left button and drag to the right until the desired range is selected. Now select the **Paste** command from the **Edit** menu. This will copy the formatting and contents of cell A8 to the range selected. Your worksheet should look similar to Figure 3-2.

Figure 3-2



Printing Your Worksheet

For printing a simple worksheet such as your student roster, which is one page or less, you need only know the basic printing commands. Select the **Print** command under the **File** menu and a dialog box will appear. The various options in this dialog box can be explored later but for our purposes, simply accept the default settings and press the OK Button.

In this case, Excel will simply print the page containing the student roster. Features that can be adjusted, including paper size, page orientation, scaling, margin sizes, alignment, header and footer contents and options, print titles, page print sequence and print area can be found under **Page Setup** under the **File** menu (see below). Before printing, the appearance of your page can be observed under **Print Preview** under the **File** menu so paper need not be wasted.

As Excel is printing your page, a small message box appears stating which page is currently being printed. This box contains a Cancel Button which can be chosen if you want to stop the printing.

Changing Excel's Printer Options

The printing options in Excel can be changed under **File/Page Setup**. Four formatting tabs will appear in the Page Setup dialog box: Page, Margins, Header/Footer, Sheet.

Under **Page** you can decide on features pertaining to individual pages being printed. The first option is Page Orientation, that is to decide whether the page should be printed in landscape (horizontal wide, vertical narrow) or portrait (horizontal narrow, vertical wide) mode. The second option permits you to scale the size of the print up or down a specific percentage, or scale to fit a certain number of pages where the computer determines the requires percentage scaling. The next option permits you to change the paper size between various standard sizes (e.g. A4, Letter, Legal). The next option allows you to change the Print Quality (in Dots Per Inch), typically available on dot matrix printers. The final option permits you to specify the starting page of printing. To print from the start, enter 1 or AUTO; otherwise enter the page number from which you wish to commence.

Under **Margins** you can change the margins which border the pages. The size of the margins on each page edge can be specified, in the units specified (inches, centimeters). The next option is the distance between the edge of the page and the header or footer. The last options are whether to centre the print subject vertically and horizontally on the page.

Under **Header/Footer** you can change the appearance of the headers and footers of the printed pages. The operation and options of headers (appearing at the top of every page) and footers (appearing at the top of every page) are the same. Each is offered a number of default sample headers, a list you can add to. Pressing **Custom Header** or **Custom Footer** permits you to change the header/footer. Each is split into thirds: left, centre and right. In each or any of these sections you can enter any text or enter any of the options offered: current page number, total pages in print, Current Date, Current Time, Document Name and Sheet Name. The fonts of any text can also be changed using an available button.

Under **Sheet** you can change the print features pertaining to the worksheet. Firstly, you can precisely specify the range on the worksheet you want to print. This is done by placing the cursor in the window provided and then highlighting the range on the worksheet. The next option lets you specify Print Titles which are either rows or columns that are to appear on every printed page. These are useful for lists or tables that extend beyond the confines of a single page. The Next group of options are general purpose ones: Print Gridlines, Print Draft Quality, Print Black and White and Print Row and Column headings. The final option is the Page Order, that is whether to print down and across or across and down.

Save COURSE.XLS with the changes you made during this tutorial session. You will need it for the next Spreadsheet Case.

Spreadsheet Case 2

Speedy Copy Service

Problem:	Develop a payroll register
Management skills:	Organization
PC skills:	Formulas Absolute and relative addressing

File: SPEED_Q.XLS

Speedy Copy Service is a fast-service printing and reproduction service operating in Stamford Connecticut. Last year its gross revenue was over $600,000. The single store currently has eight employees, although the number fluctuates during the course of the year. Business has been so good that the store's owner, Frank Losapio, is thinking about opening a second store nearby.

Frank currently processes the payroll manually. This is a fairly time-consuming process which prevents Frank from fulfilling other responsibilities. He must make all of the calculations for salary changes, deductions, and net pay using a hand-held calculator and then type the results onto a Payroll Register sheet. Frank then writes out the checks by hand. Payroll Registers are filed for internal accounting purposes and to provide information for federal and state tax reporting.

Frank would like to use his time more effectively for courting new business and keeping up with new reprographic technology. Also, there is a danger of miscalculations which are difficult to detect and correct. Frank would like to automate the process as much as possible to facilitate anticipated expansion.

Frank feels that there are so few checks to write that this part of the process could remain manual. However, many hours could be saved if he could automate all of the payroll calculations and the preparation of his Payroll Register report.

From your data diskette load SPEED_Q.XLS, which shows the basic format for the payroll register: the pay period, the names of the employees, social security numbers, hours worked and hourly pay rate. You should develop a worksheet which can automatically calculate weekly gross pay, net pay, and all deductions for each employee and provide year-to-date totals for each of these categories.

Weekly gross pay can be computed by multiplying pay rate by regular hours worked plus time and a half for overtime hours. Federal withholding tax should be set to 20% of gross pay; state withholding to 6%. FICA (the employee Social Security deduction) is currently 6.2% of the first $68,400 during the calendar year. The Medicare deduction is 1.45% of gross pay for all wages during the calendar year. Since this is the first pay period of 1999, FICA and Medicare deductions must be taken for all employees during this pay period.

Tasks

There are 5 tasks to this problem:

1. Complete the column labels to include all deductions and net pay.

2. Make all appropriate format changes for numbers and percentages. Columns containing numbers should be formatted to show 2 decimal places to the right of the decimal point.

3. Create a box for all deductions and other variables in the lower left hand corner of the spreadsheet and label it `Variable Factors'. This way changes in deductions and formulas can be made easily using the addressing function of spreadsheet software. By keeping all variable factors in a single small box, changes in the spreadsheet can be made very rapidly to respond to changing tax policy or other regulations. Listing all variables and formulas also allows all assumptions to be visible and clearly reported.

4. Use formulas to calculate gross pay, all deductions, and net pay. Be sure these formulas reference the appropriate cells in your variables box in the lower left hand corner of your spreadsheet.(e.g., $A$56) rather than actual values (e.g., 20%). Provide totals for gross pay, net pay, and each deduction category so that Frank can track his expenses for the pay period.

5. Print the spreadsheet. Try to fit the sheet on a single page.

Additional Problems

1. Frank has joined a group health plan through his local chamber of commerce. The cost to the employee for group health insurance is $15 per week. Modify the Payroll Register worksheet to implement this plan and print it out.

2. Business is so good that Frank would like to give all employees a 10% raise. How much more would he have to pay out each week to do this? Modify the Payroll Register Worksheet to provide Frank with this information and print it out again.

Time Estimates

Expert: 30 minutes
Intermediate: 1 hour
Novice: 1.5 hours

Excel Tutorial For Spreadsheet Case 2

This case draws upon all of the skills acquired in Spreadsheet Case 1 plus new skills for using formulas, formatting and absolute and relative addressing. You will need to use COURSE.XLS again for this tutorial.

Suppose you want to expand your worksheet by including each student's final grade. You will need to add an extra column and label for FINAL GRADE and you will need to calculate the final grade for each student. The final exam counts for 50% of the final grade; the midterm for 35% and the quiz for 15%.

Formulas

To compute the final grade you would need to use a formula: A formula tells Excel what manipulations to perform on specific cell contents. The cells are specified using their cell references (e.g. A11, C3). Mathematical operators specify arithmetic operations. They are:

^	Exponentiation
*, /	Multiplication, Division
+,-	Addition, Subtraction
%	Percent (i.e. 75% represents 0.75)

Operations are always performed left to right within a formula in their order of precedence. The order of precedence in Excel corresponds to the order of the above list. Exponentiation will always be performed first, followed by multiplication and division, then addition and subtraction. Percent amounts are evaluated when they are encountered.

Parentheses can be used to override the order of precedence. Operations inside parentheses will be performed before those outside the parentheses. The order of operations remains the same within the parentheses, however. When multiple sets of parentheses are employed, the operations within the innermost set of parentheses will be performed before those within the next set.

Thus, the formula for James Jackson's final grade would be:

=B9*.15+C9*.35+D9*.50

or alternatively

=B9*15%+C9*35%+D9*50%

Enter this formula in cell E9. This cell will display Jackson's final grade. Note that the first cell in the formula is preceded by a = sign. In order to be treated as a formula, rather than a label, a formula must begin with an equals (=) symbol.

Thus a formula to add the contents of cells A6 and B6 must be expressed as =A6+B6. If you try to type this formula as A6+B6, it will be treated as a label.

It is usually more accurate and reliable to create the cell references by using the cursor keys to point a mobile selecting cell with a moving border and finish with the next operator (+,-,*,/) or ENTER. Similarly, cell references can be created using the mouse cursor and left button.

Formula Errors

If you try to enter a formula with a logical or mathematical error, Excel will show a message box stating what error has occurred. To proceed you must press the OK Button. If you want further information on the error you can select the Help Button, which will provide a broader explanation of the error. You need only correct the problem to continue.

Another common problem is the circular reference, which is a formula that directly or indirectly refers back to the same cell it resides in. For example, if you tried to enter in cell B12 the formula =A12+B12, an error box will appear stating a circular reference has occurred, and a message "Circular: B12" would appear at the bottom of your screen. This is because cell B12 is an operand in the operation, as well as the cell that holds the result of the calculation. The "Circular: B12" message will disappear when the circular reference is corrected.

Absolute and Relative Addressing

Suppose that we want to make our worksheet more flexible for future changes. The professor may decide that the quiz should only contribute 10% toward the final grade and the midterm 40%. In that case, the formula for the final grade would have to be adjusted to change the percentage weight applied to the midterm and quiz.

You could, of course, re-enter the new formulas. But an easier way to keep track of the percentage weight assigned to each grade would be to list the percentages assigned to each grade in an unused portion of the spreadsheet. Formulas would reference the cell addresses where these percentages reside rather than the percentages themselves.

Set up an Assumptions section in the upper left-hand corner of your worksheet. Move the course list down so that the column labels are in row 14. Enter the label "Assumptions" in cell A7 and enter underlining in cell A8. Below that, in cells A9 through A11 enter the labels "Quiz," "Midterm," and "Final Exam." In cells B9 through B11 you would enter the percentage weights for each of these grades.

You could then develop a formula to reference the cells where these percentages resided rather than using the percentages themselves. The formula for James Jackson's grade (which should be entered in cell E16 since the worksheet was moved down) would then be:

=B16*$B$9+C16*$B$10+D16*$B$11

The $ designates an *absolute address*. An absolute address is one that will not change when that address is copied. Excel's default is to treat an address as a *relative address*, meaning that when you copy or move a formula, the addresses of the cells in the formula will be adjusted automatically to fit the new location. A relative address has no $ symbols. Any formula with multiple cell references can have absolute, relative and mixed (see next section) all in the one formula.

In other words, if you copied the formula in cell E16 for James Jackson's grade to cell E17 for Pauln Parker, the formula in E17 would automatically adjust to add the proper cell addresses for Parker's grade. The formula bar would show the formula in E17 to be =B17*$B$9+C17*$B$10+D17$B$11.

Mixed Addressing

There will be certain situations where you will want to combine relative and absolute addressing; that is, create a cell reference that is part relative and part absolute. Either the column letter or the row number remains constant.

For example, an address of $B21 means that absolute addressing will be used on the column portion of the address, but relative addressing will be used on the row portion. Conversely, an address of B$21 means that absolute addressing will be used on the row portion of the address and relative addressing on the column portion. You will need to use relative, absolute, and mixed addressing throughout your *Solve it!* Spreadsheet Cases.

Formatting

Suppose you want to express the percentages in your Assumptions section as 50% rather than .5. You can change the format in which numeric information appears by using Excel **Format/Cells** commands.

Firstly, select the cells you want to change by selecting the range B9:B11.

The **Format/Cells** command activates the Format Cells dialog box, as explained in the tutorial for Spreadsheet Case 1. This box has six formatting tabs: Number, Alignment, Font, Border, Patterns, Protection. To change the format of numbers select Number. Do this by clicking on the tab label at the top of the box. The Format Category list is divided into the different types of numerical appearance. Select Percentage with zero decimal places and press the OK Button.

You should see the effects of this formatting on the three figures in the Assumptions area of the worksheet. They will be displayed as percentage figures: 15%, 35% and 50%. Excel often provides shortcuts to frequently used commands and operations by allocating buttons on Toolbars. In this case, Excel has placed a Percent Style Button on the Formatting Toolbar. To format the current cell(s) in a Percent Style simply press the Percent Style Button:

The different Number categories are:

General format is the default format for all new worksheets. With the General format the values are displayed in their natural state, no suppression or compression of formatting is permitted. Scientific notation will be used to display numbers that are too large or too small to be displayed normally. The General format displays up to 11 digits.

Number formats display a fixed number of decimal places, negative values are red, negative values in brackets, and combinations of these characteristics.

Accounting format displays four accounting formats keeping the dollar sign to the left of the cell, showing negative values in brackets and show zero values as hyphens.

Date and **Time** formats can represent dates and times in a number of formats where they can be used by Excel date and time functions in mathematical calculations. Default date formats include:

d/mm/yy	(Example 2/02/80)
d-mmm-yy	(Example 2-Feb-80)
d-mmm	(Example 2-Feb)
mmm-yy	(Example Feb-80)
d/mm/yy h:mm	(Example 2/02/80 15:15)

Default time formats include:

h:mm AM/PM	(Example 3:15 PM)
h:mm:ss AM/PM	(Example 3:15:30 PM)
h:mm	(Example 15:15)
h:mm:ss	(Example 15:15:30)
d/mm/yy h:mm	(Example: 2/02/80 15:15)
mm:ss	(Example: 15:30)
mm:ss.0	(Example: 15:30.0)
[h]:mm:ss	(Example: 75:30:00)

In this last format, 75:30 is displayed as 75:30. In the format h:mm:ss, 75:30 would be displayed as 3:30:00. Placing square brackets around the hours displays the literal value.

In order to use date and time formats, you must use one of the Excel date and time functions (such as =DATE, =DATEVALUE, =TIME, =TIMEVALUE, or =NOW) to enter dates and times onto your worksheet. Excel date functions will be treated in more detail in Spreadsheet Case 8. A date can also be entered as a label if it is not involved in any calculations.

Percentage format displays numbers as percentages. The number of decimal places in the default formats are zero and two.

Fraction format displays numbers as fractions, separating whole numbers and the fractional parts.

Scientific format displays data in exponential scientific notation.

Text format displays the values as labels rather than values.

Currency format places a dollar sign before each number and uses commas to separate hundreds from thousands, etc. The option for zero or two decimal places and for red or black negative values.

Custom format can be used to create custom formats for numbers such as product codes.

After you have finished formatting, calculating final grades, and adding a column heading for FINAL GRADE your worksheet should look like Figure 3-3.

Totalling the Values in a Range

This case requires totals of gross pay, net pay and all deductions for the payroll register so that the employer can determine the total amount paid out for each of these categories during a particular pay period. The =SUM function of your spreadsheet software

can help you do this. The =SUM function calculates the sum of all of the values in a specified range. The form of the =SUM function is:

=SUM(range)

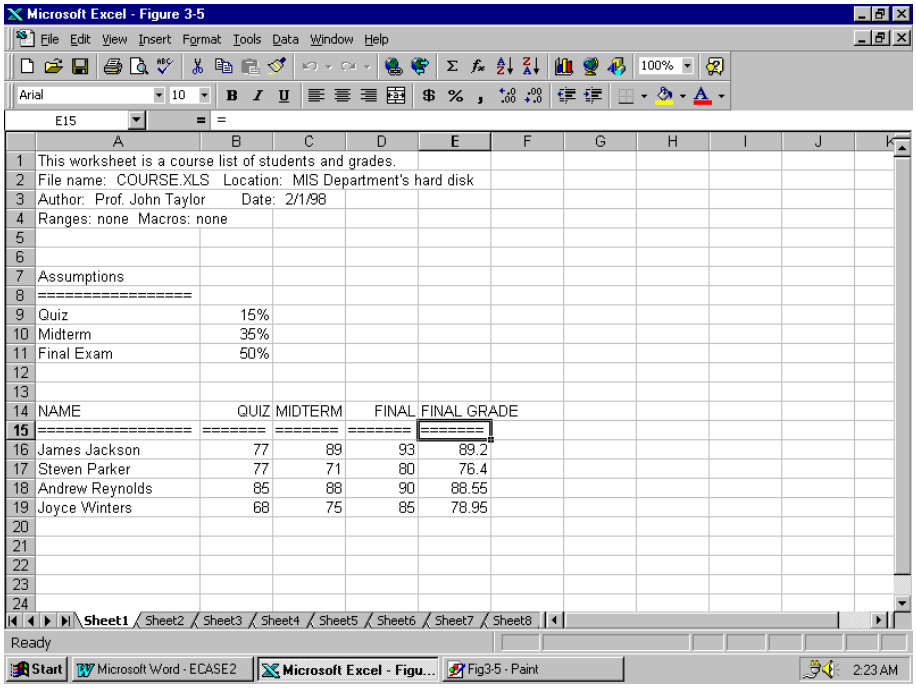
For example, if you wanted to total the percentages in the Assumptions section of your sample worksheet to make sure they added up to 100 percent, you could use the =SUM function instead of the formula =B9+B10+B11. The values in range B9:B11 could be totalled much more easily by entering in cell B15:

=SUM(B9:B11)

Excel provides a shortcut for this formula: the Sum Button [Σ]. This button appears on the Standard Toolbar and if pressed, the =SUM formula will be placed in the current cell and a sample range will be assumed, typically above or to the left of the current cell. All you have to do is press ENTER to complete the operation. Try the button as well as entering the formula yourself. When entering the formula, select the range B9:B11 using the cursor keys or the mouse, rather than typing the range. This tends to be easier and more accurate; typing is more prone to error.

Save COURSE.XLS with the changes you made during this tutorial session. You will need it for subsequent Spreadsheet Cases.

Figure 3-3



Spreadsheet Case 3

Swanson and Hayes Investment Advisers

Problem:	Analyze financial ratios
Management skills:	Deciding
PC skills:	Formulas Spreadsheet control Reporting
File:	HAYES_Q.XLS

Swanson and Hayes Investment Advisers is a small investment advisory firm which has just opened in Oak Park, Illinois. It is trying to take market share from large brokerage houses by offering custom advice to clients, rather than recommending the "stock of the week" touted by large firm research departments.

Clients may visit to check daily quotations or to seek expert advice about particular stocks they are interested in. Sylvia Pierce, a dentist with a thriving practice, has expressed an interest in two telecommunications companies and she wants to know if either make good investments. These firms are Quest Communications International Inc.and SBC Communications Inc.

Quest Communications International Inc. is a high-capacity broadband Internet communications company with two core businesses: communications services and construction services. Communications services include Internet, multimedia, data and voice services sold to businesses and consumers. Quest also provides high-volume voice and conventional private line services to other telecommunications providers, Internet Service Providers (ISPs) and other data service companies. Construction services provide turnkey fiber-optic communication systems for major telecommunications carriers and for Qwest's own use.

SBC Communications Inc. is a global telecommunications company offering a wide array of services and products in the United States. It has investments in telecommunications companies in 23 other countries around the world. Its U.S. subsidiaries include Southwestern Bell, Ameritech, Pacific Bell, SBC Telecom, Nevada Bell, SNET, and Cellular One. SBC provides a competitive mix of local, wireless, Internet, and high-speed data services to millions of customers.

In order to make a sound recommendation, Swanson and Hayes' analysts must look at a company's financial statements. The purpose of financial statements is to identify the major sources, uses, and flows of funds within an organization. The three principal financial statements used in business are income statements, balance sheets, and cash flow statements.

Income statements (also called operating statements) summarize the income, expenses, and profits of businesses for a specified period of time. The purpose of income statements is to show the profitability or unprofitability of firms during the specified period of time, usually a year, a quarter or a month.

Balance sheets identify the assets, liabilities and equity of a firm at a particular point in time. The difference between assets and liabilities is net worth or equity (literally what the organization is worth net of all other factors). Cash flow statements provide detailed information on total receipts and disbursements of cash. Cash flow statements are like checking account registers for individuals. The 1999 income statements and balance sheets of Qwest Communications and SBC Communications will be used for this case.

Certain financial ratios based on figures from financial statements have been traditionally used to assess a company's financial health and performance. There are five kinds of financial ratios that can be applied to assess the financial position of a firm.

1. Liquidity Ratios

Various liquidity ratios measure a firm's liquidity, its ability to draw on cash and other current assets to pay its financial obligations. Two commonly used liquidity ratios are the *current ratio* and the *quick ratio*, or *acid test*.

(a) Current Ratio = $\frac{\text{Current Assets}}{\text{Current Liabilities}}$

(b)Quick Ratio, or Acid Test = $\frac{\text{Current assets - Inventory}}{\text{Current liabilities}}$

The current ratio is the most commonplace measure of short-term solvency. If current liabilities are rising faster than current assets, this may be a harbinger of financial difficulty. The quick ratio measures the firm's ability to pay off short-term obligations without relying on the sale of inventory, which is the least liquid of the firm's current assets.

2. Asset Management Ratios

Another group of ratios measures how effectively a firm is managing its assets. One of these is the *total assets utilization ratio*, which measures the utilization or turnover of all of the firm's assets.

(c)Total assets utilization= $\frac{\text{Sales}}{\text{Total assets}}$

3. Debt Management Ratios

These ratios determine the extent to which a firm uses debt financing. If equity, or owner-supplied funds, accounts for only a small portion of a firm's total financing, the risks of the firm are borne mainly by creditors. On the other hand, by raising funds through debt, owners can control the firm with a smaller investment of their own. If the firm returns more on the borrowed funds that it pays in interest, the return on the owner's capital is magnified, or leveraged. An important ratio is the *debt ratio*, which measures the percentage of a firm's total funds provided by creditors.

(d)Debt ratio = $\frac{\text{Total liabilities}}{\text{Total assets}}$

Creditors prefer low debt ratios, whereas it may be advantageous for owners to seek higher debt to leverage their money and earnings. However, a debt ratio that is too high signals trouble repaying loans and too much reliance on borrowed money to pay for the firm's operations.

4. Profitability ratios

Profitability ratios illustrate the combined effects of liquidity, asset management, and debt management on profits. Important profitability ratios measure the *return on total assets (ROA)*, the *return on common equity (ROE)*, or return on stockholders' investments, and the *profit margin on sales*.

(e)Return on Total Assets (ROA) = $\frac{\text{Net profit after taxes}}{\text{Total assets}}$

(f)Return on Equity (ROE) = $\frac{\text{Net profit after taxes}}{\text{Net worth (equity)}}$

(g)Profit Margin = $\frac{\text{Net profit after taxes}}{\text{Sales}}$

5. Market Value ratios

These ratios are an indication of what investors think of the company's past performance and future prospects. The market value ratios (and stock price) will be high if a firm has strong liquidity, asset management, debt management, and profitability ratios. The most widely used market value ratio is the *price/earnings ratio*.

(h)Price/Earnings Ratio = $\frac{\text{Price per share}}{\text{Earnings per share}}$

To evaluate a firm's financial ratios properly, they must be compared to ratios for comparable businesses. Financial ratio data on comparable businesses can be found in Dun and Bradstreet's *Industry Norms & Key Business Ratios* and *Annual Statement Studies* by Robert Morris Associates. Both publications group businesses by standard industry classification codes and provide financial ratio data by standard industry classification code.

Tasks

There are 5 tasks to this case:

1. Examine the 1999 income statements and balance sheets of Qwest and SBC Communications, which can be found by loading the file HAYES_Q.XLS from your data diskette. All data are based on the publicly available 1999 annual reports of these firms.

2. Print out the financial statements so you have a hard copy to work with. Look at them very closely. Since these statements are for telecommunications companies, operating revenue is equivalent to revenue from sales and materials and supplies are equivalent to inventory.

3. Assign range names to the income statement data, the balance sheet data, and the area of the worksheet containing the financial ratios. Create a table of range names below the worksheet. Format the ranges with the income statement and balance sheet data to display a comma with zero decimal places. Format the ranges with the financial ratios, earnings per share, and stock price to display 2 decimal places.

4. Calculate the eight financial ratios outlined above for each company at the end of the financial statements. Print out the ratios for both companies.

5. Write an analysis of both companies in a single paragraph. If possible, find statistics on financial ratios for comparable businesses using either the Dun and Bradstreet or Robert Morris publication. On the basis of the information provided in this case, is each financially sound? Which would make the better investment? Review the financial statements of both companies for any items that might help explain their financial position.

Additional Problems

1. Obtain the current stock price for each of these companies and recalculate the Price/Earnings ratio.

2. Use the Web to search the EDGAR database [http://www.sec.gov/edaux/searches.htm] maintained by the Securities and Exchange Commission. This database contains companies' quarterly financial reports and other documents filed with the SEC. Use this information to recalculate the financial ratios for both companies. Would that make a difference in your decision?

Time Estimates

Expert: 45 minutes
Intermediate: 1 hour
Novice: 1.5 hours

Excel Tutorial For Spreadsheet Case 3

You do not need to use range names for the solution to this case, but they can be used if you wish when you use different formats for different ranges in your worksheet.

Let's name the range in COURSE.XLS with the student grade data GRADES and name the range with the percentages used in the Assumptions section PERCENT.

Naming Ranges

You can name a range the with the **Insert/Name/Define** command. To name the range with the percentages PERCENT, you would do the following: firstly, highlight the three cells B9:B11; then select **Name** from the **Insert** menu; and then select **Define** from the sub-menu. A Define Name dialog box will appear, already containing the percentage range reference. Type the name PERCENT and press the OK Button.

You can alternatively select **Insert/Name/Define** and then select the range once within the dialog box and type the name. Use either procedure to name the range B16:E19 GRADES.

Creating a Table of Range Names

You can document your range names in a table in your worksheet using the **Insert/Name/Paste** command. The table should be placed in an unused portion of the worksheet where it will not overlay any data. The spreadsheet design principles introduced in Chapter 2 suggest placing the range name table below the leftmost portion of the worksheet.

To create the range name table, move to the cell where you want the table to appear and select the **Insert/Name/Paste** command. A dialog box appears, listing the named ranges. The purpose of this dialog box is to paste a selected Name in the formula bar. However, we are using the alternative purpose of the dialog box: pasting a list of range names and their references onto the worksheet. To do this select the Paste List Button. This will result in a worksheet resembling that in Figure 3-4.

Spreadsheet Case 4

Slopeside Condominiums

Problem: Analyze cash flow of a condominium hotel

Management skills: Planning
Deciding

PC skills: Formulas
Spreadsheet control
Reporting

File: SLOPE_Q.XLS

Slopeside Condominiums is a 100-unit condominium complex in north-central Vermont, situated near several major ski areas. It operates as a hotel-condominium, renting out units that owners are not using through a rental office.

In most years, the complex returns a small profit each year to the owners. But there are many expenses involved in running a hotel-condominium--expenditures for electricity, heat, insurance, and maintenance staff, to name a few. At times Slopeside's manager, John Watts, has found that Slopeside has been short of cash for paying the bills.

Slopeside needs to develop a plan for managing its cash flow. In some respects, a business's cash flow is more important than its profit. If a firm can't pay its employees or suppliers or collect from its customers it may not survive long enough to generate a profit.

One way of analyzing and forecasting cash flow is to use a "cash receipts and disbursements" forecast, which estimates the amount of each category of cash receipts and disbursements. This enables businesses to track actual cash movements on a daily, weekly or monthly basis.

Essentially this type of cash flow forecast shows projected cash inflow and outflow and subtracts the outflow from the inflow. From your data diskette load the file SLOPE_Q.XLS, showing the various categories of receipts and disbursements for Slopeside Condominiums and the hotel-condominium's beginning cash on hand. (Beginning cash on hand may be cash in the checking account or other bank accounts.)

Cash inflow or outflow can be determined by subtracting total disbursements from total receipts. This resulting amount (which may be positive or negative) must then be added to cash on hand at the beginning of the period to arrive at an ending cash figure. This ending cash figure in turn becomes the beginning cash figure for the next period.

From past experience, Watts has determined the average occupancy rate for various months of the year. The occupancy rate is highest during the ski season, summer months, and during September, when visitors flock to Vermont to witness the changing fall foliage. Watts can estimate monthly room receipts by multiplying the daily unit rental receipts by the number of days in the month, and the average occupancy rate for that month. The daily unit rental receipts are

$7800 and represent the amount of receipts if all available units were rented. Slopeside also receives $8000 per month in commons fees paid by the owners for insurance and upkeep.

Watts has also come up with formulas for estimating other receipts and disbursements. Some of these are variable, and related to the occupancy rate at the hotel-condominium. Watts has figured that expenditures for telephone usage are approximately 5% of room receipts. Expenditures for room upkeep are approximately 30% of room receipts.

Other expenses are not related to occupancy rates. On an annual basis, Slopeside pays $18,500 for insurance; $275,000 in payroll for the manager, office, cleaning, and maintenance staff; $25,000 for marketing and advertising; and $250,000 for repairs and maintenance. Since furniture and appliances are constantly being damaged, Slopeside pays $6000 per month into a furniture reserve fund. Slopeside also pays between $4400 and $9500 per month for utilities and fuel. Insurance is paid in one installment every May. All other expenditures are paid on a monthly basis.

Tasks

There are 4 tasks in this case:

1. Print out and review SLOPE_Q.XLS.

2. Create an assumptions section of the worksheet to identify factors in your calculations of receipts and disbursements. Make sure formulas reference cells in the assumptions section wherever possible.

3. Calculate the receipts and disbursements for each of the categories on the worksheet for 12 months. Calculate total receipts and total disbursements. The case has been simplified so that all revenue is collected in the month it is generated.

4. Complete the cash flow worksheet by calculating the ending cash for each month and print the results. Write an analysis of the cash flow situation for Slopeside Condominiums. What explains the negative cash flows in certain months? What steps can Slopeside take to eliminate cash flow crises?

Additional Problem

Slopeside has decided to open up a small restaurant for serving breakfast and lunch. Management thinks that receipts from the restaurant would be 15% of room receipts, while expenditures for food and beverages would run about 8% of room receipts. One extra employee would have to be hired at an annual salary of $20,000 to prepare the food, but the current staff could help set up and wait on tables. Modify your worksheet to show the impact on Slopeside's cash flow. Is the restaurant a good idea?

Time Estimates

Expert: 45 minutes
Intermediate: 1.5 hours
Novice: 2 hours

There is no tutorial for this case because it uses spreadsheet software skills introduced in earlier chapters.

Spreadsheet Case 5

Choice Coffee of the Month

Problem:	Develop a breakeven analysis model
Management skills:	Planning Deciding
PC skills:	Graphics Worksheet organization

File: COFFEE_Q.XLS

Choice Coffee of the Month is a newly-formed one-person company that markets choice coffees through a monthly subscription service. For an annual subscription fee, Coffee of the Month will send subscribers a different choice coffee, such as Colombian, Mocha Java or French Dark Roast, every month. Its founder, Martha Staunton, feels this service will appeal to coffee-loving young professional couples who like to experiment with new flavors and ideas.

Martha plans to place small ads in the back of various gourmet magazines, but is unsure of the response rate. Before she invests heavily in advertising, inventory, and office space, Martha needs to know if there is a future in this type of business and at what point it will produce a profit.

This is a classic problem for all businesses: determining what objectives must be met to produce a profit or to minimize losses. What Martha must do is utilize the managerial accounting concept of Breakeven Analysis. Breakeven Analysis establishes the breakeven point, which is the number of units that must be sold to yield no profit and incur no loss. Any units sold beyond the breakeven point will represent profit, and a sales volume below the breakeven point will put the firm at a loss.

In order to perform Breakeven Analysis, a company must examine its operating costs. Some of these costs are fixed and do not change significantly over the range of the operations activity. Variable costs, on the other hand, increase with increasing production, and decrease as production decreases.

In the case of Choice Coffee of the Month, fixed costs are Martha's rent for a one-room office-storage area ($6800 per year) and the costs of her initial advertising campaign ($3000). Martha's variable costs are the cost of purchasing the coffee, shipping costs, and ongoing advertising expenses. Martha has calculated that the cost of purchasing and shipping each pound of coffee is $4.75. Martha plans to sell the coffee on an annual subscription basis at a price of $100.00 per year and there is a pound of coffee shipped each month.

Martha won't be paying herself a salary right away. Until Choice Coffee of the Month starts producing a profit, Martha is planning to live off of her savings, which amount to $10,000.

Once a product's costs have been determined, the contribution margin per unit must be calculated. The contribution margin per unit is the difference between the selling price per unit and the variable costs per unit. (The contribution margin per unit = average selling price per unit -

variable costs per unit.) Once the contribution margin per unit has been determined, one can then calculate the breakeven point.

In a company such as Choice Coffee of the Month, which only produces one product, the formula for the breakeven point would be calculated by dividing the total fixed costs by the contribution margin per unit.

Often the best way to display the results of breakeven analysis is in graphic form. It is also useful to use breakeven analysis to generate pro forma income statements which convert unit data to dollars and display projected sales revenue.

Load the data file COFFEE_Q.XLS from your data diskette.

Tasks

There are 5 tasks in this case:

1. Create a worksheet that displays the total fixed cost, variable cost per unit, and average sale price for Choice Coffee of the Month and calculates the contribution margin per unit and the breakeven point. The breakeven point should be based on number of subscriptions Martha needs to sell before she can start earning a profit.

2. Use your results to generate pro forma income statements using the framework supplied on the data file. There should be two projections of sales and income below the breakeven point and two above it. There should also be a projection of sales and income right at the breakeven point. The income data below the breakeven point should reflect zero sales and sales at half of the breakeven units. The income data at the breakeven point should reflect sales at 1 times the breakeven units. The income data above the breakeven point should reflect sales at 1.5 and 2 times the breakeven units.

3. Create a line chart (graph) to display the most important data from the pro forma income statements and the breakeven point. The X-axis of the chart should display the range displaying units sold in your pro forma income statements. The first data series should display fixed cost (which will be constant). The second data series should display total cost figures. The third data series should display revenue figures. Give your chart (graph) a title and supply titles for the X and Y axes. Supply legends for all of the data series. The point on the chart where the data lines for total cost and revenue intersect is the breakeven point.

4.Be sure to name and save your graph as well as your worksheet. Print the graph and worksheet.

5.Examine your output. Write a one-paragraph statement analyzing the results of the breakeven analysis. Is Choice Coffee of the Month a worthwhile business venture for Martha?

Additional Problem

1. Martha has been told that packaging material costs will rise 10% in a few weeks. Packaging material costs account for approximately 5% of the baking and packaging cost for each package of muffins. What impact will this have on Martha's breakeven analysis?

Time Estimates

Expert: 45 minutes
Intermediate: 1 hour
Novice: 2 hours

Excel Tutorial For Spreadsheet Case 5

This case draws upon all of the skills acquired in previous Spreadsheet Cases plus new skills for creating and printing graphs, or Charts as they are known in Excel, with spreadsheet software. You will need to use COURSE.XLS again for this tutorial.

Creating Charts with Excel

Excel provides an excellent facility for generating charts to graphically display data on your worksheets. The facility is known as the ChartWizard. The idea behind Wizards is that the program helps you through a complicated process by splitting it into easy steps with clear instructions. The ChartWizard helps you through the process of creating a Chart in Excel in five easy steps.

Charts in Excel can be stored in separate chartsheets or be Embedded in worksheets. With the Chart embedded in the worksheet, you can instantly observe the effects of changes of the data in the worksheet, and create attractive documents with the chart accompanying the original data. The steps to achieve each of these graphs are identical.

The ChartWizard can be invoked in two ways: by selecting **Insert/Chart** from the menu or by pressing the ChartWizard Button on the Standard Toolbar. The ChartWizard Button will only embed a chart in a worksheet. The menu item will give you the choice of embedding the chart in the worksheet or creating a new chartsheet.

We will create a separate chartsheet for our chart. The information we want to graph is the grades of each assessment task for every student. It is easier to select the range containing the data before selecting the ChartWizard. However, currently the data headings are separated from the values in the worksheet by a row containing the "=" symbol, in Row 15. Since we want to include the headings as Labels in the chart, it would be easier if we deleted Row 15 and had a undivided range. Select Row 15 and select the **Edit/Delete** command to delete Row 15.

Select the range A14:E18. This range contains the values representing the Students' Grades and also the Students' Names and the Assessment Tasks. Now, select **Insert/Chart** from the menu.

Excel Chart Types

The **Area Chart** shows each data series as a shaded area, each added onto the previous area.

The **Bar Chart** shows the values as solid horizontal bars of differing lengths. This type is ideally suited to categorized data.

The **Column Chart** is similar to the Bar Chart except the bars are vertically aligned. Again, this type is ideally suited to categorized data.

The **Line Chart** show trends and values, typically over a time horizon at even intervals, representing each series as points connected by a line.

The **Pie Chart** shows values as a proportion of a whole or total. The values are represented as a slice of a circular pie. This is used when there is only a single data series.

The **Doughnut Chart** shows values as a proportion of a whole or total, similar to the Pie Chart, except more than a single data series can be represented. Each data series is shown as a concentric circle.

The **Radar Chart** shows each data series as a ring arranged on axes radiating from a central point. Each axis represents a data category. This chart type is suited for comparing data series against distinct categories.

The **XY (Scatter) Chart** shows the degree of a relationship between the numeric values on both the X and Y axes, for several data series. This chart type is useful since it represents data with uneven intervals on the axes.

The **3-D Area Chart**, the **3-D Bar Chart**, the **3-D Column Chart**, the **3-D Line Chart** and the **3-D Pie Chart** are three dimensional representations of the above chart types. These types can be attractive and sometimes more effective in conveying the information contained in the numbers. For example, in a 3-D Perspective Column Chart the values are represented by the vertical (Z) axis, the data categories are represented by the horizontal (X) axis, and the data series are represented by the depth (Y) axis.

The **3-D Surface Chart Type** represents the data as a continuous sheet of data across three dimensions. The three axes usually represent continuous values rather than categories, which permits a large amount of data to be portrayed clearly.

Instructions for Windows 95 Users

You should now be faced with the ChartWizard's first dialog box. This dialog box, titled "ChartWizard-Step 1 of 4" is asking for the chart type you want to display the data. Choose the Line type of chart by selecting it with the mouse or with the cursor keys. The different chart types are defined on the previous page. You can also select a specific format for your line chart from the samples in the Chart Sub-type section of the dialog box. When you have selected Line type and the format for your Line chart, press the "Next >" Button or the ENTER key.

The second dialog box appears. It asks for the range containing the data for graphing. Since we have already selected the range, it appears in this dialog box. This stage permits you to redefine the range if you wish. (This dialog box also asks you to specify whether the individual data series are in the rows or in the columns. The data series we want to graph should be specified as in columns.) When you are satisfied with the specified range, press the "Next >" Button or press ENTER.

The third dialog box deals with Chart Options. They include Titles, Axes, Gridlines, Legend, Data Labels, and Data Table. Make sure the Titles tab is selected so that you can add a Chart Title and Axis Titles. Opt to have the Title tab selected. Enter the Chart Title "Student Grades", the X-Axis Title "Assessment Types" and the Y-Axis Title "Grade". As you enter these titles, the sample chart will incorporate them. After you have specified these titles, press the “Next.” Button or press ENTER.

The fourth and final dialog box asks for the Chart Location. Here, you can specify how you want to store your chart—“As object in” and “As new sheet”. “As an object in” means that you want to store the chart as an embedded object on the current sheet . “As new sheet”means that you want to store the chart on a separate sheet. Since we want the chart on a separate sheet, select the "As new sheet" option. When you have finished, select the "Finish" Button.

The ChartWizard has now finished. Notice the Chart1 tab at the bottom of the screen. This is where the newly created chart is stored. To move between the chartsheet and the worksheet, simply press the respective tab at the bottom of the screen.

Instructions for Windows 3.1 Users

If you are using Excel with Windows 3.1, this will show two options: On This Sheet; As New Sheet. Since we want a separate sheet, select the "As New Sheet" option.

You should now be faced with the ChartWizard's first dialog box. This dialog box, titled "ChartWizard - Step 1 of 5", is asking for the range containing the data for graphing. Since we have already selected the range, it appears in this dialog box. This stage permits you to redefine the range if you wish. When you are satisfied with the specified range, press the "Next >" Button or press ENTER.

The second dialog box then appears. This second step is asking for the chart type you want to display the data. Choose the Line type of chart by selecting it with the mouse or with the cursor keys. The different chart types were defied earlier. When you have selected it, press the "Next >" Button or the ENTER key.

The third dialog box requires you to select a format of the chart type selected. Since you selected the Line chart in the previous step, you are now presented with ten different Line formats. If you had selected a different chart type, different formats would be offered in this step. Select format type 1 and press the "Next >" Button.

The fourth dialog box asks you to specify whether the individual data series are in the rows or in the columns, and how many rows and columns the Axis Labels take. Excel displays a sample chart of your data. Notice that currently the data series are assumed to be in rows;

that is, each student's grades are a data series, with each type of assessment representing an X-Axis category. This is the layout we want for the Student Grades. Excel has assumed the first row of the selection on the worksheet contains the X-Axis Labels and the first column of the selection contains the Text for the Legend. These settings are also what we want for the chart. Experiment by adjusting the settings, observing the effects on the sample chart. When you are satisfied with the settings, press the "Next >" Button.

The fifth and final dialog box asks you whether you want a Legend and permits you to add a Chart Title and Axis Titles. Opt to have the Legend. Enter the Chart Title "Student Grades", the X-Axis Title "Assessment Types" and the Y-Axis Title "Grade". As you enter these titles, the sample chart will incorporate them. When you have finished, select the "Finish" Button.

The ChartWizard has now finished. Notice the Chart1 tab at the bottom of the screen. This is where the newly created chart is stored. To move between the chartsheet and the worksheet, simply press the respective tab at the bottom of the screen.

Graph Formatting

Excel provides extensive features for formatting of objects within charts. Although the ChartWizard has produced an attractive chart, further enhancements can be made to improve certain aspects of it. You can select virtually any object in an Excel chart and adjust its features. You can select objects in a chart using the cursor keys to step through each of the object that can be changed, or simply select the object with the mouse.

Objects that can be selected are: Axis (X and Y), Axis Title, Chart Title, Chart Area, Plot Area, Data Series, Data Point, Legend, Legend Key, Legend Entry and <Chart Type> Group.

Once the object is selected you can choose **Format/Selected <Object>** from the menu or double click on the object. This will produce the **Format <Object>** dialog box containing different tabs for adjusting different aspects of the object.

For example, if you selected the Y-Axis by clicking on it (a black square will appear at each end of the axis when it is selected) and then selected **Format/Selected Axis ...** from the menu, you will be presented with a dialog box with tabs for Patterns, Scale, Font, Number and Alignment. These tabs can also be applied to other objects in the chart but in this context they refer to the Y-Axis. For example, the Patterns tab contains settings for the axis thickness, style, colour, tick type, tick location and a sample. The Patterns tab for other objects would have different settings. Other tabs, such as Scale, refer to this type of object alone.

Select the Scale tab. The settings here permit you to change the minimum value, the maximum value, the major interval value, and the minor interval value of the Axis. Other settings also let you specify where the X-Axis crosses, whether the scale is logarithmic, whether the values appear in reverse order and whether the X-Axis crosses at the maximum value.

Currently the graph has all the data points congregating at the top of the chart. It would be preferable for the data to be spread more evenly up the chart. In order to do this we would adjust the minimum value of the Y-Axis to 60. If you have not already done so, select the Y-Axis and choose **Format/Selected Axis ...** from the menu. Now select the Scale tab. To

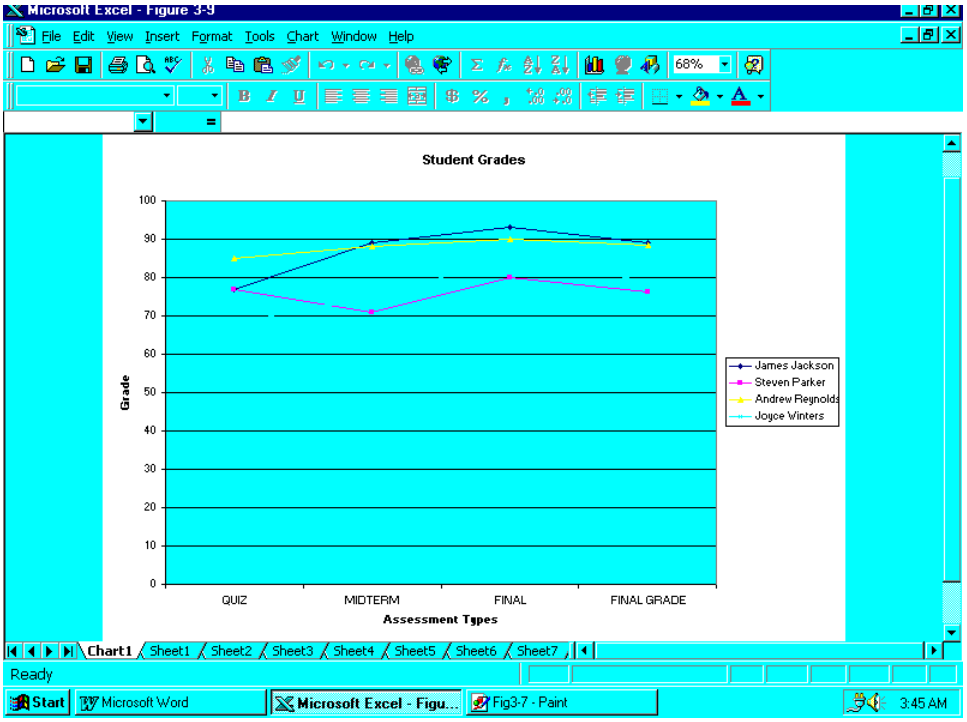
change the minimum value, first click the Auto Check box next to Minimum so a cross does not appear in the box. This permits manual entry of values. Now enter 60 in the Minimum value box and select the OK Button. Your chart should now resemble that in Figure 3-5.

The other tabs under various objects are:

Font - changing the appearance of text (size, font, style etc.)
Number - changing the presentation of values
Alignment - changing the arrangement and orientation of objects
Placement - changing the location of the Legend (Bottom, Corner, Top, Right, Left)
Y Error Bars - Lets you arrange error bars on data points
Axis - Lets you distinguish between primary and secondary axes
Data Labels - Lets you attach labels to data points
Names and Values - Lets you redefine the worksheet cells that contain the names and values of the current data series
X Values - Lets you redefine the worksheet cells that contain the X values
Subtype - Lets you define cumulative and percentage cumulative charts
Series Order - Lets you change the order of the data series
Options - further options for the current object, not included in other tabs

Most Excel chart objects can be changed merely by selecting it and double clicking to bring up object settings. Title text can be changed simply by selecting the object and placing the cursor where you want to insert or replace text.

Figure 3-5



Embedding a Chart in a Worksheet

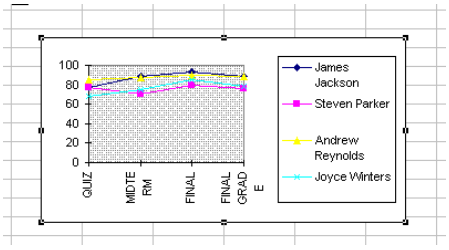
All the features discussed for a chart in a separate chartsheet apply equally to a chart embedded in a worksheet.

Users of Windows 95 could embed a chart by selecting "As object in" for their Chart Location when using ChartWizard dialog boxes.

If users of Windows 3.1 selected **Insert/Chart/On This Sheet** or used the ChartWizard Button, they would be required to define an area on the worksheet to contain the chart. Defining the chart area requires you to specify the diagonally opposite corners of the rectangular area. Do this by selecting the point for the first corner by locating the mouse cursor and depressing the left button. Select the diagonally opposite corner by dragging the mouse, keeping the button down. As this is done, you should observe the area growing and changing as you move the mouse. Release the button when you are satisfied with the defined area. ChartWizard's first dialog box will now appear.

At the completion of the ChartWizard users of both Windows 95 and Windows 3.1 will have the chart embedded in the worksheet. A selected embedded chart is identified by a thin border with black squares, known as handles, which permit you to resize the chart size. A selected chart can be seen in Figure 3-6.

Figure 3-6



To select an embedded chart, simply press it once with the mouse cursor. To resize the chart, grab one of the eight handles (on either the selected or activated chart) and drag it in the desired direction. To move the chart grab the selected chart anywhere inside the border and drag to the new location. To delete the selected chart, simply press the DEL or DELETE key.

Printing and Saving a Chart

Excel chartsheets can be printed in exactly the same way as Excel worksheets can, as described in the Tutorial for Spreadsheet Case 1. The **File/Print Preview** and **File/Page Setup** operations apply in the same way as for a worksheet also.

A chartsheet will be saved when the workbook it resides in is saved. Similarly, an embedded chart will be saved when the worksheet containing it is saved, described in the Tutorial for Spreadsheet Case 1.

Spreadsheet Case 6

Paragon Computer Office Equipment Inc.

Problem: Develop an inventory control system

Management skills: Organizing
Controlling

PC skills: Spread sheet control
Logical functions

File: PARA_Q.XLS

Paragon Computer Office Equipment Corporation, located in Easton, Pennsylvania, manufactures computer-related office equipment, such as computer tables, printer tables, and monitor stands. Paragon sells this furniture to discount and retail department store chains, which in turn market it under their own brand name. Paragon buys prefabricated components, such as legs and tops for computer tables, from other vendors and then assembles them into finished office computer furniture.

The computer office supply business is highly competitive. Paragon has acquired an impressive roster of clients by producing well-crafted computer office furniture within the budgetary limits specified by clients.

Because the company is fairly new, Harold Moyer, the production manager, has been tracking the components by hand using a ledger sheet. This process is time consuming and errors have, at times, resulted in missed production deadlines due to inadequate supplies of furniture components. In response, Harold has increased the level of component safety stock in order to avoid running out in the future. But this raises operating costs, and Paragon needs to keep these down in order to remain competitive. Harold has decided to implement his own inventory control system using a PC and spreadsheet software.

A good inventory control system will maintain an inventory level that is neither overstocked nor understocked to ensure most efficient utilization of funds. It will match existing inventory levels against desired levels so that understocked items can be reordered.

There are two basic models for accomplishing this. One is to use a *reorder level system*, which merely makes sure that required items are ordered with sufficient lead time to arrive when they are needed in the production process. The other is to use a system that determines the least expensive quantity to order, or most economic quantity. This approach is based on the *economic order quantity model*, which strikes a balance between carrying costs, such as taxes and insurance, and procurement costs, such as ordering, shipping, and receiving costs.

Ordering in large quantities reduces procurement costs but raises carrying costs. The *economic order quantity* represents the number of units where procurement costs equal carrying costs. The exact size of the economic order quantity is dependent upon the estimated amount of the product needed each year, its unit cost, the fixed cost of placing and receiving an order for the

item, and the carrying cost for the item in inventory, expressed as a percentage of inventory value. The formula for calculating an item's economic order quantity is:

$$EOQ = \frac{2\,FU}{CP}$$

where

EOQ = the item's economic order quantity
F = the fixed cost of ordering the item
U = the amount of the product needed each year
C = the item's carrying cost, expressed as a percentage of inventory value
P = the item's unit cost

The calculation of the economic order quantity often results in a fractional amount that must be rounded to the next whole number to determine the economic order quantity.

Load the data file PARA_Q.XLS from your data diskette. This file contains a list of the computer office furniture components stored by Paragon Computer Office Equipment, balance on hand, balance on order, unit cost per item, estimated annual usage per item, and the order point. The order point is the number of units of an item in inventory that triggers the decision to order more items. There is usually a lead time period (say, of two weeks) between the time an order is placed and when it is actually fulfilled. Having some items in inventory while reordering reduces the possibility of a stockout.

Order cost is assumed to be a fixed cost of $90.00 for all items in this problem.

Inventory carrying cost is assumed to be 21% of inventory value for all items in this problem.

Tasks

There are 4 tasks in this case:

1. Assign an area to hold assumptions for this model at the upper lefthand portion of the worksheet.

2. Expand the worksheet to track:
(a) Balance available which can be calculated by adding balance on hand + balance on order.
(b) Order quantity for those items in need of reordering. If the balance available is less than the order point, calculate the economic order quantity. If the balance available is greater than the order point, put a zero in the column for order quantity. Format the Order Quantity column to comma format with zero decimal places.

3. Develop a method to identify any stock items that need reordering on the worksheet. (Hint: One way is to use an asterisk to flag items that need reordering.)

4. Write a short paragraph suggesting some enhancements to this application to make it a better management tool.

Time Estimates

Expert: 1 hour
Intermediate: 1.5 hours
Novice: 2 hours

Excel Tutorial For Spreadsheet Case 6

This case draws upon all of the skills acquired in previous Spreadsheet Cases plus new skills for using the logical functions of your spreadsheet software and the =SQRT function. You will need to use COURSE.XLS again for this tutorial.

Excel includes a set of logical functions which allow the software to perform conditional tests and evaluate a condition in your worksheet. Depending on whether the condition is true or false, different values will be returned to cells.

The most important conditional function in Excel is =IF. The =IF function allows you to test one or more conditions in your worksheet and perform different tasks, depending on the outcome of the test. The form for the =IF function is:

=IF(condition, action if true, action if false)

This tests the "condition" to determine if specific results or cell contents are true or false. If the result of the test is true, the "action if true" portion contains specific instructions to execute. If the result is false, the "action if false" portion contains another set of instructions to execute. The instructions to be executed can return cell contents that are labels as well as values.

To perform conditional tests, the =IF function and other conditional functions require logical operators. These operators help establish the relationship between two numbers, strings or cell references.

Logical Operators

Operator	*Meaning*
=	Equal
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
<>	Not equal

To establish relationships between two or more conditional tests, Excel provides Logical Functions.

Logical Functions	
AND(*logical1*, *logical2*, ...)	Returns TRUE if each *logical* condition is true; returns FALSE otherwise
OR(*logical1*, *logical2*, ...)	Returns TRUE if any *logical* condition is true; returns FALSE otherwise
NOT(*logical*)	Returns TRUE if *logical* is FALSE; returns FALSE otherwise
TRUE()	Returns TRUE always
FALSE()	Returns FALSE always

The logical functions NOT, AND and OR contain conditional tests to result in a single TRUE or FALSE. The following are examples of logical statements using =IF ,the logical operators and the logical functions:

=IF(A5>20,B5,0) means that if the value in cell A5 is greater than 20, use the value in cell B5. Otherwise, assign the number zero.

=IF(AND(B11<>0,G11=1),10,0) means that if the value in cell B11 is not equal to zero and the value in cell G11 is equal to 1, assign the number 10. Otherwise, assign the number 0.

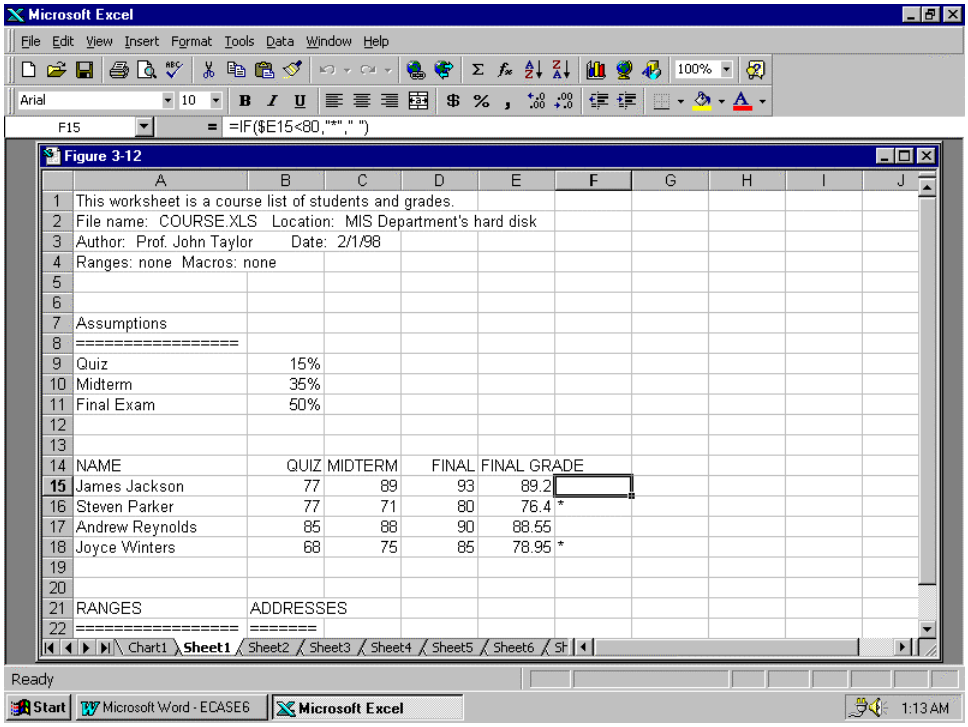
=IF(OR(E13="Profit",F15>=G15),"Surplus","Deficit") means that if either cell E13 contains the label "Profit" or the contents of cell F15 are greater or equal to the contents of cell G15, assign the label "Surplus". Otherwise, assign the label "Deficit".

The second and third examples show how logical functions can be used to have more than a single condition as the =IF conditional test.

For your student roster you can develop a conditional test to print an asterisk (*) after the name of any student whose final grade is less than 80. In cell F15, add a formula that will examine the student's final grade. If the grade is less than 80, an asterisk will be placed in the cell next to the final grade. If the grade is greater than or equal to 80, a character string consisting of a blank space will be placed in the cell. The formula for this would be:
=IF($E15<80,"*"," ")

Copy this formula into range F17:F19. Observe on your screen and in Figure 3-7 that the final grades for students Parker and Winters will be followed by an asterisk. The final grades for the other students will be left blank.

Figure 3-7



	A	B	C	D	E	F	G	H	I	J
1	This worksheet is a course list of students and grades.									
2	File name: COURSE.XLS Location: MIS Department's hard disk									
3	Author: Prof. John Taylor Date: 2/1/98									
4	Ranges: none Macros: none									
5										
6										
7	Assumptions									
8	==========									
9	Quiz	15%								
10	Midterm	35%								
11	Final Exam	50%								
12										
13										
14	NAME	QUIZ	MIDTERM	FINAL	FINAL GRADE					
15	James Jackson	77	89	93	89.2					
16	Steven Parker	77	71	80	76.4	*				
17	Andrew Reynolds	85	88	90	88.55					
18	Joyce Winters	68	75	85	78.95	*				
19										
20										
21	RANGES	ADDRESSES								
22	==========	=======								

The =SQRT Function

The =SQRT function is one of a series of functions that perform mathematical, statistical and trigonometric operations. The =SQRT function calculates the square root of a positive number. The form of this function is:

=SQRT(number or cell reference)

For example the square root of the average of James Jackson's quiz and midterm grades would be =SQRT((B15+C15)/2)

You do not need to save COURSE.XLS with the modifications you made during this tutorial session.

FunctionWizard

Excel provides a facility to make retrieving and entering functions easier, called FunctionWizard. It can be activated by selecting **Insert/Function** from the menu, or by selecting the FunctionWizard Button on the Standard Toolbar:

Press the FunctionWizard Button now. The first dialog box asks you which function you want to use. The left window contains all the function categories and the right window contains all the functions in the category selected in the left window. For example, if you wanted the tangent function, select "Math & Trig" from the Category window and then scroll down the list in the Function Name window until TAN appears. Select TAN and then select the "Next >" Button.

The second and final dialog box permits you to enter the parameters for the formula selected. If the parameter is compulsory the word "required" appears next to the parameter box. Excel permits you to enter additional functions as the parameters of the original function and the FunctionWizard Button is available to enter a further function. Entering functions as parameters for further functions is known as nesting. Excel permits seven levels of nesting. When you have satisfied the parameter requirements, select the Finish Button.

Spreadsheet Case 7

Lightning Communications Outlet

Problem:	Determine the relationship between rebates and level of sales
Management skills:	Controlling Planning
PC skills:	Regression analysis Graphics
File:	REBATE_Q.XLS

Lightning Communications Outlet is a distributor of radios, calculators, paging devices, and telephones in the upper Midwest. Telephone sales have recently been falling, so senior management decided to experiment with using cash rebates to stimulate sales. To test the effectiveness of rebates, Lightning Communications offered different rebate amounts for phones retailing for $99.00 in each of its 16 retail outlets. The results were so varied that Robert Whittaker, the firm's president, is not sure that the rebates were effective in stimulating sales.

Larry Gorman, Lightning Communication's director of sales and marketing, believes otherwise. He is convinced that even though sales fluctuated at Lightning's various stores, the sales increases were not random. The greater the rebate, the higher the sales increase. Gorman needs a way of showing that the amount of cash is directly related to the percentage of increase in sales.

Load the data file REBATE_Q.XLS from your data diskette. This file shows the data on the amount of rebate and the corresponding increase in sales in 16 stores. Gorman wants to use this data to determine the relationship between rebates and increase in sales to help management determine whether rebates should be used to increase sales in the future. He believes the results will be more convincing if they are displayed graphically.

Gorman can do this by performing a regression analysis, which is a statistical method for measuring the relationship between two or more variables. If the relationship is only between two variables, the method is called simple regression. If the relationship is between more than two variables, the method is called multiple regression. This case has been simplified for instructional purposes so that only two variables, amount of rebate, and percent increase in sales, will be analyzed.

A regression analysis results in an equation that describes the behavior of one dependent variable in terms of other variables, called independent variables. In this case, there is only one independent variable, amount of rebate, and one dependent variable, percent increase in sales. The regression analysis also produces statistics that measure the strength of the relationship between the independent and dependent variables. Regression analysis can be visualized as a way of drawing the "best line" through a series of data points.

You can then use the regression equation to forecast the impact on sales of various rebates. Some popular business applications for regression analysis are determining the

relationship between a product's price and cost of production or determining the relationship between incidence of defects in manufactured goods and production volume.

Tasks

There are 4 tasks in this case:

1. Carefully examine the template for the data file, REBATE_Q.XLS which you have just loaded. Note that percentages are entered and displayed as 3.2 rather than .032 in order to appear appropriately in the graph you will construct later. Use the regression analysis commands of your spreadsheet software to perform a simple regression analysis. Select the amount of rebate range as the independent variable and the percent increase in sales range as the dependent variable. Specify for your Output Range an unused area of your worksheet because the output will be written over any existing cell contents.

2. Use the results of the regression analysis to construct a Regression Line. The Regression Line can be calculated by multiplying the value of the X Coefficient by the each value of the independent variable (amount of rebate) and then adding the value of the constant.

3. In the regression output area, R squared values range from 0 to 1. An R squared of 0 means the relationship between the independent and dependent variables is entirely random, whereas an R squared of 1 indicates that the dependent variable moves in a lockstep relationship with the independent variable. The X Coefficient tells how much the dependent variable changes for each unit of change in the independent variable. The standard error of the X Coefficient indicates how precise the relationship is between independent and dependent variables. If the error is great, the X Coefficient isn't a good predictor of the value of the dependent variable, but if the standard error is small, the X Coefficient is a good predictor. To judge whether the error is great or small, you must add a formula for the t-statistic. The t-statistic can be calculated by dividing the X Coefficient by the standard error of the coefficient. A t-statistic of 2 means that one can be 95% certain that the X-coefficient is an accurate predictor for the behavior of the dependent variable. A lower t-statistic indicates that the independent variable may not be an accurate predictor.

Print out the worksheet and the Regression Output. In a few sentences state whether you believe that increasing rebates will increase sales and why.

4. To demonstrate the relationship between rebates and sales visually, construct an XY type graph showing the independent variable (amount of rebate) on the X-axis, the dependent variable (percent increase in sales) on the Y-axis, and the Regression Line. Use the X data range for the independent variable, the A data range for the dependent variable and the B data range for the Regression Line. Add appropriate legends and graph titles. Print out this graph to show to management.

Additional Problem

Management wants to be able to experiment with various rebate amounts and determine the effect on sales. In other words, if management asks how much of sales increase can be expected if all stores offer a $10 rebate, they should be able to receive a solid answer. Hint: Create an output area where you can display various rebate amounts and the corresponding increase in sales for each amount. You will need to use the formula you developed to calculate the Regression Line. This is necessary because even though there may be one data point for a $10 rebate, it is an isolated observation. One can be much more confident in predicting the relationship between amount of rebate and sales increases by using the regression formula.

Time Estimates

Expert: 45 minutes
Intermediate: 1.5 hours
Novice: 2.5 hours

Excel Tutorial For Spreadsheet Case 7

This case expands on graphics and other spreadsheet skills acquired previously and introduces the use of a Excel command for regression analysis.

You can use your sample student roster (COURSE.XLS) if you expand it to include a column for the students' ages. (If you saved COURSE.XLS with the formula to place asterisks in column F, you can erase range F15:F18 and then enter student age data in this column. You should also erase the range name table.)

The professor and school administration want to see if there is any correlation between student age and maturity and academic performance. Let's assign the age of 25 to James Jackson, 22 to Andrew Reynolds, 23 to Steven Parker, and 19 to Joyce Winters. You can place ages in corresponding cells in range F15:F18. Enter the column heading AGE in cell F14. You will need to widen column E to 12 positions. You may want to reduce the widths of columns B, D and F so that you can display columns A-H on your worksheet screen.

The number of students on this worksheet is actually too small a sample to be statistically valid in real life, but it will illustrate the concept of regression analysis and the Excel **Regression** Analysis Tool. Excel provides a variety of Analysis Tools that can be accessed through **Tools/Data Analysis**.

If **Data Analysis** does not appear in the **Tools** menu, the Analysis ToolPak add-in has not been loaded. An add-in is a file which provides additional functions, commands and menus. Select **Tools/Add-Ins**. In the Add-Ins dialog box, choose Analysis ToolPak and select the OK Button. Now select **Tools/Data Analysis**.

Now choose Regression from the Data Analysis dialog box by moving down the list using the scroll bar, highlighting "Regression"and pressing OK. To determine the relationship

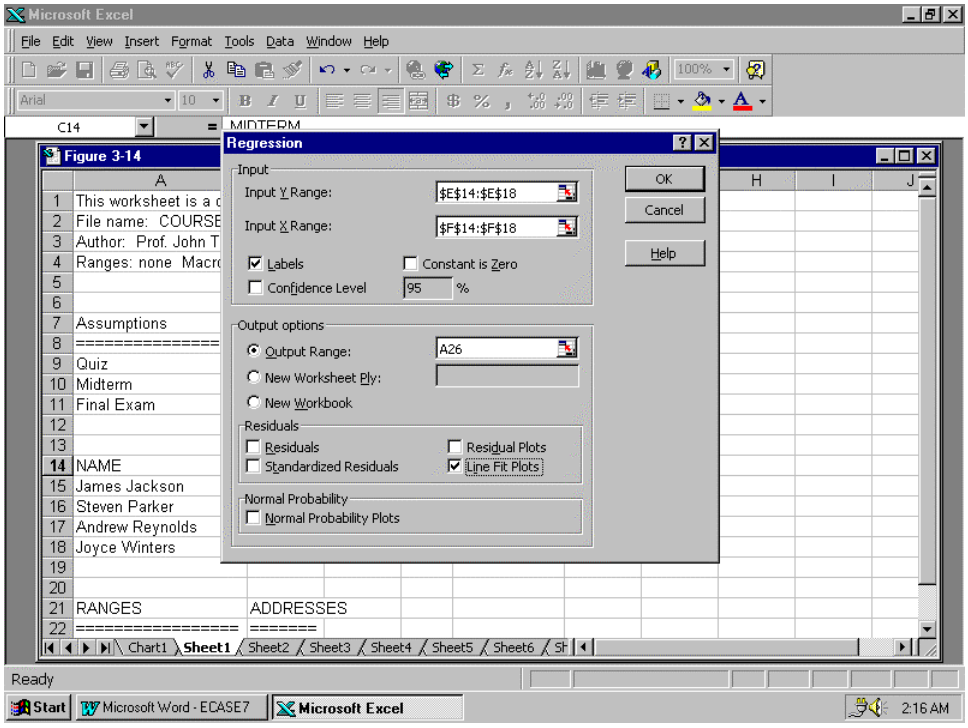
between students' ages and final grades, you must make age the independent variable and final grade the dependent variable.

The Regression dialog box (see Figure 3-8) requires that you specify the ranges that holds the dependent values and the independent values.

At the **Input Y-range** setting specify your dependent variable. In the window provided enter the range E14:E18 (the range for final grades). This dialog box permits you to select ranges on the worksheet. Select the Final Grades range using your mouse, or type the range E14:E18.

At the **Input X-Range** setting specify one or more independent variables. In the window provided enter the range F14:F18 (the range for students' ages). As with the Y-range, ranges on the worksheet can be selected with the mouse. Select the Students' Ages range or type the range F14:F18.

Figure 3-8



Check the **Labels** Check Box to indicate the selected ranges include the Labels in the first row. Another input setting available is the **Constant is zero** Check Box which lets the user force the Y intercept to zero. We will not use this setting for this problem. The **Confidence Level** Check Box can be selected if you want confidence intervals applied to the regression in addition to the default 95% levels.

The remaining settings concern the output options of the regression analysis. Three output options are available: firstly, **Output Range** which deposits the output on the same sheet and asks for the upper-left cell reference of the output range; secondly, **New Worksheet Ply** which creates a new worksheet in the same workbook and (optionally) asks you to name the new worksheet; and thirdly, **New Workbook** which creates an entirely new workbook and places the output in cell A1 of its first worksheet. For our purposes, select **Output Range** and place the cell reference A26 in the available window. Generally it is wise to allocate a worksheet area that is clear.

Excel provides four options for residual values: **Residuals**, **Standardized Residuals**, **Residual Plots**, and **Line Fit Plots**. Residuals are the differences between the actual values and predicted values using the regression coefficients for the same dependent values. The plots will be embedded charts in the worksheet where the output tables will appear. Choose the Line Fit Plots setting to achieve a comparison between the actual values and the predicted values.

The final option is for **Normal Probability Plots**. Select the OK Button to start the regression calculations.

The output tables (see Figure 3-9) contain: Regression Statistics, ANalysis Of VAriance (ANOVA) table, and Regression Coefficients. If selected, Residual Tables are also delivered. The Regression Coefficients are the values which permit you to construct the Regression line. For example, the Y-intercept of the regression you created is 40.53 and the gradient is 1.92. If you placed the following formula in cell G15, you would receive the predicted Student Grade for the age in cell F15:

```
=$B$42+F15*$B$43
```

A set of values to construct a line plot of the predicted values could then be produced. The chart could be created in a way described in the Tutorial for Spreadsheet Case 5. Excel's Regression Tool creates such a chart for you. If you selected the Line Fit Plots Check Box in the Regression dialog box, you would have a similar chart to that in Figure 3-10. We have simply adjusted the X-Axis scale to amplify the line.

You may save your worksheet, but the changes you made to COURSE.XLS for the regression analysis will not be utilized by subsequent cases.

Figure 3-9

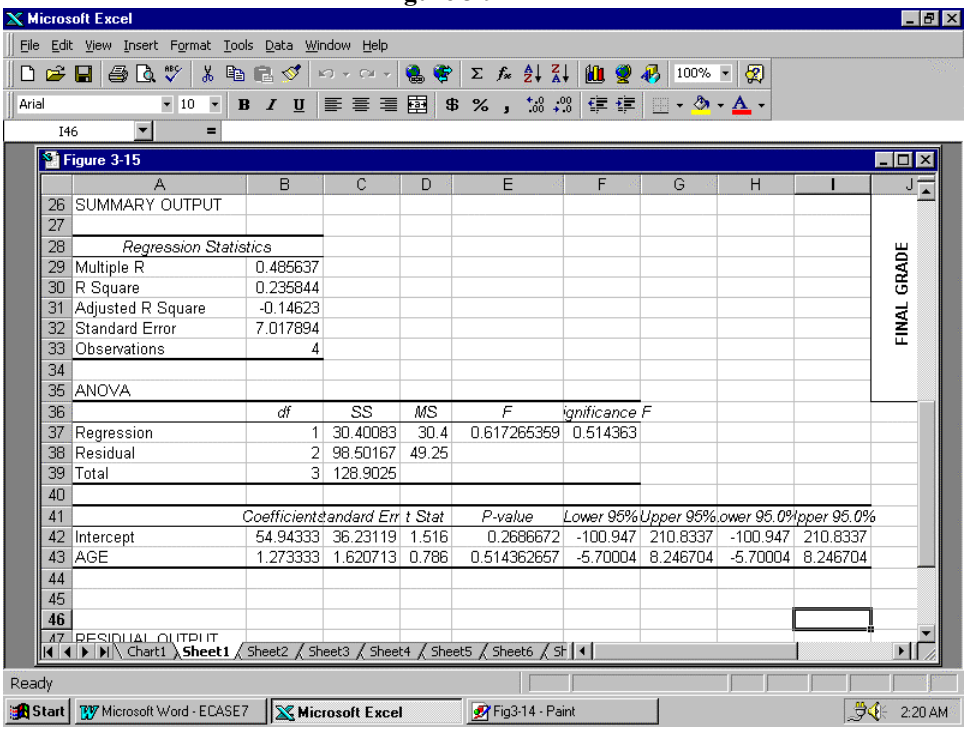
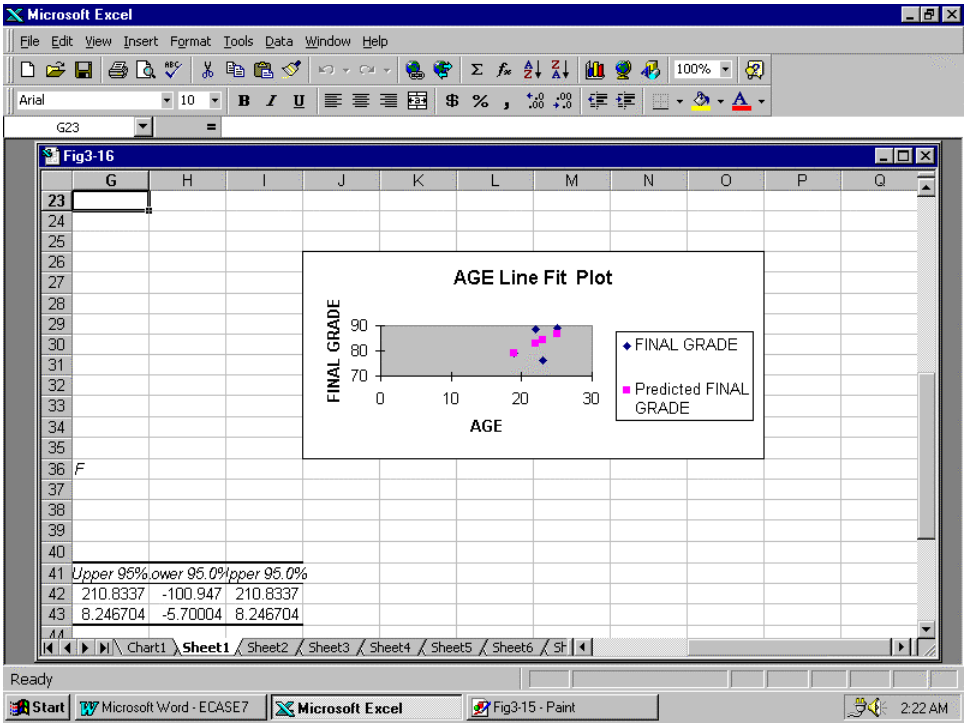


Figure 3-10



Spreadsheet Case 8

Raymond Paint Company

Problem:	Develop an accounts receivable system
Management skills:	Organize Control
PC skills:	Date functions Database querying Logical functions
File:	RAY_Q.XLS

Raymond Paint Company has been the major wholesale and retail supplier of paint and wallpaper for the Cincinnati metropolitan area since 1930. Its customers include contractors, department stores, and local families. Raymond employs 15 people, including an accountant, interior decorator, sales manager, warehouse workers, and travelling sales representatives. Gross revenues total $8,000,000 annually.

Raymond has remained essentially a family business. The original owners, Fred and Verna, have retired, and their son, Carl, has taken over. Carl is concerned that firm is not generating as much revenue as it should to pay his monthly salary. He suspects this is because too many bills are outstanding. His parents ran the business on a largely goodwill basis. All their accounting records were maintained manually, and they would remember who had not paid up. They shied away from calling late payers about their bills if they were "old timers."

In the past Raymond's suppliers were quite lenient about repayment. Now they are much more demanding of small firms, since their primary business is the large discount and home supply chains. The large manufacturers will offer Carl a 2% discount for payment within 10 days, but have threatened to stop supplying him if he does not pay within 30 days. Carl needs to collect his own bills faster in order to remain in a business dominated by giants.

Carl would like his accounts receivable file automated and organized so that he can easily locate late payers. Load the data file RAY_Q.XLS from your data diskette to see the outline of the accounts receivable file. It contains a sample of Raymond's accounts receivable list as of April 1, 1999. Carl would like to type in the date (today's date) whenever he accesses the file and have the days outstanding for each account automatically calculated. He would also like to produce an aging report that automatically classifies the records according to 4 categories of lateness: current (30 days or less), over 30 days, over 60 days, and over 90 days.

Carl would also like some reporting mechanism to identify late payers so that he can phone them and expedite reimbursement. The accounts receivable file should be sorted first by largest number of days outstanding to smallest and then by highest invoice balance to lowest. A listing of customers more than 60 days late with outstanding balances over $400 would also be useful.

Tasks

There are 6 tasks to this problem:

1. Calculate the number of days each invoice has been outstanding. This is the difference between the transaction date and today's date.

2. Classify the invoices using formulas into the four categories shown on the worksheet. Hint: you can use logical functions to classify the invoices.

3. Sort the database in descending order using the number of days outstanding as the primary sort key and the invoice balance as the secondary sort key. Print the sorted database (including the additional information you just provided on categories of lateness.)

4. Use the data base query function to extract two reports. The first report will show relevant information on customers who are more than 60 days late. The second report will show relevant information on customers more than 60 days late who owe more than $400. Criteria ranges (and output ranges for the data base query required by both reports are provided at the bottom of the worksheet.

5. Use range names for the data range, input range, criteria range, and output range of the accounts receivable database. Create a table that lists all the range names at the bottom of the worksheet. Insert rows if necessary.

6. Print the two reports resulting from your data base queries.

Time Estimates

Expert: 1 hour
Intermediate: 2 hours
Novice: 3 hours

Excel Tutorial For Spreadsheet Case 8

This case requires new skills utilizing Excel data management capabilities and date functions. You will need earlier versions of COURSE.XLS for this tutorial. If you saved it with regression analysis data or the range name table, you should erase these parts of the worksheet.

Excel Date Functions

In order to determine how many days an account is overdue, you must be able to calculate the difference between today's date and the date of a customer's invoice. Such calculations are possible using the =DATE function of Excel and by formatting the ranges involved in date calculations in **Date** format.

The =DATE function converts a date into a date number. Excel can represent any given date as a serial number equal to the number of days from December 31, 1899 to the date in question. (Excel also has a "1904" date system where the serial number represents the number of days since January 1 1904) Dates and times thus can be used in calculations like any other number in Excel.

The format of the =DATE function is:

=DATE(year number, month number, day number)

For practice purposes, let's modify your sample spreadsheet to include data about due dates for students' library books. You will then calculate the number of days overdue by subtracting the date due for each student's books from today's date.

Create a new range for book due dates in F14:F18. The first row will contain the column label DUE DATE and F15:F18 will contain date information for each student as follows:

James Jackson:	2/2/98
Steven Parker:	3/3/98
Andrew Reynolds:	4/11/98
Joyce Winters:	5/5/98

To enter this date for James Jackson using the =DATE function you would enter in cell F15 =DATE(98,2,2). F16 would then display 2/2/98 since Excel automatically formats cells containing the DATE function.

You can enter the due dates for the rest of the students using =DATE. Excel also permits you to enter dates directly and it immediately interprets them, ensuring you have adhered to the date system that is the computer default, i.e. American or British dates. Therefore you could enter Parker's date by typing

3/3/98

followed by ENTER. You will see Excel interpret this date and the formula bar will show 3/3/1998. Finish entering the remaining dates, using either method. Then enter the label REPORT DATE in cell E20 and enter the date of this student roster report (5/15/98) in F20 using =DATE. Using this date and the student book due dates, calculate the overdue days.

You will want to calculate the number of days each student's book is overdue in range G15:G18. Enter a label for DAYS OVERDUE in G14. The calculation can be performed by subtracting each student's due date from the report date. Since both dates are entered using the =DATE function, the formula for calculating DAYS OVERDUE for James Jackson is =$F$20-F15. Enter this formula into cell G15 and copy it to G16:G18. Make sure the cells in DAYS OVERDUE column are in General format. Your worksheet should look like Figure 3-11.

Database Management with Excel

The student roster you have created in A14:G18 can be treated as a database, where data can be extracted, sorted, and analyzed. Each row with data about a student can be considered one *record* in the database. Within each row, each cell represents a *field* of that record. The one-line headings at the top of each column, such as NAME or QUIZ, represent *field names*.

Any collection of data organized into records and fields can be treated as a database by Excel. Fields in a Excel database may contain either labels (such as the student names in our roster) or numeric data (such as the student grades).

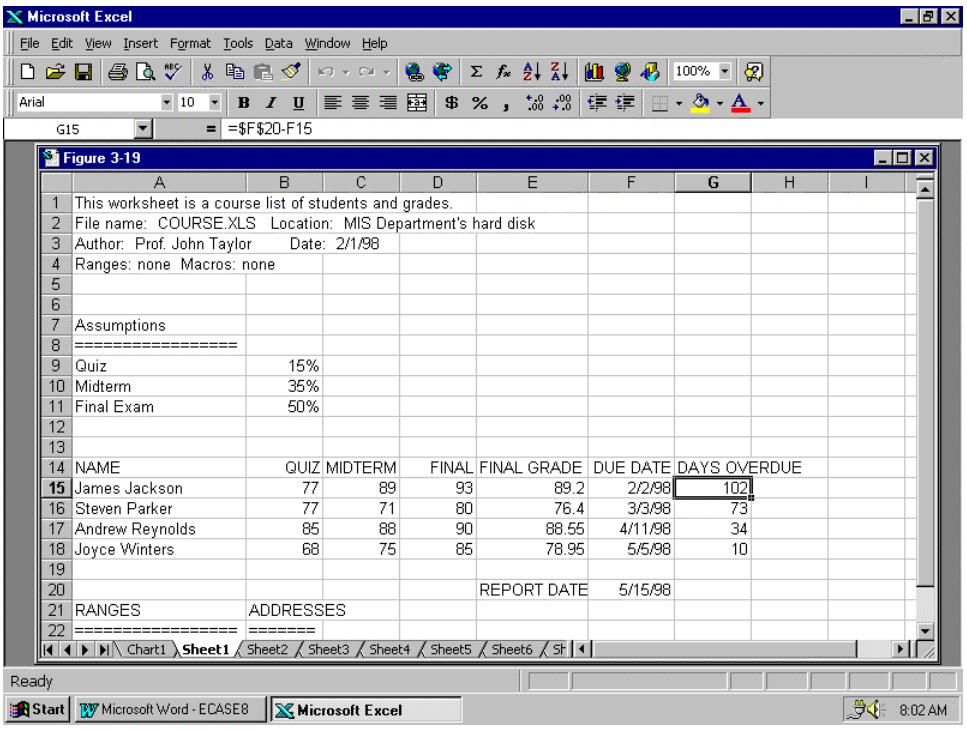
You can use various data commands with a Excel database to sort its records in numeric or alphabetical order, or to find and list records that match criteria that you specify. Let's first sort our student roster database and arrange it by number of days overdue in ascending order.

First, select the range A14:G18, which contains the data and the column headings. Now select Data/Sort from the menu. The resulting dialog box shows three sort keys are available to rearrange the sequence of the database. These are known as the *primary, secondary*, and *tertiary* sort keys. Notice, the selected range now highlights the data only, excluding the headings. Excel has recognized the selection has field names in the first row and deselected the headings.

Each of the sort keys in the dialog box has a drop-down list of field names. By pressing the downward pointing arrow the list will be revealed. Select the DAYS OVERDUE field from the primary sort key at the top of dialog box and select Ascending order for the sort. We could select a secondary sort key to perform a second sort on records that have the same entries on the primary key. However, in our list of Students' Grades, there are no equal entries on DAYS OVERDUE.

The setting at the bottom of the Sort dialog box asks whether the list includes or excludes a Header Row (containing field names). Specify that the selection (A14:G18) contains a Header Row. If you specify that the list does not contain a header row, the heading row will be sorted with the data.

Figure 3-11



To execute the sort, select the OK Button. Your course roster will then be sorted so that Joyce Winters, with 10 days overdue, will be first on the list and James Jackson, with 102 days overdue, will be last on the list. (See Figure 3-12).

Filtering a Database

You can search a database for particular records, copy or extract records from a database using the **Data/Filter/AutoFilter** or **Data/Filter/Advanced Filter** commands of Excel. For example, you can filter your student database to produce a list of all students whose books are more than 45 days overdue.

The **AutoFilter** and **Advanced Filter** do the same thing in different ways. An Excel Filter lets the user dictate some criterion or criteria which will include some records and exclude others. **AutoFilter** performs the filter operation on the list at the same location. The **Advanced Filter** gives the user the option whether to filter on the same location as the list or to copy to a new location. The **Advanced Filter** also permits more complex criteria than the **AutoFilter**, although the **AutoFilter** satisfies most demands.

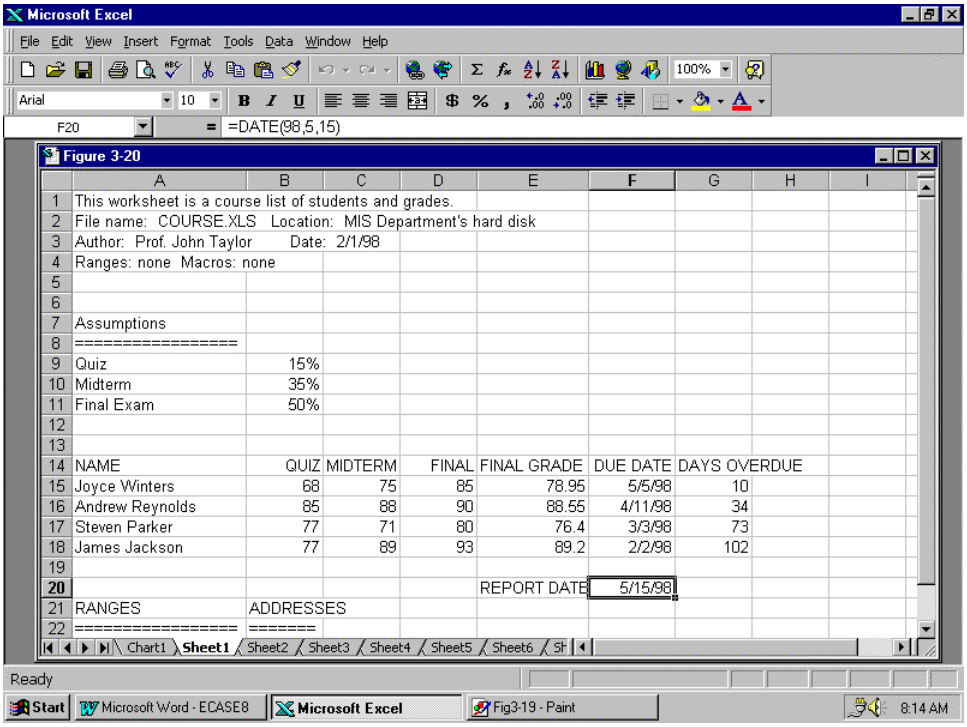
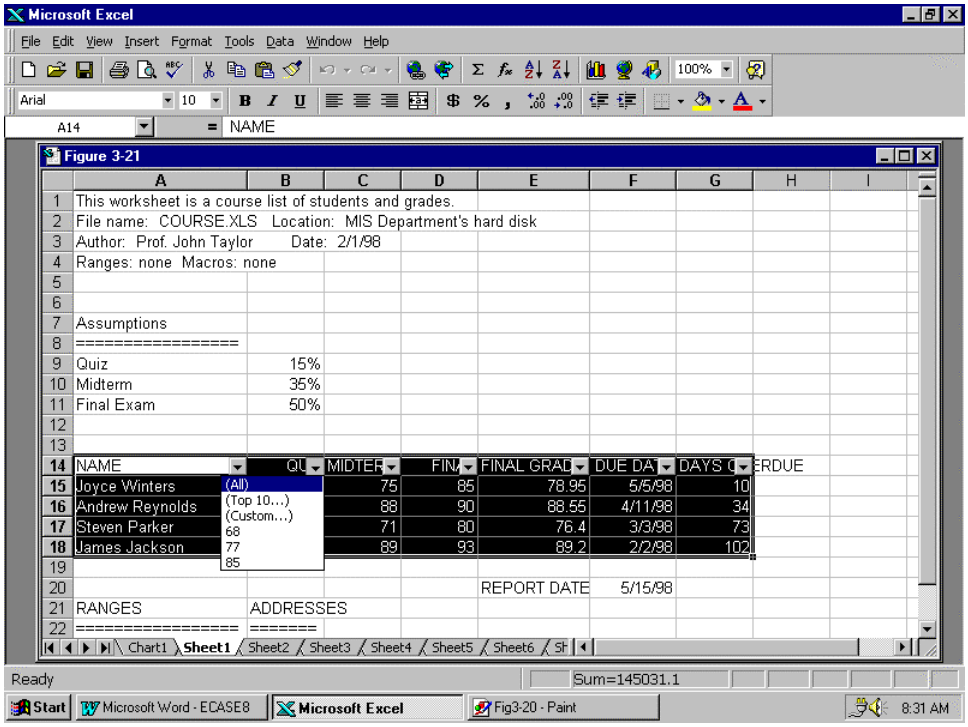


Figure 3-12

The **AutoFilter** is a simple but extremely effective tool to distil large amounts of data very quickly. To demonstrate its usefulness, highlight the database in the range A14:G18. Now select **Data/Filter/AutoFilter** from the menu. You will notice the AutoFilter drop-down arrow buttons next to each of the field names. These buttons are how the query is achieved. Press the drop-down arrow on the QUIZ field and your screen should look like Figure 3-13

The **AutoFilter** drop-down menu contains items such as: (All), (Custom...), (Top 10), (Blanks), (NonBlanks), and an entry for each unique list entry. The (All) item displays all the records; the (Custom...) item lets you enter a criterion for the current field. The (Top 10) item displays all rows that fall between the upper and lower limits you specify. The (Blanks) item displays the blank records in this field only. The (NonBlanks) item displays all the records that are not blank. These last two items appear in the AutoFilter menu only when the column you want to filter contains a blank cell. The remaining items are the entry items; selecting one of these entries will display those records with that entry only. Select 77 from the QUIZ drop-down list and observe the effects. All the entries other than those with 77 in the QUIZ field are hidden and the row numbers of the remaining entries are colored. Select (All) from the QUIZ drop-down menu to return to a full list.

Figure 3-13



Select the DAYS OVERDUE drop-down menu and select (Custom...) from the menu. The Custom AutoFilter dialog box permits one or two simple conditions connected with an OR or an AND. If the field being queried is text the box for the first logical operator on the left should be an "=" or “equal to”. For our purposes, select ">=" (greater than or equal to) as the logical operator and type 45 in the value window to the right of this logical operator box to represent DAYS OVERDUE >= 45 (greater than or equal to) 45 days. Press the OK Button to execute the query. The list should have two records displayed now: Steven Parker and James Jackson. If you wanted a separate permanent copy of the results of an AutoFilter query, you would have to manually copy them. To turn off the AutoFilter, select **Data/Filter/AutoFilter**.

The Advanced Filter permits more advanced criteria than the AutoFilter. It requires a criteria in a separate location on the worksheet. This is best done by copying the field names from the original list so no typing error can creep in. Copy the headings to Row 23, directly below the list. Type >45 under the DAYS OVERDUE heading in the criteria. (You will have to erase or move further down the range name table you created in an earlier tutorial.)

Now select **Data/Filter/Advanced Filter**. In the Advanced Filter dialog box, select Copy to Another Location to provide a separate list for the result of the query. For this Action

setting, the Copy to: window becomes active and requires a reference. Three references are required for this dialog box: the List Range, the Criteria Range and the Copy to Range. Select the ranges:

List Range	A14:G18
Criteria Range	A23:G24
Copy to Range	A27:G27

Now select the OK Button to execute the query. The resulting screen should look like Figure 3-14.

The **Criteria Range** setting in the Advanced Filter dialog box tells Excel which records to search for in the database. Your search criteria may include one or several fields in the database. The criteria range will have at least two rows: one for the heading and one for the selection criteria.

The first criteria row *must* be the field names of all of the fields that will be referred to in your search criteria. The second row of the criteria range is where the various selection criteria are entered. Each criterion must be entered directly below the field name to which it applies. Criteria may be numbers, labels, or formulas. A criteria range can be two or more rows long.

To search for exact matches of labels, enter the label used as a criterion exactly as it appears in the database. You can also search for similar, but not identical, label entries using special characters called *wildcards*. Wildcards can be used in both Advanced Filter criteria and AutoFilter criteria:

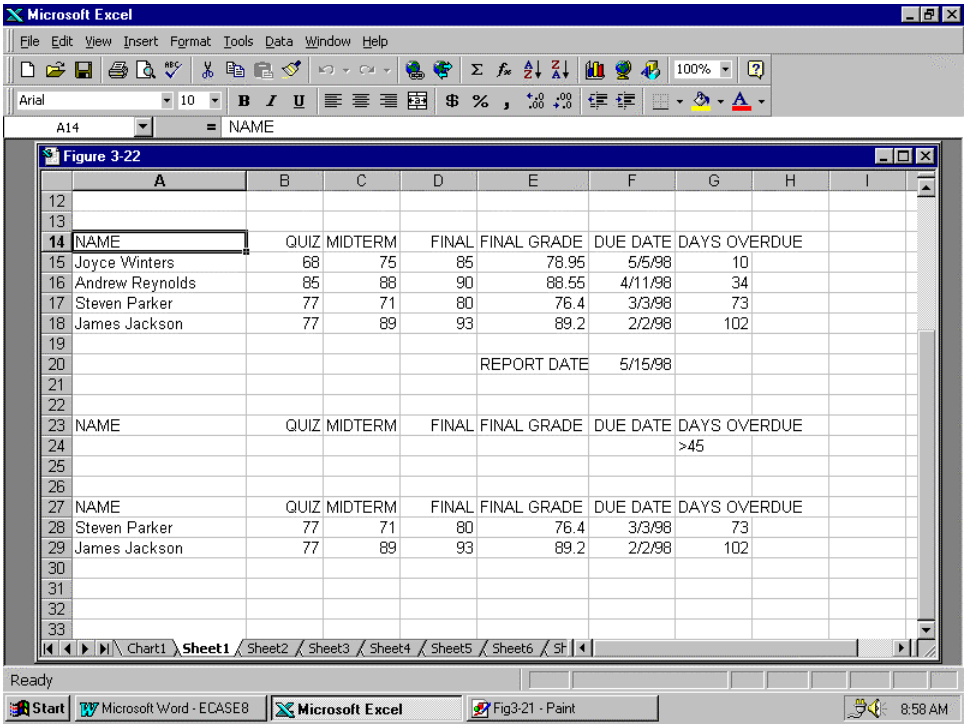
? instructs Excel to accept any character in that specific position and can only be used for fields of the same length. (For example, b?t matches bit or bat but not beet.)

* instructs Excel to accept any and all characters that follow and can therefore be used for fields of unequal length. (For example, bat* matches batch or batter, but not butter.)

The Copy to Range determines the destination of the extracted records and the field names are copied as well. This option is only active when the Copy To Another Location has been selected. The Copy to Range should be an unused area of the worksheet.

Be sure to save COURSE.XLS with the changes resulting from the Advanced Filter operation. It will be required by the tutorial for Spreadsheet Case 10.

Figure 3-14



	A	B	C	D	E	F	G	H	I
12									
13									
14	NAME	QUIZ	MIDTERM	FINAL	FINAL GRADE	DUE DATE	DAYS OVERDUE		
15	Joyce Winters	68	75	85	78.95	5/5/98	10		
16	Andrew Reynolds	85	88	90	88.55	4/11/98	34		
17	Steven Parker	77	71	80	76.4	3/3/98	73		
18	James Jackson	77	89	93	89.2	2/2/98	102		
19									
20					REPORT DATE	5/15/98			
21									
22									
23	NAME	QUIZ	MIDTERM	FINAL	FINAL GRADE	DUE DATE	DAYS OVERDUE		
24							>45		
25									
26									
27	NAME	QUIZ	MIDTERM	FINAL	FINAL GRADE	DUE DATE	DAYS OVERDUE		
28	Steven Parker	77	71	80	76.4	3/3/98	73		
29	James Jackson	77	89	93	89.2	2/2/98	102		
30									
31									
32									
33									

Chart1 Sheet1 Sheet2 Sheet3 Sheet4 Sheet5 Sheet6

Spreadsheet Case 9

Garwood Properties Inc.

Problem: Develop a sensitivity analysis

Management skills: Planning

Deciding

PC skills: Data base creation
Data table building
Financial functions

File: GARW_Q.XLS

Garwood Properties is one of the hottest real estate developers in the San Diego metropolitan area. It handles both residential and commercial properties, and has an unmatched reputation for spotting the best new areas to develop ahead of the competition. Garwood Properties has managed to purchase the right properties in the right locations right before development took off.

Garwood's president, Michael Clausen, is considered one of the shrewdest evaluators of properties in the business. Garwood is considering purchasing an apartment building in an upcoming residential area. The building was renovated in the early 1990's to attract young working professionals and it anticipates a steady growth in revenue from rentals to this group.

The owners of this property are asking $12,000,000. Thanks to economic revival and gentrification, properties had been appreciating at close to 9% a year in real terms. At this rate of appreciation, they claim that the building will have a market value of over $26,000,000 in ten years.

Garwood never bets on historic costs of appreciation or on sentiment, history, and location. The firm insists that buildings must be evaluated solely in terms of what they produce in income. A real estate investment is worth no more than the rents it will produce and the final sale price over the next ten years.

Garwood prefers to use the discounted cash flow method for determining the value of a real estate investment. The value of the investment results from cash flow from rental income produced over a period of years plus the value of the property when sold.

This analysis calls for basing the anticipated sale price of the property in 2008 on the cash flow for the year 2009 and the anticipated interest rate which investors could receive on the open market. In a nutshell, Garwood analysts must determine the 2008 sale price by dividing the cash flow in year 2009 by the "going out rate."

These values in turn must be discounted to account for the declining time value of money. A dollar earned ten years from now will not be worth one of today's dollars because of inflation. To arrive at the value of the investment today's dollars, Garwood first calculates the present value of the expected cash flows from the building and expected sale price. The purchase price of the property in today's dollars is then subtracted from present value of the future payments or earnings to arrive at the net present value of the investment. The net present value can then be compared to the asking price and to what historic appreciation would suggest the building is worth in 10 years.

Since real estate investment is highly sensitive to changes in interest rates, the firm insists on examining net present value under a wide range of situations. Net present value of the property must be calculated using a variety of capitalization and discount rates.

On your data file GARW_Q.XLS the assumptions for Garwood's analysis are displayed in the upper left hand corner of the template. The discount rate is the rate at which money loses value over time (roughly the rate of inflation.) The "going out rate," or capitalization rate, is in essence the prevailing interest rate of Treasury bills. Garwood depends very heavily on this rate for its analysis. The growth rate is the rate at which net rental income from the property is expected to grow each year. There are invariably costs for selling a building. In this case, 2.25% of the building's market price must go to pay the costs of selling.

Garwood must adjust its sale price by the cost of selling the property and add the result to revenue from the ten years of cash flow produced by building rentals. Garwood's analysts will then establish a present value for the sum of the adjusted sale price plus ten years of cash flows. The difference between the present value of these future earnings and the purchase price is the net present value. If the net present value is positive, the investment could be acceptable. If the net present value is negative, the investment should be rejected.

Since real estate investments are highly sensitive to changes in interest rates, Garwood analysts must also perform a *sensitivity analysis* in the form of a data table which shows what impact changes in the discount rate and "going out rate" have on the net present value of the building.

Tasks

There are 7 tasks in this case:

1. Print out the data file GARW_Q.XLS to see the assumptions for the problem and the basic outline of the template.

2. Calculate the net income (cash flow) of the property for years 1999-2009.

3. Calculate the sale price of the property using Garwood's method of dividing the cash flow in 2009 by the "going out rate".

4. Calculate the total value of the investment by adjusting the sale price by the cost to sell, and adding the result to the total cash flow for 10 years. Calculate the present value of this amount using the =NPV function of your spreadsheet software. Then calculate the net present value of the investment by subtracting the purchase price from the present value. Round the net present value using the =ROUND function of your spreadsheet software and divide the rounded figure by 1000.

5. Develop a data table that shows the impact of different discount rates and "going out rates" on the net present value of the investment. The discount rate should be on the X axis and the "going out rate" should be on the Y axis. The discount rate should begin with a value of 3% and end with 11% in half percent increments. The "going out rate" should begin with 6% and end with 12% in half per cent increments. It is helpful to use the Edit/Fill/Series commands of your spreadsheet software to set up these values on the data table.

7. Write a one-paragraph evaluation of the proposed investment. Should it be purchased? Should it be rejected at all costs? Or are there are conditions under which it would be a worthwhile investment?

Time Estimates

Expert: 1 hour
Intermediate: 2 hours
Novice: 3 hours

Excel Tutorial For Spreadsheet Case 9

This case requires knowledge of the table-building features of spreadsheet software and of the =ROUND and =NPV functions of Excel.

Sensitivity analysis is the process of exploring various "what-if" situations in order to determine the impact of one or several variables on a model. Table-building automates the "what-if" process so that sensitivity analysis does not have to be performed manually. Instead of performing repeated "what-if" analyses, the Excel table-building function allows various values to be substituted for existing values in your worksheet. A table will then be generated to detail the results.

To demonstrate, let's create a new worksheet that calculates sales commissions. You want to see the impact on commissions on sales of $5000 when they are based on different percentage rates. In order to fully display the data table on your worksheet screen, we will not enter documentation into the first four rows.

Set up your worksheet so that the labels PERCENT, SALES, and COMMISSION are in cells A1, B1 and C1, respectively, and the values for percent (5%) and sales ($5000) that we want to use in our calculation are in cells A2 and B2, respectively. In cell C2, place the formula, =A2*B2, for calculating the commission.

You could enter different percentages in the A column and then copy the formula for commission calculation to appropriate cells in the C column. However, you can also use the **Data/Table...** commands to perform this analysis automatically.

The Edit/Fill/Series Command

Before using these commands, you must enter the different percentage values in a column. Instead of entering each value individually, you can use the **Edit/Fill/Series** command. This command fills a range of cells with a series of numbers or dates that increase or decrease by a specified increment or decrement, or increase or decrease with a multiplicative growth factor.

Firstly enter .05 in cell A9. After accessing the **Edit** menu, select the **Fill** item, and select **Series** in the submenu. In the dialog box, check the Series in Columns setting and the Type is Linear. In the Step window type .01 and in the Stop window type .1. Press the OK Button when finished.

Alternatively, if you did not have a required Stop value but have a desired range for the series, you could select the range with the first cell containing the starting value and select **Edit/Fill/Series**. In the dialog box do not specify a Stop value; Excel will finish Filling when it reaches the end of the selection.

Data/Table Commands

After generating the column of interest rates in range A9:A14, we must enter either the formula for calculating commissions or the cell address from which to draw the formula. *This entry goes next to the column of percentages and one row above the first entry.*

Enter =C2 in cell B8. You could also have entered the formula for computing commission in B8.

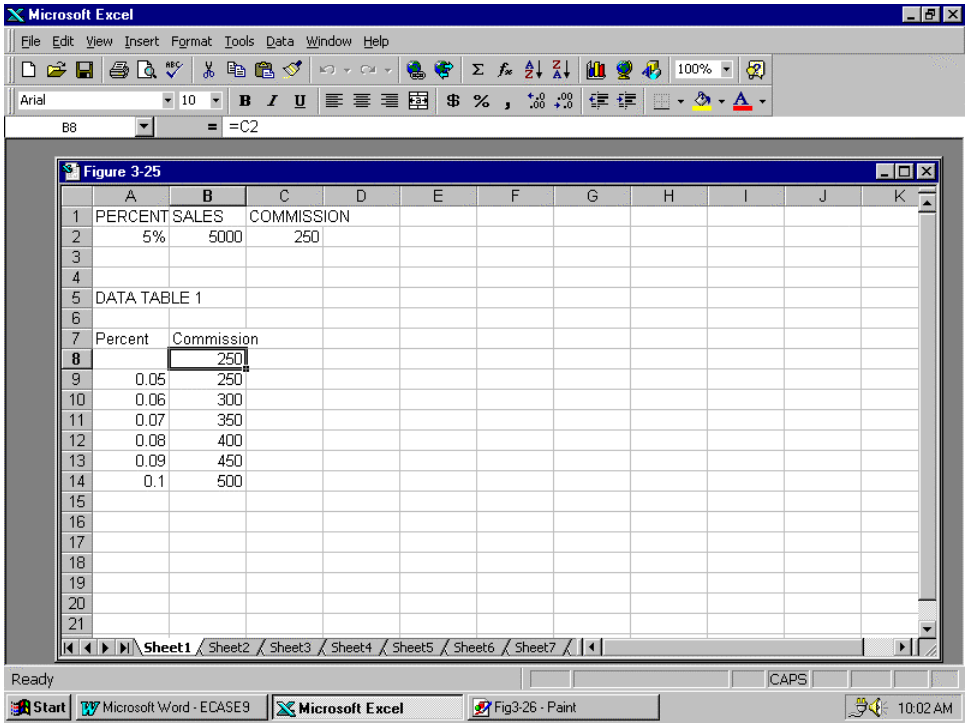
Select the range A8:B14, the range containing the percentage values, the formula and the blank cell where the results will go. Next, select the **Table** item from the **Data** menu. The Table dialog box has two available settings: the Row Input Cell and the Column Input Cell. Since you are only examining the impact of changes in one variable -- percentage -- you will enter a cell reference in only one of these. The variable numbers you are examining is in a column so the cell reference for the percentage variable should be entered. Enter the reference A2 in Column Input Cell and select the OK Button.

By entering only a single input cell reference in the dialog box, you are examining the impact of changes to one value in a formula. If you entered both the Row Input Cell and Column Input Cell, you would show the impact of changes to two values in a formula.

The results should look like Figure 3-15.

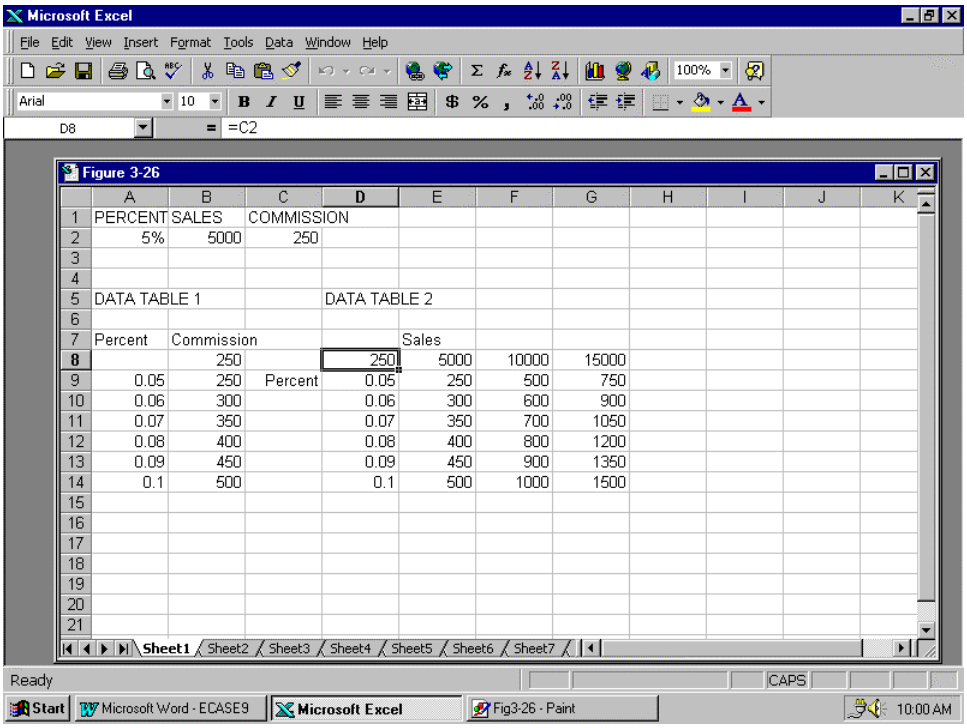
Now examine the impact on commissions if the sales amount, as well as the percentage, is variable. What happens when sales are $5000, $10,000 and $15,000? You would enter the values for our second variable (Sales) in the row just above the first entry of our first variable (Percentage). Enter the Sales values in range E8:G8 and the Percentage values in range D9:D14. *The address of the formula (=C2), must be entered in the cell directly above the first entry of the first variable.* (In your example, this would be in cell D8.)

Figure 3-15



Select the range of the table (D8:G14) and then choose **Data/Table**. In the Table dialog box choose cell reference A2 as the Row Input Cell and cell reference B2 as the Column Input Cell. Then select the OK Button. The two-dimensional data table should look like Figure 3-16:

Figure 3-16



	A	B	C	D	E	F	G	H	I	J	K
1	PERCENT	SALES	COMMISSION								
2	5%	5000	250								
3											
4											
5	DATA TABLE 1			DATA TABLE 2							
6											
7	Percent	Commission			Sales						
8		250		250	5000	10000	15000				
9	0.05	250	Percent	0.05	250	500	750				
10	0.06	300		0.06	300	600	900				
11	0.07	350		0.07	350	700	1050				
12	0.08	400		0.08	400	800	1200				
13	0.09	450		0.09	450	900	1350				
14	0.1	500		0.1	500	1000	1500				
15											
16											
17											
18											
19											
20											
21											

=NPV and =ROUND Functions

Spreadsheet Case 9 requires that you compute the projected value of an investment in today's dollars. The **=NPV** function computes the present value of a stream of cash flows discounted at a fixed periodic interest rate. The form of this function is:

=NPV(*Discount Rate*, *Value1, Value2, ...*)

The discount rate is the interest rate, and the Values are the series of cash flows to be discounted. The interval between cash flows must be constant and must agree with the period of the discount rate. The Value parameters can be both values in the formula, references to cells, and references to ranges of cells. Up to 29 Value parameters are allowed, although each can of course be several values in a range on the worksheet. It is important to realize that Excel takes the order of Value1, Value2 and so on, as the sequence of the cash flows.

The **=ROUND** function rounds numbers to a specified number of places. The form of this function is:

=ROUND(number, num_digits)

Number signifies the number you want to round. Num_digits specifies the number of digits to which you want to round the number.

If num_digits is greater than 0, the number is rounded to the specified number of decimal places.

If num_digits is less than 0, the number is rounded to the left of the decimal point.

If num_digits equals 0, the number is rounded to the nearest integer.

For example,
=ROUND(2.15, 1) equals 2.2

=ROUND(2.149, 1) equals 2.1

=ROUND(21.5, -1) equals 20

=ROUND(-1.475, 2) equals -1.48

Spreadsheet Case 10

Levit Brothers Investment Banking

Problem: Develop a money market support system

Management skills:	Controlling Deciding
PC skills:	Macro building Database query and extract
File:	LEVIT_Q.XLS

Levit Brothers is a distinguished investment banking firm on Wall Street. With the recent uncertainty in the stock market, its Money Market department has been flourishing. Individual and corporate investors are investing more conservatively. Money market funds are considered a safe place for parking funds until market conditions improve.

The Levit Brothers Money Market Department would like to capitalize on this trend by making sure it can get the most leverage out of the firm's commercial paper portfolio. This portfolio consists of overnight to six month obligations valued at well over a billion dollars.

Levit Brothers does not own all of the commercial paper in its portfolio outright. It leverages its profits by financing a large portion of its holdings using overnight loans from major banks and "repo" agreements from major pension funds and private institutions. (In a repo agreement, an outside institution purchases a part of the portfolio for a short period, from one day to a few months.) The loans from these other institutions are collateralized by the same money market instruments they are financing (much as a house is used as collateral in a mortgage.) In this manner, Levit Brothers can maximize its return on a billion dollar portfolio using only a small amount of its own equity.

The Money Market Department is responsible for refinancing the portfolio daily, taking advantage of changing interest rates to keep abreast of the market. The Money Market Department must track approximately 500 purchase or sale transactions per day and monitor their impact on the firm's portfolio position.

Ted Samuels, the head of the Money Market Department, is worried that the volume of purchase and sale transactions is too high to be tracked manually. Traders can no longer record individual transactions with paper and pen and expect them to be listed, summarized, and analyzed the next morning. The firm can't take maximum advantage of the portfolio for fear of double pledging collateral and overextending its position.

The Money Market Department needs an automated system that can track the flow of funds and immediately, on demand, provide traders with a listing of all the inventory held by an inquiring bank or institution. First, it would like to be able to sort the commercial database by net amount in descending order and by the name of the institution in alphabetical (ascending) order. Second, it would like to be able to enter the identification code of an inquiring bank or institution

and have a report of that institution's holdings generated immediately. The Money Market Department also needs a listing of what inventory is still available to be financed or sold.

Load the data file LEVIT_Q.XLS from your data diskette. This file consists of three separate databases: 1) a Commercial Paper Database, showing a sample of Levit Brothers' money market portfolio; 2) a Bank Loans Database, listing all the banks that have made loans on commercial paper holdings; and 3) a Repo Name Data Base, listing all of the institutions with whom Levit Brothers has repo agreements.

Most of the fields in the three databases need no explanation. However, you should pay particular attention to the CODE field in the Commercial Paper Data Base. A code of 'S' designates inventory that has been sold. Numeric codes on the Commercial Paper Data correspond to the Bank I.D. number on the Bank Loans database. These numeric codes designate inventory that has been pledged as collateral on loans from banks. For instance, a code of 12 on the Commercial Paper Data Base refers to Bank I.D. 12 (Bank of America).

Codes such as R3 correspond to identification codes on the Repo Name Database. These codes designate inventory that has been pledged to various organizations in repo agreements. The code 'OPEN' has been assigned to any piece of inventory that is available to be financed or sold.

Tasks

There are 4 tasks in this problem:

1. Print out the data file LEVIT_Q.XLS so you can see the three databases and their structures.

2. Construct two macros to sort the Commercial Paper database. One macro should sorts the database by the amount of paper from each institution (NET AMOUNT) in descending order. Another should sort the database by the name of the institution (AVAILABLE) in ascending order.

3. Construct a macro for a "Holdings Report" The macro should automatically produce a report that searches for and extracts records representing the inventory held by an inquiring bank or institution with a repo agreement. The macro should pause for the user to type in the identification code of the inquiring bank or institution with a repo agreement as the criteria for extracting records.

4. Construct a macro for an "Open Inventory" report. The macro should automatically produce a report of all items in inventory with the code OPEN.

(Hint: Pay attention to label-prefix characters. Your macros may not work properly if you do not clear the criteria range or table before your macro enters criteria for either report. Be sure to have each macro erase the criteria remaining in the criteria range or table as the first step in the macro.)

Excel Tutorial For Spreadsheet Case 10

This case requires that you use Excel to build a *macro* to extract records from a database to produce a report automatically, and to perform a sort on a list. A macro is, in essence, a collection of commands. The commands are contained in a module sheet which can be stored in a workbook. The macros can be executed through the **Tools/Macro** menu item, through a custom Toolbar button, through a customized menu item, or through a key combination (e.g. CTRL-A).

For this exercise you will need COURSE.XLS with some of the changes you made during the Spreadsheet Case 8 tutorial session.

Current versions of Excel have an extensive macro language called Visual Basic. This language is intended to be used by several Microsoft applications. Excel allows macros to be recorded by transcribing a series of operations acted out by the user to the equivalent Visual Basic commands.. The commands are stored in a module. (In Excel versions for Windows 3.1, macros are stored on a separate sheet in the workbook called a Module sheet. When you record a macro, Excel inserts a Module sheet and places the commands on the sheet.)

A macro can be recorded by accessing the command **Tools/Macro/Record New Macro** (**Tools/Record Macro/Record New Macro)** from the menu and naming the macro.

You can create a macro, for example, to automate the sorting of your student roster database by number of days overdue in ascending order, the same task you performed during the tutorial for Spreadsheet Case 8.

To record an Excel macro, select the **Macro (Record Macro)** command from the Tools menu. This opens a sub-menu with more choices. From this menu, select **Record New Macro**. This opens a dialog box with the same name. The Macro Name text-entry box is highlighted, with the default name "Macro" followed by a number. Type in a name for the macro. The highlighted default name for the macro is automatically deleted and replaced with the new name you assigned. Note that macro names cannot have any spaces or punctuation marks, and must begin with a letter. Otherwise, they can contain letters, numbers and underscores. Below the name box is a documentation box where you describe what your macro does. Excel automatically includes the name of the spreadsheet's author and the date in this area.

To start recording, click the OK button. Any action you perform with the mouse or keyboard will be recorded by Excel as a macro command that will be stored in a worksheet added to the end of the workbook. Notice that a button with a small black square on it has appeared over the worksheet:

This is the Stop Macro button which you can click on to turn macro recording off. The same command is available as a menu item (select **Tools/Record Macro/Stop Recording** from the menu). Excel won't record you selecting it as part of the macro.

The actions to record for automating the sorting of your student roster database by number of days overdue in ascending order would be as follows:

Select the range A14..G18.
Select **Data/Sort** from the menu.
Select the DAYS OVERDUE field as the primary sort key.
Select Ascending Order for the sort.
Makes sure that Header Row is selected.
Click the OK button to execute the sort.

To finish, click on the Stop Macro button. Notice that the Stop Macro button disappears, indicating that the macro has stopped recording.

To test the macro, undo the changes you made while creating the macro. To run the macro, open the Tools menu and click on the Macro command. The Macro dialog box opens. Beneath the Macro/Name Reference text box is a larger box listing all available macros. Click on your macro's name in this box. The name is displayed in the text box above and the macro's description appears at the bottom of the dialog box. Then click the Run button.

If the macro runs correctly, the database should be sorted in the appropriate order. For most purposes, recording a macro is the best way of creating macros and learning the language.

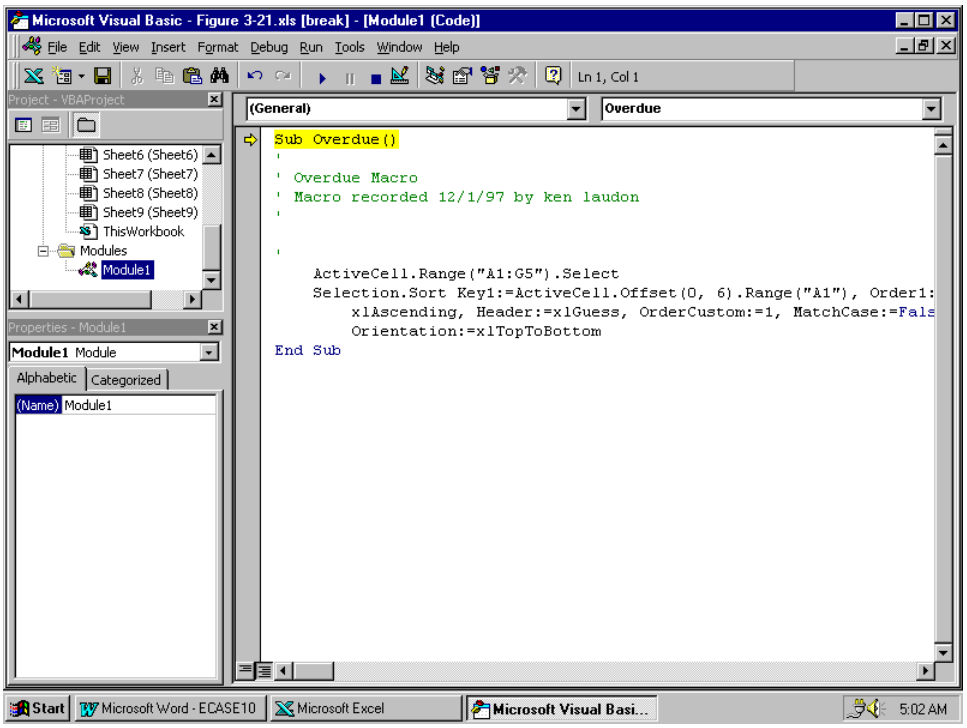
Viewing the Macro

Users of Excel for Windows 3.1 can see the macro commands that were recorded by clicking on the tab labeled "Module 1". The worksheet appears along with a new group of icons beneath the second tool bar.

Users of Excel versions for Windows 95 can view their macro commands by selecting Tools/Macro/Macros, then selecting the Edit option from the Macro dialog box. The Visual Basic module for the macro appears and can be edited. The module for the macro to sort the student roster database would look like Figure 3-17.

As you can see, modules have a non-tabular format, and lack the rows and columns typical of regular worksheets. Instead, data is displayed as lines of text against a plain, white background.

Figure 3-17


Microsoft Visual Basic - Figure 3-21.xls [break] - [Module1 (Code)]
File Edit View Insert Format Debug Run Tools Window Help
Ln 1, Col 1
Project - VBAProject
Sheet6 (Sheet6)
Sheet7 (Sheet7)
Sheet8 (Sheet8)
Sheet9 (Sheet9)
ThisWorkbook
Modules
Module1
Properties - Module1
Module1 Module
Alphabetic
Categorized
(Name) Module1
(General)
Overdue
Sub Overdue()
'
' Overdue Macro
' Macro recorded 12/1/97 by ken laudon
'
'
ActiveCell.Range("A1:G5").Select
Selection.Sort Key1:=ActiveCell.Offset(0, 6).Range("A1"), Order1:
xlAscending, Header:=xlGuess, OrderCustom:=1, MatchCase:=Fals
Orientation:=xlTopToBottom
End Sub
Start
Microsoft Word - ECASE10
Microsoft Excel
Microsoft Visual Basi...
5:02 AM

4

Introduction to Database Software

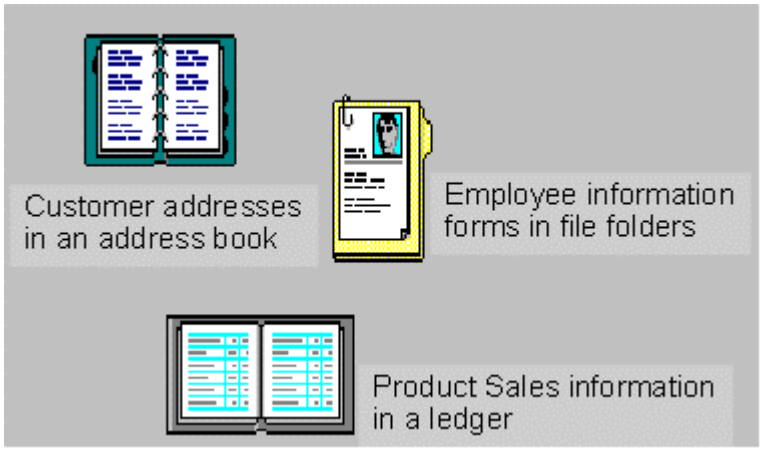
This edition of the *Solve it!* database cases was developed for Microsoft Access 2000 and Access 97.

This chapter describes the elements of a computerised database and how to use database software. The name "database" sounds formidable, but all of us have used databases before.

Some examples of common manual databases we all use every day are a telephone book, an address book, and the card catalog in a library.

In the business world some common manual databases are a list of customers and customer addresses, a list of suppliers, a list of products sold and their respective prices, a list of products in inventory, and a filing cabinet which contains invoices arranged in numerical or date order.

Figure 4.1



4.1 What is a Database?

These examples of common manual databases can help provide an initial definition of a database as any organised list or *file* of information pertaining to people, places, or things (see Figure 4-1).

For spreadsheet software the central metaphor is that of a matrix where quantitative information is organised in rows and columns, but in a database the central metaphor is that of a list or file where information is organised in rows (records) and columns.

Any problem in the real world which can be expressed as a problem of lists or files is potentially amenable to solution by a computerised database.

There are three basic types of databases: hierarchical, network, and relational. The student should consult a textbook for a detailed description of the various kinds of databases. Here we will describe only *relational databases*.

A simple example of a database file is a customer file:

Record #	Field 1 Last Name	Field 2 First Name	Field 3 Address
2			
3			
...			
...			

As you can see, in a database file all the information you have on an entry is called a *record*. Each record is composed of a number of *fields* which constitute the information stored on each entry. A collection of records is called a *file.* Below you will learn how to create a computerised database file. But modern database packages offer much more than just computerised files.

Two excellent Microsoft whitepapers on relational database basics and why businesses use databases can be downloaded from the World Wide Web at:

ftp://www.mbs.unimelb.edu.au/pub/slides/solveit/dbback.ppt
ftp:// www.mbs.unimelb.edu.au/pub/slides/solveit/whyuse.ppt

4.2 What is a Database Management System?

A database management system (DBMS) is a software package which, at a minimum, allows the user to create several different database files *and* relate information in one file to information in another file. Second, a modern DBMS provides a number of related tools needed to develop complete information systems.

One important advantage of *relational* database management systems over a manual filing cabinet is the ability to easily combine specific pieces of information *(fields)* from several different database files into a new file.

For instance, what if you wanted to determine a daily inventory which was calculated by deducting daily sales of all products from a beginning inventory?

Here, you would want to find out from the invoice file at the end of each day exactly which items were sold. You would then want to go to the inventory file and debit the existing inventory. In a truly sophisticated system you might want to go into a third file which contained the names and addresses of your suppliers and generate a purchase order and mailing label for those products where inventory was low.

In a manual record system stored in the traditional filing cabinet, a lengthy search process would be required to solve this problem. In a contemporary relational database management system, the job can be done in minutes.

A second feature of a relational DBMS is a set of powerful tools which can be used to develop a complete information management system. Included here are facilities to create and store memos and notes, data entry screens (forms in Access), reports, labels, and programs. These features are controlled through a powerful fourth generation language or menus which require little or no programming knowledge.

Briefly, a contemporary DBMS for the PC is a system development tool which permits the user to create complete management systems suitable for a small business or for an office within a large business organisation.

4.3 Comparative Advantages of Database Packages

Once students learn a relational database package, they often find it more powerful and useful than spreadsheet packages. Yet the programs really have quite different strengths (and weaknesses).

Spreadsheet programs are very good at manipulating quantitative data, but they are poor at storing and manipulating lists or extracting parts of files from a larger data set. They generally are quite poor at combining information from several different files, and they typically have very limited macro or programming languages, although recent releases of popular spreadsheets defy this.

Database packages are very good at creating and manipulating lists of information, especially text information. Contemporary relational database packages all have very friendly, easy-to-learn, menu-driven command systems which permit the novice to accomplish many if not all of the program's functions.

In addition, relational database packages are accompanied by powerful fourth generation languages which are easy to learn and which permit intermediate users to build complete information systems suitable for a small business or an office within a larger business.

However, database software is comparatively weak at manipulating quantitative data. This gap is however rapidly closing with the advent of recent interactive Windows-based database products such as Microsoft Access.

To operate a contemporary business, both kinds of packages are needed. Data can be exchanged between the packages so that the most appropriate software is used.

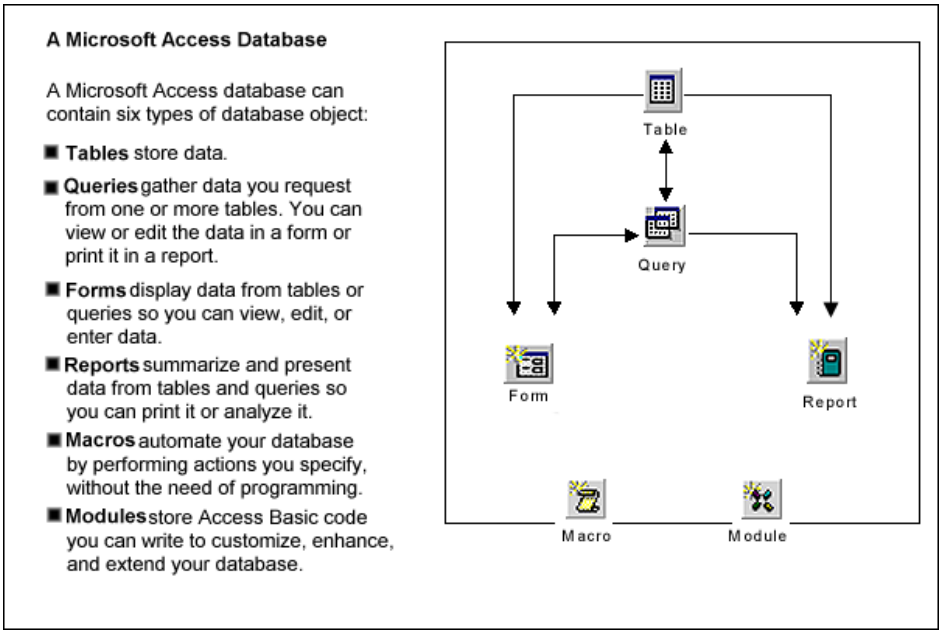
4.4 Windows-based Databases

Database products developed for the graphical, Windows environment such as Borland's dBase for Windows, or Microsoft's Access for Windows utilise a user friendly, non-programming approach for many tasks which would have required programming in DOS-based (ie:non-Windows) database packages. For Access users, none of the cases in this edition of *Solve it!* will require explicit programming (although plenty of code will be generated in background).

4.5 Introducing Microsoft Access for Windows

An Access database is represented as a collection of objects. One .MDB file encloses all the tables, queries, forms, reports and other objects associated with a particular database. A brief description of these objects, the relationships between them within the database envelope, and the graphical buttons that identify them is shown in Figure 4-2

Figure 4-2



Access object names are not subject to DOS file-naming conventions which limit file names to 8 characters. Access object names (and field names in tables) can be up 64 characters in length, and can include spaces as well as a mixture of upper and lowercase letters. Access is also not case sensitive, and will find data within a field (as long as the spelling is correct !) regardless of the case used to originally enter it.

Toolbar Buttons

Access makes extensive use of Toolbar buttons. Each object view in Access has one or more toolbars associated with it by default. A segment of the Table object toolbar is shown above. The buttons on these toolbars provide shortcuts to menu commands, enabling the user to carry out frequently performed tasks on the click of a button. Another nice feature is that as the mouse pointer is positioned over a toolbar button, Access will display a short label describing the button's purpose.

Access 2000 Users:	**Access '97 Users:**
Press the F1 function key and from the INDEX, type in the word *toolbar* for more information about using toolbars.	Press the F1 function key and from the INDEX, type in the word *toolbar* for more information about using toolbars.

Get Help At Any Time!

Access 2000 Users:	**Access 97 Users:**

(Office Assistant, Help System and Wizards)	***(Office Assistant, Help System and Wizards)***
The Office Assistant is a feature included with Access and other Office 2000 applications which provides tips, and attempts to interpret the help information you may need based on your current actions.	Access 97's help features also include Office Assistant, which provides tips, and attempts to interpret the help information you may need based on your current actions.
Access 2000 also includes two other interactive help tools: an extensive context sensitive help system (press the F1 key anywhere, at any time), and a series of Wizards which offer help in creating Access objects such as tables, queries, forms, reports and macros. Unlike the main help system, which you need to search or browse to find answers to questions, wizards ask you relevant questions and actually create a customised version of the object in question according to your responses. A summary of the wizards used in this edition of *Solve it!* is presented in Figure 4-3 below.	Access 97's help features also include context sensitive help (using the F1 key), and wizards. A summary of the wizards used in this edition of *Solve it!* is presented in Figure 4-3 below.

Figure 4-3

Table Wizard	walks the user through table setup and design. Users can choose from dozens of predefined sample tables, and hundreds of sample fields which can then be used to generate tables.
Query Wizard	helps users construct complex queries for common database management tasks such as the merging two tables or performing crosstabulations
Form Wizard	a tool for creating forms in a variety of predefined formats and presentation styles
Report Wizard	a tool for creating reports in a variety of predefined formats and presentation styles
Macro Wizard	a tool used for creating or editing a macro from within an event procedure in a form or report
Control Wizard	creates code behind command buttons, option groups, list and combo boxes used in forms or reports

Access Setup Requirements

Access 2000 Users	**Access 97 Users**
To use Access 2000 you will need:	To use Access 97 you will need:

- a personal computer with Pentium 133-MHz or faster processor
- available hard-disk space usage will vary depending on configuration (167 MB recommended minimum)
- 64mb of RAM
- a Microsoft mouse or compatible pointing device

Operating Systems:

- Microsoft Windows NT (Server or Workstation) version 4.0 or later operating system, with Service Pack 4 or later
  Windows Version 3.1 or later

- an IBM compatible 486-based PC or better (a Pentium is strongly recommended)
- Microsoft Win95 operating system or Windows NT 3.51 or later
- minimum of 12 mb of RAM for use on Win95; 16mb is recommended for Windows NT
- a hard disk with at least 28-60 mb of free disk space. A typical installation requires 40 mb
- a Microsoft mouse or compatible pointing device

Terminology and Conventions Used in the Access Tutorials:

- *Double-click* means to press the left mouse button twice in rapid succession.
- *Click* means to press the left mouse button once only.
- *Righ- click* means to click the right mouse button once.

- New terms, new filenames, key concepts and action words appear in *italics*.
- Menu choices and new object names are shown in capitals (eg: FILE/SAVE or FRIENDS.MDB).
- Toolbar buttons are shown next to or within the paragraph in which they are first described.

4.6 Getting Started - Access 2000 and Access 97 for Windows

To Start Access 2000	**To Start Access 97**
1. From Windows , click on the *Start* button at bottom left of the screen, and then select the *Programs* option.	1. From Windows , click on the *Start* button at bottom left of the screen, and then select the *Programs* option.
2. Within Programs, select the *Microsoft Access* option. This last action will load the Access program.	2. Within Programs, select the *Microsoft Access* option. This last action will load the Access program.

Access 2000 Users - Creating a Database

Practice: *Create a new Access database called CONTACTS.*

1. Click the *New Database* toolbar button or select FILE/NEW DATABASE from the menu. Access displays the New Database dialog box.

2. In the File Name box, type a name for your database (eg: CONTACTS) after selecting the appropriate path (eg: a:\CONTACTS). Click OK. Access creates a new database file, and opens the *Database Window* (see Figure 4-4).

Figure 4-4

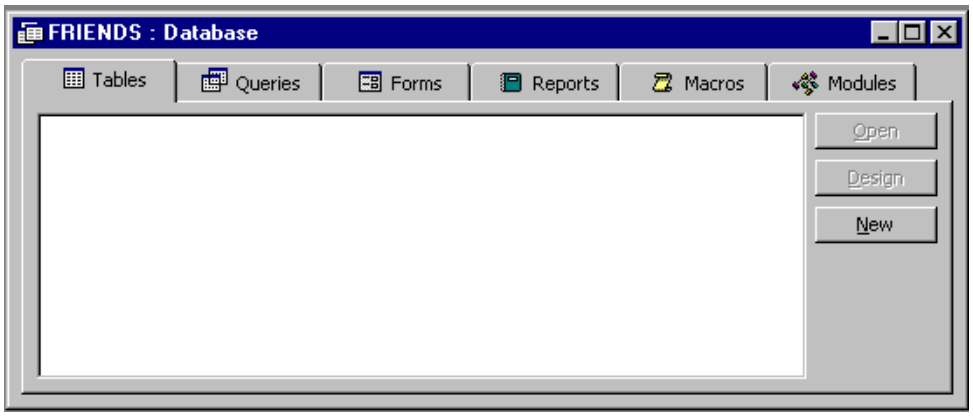
Access 97 Users – Creating a Database

Practice: *Create a new Access database called CONTACTS.*

1. After loading the Access program, you will be confronted with a dialog box asking if you would like to create a new database. Click OK and give the database a name (eg: CONTACTS).

2. Access creates the new database, and opens the *Database Window* (see Figure 4-5).

Figure 4-5



Access 2000 and Access 97 Users - Building a Database Structure

Access allows you to create a structure for storing your data in three different ways:

- by *creating Tables*, and entering data into the table structure
- by *Importing Data* from another application or database file
- by *Linking Data* from an application other than Access

Access 2000 Users	**Access 97 Users**
Access 2000 allows exporting, importing or linking from a number of different formats, as well as other Access databases (version 2.0, 7.0/95, 8.0/97, 9.0/2000). These include: • fixed width text files • a dBase III, III+, IV, and 5 files • Microsoft Excel spreadsheet 2.9 Millennium, 4.0, 5.0, 7.0/95, 8.0/97, and 9.0/2000 • Paradox and Paradox for Windows files • Btrieve tables	Access 97 can export, import or link table data from other Access databases (version 1.x, 2.0, 7.0/95, and 8.0/97), as well as data from other programs and file formats. It is also possible to import or link HTML tables and lists, which can reside on your local computer, a network server, or an Internet server. Access 97: • directly *imports*, *exports,* and *links* to: • Microsoft Excel version 2.9 Millennium or later, Foxpro

- SQL tables, Microsoft Visual FoxPro, and data from other programs and databases that support the ODBC protocol
- Lotus 1-2-3 files

version 2.x or later, SQL Server, dBase III+, dBase IV, dBase 5, Paradox versions 2.9 Millennium to 5.0, ASCII text, and all ODBC-compliant databases

- directly *imports* and *exports* to:
  - Microsoft Visual Foxpro version 2.9 Millennium and Lotus 1-2-3

If you have data in dBase or Paradox databases, and you want to leave your data in its original format, choose the *Linking* option in Access. This way, files can still be used in their original application. If you plan to use your data only in Access from now on, you should choose the *Import Data* function.

Importing data creates a copy of the information in a new table in your Access database. The original source table or file is not altered in this process. *Linking data* enables you to read and in most cases update data in the external data source without importing. The external data source's format is not altered so that you can continue to use the file with the program that originally created it, but you can add, delete, or edit its data using Microsoft Access as well.

Access uses different icons to represent linked tables and tables that are stored in the current database. If you delete the icon for a linked table, you delete the link to the table, not the external table itself.

Creating Tables in Access

You can create tables in two different ways:

- via the *Table Wizard*. The Table Wizard is the easiest way and most usual way for creating tables
- create one from scratch. You may want to do this if you plan to store unusual information in your table

Tip: To return directly to the Database Window at any time, click the  button on the toolbar or press the F11 key.

To create a Table using a Wizard/Adding Records

1. From an open Database Window, click the *Tables* button 

in the Database Window.

2. Click the *New* button. In the New Table dialog box, click the Table Wizards option. The Table Wizard presents a listing of sample table templates which are already set up with fields. Toggle between the Business and Personal radio buttons to get a flavour of the type of tables the Table wizard can automatically create for you. The Table Wizard works as a series of dialog boxes. Follow the instructions to create a table to suit your purpose.

Practice: Create a simple table within the CONTACTS database using the Table Wizards. The scenario is as follows. You are a Phd student who has collected a number of business cards as a result of attending conferences and visiting organisations in the course of your research. On the back of each business card, you have made rough notes about where and when you met the person in question, and what you talked about. You would like to compile a simple database of professional contacts who you think may be useful people to know.

The fields you might need in your table could include:

First Name	State	Last Meeting Date
Last Name	Zip	Action Items
Organisation Name	Work Phone	Notes
Address	Fax Number	

Create this table now. (Hint: use the Contacts table sample under the Business listing in the wizard. Call your new table Contacts). Add five new records to your new table by following the instructions provided in the Access Tutorial for database Case 1 in Chapter 5 of Solve it!

To Create a Table from Scratch/Adding Records

1. From the Database Window, click the *Table* button 

and then click *New*.

2. In the New Table dialog box, select the *Design View* option and then click OK. Access will open the new table in table design view. Create fields for your table (refer Access Tutorial in database Case 1 for details on how to do this). Save your table by selecting FILE/SAVE from the menu, or clicking on the *Save* toolbar button. Add records to your new table following the instructions provided in the Access Tutorial for database Case 1.

Access Field Data Types

A data type defines the type and range of values you can enter in a field. For example, Access will not allow you to enter text in a field which is set to a Currency data type. Access 2000 amd Access 97 use the following data types:

Text	Use for text, or combinations of text and numbers, (such as an address), or for numbers you don't intend to perform calculations on (such as phone numbers or part numbers)

Memo	Use for longer text, such as notes and comments
Number	Use for data you might want to perform calculations on, unless it's money
Date/Time	Use for dates and times
Currency	Use for large numbers requiring rapid calculation, or for numbers that require highly accurate rounding, such as money
Counter	Use to automatically assign consecutive or unique values, such as invoice numbers
Yes/No	Use for true/false, yes/no, or on/off values
OLE Object	Use for graphical objects such as pictures and charts, and for Excel spreadsheets
Hyperlink	Use to store text or combinations of text and numbers as a hyperlink address. This is typically used for connection to data stored on the public World Wide Web or a firm's internal Intranet.
Lookup Wizard	Use to create a field that allows selection of a value from another Access table or from a list of values.

Note: If you change the data type of a field that contains data, you may lose some data. If you attach tables from an external application rather than importing them, Access will not allow you to change the data types.

To set or change the data type of a field:

1. In Table design view, click the Data Type box
2. Click the button to the right of the box
3. From the drop down list that appears, select a data type from the list

Tip: Access object (tables, queries, reports, forms, etc) names, are not restricted by the usual 8 character DOS file-naming rule. Names can be up to 64 characters long and can include spaces between words. After attaching or importing a table, you can rename your new Access table with a more expressive title.

Modify a Table Structure

After you create a Table with the Table Wizard or import or attach a file from an external application, you may want to change its structure by adding or deleting fields, setting primary

keys, creating indexes to speed up data searches, or setting field or table properties to determine the types of data stored in the table and their method of storage.

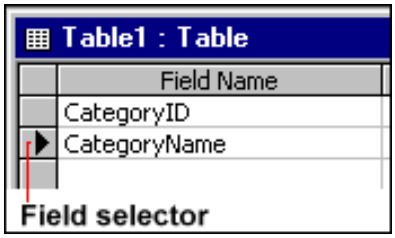
Modifications of this nature are done in table design view. Press the *Table Design* button on the toolbar to change to this view. The Access tutorial for Case 1 contains instructions for working in Table design view. We will be looking at creating primary keys in the next section of this tutorial, and at other table structure modifications in later *Solve it!* tutorials.

Setting a Primary Key

Access uses a unique tag called a *primary key* to identify each record in a table. As a licence plate is unique to car, or a fingerprint is unique to an individual, the primary key uniquely identifies each record. Every table should have at least one primary key. Not all fields are good candidates for primary keys. For instance, if you chose a LastName field as a primary key, your table would not be able to contain two records with the LastName "Jones". Some number fields make good primary keys. For example, an Employee ID number would uniquely identify each record in an Employee table.

To set a Primary Key:

Figure 4-6



1. In *Table Design* view click the *field selector* (refer Figure 4-6) for the intended primary key field. (If you need more than one primary key, select the wanted fields by holding down your Ctrl key as you click on the field selectors).

2. Click the Primary Key button on the toolbar. (A key symbol will appear on the field selector of every primary key field you highlight).

Note:	If you attempt to save a new table without creating a primary key, Access will prompt you to create one.

Creating Relationships Between Tables

Once you have created or attached or imported the tables that you want in your database, it's a good idea to establish relationships between them. You create relationships so you can view data from more than one table in the same report or form. (To create a relationship, you will need to have at least two tables in your database).

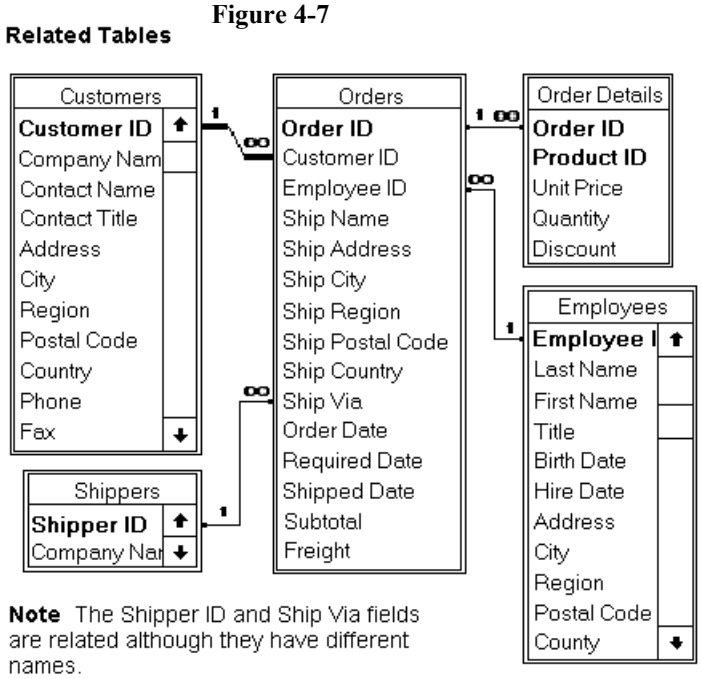
To Create Relationships:

1. From the Database Window, click the *Relationships* button on the toolbar. Access displays the Relationships window. If you are creating relationships for the first time, Access will display the Show Table dialog box. If this does not occur, click the Show Table button on the toolbar. In the Table/Query box, double click on the table or query you want, and Access will add the table to the Relationships window. Click Close.

2. Click a field in the first table. This will usually be the primary key (always displayed in bold). Hold down the left mouse button, and drag the field to a field containing the same type of data in another table. Access displays the Relationships dialog box with the field names filled in. Click the Create button, and Access draws a line to join the two related tables (see for example, Figure 4-7).

Note that relationships between tables can also be created at query design level. We will be doing this in a later tutorial.

Figure 4-7



Exiting from Access

To exit from Access, press the F11 key to return to the database window, and then select FILE/EXIT from the menu.

5

Database Management Software Cases

Access Case 1

Brennan Abrasives Inc.

Problem: Create and modify a sales support system.

Management Skill: Organize

Access Skills: Data Table Setup
Data Input and Editing
Selecting Data Subsets
Printing

Data Table: BRENNAN

Dolly Chen had always been competitive. As long as she could remember there was always something to strive for: better grades, faster times in her long distance running and outdoing her two brothers. It therefore gave her great satisfaction to learn last week that she was "Salesperson of the Year". She had sold more abrasives, grinding wheels and sandpaper last than any of the other 120 sales staff at Brennan Abrasives.

Brennan Abrasives was founded in Wisconsin after the First World War. The founders recognised the growing need by metal and wood manufacturers for abrasives in the finishing of their products. The business had performed solidly between the World Wars and grew especially fast after the Second World War, particularly during the 1980's when capital equipment expenditures boomed. Although this was curtailed by the economic downturn of the late 80's and early 90's, by the late 1990's Brennan was again experiencing growth as the economic situation eased.

Recent advances in computer controlled machinery, along with robotics and flexible manufacturing systems, provided a strong growth period for Brennan. The new machines required large quantities of high quality abrasives, particularly sandpaper belts. These expensive machine tools were used on a 24 hour basis. To minimise downtime and maintain quality high, sandpapers were changed frequently, often before wearing out.

Although very profitable, the abrasive supply industry had become very competitive. Brennan had many competitors who undercut them on particular items, but only 3M had as large a range of products. Brennan follows a high product quality and premium pricing strategy based on excellent customer service. Brennan's competitive advantage lies in three areas:

- a comprehensive product range,
- a hard working and knowledgeable sales force,
- guaranteed overnight delivery for stock items.

Dolly has been very successful in presenting Brennan as a provider of the full range of a client's abrasive needs. Rather than approach new companies she has concentrated on increasing the average purchase of existing clients.

In the five years Dolly Chen has been with Brennan, her rise to success has been fast. It seemed only yesterday that she started out as a trainee sales representative with old Lars "the gentleman" Hansen. Lars preached that there were only two important things in sales*: "get close to the customer and know the products inside-out."*

Dolly kept her client information in a large ledger stored alphabetically by the client's company name. She now had over 200 companies in her sales territory of Michigan, Indiana, Illinios, Wisconsin and Minnesota.

Within each company, she called on an average of four different groups of people in purchasing, engineering, production and the workshop. So she had nearly 800 names, in addition to details on the products they ordered and all sorts of other useful information. Dolly often asked about the client's wife or husband by name and always knew the pattern of the last few orders. If the orders were decreasing she asked why and found out if one of her competitors had been better able to meet the client's needs.

Dolly's ledger was becoming too heavy to carry, and she was always wanting to know things that were in the ledger but not easily accessible. For example, last month a new low cost sandpaper belt for soft woods was introduced. Dolly wanted to know which of her clients ordered the product it superseded. Then she could write or call on all those clients and demonstrate the new product. To find that information from her ledger would have taken a week of searching. So she relied on her memory and was sure some major users were overlooked.

Dolly had tried asking Brennan's overworked Information Systems department to get the information for her from past invoices, but the IS Manager replied *"we'd love to, but we can't even finish our own work and anyway I don't think it's possible. You'd be better off putting it on a Notebook computer using and using a Windows-based database package like Access or Paradox. Much more flexible, cheaper and portable also."*

One of the perks of winning the "Salesperson of the Year" award was a new ideas budget of $25,000 which Dolly has decided to devote to her new Sales Support System. Dolly has purchased a Pentium-based notebook computer and a Windows-based database package. She wants you to prepare a prototype system.

Dolly wants all of the important data to be entered into the database so she can look up information in many different ways, even while on the road. Then at night in her hotel room, she can update the database after visiting clients. Dolly feels certain the new system will enable her to keep improving her performance and maybe even win her "Salesperson of the Year" again.

The important fields for the prototype system are:

Company Name	**Client Name**	**Spouse Name**
Address	**Department**	**Order Pattern**
City	**Title**	**Company Number**
State	**Last Order Date**	
Zip	**Product Group**	

Brennan's products are classed into three major groups: grinding wheels (G), sandpaper (S), other abrasives (O). Within each group are up to 99 subgroups, i.e. S01 to S99. Order patterns are coded as either increasing (I), decreasing (D), or stable (S). Company numbers are 5 numerical digits long.

A portion of Dolly's database has been started for you in the data table BRENNAN in SOLVEIT.MDB. Create a new Access database and import this object now.

Tasks: There are seven tasks in this case:

1. Complete the data table structure to include the information desired by Dolly.

2. (a) Enter data for the new fields to complete the existing records. (b) Enter data for three new clients into the data table.

3. Drew Dent, the Maintenance Manager for Outboard Motors in Port Huron, MI, has retired and has been replaced by Sam Jackson. Sam's wife's name is Gina, and he will now be known as the Engineering Director. Update the data table to reflect these changes.

4. Print a listing of the entire data table.

5. Print all of the records in the table but only include the following fields: Company, City, Client, Last Order Date.

6. Dolly is travelling to Port Huron to meet Sam Jackson at Outboard Motors and wants to know her other clients in Port Huron. Create a query and then print a listing of all clients based in Port Huron. The list should only include the fields: Company, Client and Title.

*7. After using the client database for three months, Dolly reports tremendous savings in time and is able to do her job better. However, Dolly has found the database does not handle customers that buy more than one product group. For example, 3M purchases products from the G10, S20, S35, S40, S30, O50 and O70 product groups. Find a way to incorporate this extra information into the database.

Time Estimates (excluding unstructured * problem)

Expert:	30 minutes
Intermediate:	1 hour
Novice:	1.5 hours

Tutorial For Database Case 1 Using Access 97

In order to learn the skills needed for this case, and those following, you will need to:

1. Create a new empty database called FRIENDS.MDB. The instructions on how to do this are provided in Chapter 4.

2. Import the FRIENDS table object provided in SOLVEIT.MDB on your *Solve it!* disk to your new database. The instructions on how to do this are also provided in Chapter 4. The FRIENDS.MDB database is used for most Access tutorials in *Solve it!* **This tutorial and those following assumes this database has been created.**

Remember that:
Double click means to press the left mouse button twice in rapid succession.
Click means to press the left mouse button once only.

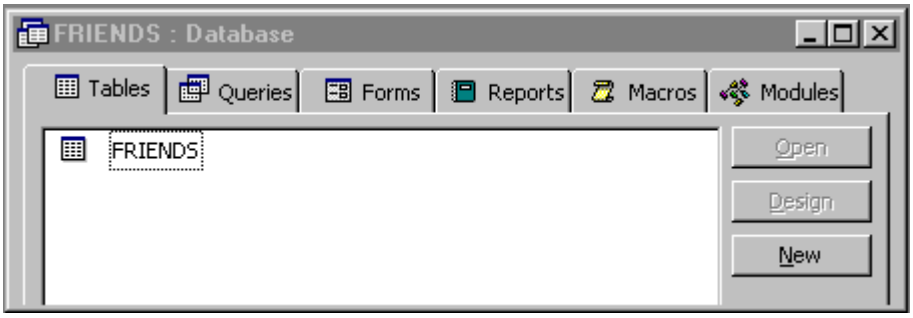
You can use the *Office Assistant* online coach (click the **?** button) at any time to provide assistance with most Access procedures.

To start Access:
1. Enter the Windows environment, and click on START to open the Start Menu (if not automatically displayed), then point to PROGRAMS, followed by MICROSOFT ACCESS in the sub-menu. Sometimes the Microsoft Access program is within a sub-menu of MICROSOFT OFFICE. If you are unsure how to enter Windows, your instructor will assist.

2. Click to load the Access program.

3. If the OPEN DATABASE window does not appear, then using your mouse, select FILE/OPEN DATABASE from the menu or double click on the *Open Database* button shown on the right.

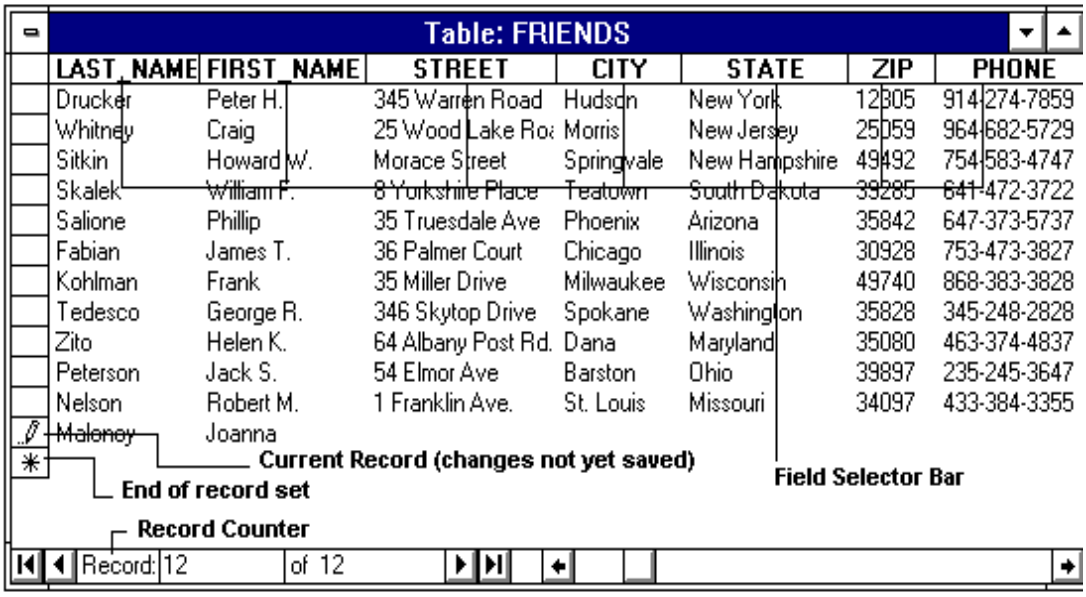
Figure 5-1



4. From the OPEN DATABASE window type in the path and filename eg: A:\FRIENDS.MDB and click on OK. This action will result in display of the FRIENDS database window. Refer Figure 5-1.

5. Double click on the FRIENDS *Table* object highlighted in the Database Window, to open the table and display the default datasheet view. Refer Figure 5-2.

Figure 5-2 Access Datasheet View



A short explanation for each of the object types shown in the Database Window is provided in Chapter 4.

Access displays table data in columns and rows, similar to a spreadsheet. Each row in an Access table represents a *record*, and each column represents a *field*. For example, Figure 5-2 shows that there are 12 records in the FRIENDS table, and that the *record counter* is currently on record 12 of 12. Above the record rows is the *field selector* bar containing columns of fieldnames. Fields describe the various categories of information (eg: LastName, FirstName) that make up a record.

Table data can displayed in two ways. Figure 5-2 shows the default *datasheet* view, represented by the toolbar button shown at the right. Datasheets are used for displaying, editing, adding and deleting records, and for simple table view printing.

Later on in this tutorial, we will also be using the *table design* view, represented by the toolbar button shown at the right. This is used for adding, editing and deleting fields, and for defining field properties and indexes.

Navigating within a Datasheet

Use your *Tab* key or click with your mouse, to move to a particular column within a record. Use the horizontal and vertical scroll bars to practice viewing records and fields in the datasheet. Use the *goto record* keys shown in Figure 5-3 to move to specific record positions on the datasheet. Note also in Figure 5-3, the symbols used for indicating current records and the end of a record set.

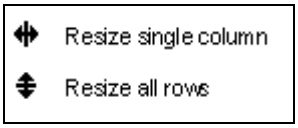
Figure 5-3

Resizing Columns and Rows in an Access Datasheet

To resize a column (field), position your mouse on the right hand side of a column at the field selector level (eg: LastName) until the *resize column* symbol shown in Figure 5-4 appears. Double click to automatically resize the width and display all data in that column. Alternatively, select FORMAT and COLUMN WIDTH from the menu.

Record rows can be resized by choosing FORMAT/ROW HEIGHT from the menu, or by positioning the mouse pointer on the border of any record selector, until the *resize rows* symbol shown in Figure 5-4 appears. Click and drag either up or down to make rows taller or narrower. *Warning: unlike column resizing, these actions resize <u>all</u> rows*

Figure 5-4



Saving Datasheet Layout Changes

From the FILE menu, choose SAVE

Printing a Datasheet

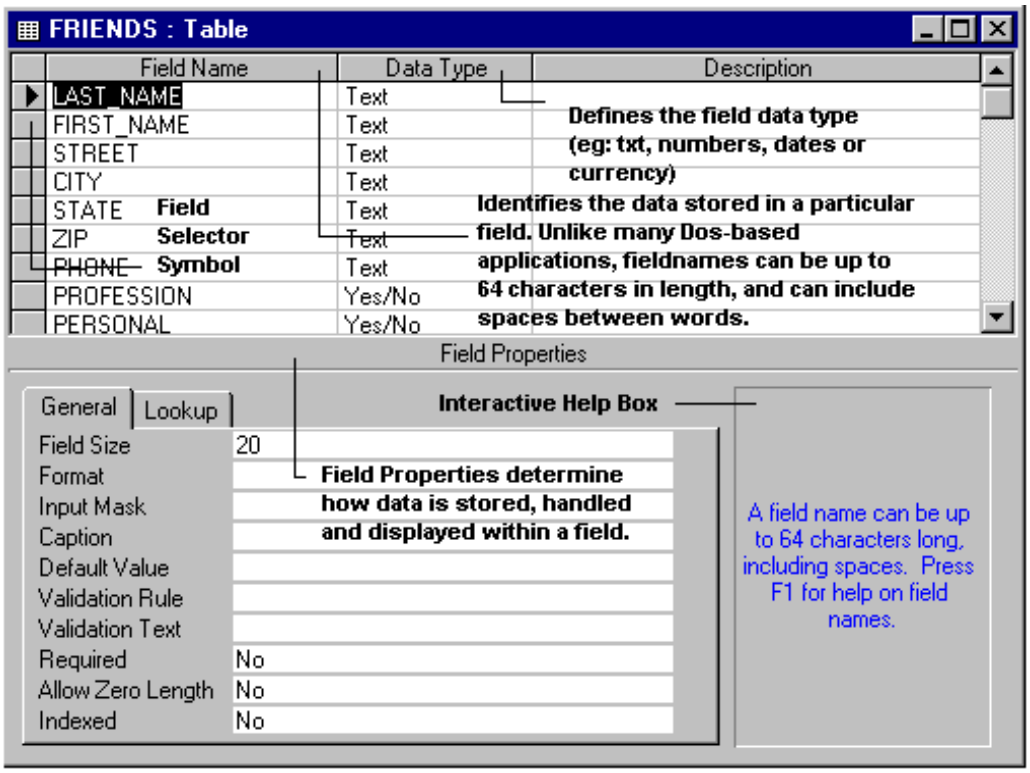
With the table open in datasheet view, choose FILE/PRINT from the menu or click the *Print* button shown on the toolbar. This will display the Print Dialog box. Choose between portrait or landscape orientations. Click on OK and Access will print the table.

How to Change the Structure of an Access Table

Using your mouse, select VIEW/DESIGN VIEW from the menu or click on the *Design View* button to display the field structure of the table. See Figure 5-5.

The Table Design view is used for establishing or modifying the field structure of a table. As you move around this window, the *help box* messages change to describe the purpose of the various elements in the window. Let's add a new field called TITLE to the table in order to store titles like Mrs, Ms, Mr and Dr.

Figure 5-5 Table Design View



1. This field needs to be added before the Last_Name field. Position your mouse on the *field selector* for the Last_Name field and click once to select and display the field select symbol (refer Figure 5-5). Choose EDIT/INSERT ROW from the menu, or click once on the *Insert Row* button shown at the right to insert a blank row above the LastName field.

2. Click in the blank *Field Name* column and type the word TITLE.

3. Click in the blank *Data Type* column and choose the field type. In this instance, the default *Text* field type is the one we need. Notice however, the ↓ button at the far right of this column. Clicking on this activates a drop down list of other data types to select from. Recap the Access data types section in Chapter 4 for an explanation of these.

4. Click on the *Field Size* bar in the Field Properties box. The default width for text fields is 50 characters. Since our new field will only contain abbreviated terms, change the field size to 5 characters.

5. Select FILE/SAVE from the menu or click on the *Save* button to save the changes we have made.

6. Select VIEW/DATASHEET or click on the *Datasheet* button to return to the datasheet view

How to Enter New Records

Change to datasheet view by selecting VIEW/DATASHEET from the menu or by clicking on the Datasheet button from the toolbar. Scroll down to the end of the record set indicated by the ✳ symbol, and click inside the first empty field (eg: Title). Add one record to the FRIENDS table. Move between fields by using your Tab key. Press Tab to move to the first field of the next record row. Access will automatically save your data when you leave a record, either by moving to another record or by closing the table.

How to Edit Existing Records

In datasheet view, move to the empty Title field created earlier in this tutorial and add appropriate titles for each of the records in the FRIENDS table. Click in the Title field of the first record (ie: Peter Drucker), and enter the title Dr. Notice that the pencil symbol appears in the record selector, indicating that this is the current record. Press the ↓ cursor key to move to the next record within the same field, and add titles for the remaining records. As you move to different records, the pencil symbol will move with you and Access will automatically save any changes changes made.

Editing Tips:

- Press the *Esc* key to undo changes made to the current field or record.
- Press the *Tab* or *Shift+Tab* keys to move to the next or previous field within a record.

To Close a Table

Select FILE/CLOSE from the menu, or double click the Control Menu box in the upper left hand corner of the table window. If you have unsaved changes in either datasheet or table design view, Access will prompt you to save the changes. Choose the *Yes* button.

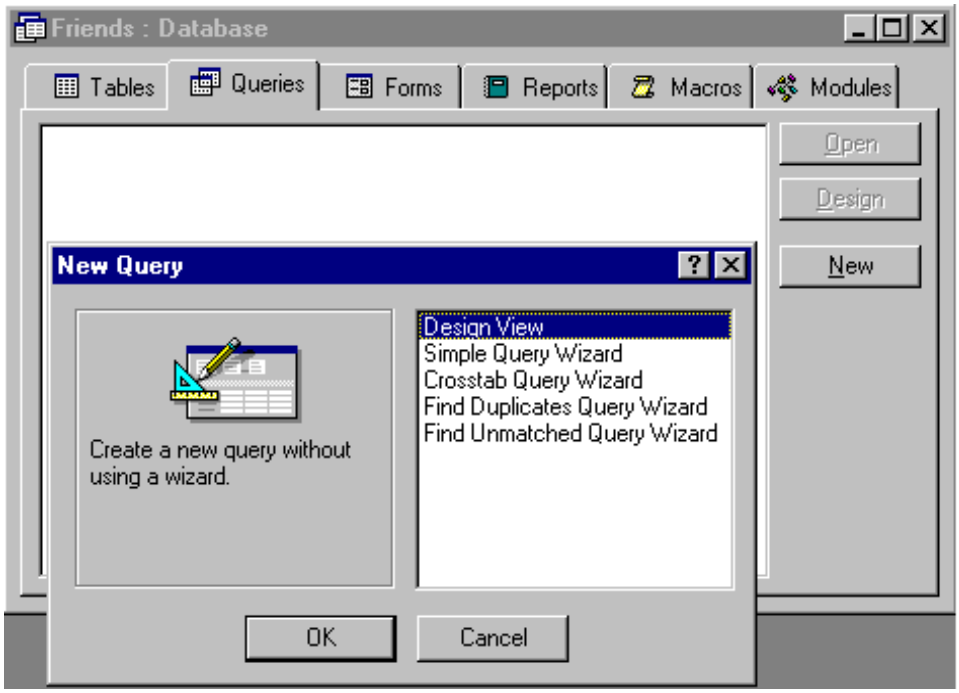
How to Display and Print Selected Fields: Building a Query 1

A query is a question that is asked about certain data in a database, such as "How many customers live in Melbourne?" or "What were our sales figures for last month?". Alternatively, you may just want to view selected fields of a database in a certain order. Access' *Select Query* window (see Figure 5-6) is a graphical query-by-example (QBE) tool. This means you can use a mouse to select and drag and manipulate objects in the query window to define an example of the records you wish to see. Access queries are very simple and quick to create, and can include data from several different databases.

Let's say you want display and print a report of just three fields from the FRIENDS table: Last_Name, First_Name and City. Here's how to set up an Access query to do this.

1. From the Database Window (see Figure 5-6), click once on *Query*, and then click once on the *New* button. (If you still have the FRIENDS table open, press the F11 function key to return to the Database window). This activates the New Query box. Click on the *Design View button*, and Access will open up a new query window. (The *Query Wizards* buttons in the New Query box are used for performing specialised queries such as crosstabs, or for copying records from an existing table to a new one).

Figure 5-6



Like most Access windows, the query function has its own set of toolbar buttons, menu bar, and other options. The various parts of the query window itself are shown in Figure 5-7.

2. From the *Add Table* box that appears in front of the query window, click and highlight FRIENDS, and then click on the *Add* button. This action adds a field list from the FRIENDS table to the query window (see Figure 5-7). Click the *Close* button to close the Add Table box.

You are Writing a Program Already !!

3. Using your mouse click and highlight the Last_Name field on the FRIENDS field list. Hold down your Ctrl key, and click and highlight First_Name. Still holding down the Ctrl key, click and highlight the City Field. (The Ctrl key is handy to use when selecting discontiguous fields for a query). Release the Ctrl key.

4. Click anywhere in the highlighted area of the field list, and then holding down your mouse button, drag the three fields down to the *Field* row in the *QBE grid* (see Figure 5-7). Release the mouse button.

5. Click on the Run button  or select QUERY/RUN from the menu to generate the results of your query.

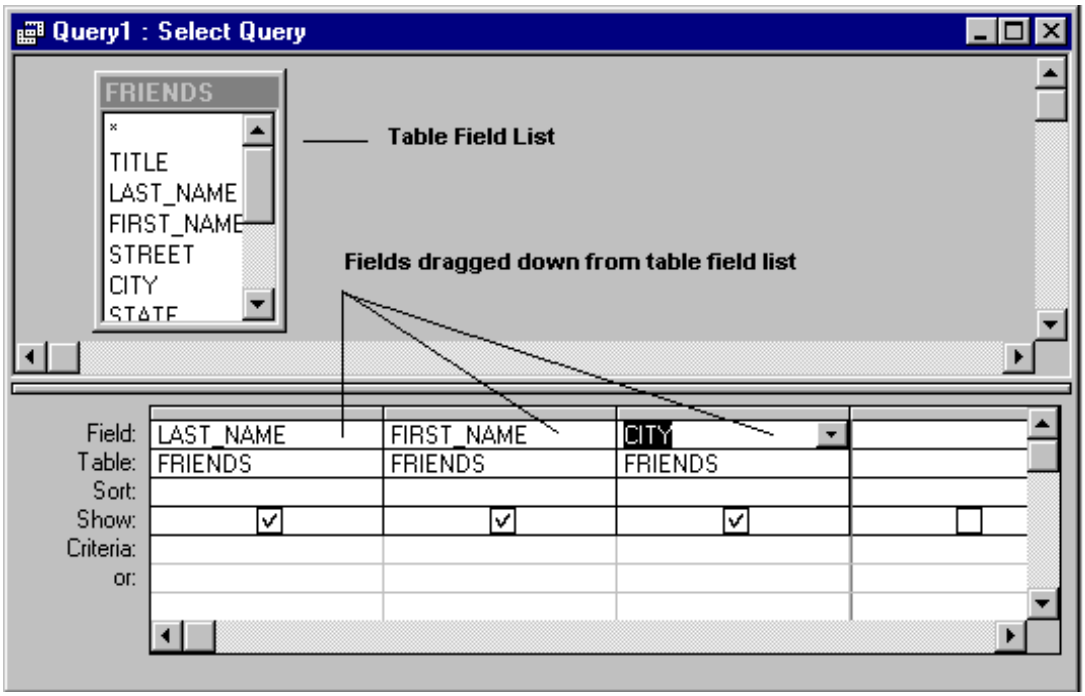
6. Click on the Print button  or choose FILE/PRINT to send your query to print.

7. Click on the Design View  button to return to the query design window.

8. Save your query by clicking on the Save button  or selecting FILE/SAVE from the menu. This will activate the Save As box. Type in a name (eg: *Tute1 Query1*) for your query and click OK.

Click on the Structured Query Language button  and Access will display the Access Basic program code generated by the query we have just created. This button is located by clicking on the drop down arrow next to the design view/datasheet view button at the left hand top of the screen.

Figure 5-7 Access Query Design Window




```
SELECT DISTINCTROW FRIENDS.LASTNAME, FRIENDS.FIRSTNAME, FRIENDS.CITY
FROM FRIENDS;
```

This code can be cut and pasted into an event procedure within a macro or module, and used as part of a larger program. Access provides a number of useful programming shortcuts. We will be looking at some of these in later cases.

How to Display Selected Records : Building a Query 2

Having constructed a simple query, let's go one step further and search the FRIENDS table for records meeting a particular criteria. In this example, let's extract only those records which contain "Mr" in the Title field.

1. In *Query Design* view, click and highlight the Title field in the FRIENDS field list. Holding down your mouse button, drag the highlighted field down to the Last_Name field in the QBE Grid, and release the mouse button. Notice how Access automatically moves the existing fields to the right to accomodate inclusion of the new field.

2. In the *Criteria:* bar of the Title field, type Mr and then click on the *Run* query button. Notice that Access automatically encloses your text in quotation marks, indicating that it is searching for text within a field. Unlike many DOS-based database packages, Access is not case sensitive. Entering Mr in capitals (or in lower case) would produce the same search result. Figure 5-8 displays the results of the search.

Figure 5-8 Query Results

Original query showing all records

Query1 : Select Query

LAST_NAME	FIRST_NAME	CITY
Drucker	Peter H.	Hudson
Fabian	James T.	Chicago
Kohlman	Frank	Milwaukee
Nelson	Robert M.	St. Louis
Peterson	Jack S.	Barston
Salione	Phillip	Phoenix
Sitkin	Howard W.	Springvale
Skalek	William F.	Teatown
Tedesco	George R.	Spokane
Whitney	Craig	Morris
Zito	Helen K.	Dana

Record: 1 of 11

Query after applying a selection criteria

friends1 : Select Query

TITLE	LAST_NAME	FIRST_NAME	CITY
Mr	Tedesco	George R.	Spokane
Mr	Kohlman	Frank	Milwaukee
Mr	Fabian	James T.	Chicago
Mr	Sitkin	Howard W.	Springvale
Mr	Whitney	Craig	Morris

Record: 1 of 5

3. To print your new query, click on the *Print* button.

4. Save your new query with a new name (eg: *Tute1 Query2*) by choosing the FILE/Save As/Export… option from the menu. Select the radio button *Save Query within the Current Database as New Name*, then click OK. You have now created two simple queries. We will be constructing more complex queries in later cases. Click on the SQL button, to see the effect that applying a search criteria has had on your initial programming code.

5. Press F11 to return to the Database Window. Exit Access by selecting FILE/EXIT from the menu.

Tutorial For Database Case 1 Using Access 2.0

In order to learn the skills needed for this case, and those following, you will need to use the practice database CONTACTS.MDB created in Chapter 4. If you did not create this database, use the sample database FRIENDS.MDB provided on your *Solve it!* disk which has a similar structure. The FRIENDS database is used for most Access tutorials in *Solve it!*

Remember that:
Double click means to press the left mouse button twice in rapid succession.
Click means to press the left mouse button once only.

You can use the *Cue Cards* online coach (select HELP/CUE CARDS from the menu) at any time to provide assistance with most Access procedures.

To start Access:

1. Enter the Windows environment, and double click to open the Program Manager (if not automatically displayed). If you are unsure how to enter Windows, your instructor will assist.

2. Double click to open the Access program group (or Microsoft Office program group), and then double click once again on the Access application icon (as shown at the top of this page). This will load the Access program.

3. Using your mouse, select FILE/OPEN DATABASE from the menu or double click on the *Open Database* button shown on the right.

4. From the OPEN DATABASE window type in the path and filename eg: A:\FRIENDS.MDB and click on OK. This action will result in display of the FRIENDS Database Window. Refer Figure 5-9.

5. Double click on the FRIENDS *Table* object highlighted in the Database Window, to open the table and display the default datasheet view. Refer Figure 5-10.

A short explanation for each of the object types shown in the Database Window is provided in Chapter 4.

Access displays table data in columns and rows, similar to a spreadsheet. Each row in an Access table represents a *record*, and each column represents a *field*. For example, Figure 5-10 shows that there are 12 records in the FRIENDS table, and that the *record counter* is currently on record 12 of 12. Above the record rows is the *field selector* bar containing columns of fieldnames. Fields describe the various categories of information (eg: LastName, FirstName) that make up a record.

Figure 5-9

Table data can displayed in two ways. Figure 5-10 shows the default *datasheet* view, represented by the toolbar button shown at the right. Datasheets are used for displaying, editing, adding and deleting records, and for simple table view printing.

Later on in this tutorial, we will also be using the *table design* view, represented by the toolbar button shown at the right. This is used for adding, editing and deleting fields, and for defining field properties and indexes.

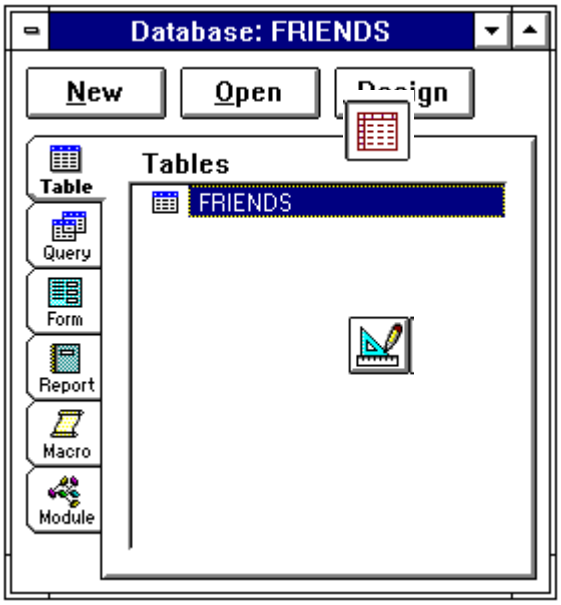
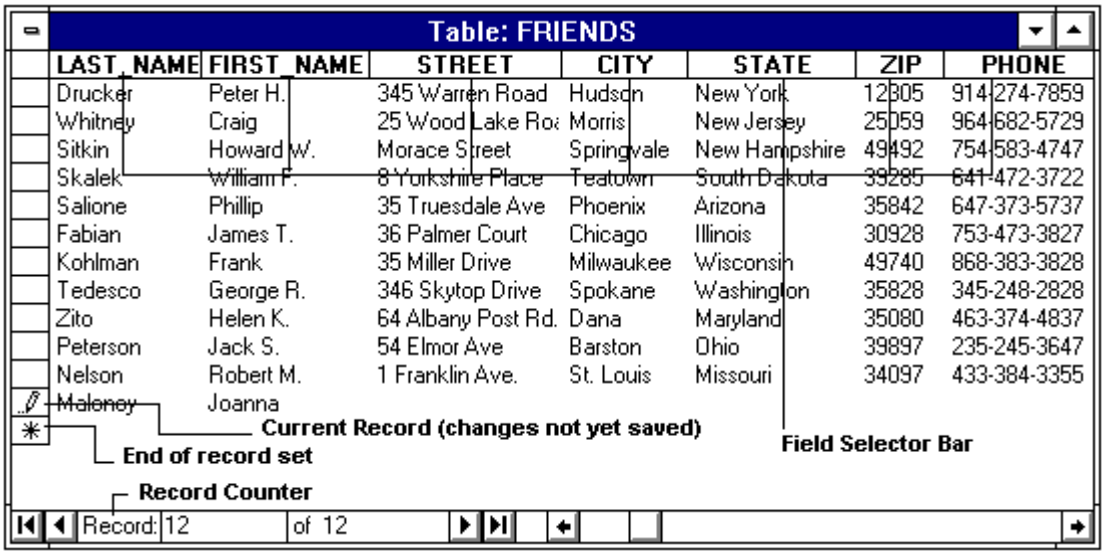


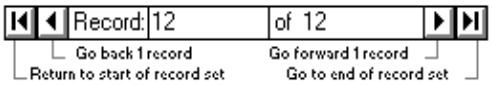
Figure 5-10 Access Datasheet View



Navigating within a Datasheet

Use your *Tab* key or click with your mouse, to move to a particular column within a record. Use the horizontal and vertical scroll bars to practice viewing records and fields in the datasheet. Use the *goto records keys record* keys shown in Figure 5-11 to move to specific record positions on the datasheet. Note also in Figure 5-3, the symbols used for indicating current records and the end of a record set.

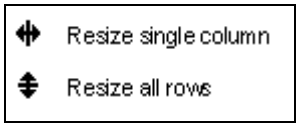
Figure 5-11



Resizing Columns and Rows in an Access Datasheet

To resize a column (field), position your mouse on the right hand side of a column at the field selector level (eg: LastName) until the *resize column* symbol shown in Figure 5-12 appears. Double click to automatically resize the width and display all data in that column. Alternatively, select FORMAT and COLUMN WIDTH from the menu.

Figure 5-12



Record rows can be resized by choosing FORMAT/ROW HEIGHT from the menu, or by positioning the mouse pointer on the border of any record selector, until the *resize rows* symbol shown in Figure 5-12 appears. Click and drag either up or down to make rows taller or narrower. *Warning: unlike column resizing, these actions resize all rows*

Saving Datasheet Layout Changes

From the FILE menu, choose SAVE TABLE

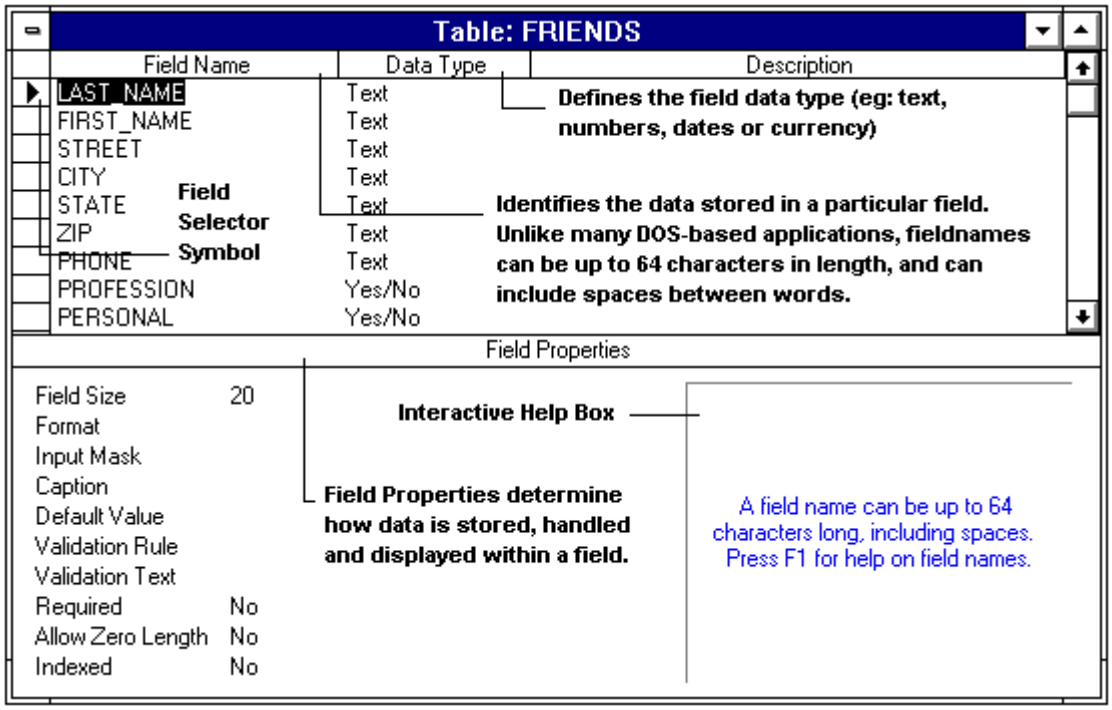
Printing a Datasheet

With the table open in datasheet view, choose FILE/PRINT from the menu or click the *Print* button shown on the toolbar. This will display the Print Dialog box. Choose between portrait or landscape orientations. Click on OK and Access will print the table.

How to Change the Structure of an Access Table

Using your mouse, select VIEW/TABLE DESIGN from the menu or click on the *Design View* button to display the field structure of the table. See Figure 5-13.

Figure 5-13 Table Design View



The Table Design view is used for establishing or modifying the field structure of a table. As you move around this window, the *help box* messages change to describe the purpose of the various elements in the window. Let's add a new field called TITLE to the table in order to store titles like Mrs, Ms, Mr and Dr.

1. This field needs to be added before the Last_Name field. Position your mouse on the *field selector* for the Last_Name field and click once to select and display the field select symbol (refer Figure 5-13). Choose EDIT/INSERT ROW from the menu, or click once on the *Insert Row* button shown at the right to insert a blank row above the LastName field.

2. Click in the blank *Field Name* column and type the word TITLE.

3. Click in the blank *Data Type* column and choose the field type. In this instance, the default *Text* field type is the one we need. Notice however, the ↓ button at the far right of this column. Clicking on this activates a drop down list of other data types to select from. Recap the Access data types section in Chapter 4 for an explanation of these.

4. Click on the *Field Size* bar in the Field Properties box. The default width for text fields is 50 characters. Since our new field will only contain abbreviated terms, change the field size to 5 characters.

5. Select FILE/SAVE from the menu or click on the *Save* button to save the changes we have made.

6. Select VIEW/DATASHEET or click on the *Datasheet* button to return to the datasheet view 

How to Enter New Records

Change to datasheet view by selecting VIEW/DATASHEET from the menu or by clicking on the Datasheet button from the toolbar. Scroll down to the end of the record set indicated by the ✳ symbol, and click inside the first empty field (eg: Title). Add one record to the FRIENDS table. Move between fields by using your Tab key. Press Tab to move to the first field of the next record row. Access will automatically save your data when you leave a record, either by moving to another record or by closing the table.

How to Edit Existing Records

In datasheet view, move to the empty Title field created earlier in this tutorial and add appropriate titles for each of the records in the FRIENDS table. Click in the Title field of the first record (ie: Peter Drucker), and enter the title Dr. Notice that the pencil symbol appears in the record selector, indicating that this is the current record. Press the ↓ cursor key to move to the next record within the same field, and add titles for the remaining records. As you move to different records, the pencil symbol will move with you and Access will automatically save any changes changes made.

Editing Tips:

- Press the *Esc* key to undo changes made to the current field or record.
- Press the *Tab* or *Shift+Tab* keys to move to the next or previous field within a record.

To Close a Table

Select FILE/CLOSE from the menu, or double click the Control Menu box in the upper left hand corner of the table window. If you have unsaved changes in either datasheet or table design view, Access will prompt you to save the changes. Choose the *Yes* button.

How to Display and Print Selected Fields: Building a Query 1

A query is a question that is asked about certain data in a database, such as "How many customers live in Melbourne?" or "What were our sales figures for last month?". Alternatively, you may just want to view selected fields of a database in a certain order. Access' *Select Query* window (see Figure 5-14) is a graphical query-by-example (QBE) tool. This means you can use a mouse to select and drag and manipulate objects in the query window to define an example of the records you wish to see. Access queries are very simple and quick to create, and can include data from several different databases.

Let's say you want display and print a report of just three fields from the FRIENDS table: Last_Name, First_Name and City. Here's how to set up an Access query to do this.

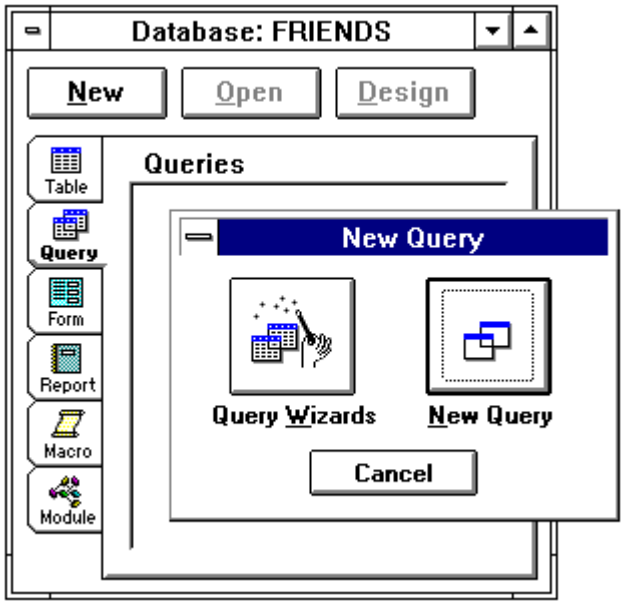
1. From the Database Window (see Figure 5-14), click once on *Query*, and then click once on the *New* button. (If you still have the FRIENDS table open, press the F11 function key to return to the Database window). This activates the New Query box. Click on the *New Query button*, and Access will open up a new query window. (The *Query Wizards* button in the New Query box is used for performing specialised queries such as crosstabs, or for copying records from an existing table to a new one).

Like most Access windows, the query function has its own set of toolbar buttons, menu bar, and other options. The various parts of the query window itself are shown in Figure 5-15.

2. From the *Add Table* box that appears in front of the query window, click and highlight FRIENDS, and then click on the *Add* button. This action adds a field list from the FRIENDS table to the query window (see Figure 5-15). Click the *Close* button to close the Add Table box.

3. Using your mouse click and highlight the Last_Name field on the FRIENDS field list. Hold down your Ctrl key, and click and highlight First_Name. Still holding down the Ctrl key, click and highlight the City Field. (The Ctrl key is handy to use when selecting discontiguous fields for a query). Release the Ctrl key.

Figure 5-14



4. Click anywhere in the highlighted area of the field list, and then holding down your mouse button, drag the three fields down to the *Field* row in the *QBE grid* (see Figure 5-15). Release the mouse button.

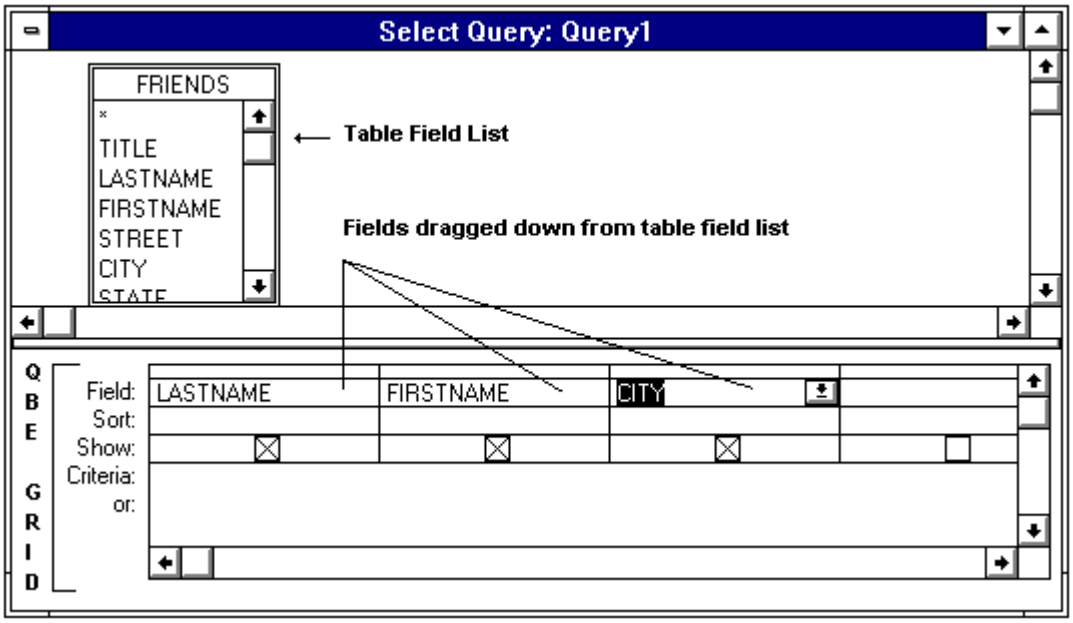
5. Click on the Run button  or select QUERY/RUN from the menu to generate the results of your query.

6. Click on the Print button  or choose FILE/PRINT to send your query to print.

7. Click on the Design View  button to return to the query design window.

8. Save your query by clicking on the Save button  or selecting FILE/SAVE from the menu. This will activate the Save As box. Type in a name (eg: *Tute1 Query1*) for your query and click OK.

Figure 5-15 Access Query Design Window



You are Writing a Program Already !!

Click on the Structured Query Language button SQL and Access will display the Access Basic program code generated by the query we have just created.

```
SELECT DISTINCTROW FRIENDS.LASTNAME, FRIENDS.FIRSTNAME, FRIENDS.CITY
FROM FRIENDS;
```

This code can be cut and pasted into an event procedure within a macro or module, and used as part of a larger program. Access provides a number of useful programming shortcuts. We will be looking at some of these in later cases.

How to Display Selected Records : Building a Query 2

Having constructed a simple query, let's go one step further and search the FRIENDS table for records meeting a particular criteria. In this example, let's extract only those records which contain "Mr" in the Title field.

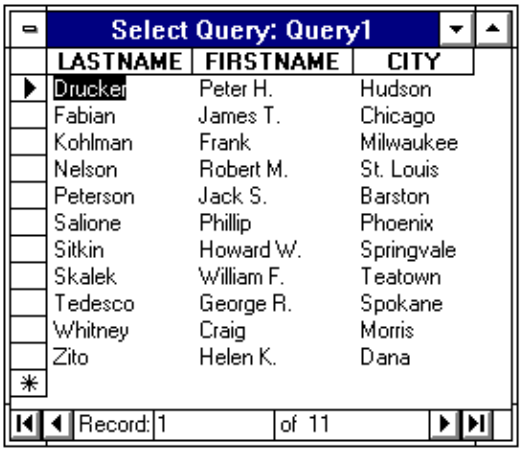
1. In *Query Design* view, click and highlight the Title field in the FRIENDS field list. Holding down your mouse button, drag the highlighted field down to the Last_Name field in the QBE Grid, and release the mouse button. Notice how Access automatically moves the existing fields to the right to accomodate inclusion of the new field.

2. In the *Criteria:* bar of the Title field, type Mr and then click on the *Run* query button. Notice that Access automatically encloses your text in quotation marks, indicating that it is searching for text within a field. Unlike many DOS-based database packages, Access is not case

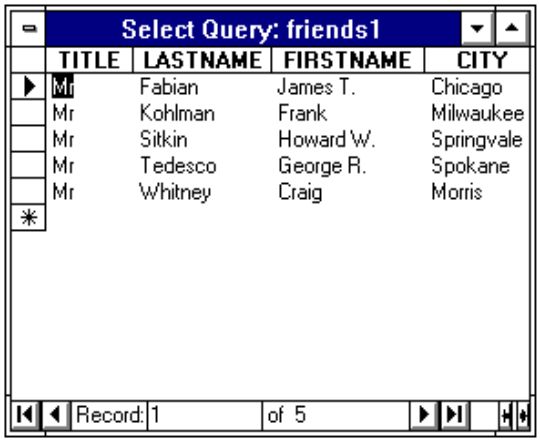
sensitive. Entering Mr in capitals (or in lower case) would produce the same search result. Figure 5-16 displays the results of the search.

Figure 5-16 Query Results

Original query showing all records



Query after applying a selection criteria



3. To print your new query, click on the *Print* button.

4. Save your new query with a new name (eg: *Tute1 Query2*) by choosing the FILE/SAVE AS option from the menu. You have now created two simple queries. We will be constructing more complex queries in later cases. Click on the SQL button, to see the effect that applying a search criteria has had on your initial programming code.

5. Press F11 to return to the Database Window. Exit Access by selecting FILE/EXIT from the menu.

Database Case 2

Capstone Associates - Attorneys-at-Law

Problem: Construct a Personnel Database

Management skill: Coordinate

Access Skills: Data Table Setup
Data Input
Select Queries
Report Design
Printing

Data Table: CAPSTONE

Capstone Associates (CA) is a large legal practice with offices in San Francisco, Los Angeles, Carson City, Salt Lake City and Phoenix, which specializes in corporate law. CA's Los Angeles office employs more than 300 freelance and contract personnel, including expert witnesses (EW) from a wide variety of fields and disciplines, private investigators (PI), process servers (PS) and legal researchers (LR). CA hires these people on an as needed basis to assist in the conduct of cases on behalf of the many client corporations the company represents.

CA has 39 Principal attorneys who are also the main partners of the firm. Each Principal has a staff of assistant lawyers. In addition, each Principal keeps a separate list of contractors and freelancers they have successfully worked with in the past. Typically, these lists reside on 3" x 5" cards at the back of a drawer in each partner's desk. Whenever a client's case requires certain outside expertise, the partners reach for these cards and select their favorite people.

Because the partners are paid in part on performance, the more successful they are on cases, the more remuneration they receive. Moreover, the more and better freelancers and contractors they know, the more clients will seek them out for assistance. In this setting, there is a natural tendency to conserve the good freelancers and contractors for one's own accounts. However, there is also the possibility that by sharing this information, everyone would benefit.

Andrew Capstone, Managing Partner of CA, started the law firm 30 years ago, and he knows how the freelancing system works. The problem is that each attorney hoards the people they think are really excellent. There is no sharing of the talent pool. Sometimes second rate people are used on a case because only one or two attorneys know first rate persons. If a partner hires a poor freelancer, and the case verdict goes badly, the client may move to another legal firm and CA's profits and reputation may suffer.

Capstone wants to establish some sort of centralized repository for this information. Too much time is wasted searching for people with particular skills or knowledge that other partners may already know about. The central repository could also contain comments on the performance of the freelancer or contractor on prior occasions and other relevant information. Perhaps some sort of scoring system could be developed so that freelancers and contractors could be rank ordered. That way only the best people would be used to assist the attorneys on cases.

The other partners do not like this idea. One of the ways they look good is by knowing highly capable people on the outside. Sharing the name of good freelancers and contractors with other partners may mean these people will not be available when they need them. On the other hand, a central repository might reveal valuable resources they do not know about.

With somewhat mixed feelings the partners respond to a memo sent out by Capstone, asking them to list the information they would like to have in a central database.

The following fields were identified by summarizing memo responses received: Name, Skill, Hourly Rate, the number of hours employed for each quarter in 1998, and Phone Number. Capstone reviewed the responses to the memo and wanted to add to the database a place for text comments that would incorporate a simple rating system from "excellent" to "good", "ok", and "poor". The last partner the freelancer or contractor worked with could input these details.

Capstone did not know if free form text could be attached to everyone's record in the database, but he knew any software which did not have this capability was not too useful. At least with a centralized manual system on 3" x 5" cards, short notes could be attached to each person's record.

Parts of the database have been identified in the data table CAPSTONE in SOLVEIT.MDB. Create a new Access database and import this object now.

Tasks: There are seven tasks in this case:

1. Complete the design of the database to carry the information specified by Capstone. Be sure to devise some way to record free-form text comments, remarks or ratings from the attorneys.

2. (a) Fabricate and enter hours worked and comments for the existing freelancers or contractors. (b) Enter data for three new freelancers or contractors such as: private investigators (PI), process servers (PS), legal researchers (LR) and expert witnesses (EW).

   (c) Enter data for at least the first three freelancers or contractors in the comments field. Make sure that the author of the comments, the partner, is listed in the field. You will have to make up the name of the partner contributing the remarks.

3. (a) Print out the entire table. (b) Create a query and then print out a listing showing only the last name and comment fields.

4. Design and create a report to list all the freelancers and contractors with just the following fields: Last_Name, Hourly_Rate, and the total number of hours worked in 1998.

5. (a) Produce and print a report of process servers (PS) who charge more than $120.00 per hour. (b) Also include in this report the total number of hours this particular group of freelancers worked for CA in 1998. (c) Print the report.

*6. There are a number of areas where this system can be enhanced and improved. Make a list of all the improvements you would make and then pick one improvement and implement it in the database.

*7. Currently with the comments in a memo field, no analysis or manipulation may be performed upon the ratings of the freelancers. There may also be conflicting opinions on the merits of any one freelancer, if they have worked for more than one partner. Devise a method to accommodate these opinions from different attorneys in such a way that meaningful analysis may be performed.

Time Estimates (excluding task marked with *):

Expert:	45 minutes
Intermediate:	1.25 hours
Novice:	2 hours

Tutorial For Database Case 2 Using Access 97

You have already learned in the previous case how to use the table design view to change the structure of a database and add new fields, and to use the datasheet for adding new records and editing existing ones. This case introduces the *memo data type* and *report design*. Load the file you used for the Case 1 tutorial (CONTACTS.MDB or FRIENDS.MDB) to practice the skills you will need for this case.

Using Memo Fields

Memo fields are very useful in databases for storing free form text and notes, or annotations against particular records. Unlike DOS-based database packages such as dBase III+, creating, entering and saving data in an Access memo field is a very simple procedure. You can store up to 64,000 characters in an Access memo field. Because of their unstructured nature, memo fields cannot be *indexed* or *sorted*, but they can be searched. We will be looking at indexing, sorting and searching in later cases.

1. From the Database Window, click on the Table object and double click on either the ADDRESS or FRIENDS table. From the default datasheet view, click on the table design toolbar button or select VIEW/TABLE DESIGN from the menu.

2. Let's add a new field to the end of the existing database. The name for the new field will be "Comments", and the data type will be Memo. Recap the tutorial for Case 1, if you are unsure how to add new fields and select data types. Notice that the Field Size bar in the properties box does not appear when you select a memo data type, indicating the unstructured nature of this field. Save your changes to the table structure by clicking on the *Save* toolbar button, or selecting FILE/SAVE from the menu.

3. Return to datasheet view by clicking on the datasheet toolbar button, or selecting VIEW/DATASHEET from the menu. Scroll or Tab over to the new Comments field, and enter text against each record (eg: whether the person in question is a Personal or Professional friend). Press the ↓ cursor key on your keyboard to move to the next record. Remember that as you move to a different record, Access will automatically save any changes.

Printing Memo Fields

Memo fields can be printed like any other field in a database. For a quick report including memo fields:

1: From datasheet view, press the F11 key to return to the Database window. Click the *Query object* and click *New*. From the New Query box, click on the *New Query* button, and Access will open up a new query window. From the Add Table box click and highlight the FRIENDS table, and then click *Add*, and then click *Close*.

2. From the FRIENDS *field list*, select and drag the LastName and Comments fields down onto the Field bar in the *QBE Grid* of the query window.

3. Click on the *Run* toolbar button or select QUERY/RUN from the menu to display the contents of the Last Name and Comments fields print (see Figure 5-17)

4. Click on the *Print* button or choose FILE/PRINT to send your query to. Select FILE/CLOSE from the menu and click *No* to the Save Query request to return to the Database Window without saving the query.

Figure 5-17

Query1 : Select Query

LAST_NAME	COMMENTS
Drucker	Professional
Fabian	Professional
Kohlman	Father-in-Law
Nelson	Professional
Peterson	Personal
Salione	Personal
Sitkin	Professional
Skalek	Professional
Tedesco	Professional
Whitney	Personal
Zito	Professional

Record: 1 of 11

Creating Reports in Access

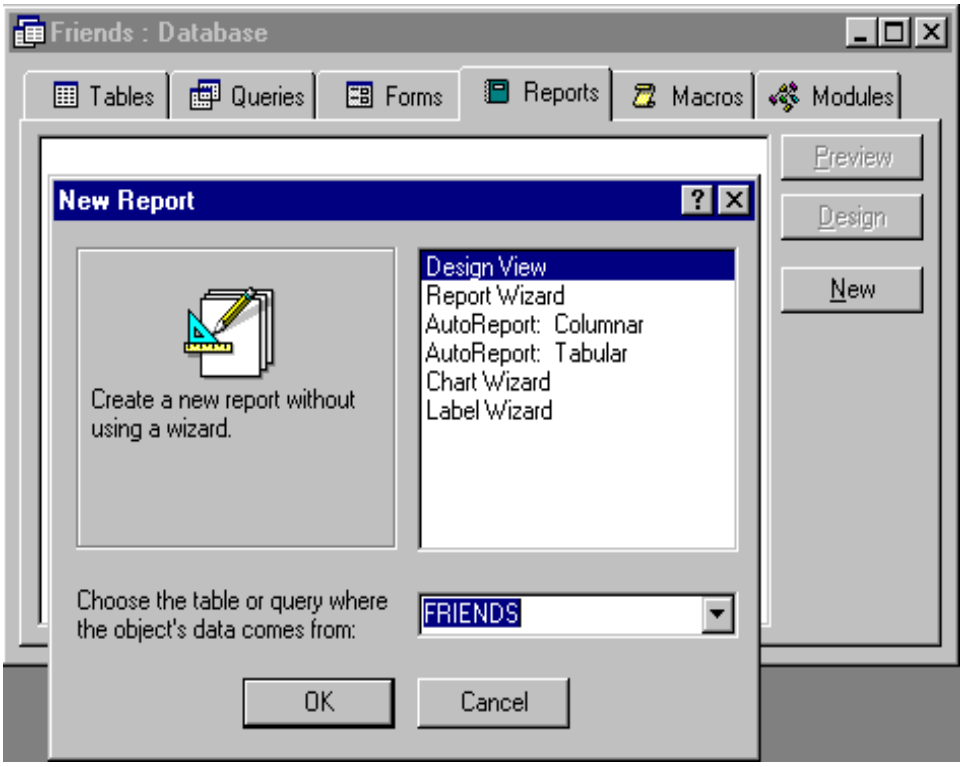
A polished look to your printouts can be achieved using the Access report function. Reports can be based on tables and/or queries, and are used to provide sub-totals and grand totals of numeric fields, and to produce summaries, mailing labels, and presentation quality display of your data. Report Wizards speed up the creation, display and printout of reports by providing a series of popular style templates to choose from.

The items on a report that display or print data are known as *controls*. With a control, you can display data

from fields, calculation results, text for report headings, and include graphs, pictures and other Access objects.

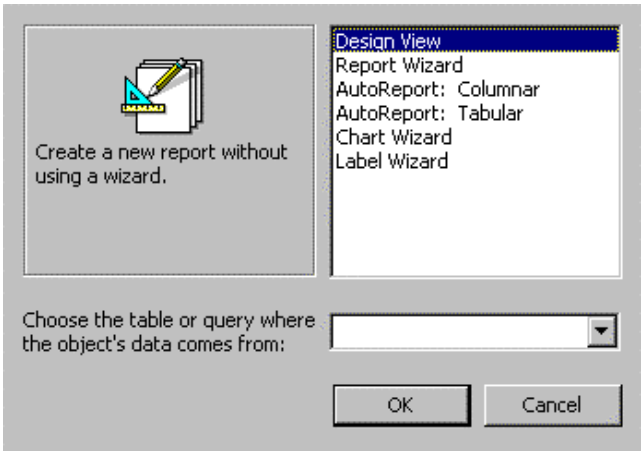
1. From the Database Window (press the F11 key if you are not in this position), click on the *Report object*, and then click *New*. From the New Reports box, click on the ↓ button beneath the *Report Types List*, and click and highlight on *FRIENDS* (see Figure 5-18). This action tells Access that the new report will be based on data contained in the FRIENDS table.

Figure 5-18



2. The *New Report box* allows selection from Design View, the Report Wizard (see Figure 5-19) and four other report templates. By clicking on each one in turn , Access will provide you with a brief description of the type of report each template will produce. Click on *Report Wizard*, and then click *OK*.

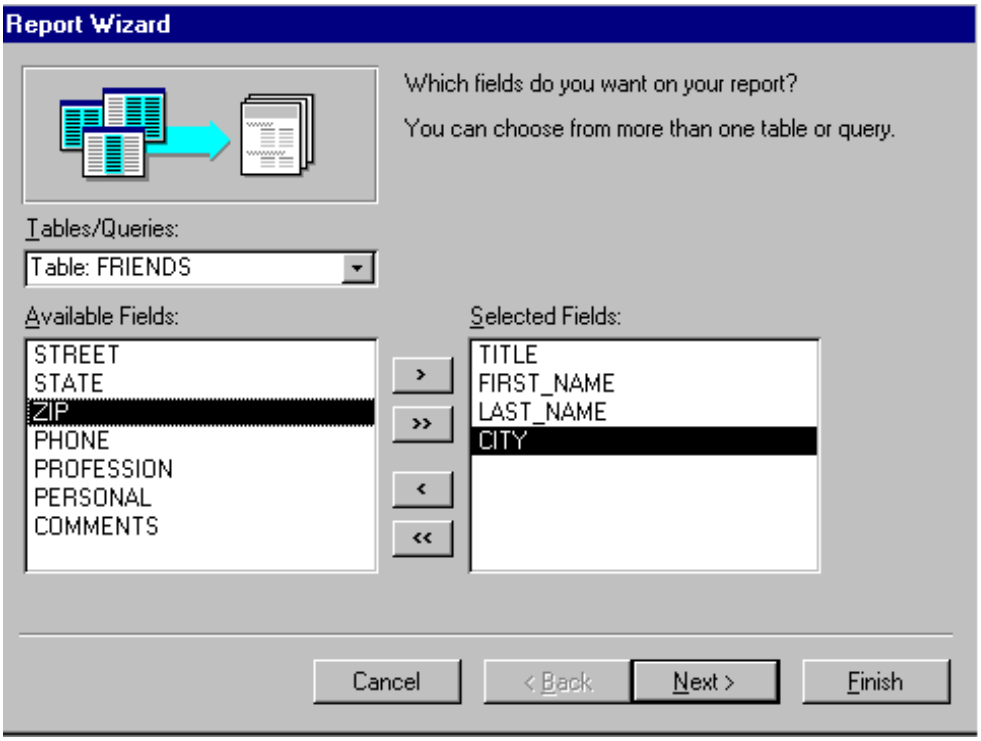
Figure 5-19



3. This drops you into the first of six Wizard setup windows (see Figure 5-20). This first window enables you to choose the fields that will appear in the report, and to decide the order in which they will be displayed. Let's select four fields - Title, FirstName, LastName and City for inclusion in the report.

Click and highlight the Title field in the field list (refer Figure 5-20), and then click on the  button to lodge the field onto the report list. Repeat this action for the remaining three fields, until the *Selected Fields box* looks like the one shown in Figure 5-20. (If you accidently lodge the wrong field in the report list, highlight the unwanted field, and then click on the  button to return it to the FRIENDS field list). When done, click the *Next* > button.

Figure 5-20



4. The second of the six report wizard windows is an optional action, and allows you to add grouping levels for the data in your report based on one or more fields. Click *Next* >. The third of the report wizard windows is also optional, and allows you to impose a sort order on the data in your report based on one or more fields. Select City from the drop down box, and then click on the > button to lodge this into the *Sort order of records*. Click *Next.*

5. The fourth wizard window allows you to determine your report layout. Radio buttons offer a choice between Portrait and Landscape orientation, and Vertical or Tabular Layout. Choose *Tabular*. Also click in the box next to *Adjust the field width so all fields fit on a page,* then click on the *Next* > button.

The fifth wizard window allows you to specify how your report will look when printed out. Styles available are Bold, Casual, Compact, Corporate, Formal, or Soft Gray. For the purposes of our simple report, accept the default by clicking on the *Next* > button.

6. The final report wizard window allows you to enter a customized title for your report. By default, Access enters in a title based on the name of the table or query from which the report is derived - in this case FRIENDS. Type in a new title: "Report of my Friends".

7. Click on the *Finish* button, and Access will proceed to generate the report, and display it in Print Preview mode. Note that the records are displayed alphabetically by City according to the sort order we imposed. Click on the *Print* toolbar button or select FILE/PRINT from the menu to send your report to print.

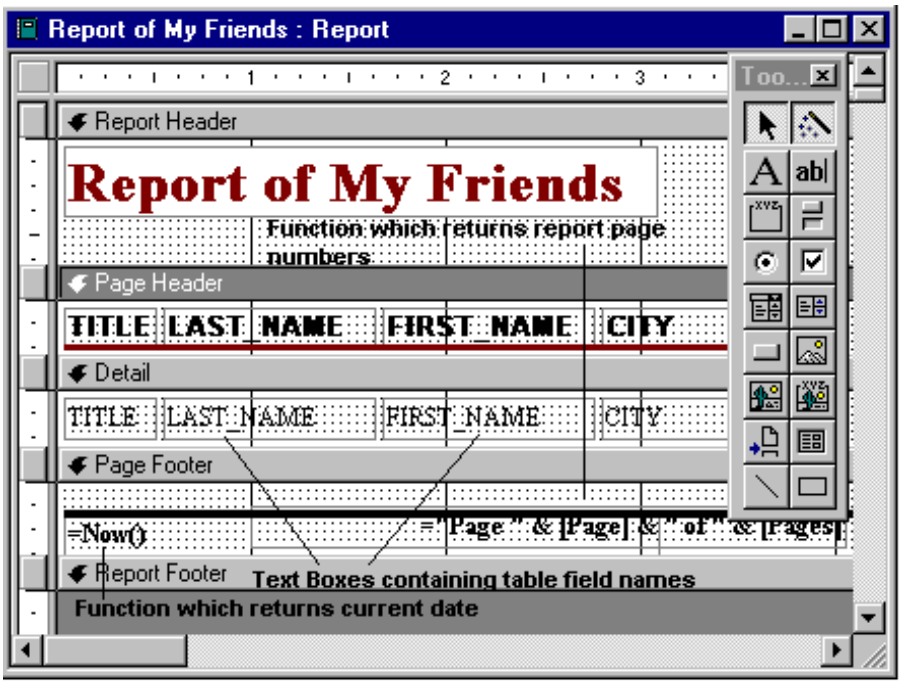
8. Save your report by clicking on the *Save* toolbar button or selecting FILE/SAVE from the menu. Type a name for your report (eg: *Tute2 Repo1*) in the Save As box and click *OK*. Press *F11* to return to the Database Window.

How to Use Reports to Display and Print Selected Fields.

Existing reports can be assigned to tables and/or queries other than the one they were originally based on. Let's say you wanted a report based on the results of the second query created in the tutorial for Case 1 (eg: *Tute1 Query2*). This query filtered out all records in the FRIENDS table that did not have the title of "Mr".

1. From the Database Window, click on the *Report object*, and click and highlight the report we have just created. Click on the *Design* button in the Database Window, and this will drop you into report design view (see Figure 5-21).

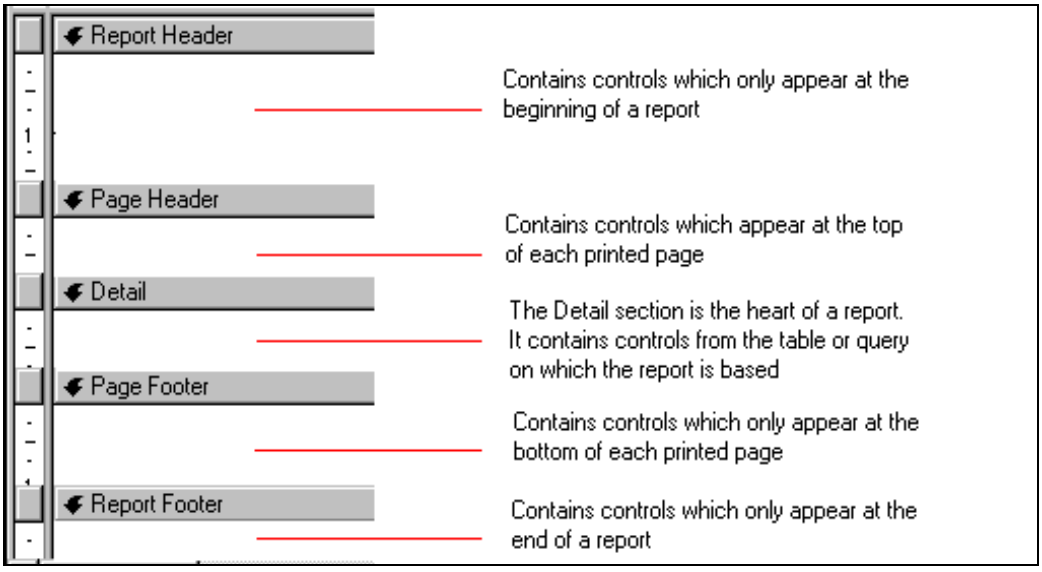
Figure 5-21



The design view comprises two main elements:

- a report design sheet divided into a number of sections. The purpose of each of these sections is briefly described in Figure 5-22
- a collection of control buttons known as a Toolbox. Descriptions for each of these buttons are provided at the end of this tutorial.

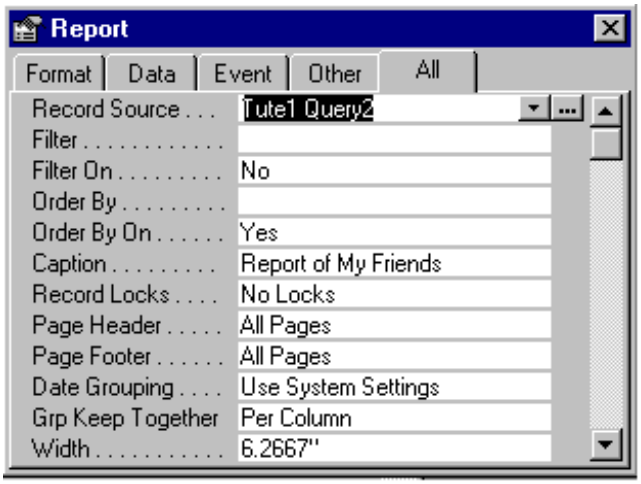
Figure 5-22



In addition to the above, sections can also be defined for headers and footers that are based around *groups* of records with like values. We will be looking at report groups in a later tutorial.

2. From the EDIT menu, choose SELECT REPORT and then click the *Properties* toolbar button, or choose VIEW/PROPERTIES from the menu. This activates the report properties sheet (see Figure 5-23). Report properties define the data source for a report as well as its overall appearance.

Figure 5-23



3. On the properties sheet, click the ↓ button to the right of the Record Source bar. This activates a drop down list of all tables and queries associated with the FRIENDS database. Click on the second query you created in the Case 1 tutorial to change the record source for the report.

4. Close the properties sheet by clicking on the ☒ box in the top right hand corner.

5. Click on the Print Preview toolbar button, or select FILE/PRINT PREVIEW from the menu. This shows a reduced data set as per the requirements of the query, with the sort order of the City field and presentation style imposed by the original report setup.

6. Click on the *Print* toolbar button to send your report to print.

7. Save your report by clicking on the *Save* toolbar button or selecting FILE/SAVE from the menu. Choose a name for your report (eg: *Tute2 Repo2*). Press *F11* to return to the Database Window.

Using Expressions in Reports

Expressions are used to get information that cannot be directly obtained from the tables in a database. An *expression* is a calculation that results in a single value. Expressions are comprised of a combination of operators (such as + and =), constant values, field names and functions. For example the expression, =[UnitCost]*[Quantity] multiplies the contents of two fields, named UnitCost and Quantity, and then displays a single value result. A *function* is a procedure or routine that returns a value. For example, the *=Now()* function shown in the Report Header section of Figure 5-18, returns the current date as stored in your computer's system clock.

Access offers three ways of entering expressions in reports:

- by typing an expression in a field text box (see Figure 5-21)
- by using any control (graphical object) that has a Control Source property; or
- by using the Access Expression Builder.

To learn and practice the skill of entering expressions, we will use the first of these three methods, and the sample data table HARDWARE provided on SOLVEIT.MDB. You will need to create a new empty database (eg: HARDWARE.MDB) and then import the HARDWARE table.

1. Load HARDWARE.MDB. *(Warning: If you still have tables or reports open from the FRIENDS database, you should save them and close them before opening the HARDWARE database).*

2. From the *Table* object, double click to open the HARDWARE table and examine the record contents. There are four fields: Invoice, Item, UnitCost, and Quantity. Press F11 to return to the Database Window.

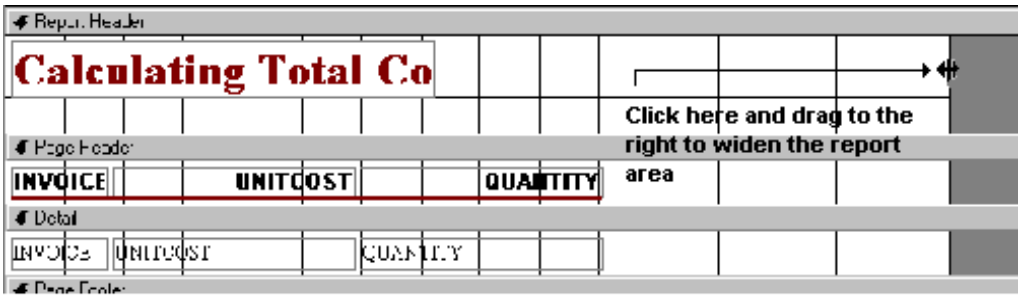
3. Click on the *Report* object and create a new report based on the HARDWARE table. Include the fields Invoice, UnitCost and Quantity. In the final wizard window, give your new report a name - *Calculating Total Cost*, and click Finish. Access will generate the report and display it in Print Preview mode.

4. Switch to report design view by clicking on the Close toolbar button.

Changing the Report Margins

5. Widen the working area of your report by extending the margins. To do this, move your mouse over to the right margin of the report area until the ⇹ symbol appears (refer Figure 5-24). Click and drag to the right to widen the report area. Release your mouse button, and Access will adjust the report width.

Figure 5-24



Creating Calculated Controls in Reports

6. A *calculated control* contains an expression and displays the results of that expression when a report is run. To practice creating a calculated control, let's multiply UnitCost by Quantity. Since this calculation will be a composite of two existing fields, we need to create an area to contain the calculation.

7. To do this, click once on the text box button ab in the *Toolbox*. Now click once in an empty area on the right hand side of the Detail section of the report (ie: next to the Quantity field). This action generates an unbound box. An unbound box is one not associated with any existing

field of the table upon which the report is based. Unbound boxes can be used for showing text, the results of calculations, or graphics.

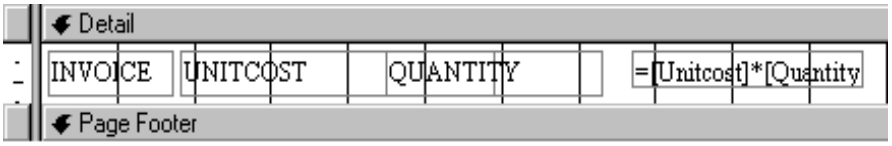
8. Click inside the unbound text box to get a blinking insertion point (the *I-beam*). Type: =Unitcost*Quantity and press the Enter key. Access will automatically add the syntax for you by enclosing each field in square brackets. Access is not case sensitive, so it does not matter whether you enter your field names in upper or lower case. You must however, spell them correctly !!

Note:
If field names contain spaces you must enclose them in square brackets
eg: [Time to Market]

Any expression in Access must commence with the = sign. The asterisk (*) sign is used to denote multiplication. A listing of other operators and functions used in Access is presented at the end of this tutorial.

9. You have now created a calculated control. The Detail section of your report should look similar to Figure 5-25. Click on the Print Preview toolbar button, and run your report.

Figure 5-25



10. Give your calculated contol a heading. Click on the label  button on the toolbox to create a label for displaying descriptive text, and then click in the Page Header section of the report, and position above the calculated control. Click to activate the I-beam, and then enter the heading TOTAL. Rerun your report to see the difference. Your report should look similar to Figure 5-26.

11. Send your report to print. Save your report (eg: *TCost*) by clicking on the Save toolbar button, or by selecting FILE/SAVE from the menu.

12. Press F11 to return to the database window.

Figure 5-26

Calculating Total Cost

INVOICE	UNITCOST	QUANTITY	TOTAL
1234	25	2	50
1235	15	1	15
1236	12.5	4	50

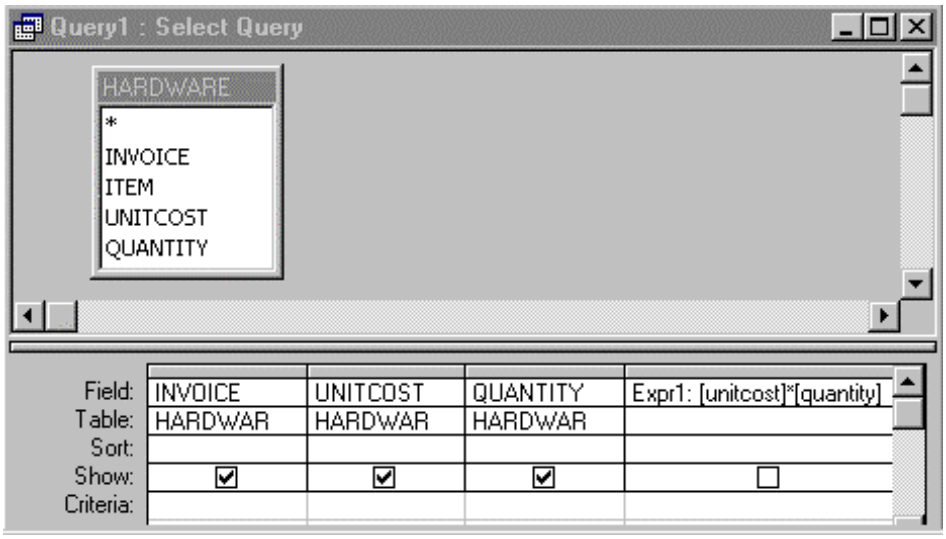
Calculations in Queries

A more rapid way of achieving the same result, is to use the Query window for creating your calculation, and then to use the query as the basis for a report.

1. From the HARDWARE database window, create a new query. From the HARDWARE field list, drag the 3 fields Invoice, UnitCost, and Quantity down into the field bar of the QBE grid.

2. Type the expression **: =[UnitCost]*[Quantity]** into the next blank field (to the right of the Quantity field), and press Enter. Your query window should now look similar to Figure 5-27.

Figure 5-27



3. Press the query run toolbar button to see the results of your query. Save the query, and then create a new report, using your new query as the basis for the report.

Expressions play many roles in Access, and can be used with virtually every type of database object. For example, you can use expressions in *Table design view* to define default values for fields. In *queries,* expressions can be used to create calculated fields and criteria. Expressions can also be used in *form and report design* windows to create calculated controls with the text box toolbox tool.

Operators and Functions in Access

Operators in an expression describe the type of action the expression should perform, or how a comparison between two values should be carried out. There are four types of operators in Access. The table below lists only those likely to be used in *Solve it!.* For a more complete list, consult the *Office Assistant*, or search the Help system using the term *operators*.

Arithmetic and Text	
^	raise one number to the power of another
*	multiply two numbers
/	divide two numbers
+	add two numbers
-	subtract two numbers, or negate a number
Mod	divide two numbers and return the remainder
&	concatenate: join two strings of text
Comparison	
< and <=	less than; less than or equal to
> and >=	greater than; greater than or equal to
= and <>	equal to; not equal to
Logical	
And	both comparisons are True
Or	one comparison or the other is True
Xor	one comparison or the other is True, but not both
Not	the comparison is not True
Miscellaneous	
Like	text matches a pattern (use with wildcard symbols ? and *)
Is	comparison is True (eg: is Null)
Is Not	comparison is not True (eg: not Null)

A *function* in Access performs a calculation on data, and then returns the result of that calculation. There are over 100 functions available in Access, and there are eight different types. The table below lists only those likely to be used in *Solve it!.* For a more complete list, consult the *Office Assistant*, or search the Help system using the term *functions reference*.

Date/Time	
Date	returns current date
Now	returns current date and time
Logical	
IIF	tests and returns a value based on whether an argument is true or false
Choose	selects a value from a list based on the content of its first argument
Aggregate	
Avg	average

Count	count how many
Sum	sum total

The Access Form and Report Toolbox

	**Select Objects**	Use to select a control, section, or form. Click this tool to unlock a toolbox button that you've locked down.
	**Control Wizards**	Turns control wizards on or off. Use control wizards to help you create a list box, combo box, option group, command button, chart, subreport, or subform.
	**Label**	A control that displays descriptive text, such as a title, a caption, or instructions on a form or report. Access automatically attaches labels to the controls you create.
	**Text Box**	Use to: Display, enter, or edit data in a form's or report's underlying record source, display the results of a calculation, or accept input from a user.
	**Option Group**	Use along with check boxes, option buttons, or toggle buttons to display a set of alternative values. For example, you can use an option group to specify whether an order is shipped by air, sea, or land.
	**Toggle Button**	Use as a stand-alone control bound to a Yes/No field, an unbound control for accepting user input in a custom dialog box, or part of an option group.
	**Option Button**	Use as a stand-alone control bound to a Yes/No field, an unbound control for accepting user input in a custom dialog box, or part of an option group.
	**Check Box**	Use as a stand-alone control bound to a Yes/No field, an unbound control for accepting user input in a custom dialog box, or part of an option group.
	**Combo Box**	Combines the features of a list box and a text box. You can type in the text box or select an entry in the list box to add a value to an underlying field.
	**List Box**	Displays a scrollable list of values. In Form view, you can select from the list to enter a value into a new record or to change the value in an existing record.
	**Command Button**	Use to perform actions, such as finding a record, printing a record, or applying a form filter.
	**Image**	Use for displaying a static picture on a form or report. Because a static picture is not an OLE object, you can't edit the image inside Microsoft Access once you've added it to a form or report.
	**Unbound Object**	Use to display an unbound OLE object, such as a Microsoft Excel spreadsheet, on a form or report. The object remains constant as you move from record to record.
	**Bound Object**	Use to display OLE objects, such as a series of pictures, on a form or report. This control is for objects stored in a field in the form's or report's underlying record source. A different object displays on the

		form or report as you move from record to record.
	**Page Break**	Use to begin a new screen on a form, a new page on a printed form, or a new page of a report.
	**Tab Control**	Use to create a tabbed form with several pages or tabbed dialog box (such as the Options dialog box on the Tools menu). You can copy or add other controls onto a tab control. Right-click on the Tab control in the design grid to modify the number of pages, the page order, the selected page's properties, and the selected tab control properties.
	**Subform/Subreport**	Use to display data from more than one table on a form or report.
	**Line**	Use on a form or report, for example, to emphasize related or especially important information.
	**Rectangle**	Use for graphic effects such as grouping a set of related controls on a form, or emphasizing important data on a report.
	**More Controls**	Adds an ActiveX control (such as the Calendar control) to a form or report. ActiveX controls are stored as separate files.

Tutorial For Database Case 2 Using Access 2.0

You have already learned in the previous case how to use the table design view to change the structure of a database and add new fields, and to use the datasheet for adding new records and editing existing ones. This case introduces the *memo data type* and *report design*. Load the file you used for the Case 1 tutorial (CONTACTS.MDB or FRIENDS.MDB) to practice the skills you will need for this case.

Using Memo Fields

Memo fields are very useful in databases for storing free form text and notes, or annotations against particular records. Unlike DOS-based database packages such as dBase III+, creating, entering and saving data in an Access memo field is a very simple procedure. You can store up to 64,000 characters in an Access memo field. Because of their unstructured nature, memo fields cannot be *indexed* or *sorted*, but they can be searched. We will be looking at indexing, sorting and searching in later cases.

1. From the Database Window, click on the Table object and double click on either the ADDRESS or FRIENDS table. From the default datasheet view, click on the table design toolbar button or select VIEW/TABLE DESIGN from the menu.

2. Let's add a new field to the end of the existing database. The name for the new field will be "Comments", and the data type will be Memo. Recap the tutorial for Case 1, if you are unsure how to add new fields and select data types. Notice that the Field Size bar in the properties box does not appear when you select a memo data type, indicating the unstructured nature of this field. Save your changes to the table structure by clicking on the *Save* toolbar button, or selecting FILE/SAVE from the menu.

3. Return to datasheet view by clicking on the datasheet toolbar button, or selecting VIEW/DATASHEET from the menu. Scroll or Tab over to the new Comments field, and enter text against each record (eg: whether the person in question is a Personal or Professional friend). Press the ↓ cursor key on your keyboard to move to the next record. Remember that as you move to a different record, Access will automatically save any changes.

Printing Memo Fields

Memo fields can be printed like any other field in a database. For a quick report including memo fields:

1: From datasheet view, press the F11 key to return to the Database window. Click the *Query object* and click *New*. From the New Query box, click on the *New Query* button, and Access will open up a new query window. From the Add Table box click and highlight the FRIENDS table, and then click *Add*, and then click *Close*.

2. From the FRIENDS *field list*, select and drag the LastName and Comments fields down onto the Field bar in the *QBE Grid* of the query window.

3. Click on the *Run* toolbar button or select QUERY/RUN from the menu to display the contents of the Last Name and Comments fields print (see Figure 5-28)

4. Click on the *Print* button or choose FILE/PRINT to send your query to. Select FILE/CLOSE from the menu and click *No* to the Save Query request to return to the Database Window without saving the query.

Figure 5-28

Select Query: Query1

LASTNAME	COMMENTS
Drucker	Professional
Whitney	Personal
Sitkin	Professional
Skalek	Professional
Salione	Personal
Fabian	Personal
Kohlman	Father-in-Law
Tedesco	Professional
Zito	Professional
Peterson	Personal
Nelson	Personal

Record: 1 of 11

Creating Reports in Access

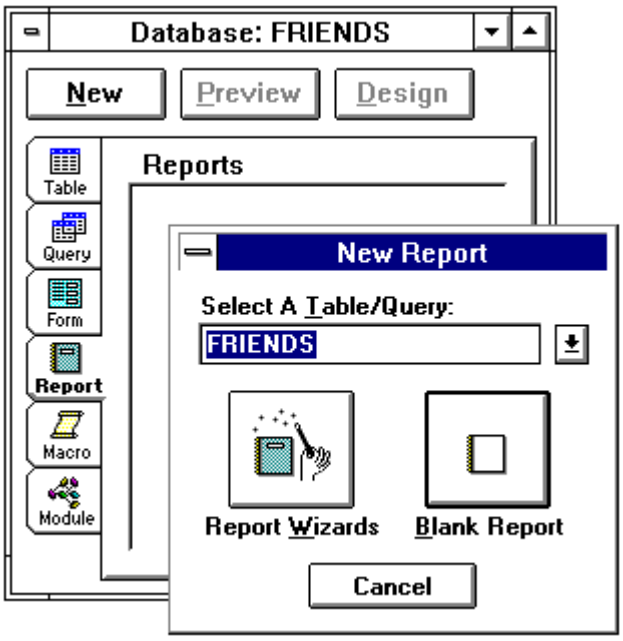
A polished look to your printouts can be achieved using the Access report function. Reports can be based on tables and/or queries, and are used to provide sub-totals and grand totals of numeric fields, and to produce summaries, mailing labels, and presentation quality display of your data. Report Wizards speed up the creation, display and printout of reports by providing a series of popular style templates to choose from.

The items on a report that display or print data are known as *controls*. With a control, you can display data from fields, calculation results, text for report headings, and include graphs, pictures and other Access objects.

1. From the +Database Window (press the F11 key if you are not in this position), click on the *Report object*, and then click *New*. From the New Reports box, click on the ↓ button to the

right of the *Select a Table/Query* Bar, and click and highlight on *FRIENDS* (see Figure 5-29). This action tells Access that the new report will be based on data contained in the FRIENDS table.

Figure 5-29



2. Click on the *Report Wizards* button. This generates the Report Wizards box (see Figure 5-30) and displays a list of six different report templates to choose from. By clicking on each one in turn , Access will provide you with a brief description of the type of report each template will produce. Click on *Tabular*, and then click *OK*.

3. This drops you into the first of four Wizard setup windows (see Figure 5-31). This first window enables you to choose the fields that will appear in the report, and to decide the order in which they will be displayed. Let's select four fields - Title, FirstName, LastName and City for inclusion in the report.

Click and highlight the Title field in the field list (refer Figure 5-31), and then click on the  button to lodge the field onto the report list. Repeat this action for the remaining three fields, until the *Field order on report box* looks like the one shown in Figure 15-17. (If you accidently lodge the wrong field in the report list, highlight the unwanted field, and then click on the  button to return it to the FRIENDS field list). When done, click the *Next* > button.

Figure 5-30

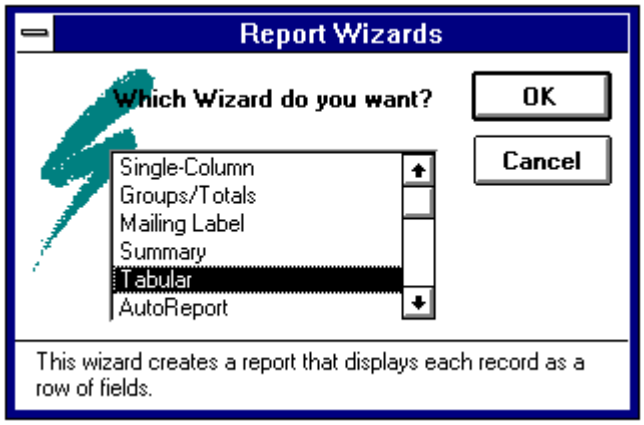
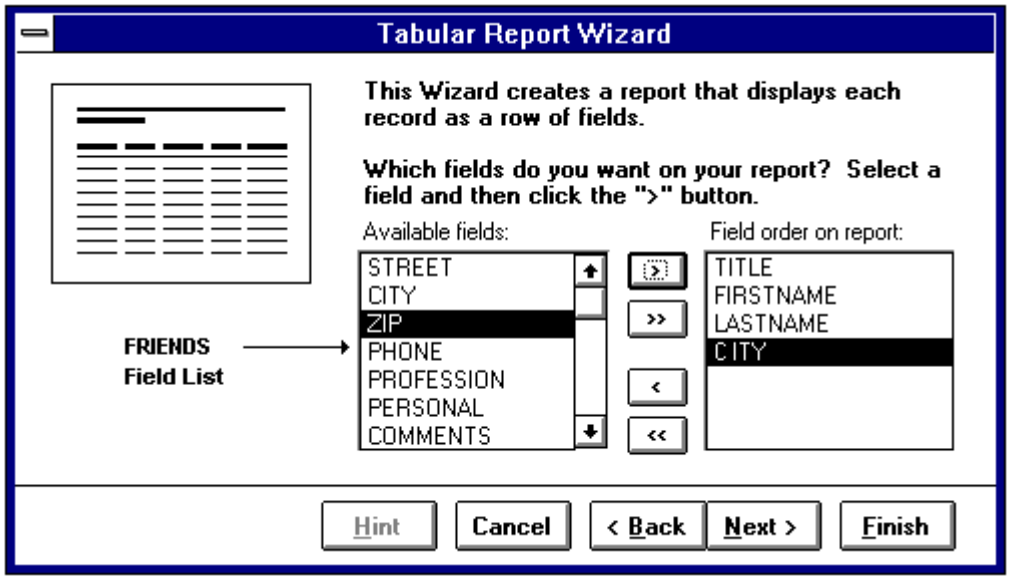


Figure 15-31



4. The second of the four report wizard windows is an optional action, and allows you to impose a sort order on the data in your report based on one or more fields. Click on City in the FRIENDS field list and then click on the  button to lodge this into the *Sort order of records* box. Click *Next* >.

5. The third wizard window allows you to determine how your report will look when printed out. Radio buttons offer a choice between Portrait and Landscape orientation, and Ledger, Presentation or Executive report styles. For the purposes of our simple report, accept the default by clicking on the *Next* > button.

6. The final report wizard window allows you to enter a customised title for your report. By default, Access enters in a title based on the name of the table or query from the the report is derived - in this case FRIENDS. Type in a new title: "Report of my Friends". Click in the box next *to See all the fields on one page* to force Access to confine field display to one page.

7. Click on the *Finish* button, and Access will proceed to generate the report, and display it in Print Preview mode. Note that the records are displayed alphabetically by City according to the sort order we imposed. Click on the *Print* toolbar button or select FILE/PRINT from the menu to send your report to print.

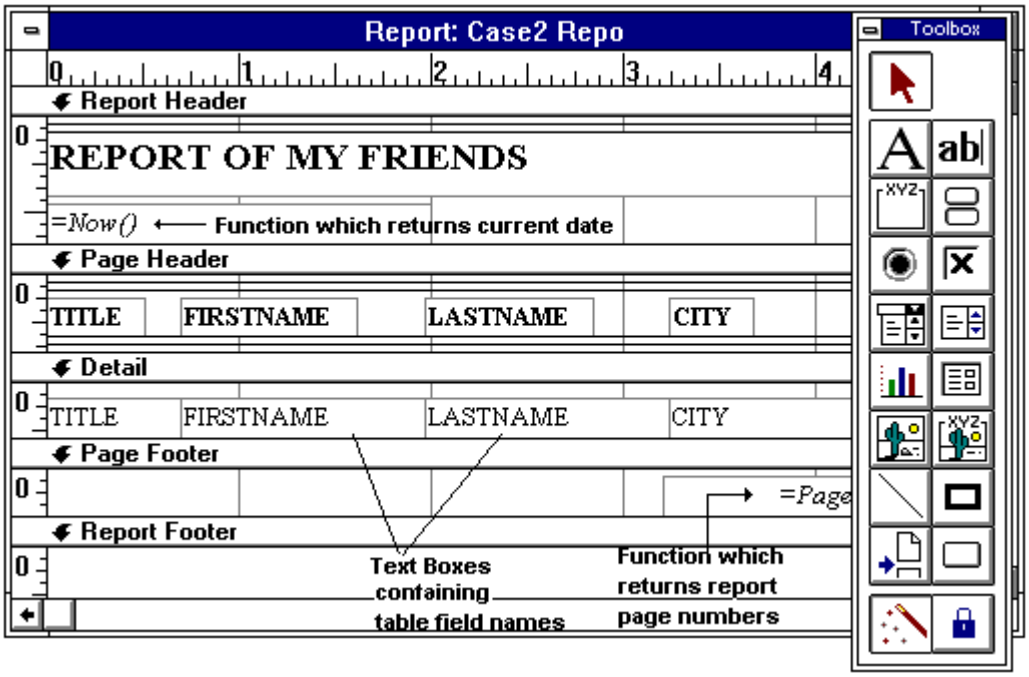
8. Save your report by clicking on the *Save* toolbar button or selecting FILE/SAVE from the menu. Type a name for your report (eg: *Tute2 Repo1*) in the Save As box and click *OK*. Press *F11* to return to the Database Window.

How to Use Reports to Display and Print Selected Fields.

Existing reports can be assigned to tables and/or queries other than the one they were originally based on. Let's say you wanted a report based on the results of the second query created in the tutorial for Case 1 (eg: *Tute1 Query2*). This query filtered out all records in the FRIENDS table that did not have the title of "Mr".

1. From the Database Window, click on the *Report object*, and click and highlight the report we have just created. Click on the *Design* button in the Database Window, and this will drop you into report design view (see Figure 5-32).

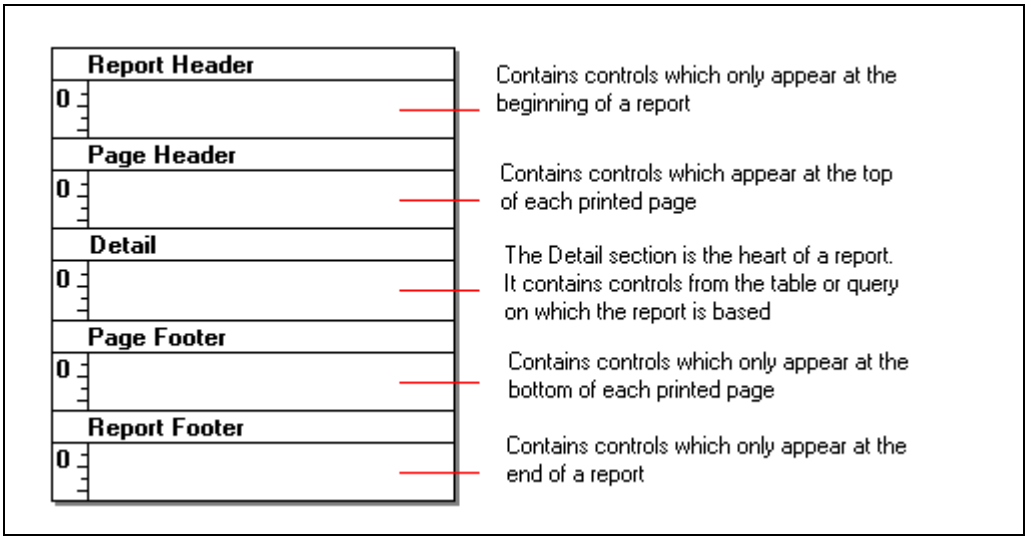
Figure 5-32



The design view comprises two main elements:

- a report design sheet divided into a number of sections. The purpose of each of these sections is briefly described in Figure 5-33
- a collection of control buttons known as a Toolbox. Descriptions for each of these buttons is provided at the end of this tutorial.

Figure 5-33



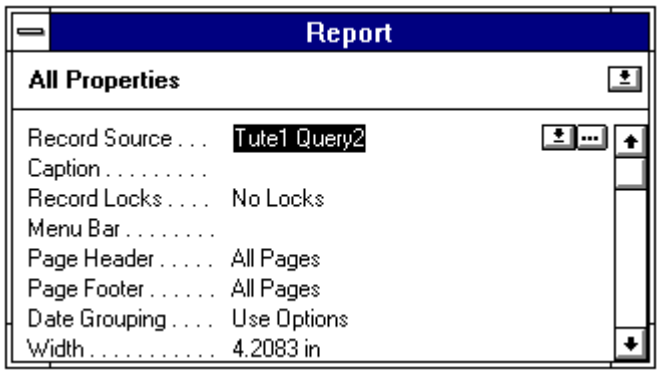
In addition to the above, sections can also be defined for headers and footers that are based around *groups* of records with like values. We will be looking at report groups in a later tutorial.

2. From the EDIT menu, choose SELECT REPORT and then click the *Properties* toolbar button, or choose VIEW/PROPERTIES from the menu. This activates the report properties sheet (see Figure 5-34). Report properties define the data source for a report as well as its overall appearance.

3. On the properties sheet, click the ▼ button to the right of the Record Source bar. This activates a drop down list of all tables and queries associated with the FRIENDS database. Click on the second query you created in the Case 1 tutorial to change the record source for the report.

4. Close the properties sheet by double clicking on the  box in the top left hand corner.

Figure 5-34



5. Click on the Print Preview toolbar button, or select FILE/PRINT PREVIEW from the menu. This shows a reduced data set as per the requirements of the query, with the sort order of the City field and presentation style imposed by the original report setup.

6. Click on the *Print* toolbar button to send your report to print.

7. Save your report by clicking on the *Save* toolbar button or selecting FILE/SAVE from the menu. Choose a name for your report (eg: *Tute2 Repo2*). Press *F11* to return to the Database Window.

Using Expressions in Reports

Expression*s* are used to get information that cannot be directly obtained from the tables in a database. An *expression* is a calculation that results in a single value. Expressions are comprised of a combination of operators (such as + and =), constant values, field names and functions. For example the expression, =[UnitCost]*[Quantity] multiplies the contents of two fields, named UnitCost and Quantity, and then displays a single value result. A *function* is a procedure or routine that returns a value. For example, the *=Now()* function shown in the Report Header section of Figure 5-18, returns the current date as stored in your computer's system clock.

Access offers three ways of entering expressions in reports:

- by typing an expression in a field text box (see Figure 5-32)
- by using any control (graphical object) that has a Control Source property; or
- by using the Access Expression Builder.

To learn and practice the skill of entering expressions, we will use the first of these three methods, and the sample database HARDWARE.MDB provided on your *Solve It!* disk.

1. Load HARDWARE.MDB. *(Warning: If you still have tables or reports open from the FRIENDS database, you should save them and close them before opening the HARDWARE database).*

2. From the *Table* object, double click to open the HARDWARE table and examine the record contents. There are four fields: Invoice, Item, UnitCost, and Quantity. Press F11 to return to the Database Window.

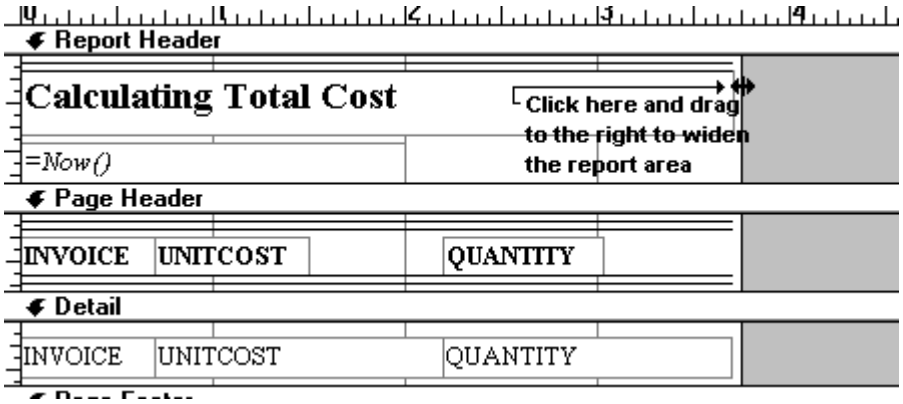
3. Click on the *Report* object and create a new report based on the HARDWARE table. Include the fields Invoice, UnitCost and Quantity. In the final wizard window, give your new report a name - *Calculating Total Cost*, and click Finish. Access will generate the report and display it in Print Preview mode.

4. Switch to report design view by clicking on the Close Window  toolbar button.

Changing the Report Margins

5. Widen the working area of your report by extending the margins. To do this, move your mouse over to the right margin of the report area until the ⇹ symbol appears (refer Figure 5-35). Click and drag to the right to widen the report area. Release your mouse button, and Access will adjust the report width.

Figure 5-35



Creating Calculated Controls in Reports

6. A *calculated control* contains an expression and displays the results of that expression when a report is run. To practice creating a calculated control, let's multiply UnitCost by Quantity. Since this calculation will be a composite of two existing fields, we need to create an area to contain the calculation.

7. To do this, click once on the text box button ab in the *Toolbox*. Now click once in an empty area on the right hand side of the Detail section of the report (ie: next to the Quantity field). This action generates an unbound box. An unbound box is one not associated with any existing field of the table upon which the report is based. Unbound boxes can be used for showing text, the results of calculations, or graphics.

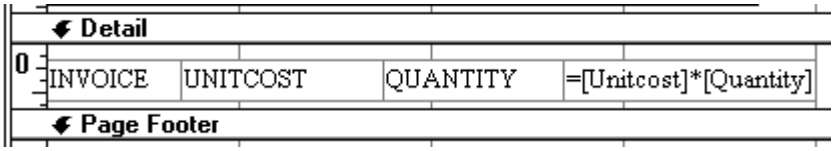
8. Click inside the unbound text box to get a blinking insertion point (*I-beam*).
Type: =Unitcost*Quantity and press the Enter key. Access will automatically add the syntax for you by enclosing each field in square brackets. Access is not case sensitive, so it does not matter whether you enter your field names in upper or lower case. You must however, spell them correctly !!

Note: If field names contain spaces you must enclose them in square brackets eg: [Time to Market]

Any expresssion in Access must commence with the = sign. The asterisk (*) sign is used to denote multiplication. A listing of other operators and functions used in Access is presented at the end of this tutorial.

9. You have now created a calculated control. The Detail section of your report should look similar to Figure 5-36. Click on the Print Preview toolbar button, and run your report.

Figure 5-36



10. Give your calculated contol a heading. Click on the label  button on the toolbox to create a label for displaying descriptive text, and then click in the Page Header section of the report, and position above the calculated control. Click to activate the I-beam, and then enter the heading TOTAL. Rerun your report to see the difference. Your report should look similar to Figure 5-37.

Figure 5-37

Calculating Total Cost

01-Feb-95

INVOICE	UNITCOST	QUANTITY	TOTAL
1234	25	2	50
1235	15	1	15
1236	12.5	4	50

11. Send your report to print. Save your report (eg: *TCost*) by clicking on the Save toolbar button, or by selecting FILE/SAVE from the menu.

12. Press F11 to return to the database window.

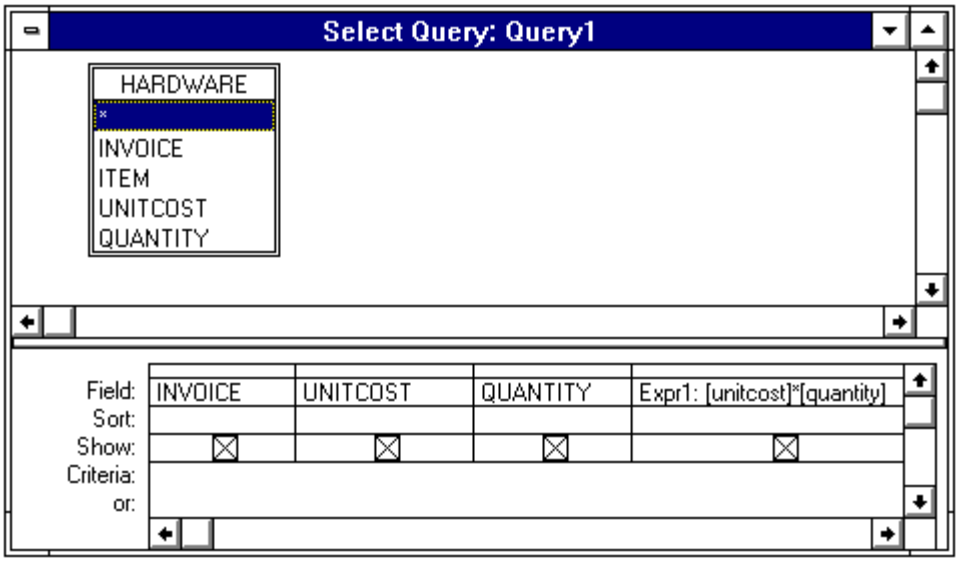
Calculations in Queries

A more rapid way of achieving the same result, is to use the Query window for creating your calculation, and then to use the query as the basis for a report.

1. From the HARDWARE database window, create a new query. From the HARDWARE field list, drag the three fields Invoice, UnitCost, and Quantity down into the field bar of the QBE grid.

2. Type the expression: =[UnitCost]*[Quantity] into the next blank field (to the right of the Quantity field), and press the Enter key. Your query window should now look similar to Figure 5-38.

Figure 5-38



3. Press the query run toolbar button to see the results of your query. Save the query, and then create a new report, using your new query as the basis for the report.

Expressions play many roles in Access, and can be used with virtually every type of database object. For example, you can use expressions in *Table design view* to define default values for fields. In *queries,* expressions can be used to create calculated fields and criteria. Expressions can also be used in *form and report design* windows to create calculated controls with the text box toolbox tool.

Operators in an expression describe the type of action the expression should perform, or how a comparison between two values should be carried out. There are four types of operators in Access. The table below lists only those likely to be used in *Solve it!*. For a more complete list, consult the Access Cue Cards, or search the Help system using the term *operators*.

Arithemetic and Text	
^	raise one number to the power of another
*	multiply two numbers
/	divide two numbers
+	add two numbers
-	subtract two numbers, or negate a number
Mod	divide two numbers and return the remainder
&	concatenate: join two strings of text
Comparison	
< and <=	less than; less than or equal to
> and >=	greater than; greater than or equal to
= and <>	equal to; not equal to
Logical	
And	both comparisons are True
Or	one comparison or the other is True
Xor	one comparison or the other is True, but not both
Not	the comparison is not True
Miscellaneous	
Like	text matches a pattern (use with wildcard symbols ? and *)
Is	comparison is True (eg: is Null)
Is Not	comparison is not True (eg: not Null)

A *function* in Access performs a calculation on data, and then returns the result of that calculation. There are over 100 functions available in Access, and there are eight different types. The table below lists only those likely to be used in *Solve it!*. For a more complete list, consult the Access Cue Cards, or search the Help system using the term *functions: reference*.

Date/Time	
Date	returns current date
Now	returns current date and time
Logical	
IIF	tests and returns a value based on whether an argument is true or false
Choose	selects a value from a list based on the content of its first argument
Aggregate	
Avg	average
Count	count how many
Sum	sum total

Database Case 3

Harrington Furniture

Problem: Develop an automated Bill of Materials

Management Skill: Plan
Coordinate

Access Skills: Queries
Sorting (tables, queries, reports)

Data Table: HARRING

Richard Devery, Production Manager of Harrington Furniture, walked out of the Chief Executive Officer's office muttering to himself. He had just left a meeting between Tony Cunningham, the CEO, Hank Alter, the Chief Financial Officer and Wendy Wang, the Purchasing Officer. They had more or less ordered him to computerize the bill of materials (BoM) and purchasing systems. He argued had that such a simple command was easier given than implemented. They had finally compromised on automating a section of the factory's assembly facilities as a pilot project. The pilot project would provide an opportunity to improve the design of the new system and measure the benefits before moving to complete implementation in the entire factory.

Harrington, based in Spokane, Washington, had experienced steady sales for its range of quality household furniture until the early 1970s when less expensive imports of similar quality began to capture significant market share. Costs had to be reduced, and senior management decided that increased production volume was needed before their household furniture products could regain competitiveness in such a capital intensive industry. These increased volumes would provide economies of scale in materials purchases and enable new, more efficient plant to be purchased. The increased volume would be obtained by entering the high-end commercial office furniture market. Harrington's quality hardwood tables would be particularly suitable in such a highly discriminating market. In addition, a range of desks was developed. The strategy proved fruitful; Harrington's business products were acclaimed in executive offices and boardrooms across the country. The strategy, together with a continuous improvement program initiated at the same time, reduced production costs and gains were made in both the household and business markets.

Management found that production scale had outpaced the manual, paper-based production control system which proved inadequate for market intelligence, activity-based costing and for efficient production administration.

Devery entered his own office, considering how to start the process. Even if his budget permitted a full-blown Materials Requirement Planning system, the jump would be too large for Harrington. He decided that creating a system in-house on a PC/Windows-based database would be the most suitable solution, since Cunningham and Alter were not specific about what computer platform to use and future requirements were unclear. This approach would provide the necessary

flexibility for incremental development and expansion. His knowledge of PC applications being limited, Devery rang the Business School at the University of Spokane. The placements office at the School referred him to Kylie Gates, a first-year MBA student seeking a summer internship in a manufacturing enterprise. Gates agreed to drive over to Spokane the following day.

Although from a fine arts background, Gates had undertaken an Information Systems course in her graduate program and was keen to learn about production systems. Devery explained what was required in the preliminary stages. Currently, the BoM for each product type was recorded on paper and used exclusively in the production area. Devery was sure that a BoM could be constructed for use by both production and the purchasing department. Currently, the purchasing department used a separate list of components, which often resulted in difficulties in attributing supply costs to particular product lines.

A BoM, Devery explained, was a list of materials, sub-parts and quantities used in a finished product. Typically, the list would be arranged to represent the sequence of fabrication and assembly, and highlighted manufacturing dependencies. For the pilot project, this would not be needed because of the relative simplicity of the production sequence.

Gates decided to use the company's range of senior executive tables and desks for the pilot project, and run the new system parallel with the existing system. In this way Harrington could refine and cost the system before committing to a complete implementation. This range had four different products: two tables, the Oxford and the Oxford Deluxe; and two desks, the Director and the Chairman. These four products were made up of 15 different components, excluding tapper screws.

The design for each product was very similar. The Oxford tables were made of six legs supporting four 5.5' beams and five 3.75' beams which in turn supported two 6'x 4' table tops which were laid end to end. Eight 3' and four 2' struts were added for additional strength and aesthetic considerations. The other two products in this range, Director and Chairman, are personal desks which have four legs, two 5.5' beams, three 3.75' beams, four 2' struts, eight 3' struts and a double table top. Director desks have three drawers, while the Chairman has six. The Oxford Deluxe table and Chairman desk also have leather tops; two pieces for the Deluxe and one piece for the Chairman.

Harrington's designers decided to use bolts to join the tables for strength and reliability rather than the mixture of bolts, nails and adhesives used by their competitors. Two 1.5"x 8" bolts are used for each leg, each 5.5' beam and each 3.75' beam. Two 1"x 6" bolts are used in each 2' strut and 3' strut. For each bolt, a matching nut and spacer are used. Each product has differing levels of carved detail and finish.

Harrington manufactures the tops and legs, purchases the struts from Trident Timbers, the leather from Lowman Leathers and the hardware from Ironmongers.

Gates listed each of the components as a record in a PC/Windows-based database, with the number used for each product listed in separate fields. The number of each component currently in inventory and the unit cost were also to be listed.

This partial bill of materials is supplied for you in the data table HARRING on SOLVEIT.MDB. Create a new Access database and import this file now.

Tasks: There are seven tasks in this case:

1. (a) Complete the supplied bill of materials by adding fields to represent inventory holdings and unit costs of the materials and fabricating appropriate data. (b) Also add new part codes, quantities used and the other information for the bolts, nuts and spacers. Note that the PART_CODE field is made up of a letter (representing the supplier: H for Harrington, T for Trident Timbers, L for Lowman Leathers and I for Ironmongers) followed by four numbers. (c) Print out the completed table.

2. Managers in a number of functional areas have heard of Gate's progress and have requested information from the computerized BoM. Managers from Sales want to know the total amount, in dollars, of the materials costs in each product type so that they may more easily set appropriate prices for these products. Managers from Accounting want to know the total dollar value of each component type in each table or desk. Devery wants an estimate of the investment in dollars in inventory holdings of the component types and a total dollar value of inventory holdings. Gates is certain that these needs may be furnished within a single report.

   (a) Create a report to satisfy these needs and (b) print out this report.

3. Gadsden Corporation, a regular and valued customer, have ordered tables for their new Albuquerque office. They require 3 Oxford tables and 42 Director desks. Cunningham the CEO, and Wendy Wang from Purchasing want a detailed Materials Requirement report for this order, showing the number of components required for each of the two products ordered, a total components required field, and the amount held in inventory. Create and print the report.

4. (a) Create a query which filters out the parts that have adequate supply in inventory. (b) Create and print out a report to show this information. Include all necessary fields.

5. Wendy Wang has heard rumours that Trident Timbers is having financial difficulties. Cunningham is worried by these rumours and the implications this may have for the Gadsden contract. Print out a listing of components sorted by PART_CODE to facilitate the investigation.

6. Engineering and Design wishes to obtain a list of the components, sorted by value. (a) Create a query and (b) print the resulting listing.

*7. Suggest any improvements or additions to the Bill of Materials as it currently exists. Could the BoM be extended to encompass other aspects of Production Management or any other associated domains?

Time Estimates (excluding task marked with *):

Expert: 45 minutes
Intermediate: 1.5 hour
Novice: 3 hours

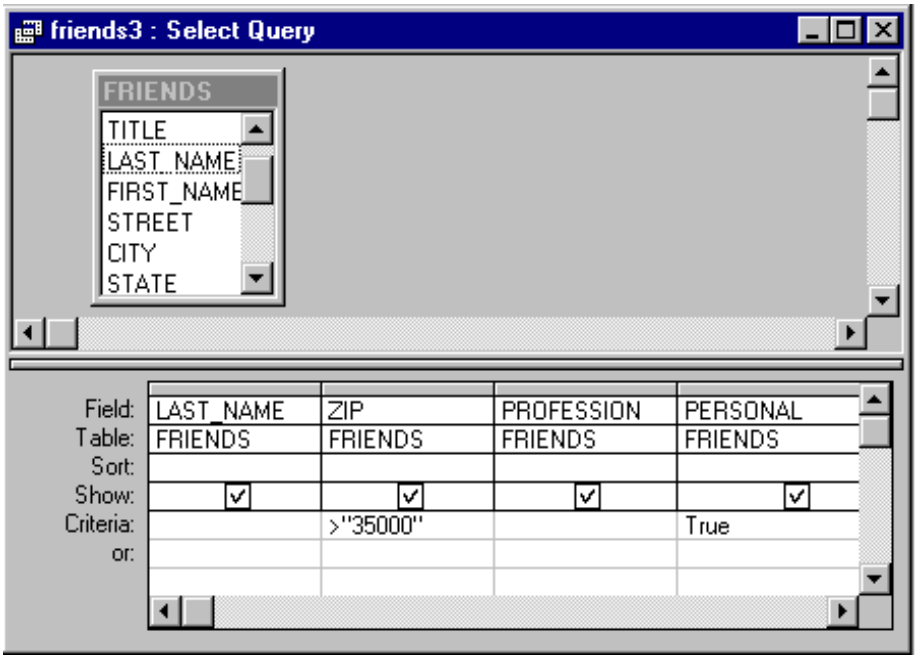
Tutorial For Database Case 3 Using Access 97

Using Query Files With Multiple Conditions

In the Tutorial for Case 1, we looked at creating queries for viewing and printing certain fields in the FRIENDS table, and fields which met a simple selection criteria. In this tutorial, we will look at constructing more complex queries using logical operators.

1. Load the practice database FRIENDS.MDB, and create a new query following the instructions given in the Tutorial for Case 1. Include the fields LastName, Zip, Profession and Personal in the query.

Figure 5-38



2. The Criteria: and or: bars of the QBE Grid allow for multiple conditions involving the logical operators "AND" and "OR". Any conditions entered on the first line of the Criteria: bar are "ANDed". Let's say you wanted a listing of all Personal friends with Zip codes greater than 35000. In the Criteria: bar of the QBE Grid (see Figure 5-38), type >35000 under the Zip field and enter True under the Personal field.

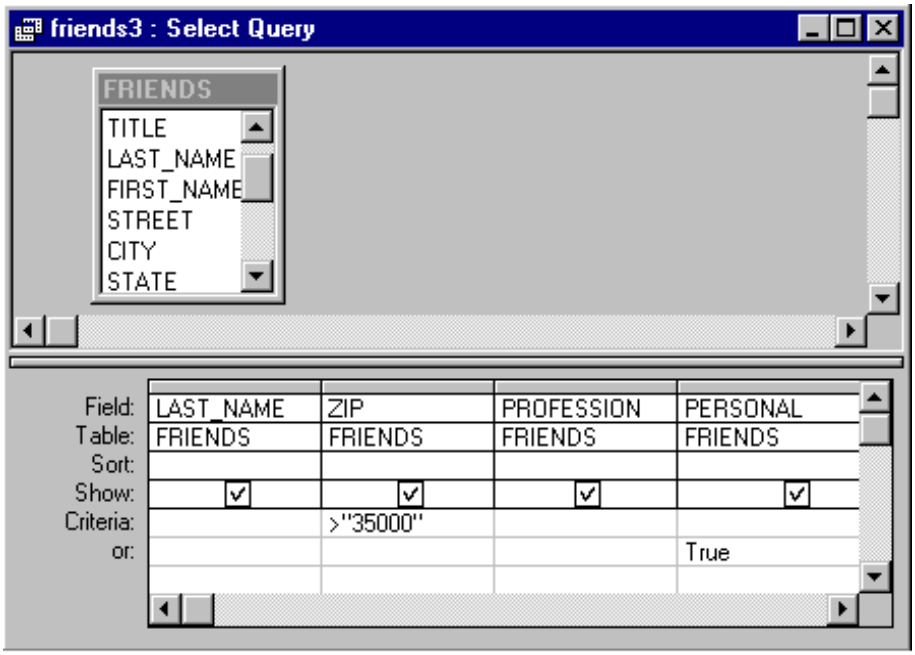
3. Run the query by clicking on the Run toolbar button, or selecting QUERY/RUN from the menu. Results should show that there are four records in the FRIENDS table which match the chosen criteria.

4. Click on the Design View toolbar button to return to the query design window.

5. Now try using the "OR" operator by deleting True from the Criteria: bar under the Personal field, and entering it on the or: bar on the next line down (see Figure 5-39) In this instance, we are searching the table for people whose Zip code is greater than 35000 *or* people who are personal friends. Run the query again. This is a less stringent condition, and the resulting dynaset should show that there are nine records which match the chosen criteria.

6. Print your queries by clicking on the Print toolbar button, or selecting FILE/PRINT from the menu. Save your query (eg: *Tute3 Query1*), and press F11 to return to the Database Window.

Figure 5-39



Sorting Tables in Access

It is often useful to sort a table by one or more fields. For instance, you may find it helpful to sort the records in your table alphabetically by Last Name, or to sort records in chronological order, or by some other numerical order. There are a number of different ways of sorting data in Access. A quick sort on a single field in either ascending or descending order, can be achieved in table datasheet view, or you can sort on multiple fields using either the query or report functions.

To sort on a single field:

1. From the FRIENDS Database Window click on the *Table* object, and then double click on the FRIENDS table to open in datasheet view. Let's say you would like to sort the records alphabetically by Last Name. To do this, move the mouse to the Last Name field selector until the ↓ arrow appears. Click to select the Last Name column (see Figure 5-40).

2. Click the Sort Ascending toolbar button, or select RECORDS/SORTand ASCENDING from the menu, and Access will sort the records into alphabetical order. (Clicking on the *Sort Descending* toolbar button will sort in reverse order).

Figure 5-40

FRIENDS : Table

LAST_NAME	FIRST_NAME	STREET	CITY	STATE
Drucker	Peter H.	345 Warren Road	Hudson	New York
Whitney	Craig	25 Wood Lake Roa	Morris	New Jersey
Sitkin	Howard W.	Morace Street	Springvale	New Hampshir
Skalek	William F.	8 Yorkshire Place	Teatown	South Dakota
Salione	Phillip	35 Truesdale Ave	Phoenix	Arizona
Fabian	James T.	36 Palmer Court	Chicago	Illinois
Kohlman	Frank	35 Miller Drive	Milwaukee	Wisconsin
Tedesco	George R.	346 Skytop Drive	Spokane	Washington
Zito	Helen K.	64 Albany Post Rd	Dana	Maryland
Peterson	Jack S.	54 Elmor Ave	Barston	Ohio
Nelson	Robert M.	1 Franklin Ave.	St. Louis	Missouri

Record: 1 of 11

Warning: this is a dynamic sort and cannot be saved.

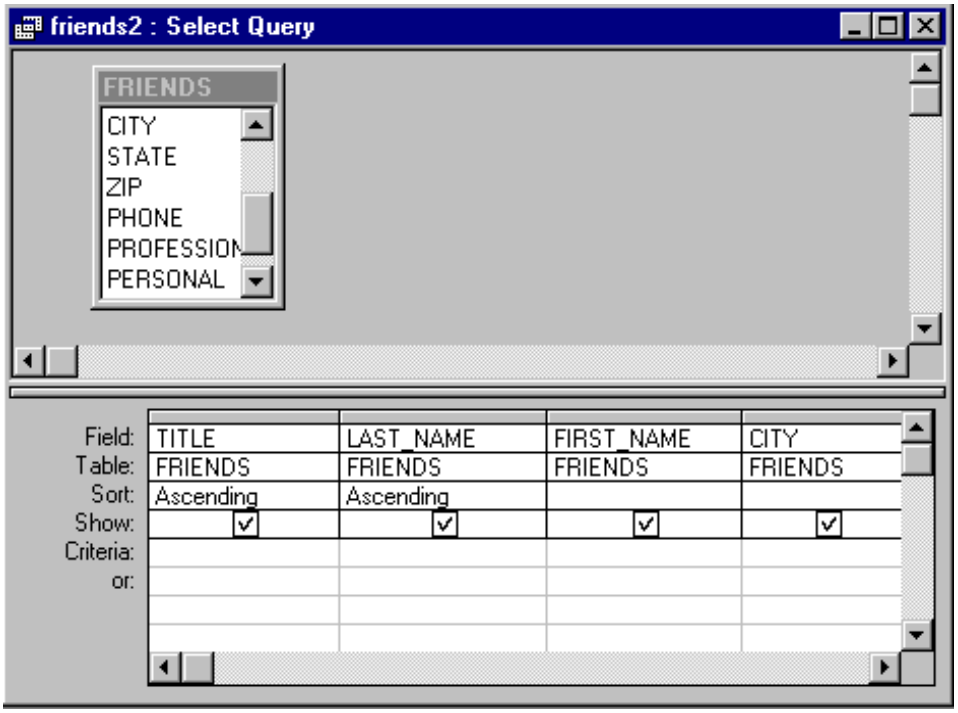
Sorts-within-Sorts

You can sort on multiple fields in a query to achieve a sort within a sort. This is analogous to a telephone directory where entries are firstly alphabeticised by Last Name, and then by First Name.

1. Press F11 to return to the Database Window. Click on the Query object and then double click on the *Tute1 Query2* query we created in Tutorial 1.

2. Let's sort the Title field and then within this, the Last Name field in ascending order. Before we do this, first delete "Dr" from the Criteria: bar in the title field. Then click on the Sort: bar in the Title field and select Ascending order. Repeat this action for the Last Name field. Your screen should now look like the one shown in Figure 5-41.

Figure 5-41



3. Click on the Run toolbar button, or select QUERY/RUN from the menu to see the results of your query. The records should be sorted firstly by Title (ie: Dr before Mr, Ms or Prof), and then within each Title grouping, in ascending Last Name order.

4. Save your query with a new name by selecting FILE/SAVE AS from the menu, and typing in a new name (eg: *Tute3 Query2*). Press F11 to return to the Database Window.

Sorting Data in Reports

When you print a report, you usually want to order the records in a particular way. For example, if you were printing out a list of suppliers, you may wish to sort the records alphabetically by company name. When you are setting up the parameters for a new report with Report Wizards, Access gives you the opportunity to specify a field(s) sort order. If you change your mind after the report has been created, use the Sorting and Grouping tool in report design view. This is a very powerful feature of Access.

The Sorting and Grouping tool allows you to sort on up to 10 fields and expressions, and you can sort on the same field or expression more than once. For example in a five character field, you could sort on ascending order on the first three characters, and in descending order on the last two characters.

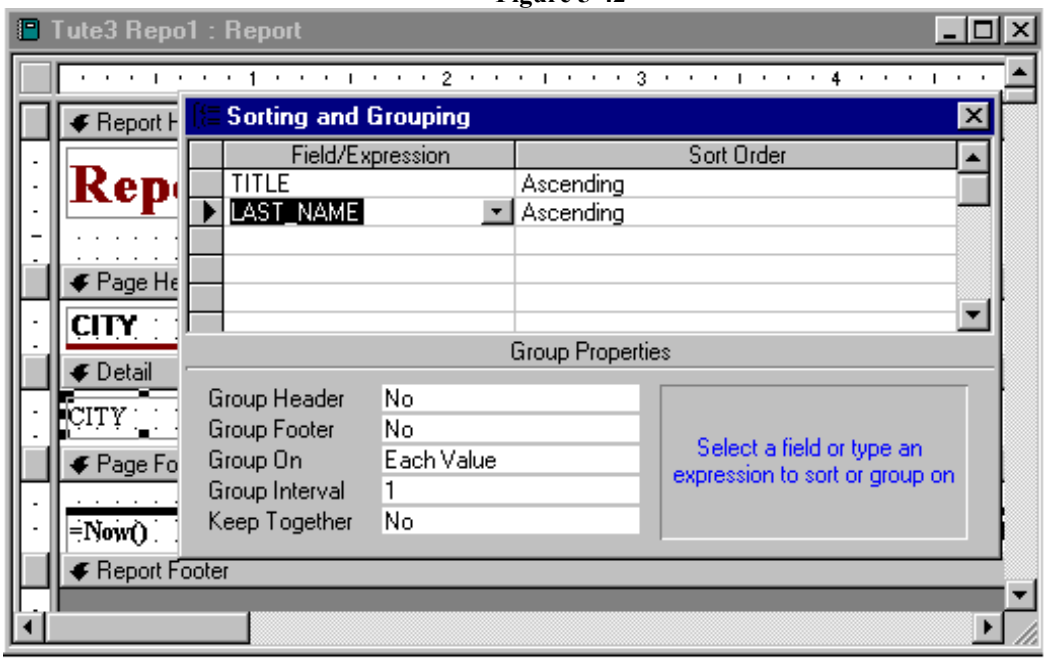
Let's repeat the query sort we have just performed in report mode. (We could of course simply rename the record source for the report using the query we have just created to achieve the same results).

1. From the Database Window, click on the Report object, and click to highlight the *Tute2 Repo1* report created in the Tutorial for Case 2. Click on the Design button to go to design view for this report.

2. Click on the Sorting and Grouping toolbar button, or select VIEW/SORTING AND GROUPING from the menu.

3. Click on City in the Field/Expression column, and change the field to Title. Click on the blank row under this to display a list of fields and select Last Name. Note that the default Sort Order is ascending which is perfect for our needs. Your screen should now look similar to Figure 5-34.

Figure 5-42



4. Click on the Print Preview button or select FILE/PRINT PREVIEW to see the effect that imposing a sort order has had on the report. Send your report to print.

5. Save the report with a new name (eg: *Tutorial3 Repo1*) by selecting FILE/SAVE AS from the menu. Press F11 to return to the Database Window, and exit Access by choosing FILE/EXIT from the menu.

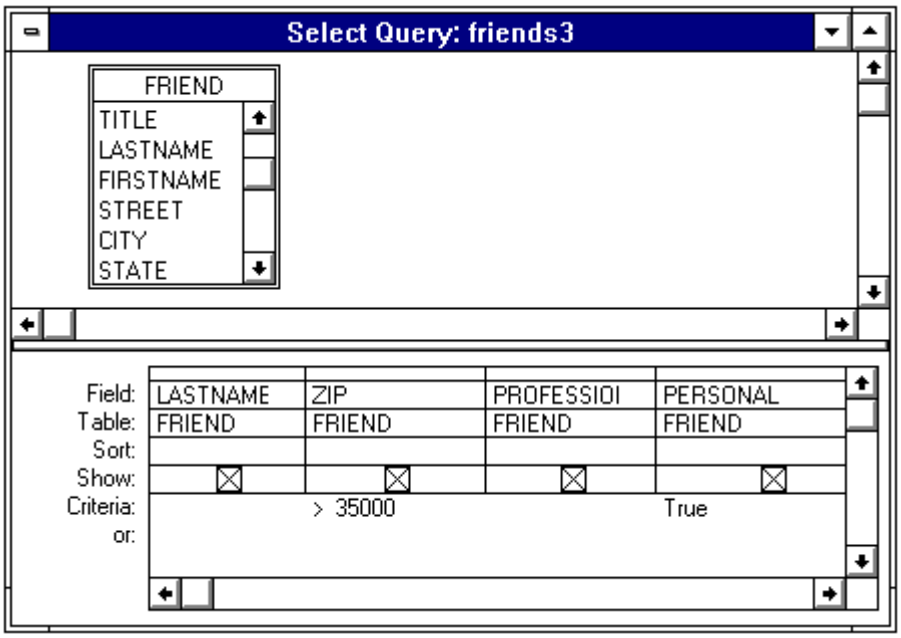
Tutorial For Database Case 3 Using Access 2.0

Using Query Files With Multiple Conditions

In the Tutorial for Case 1, we looked at creating queries for viewing and printing certain fields in the FRIENDS table, and fields which met a simple selection criteria. In this tutorial, we will look at constructing more complex queries using logical operators.

1. Load the practice database FRIENDS.MDB, and create a new query following the instructions given in the Tutorial for Case 1. Include the fields LastName, Zip, Profession and Personal in the query.

Figure 5-43



2. The Criteria: and or: bars of the QBE Grid allow for multiple conditions involving the logical operators "AND" and "OR". Any conditions entered on the first line of the Criteria: bar are "ANDed". Let's say you wanted a listing of all Personal friends with Zip codes greater than 35000. In the Criteria: bar of the QBE Grid (see Figure 5-43), type >35000 under the Zip field and enter True under the Personal field.

3. Run the query by clicking on the Run toolbar button, or selecting QUERY/RUN from the menu. Results should show that there are four records in the FRIENDS table which match the chosen criteria.

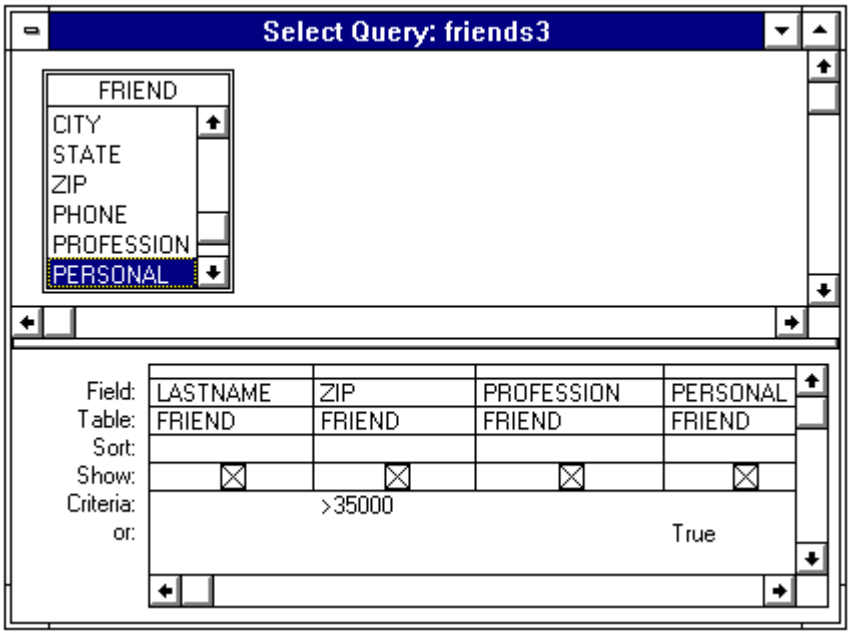
4. Click on the Design View toolbar button to return to the query design window.

5. Now try using the "OR" operator by deleting True from the Criteria: bar under the Personal field, and entering it on the or: bar on the next line down (see Figure 5-44) In this

instance, we are searching the table for people whose Zip code is greater than 35000 *or* people who are personal friends. Run the query again. This is a less stringent condition, and the resulting dynaset should show that there are nine records which match the chosen criteria.

6. Print your queries by clicking on the Print toolbar button, or selecting FILE/PRINT from the menu. Save your query (eg: *Tute3 Query1*), and press F11 to return to the Database Window.

Figure 5-44



Sorting Tables in Access

It is often useful to sort a table by one or more fields. For instance, you may find it helpful to sort the records in your table alphabetically by Last Name, or to sort records in chronological order, or by some other numerical order. There are a number of different ways of sorting data in Access. A quick sort on a single field in either ascending or descending order, can be achieved in table datasheet view, or you can sort on multiple fields using either the query or report functions.

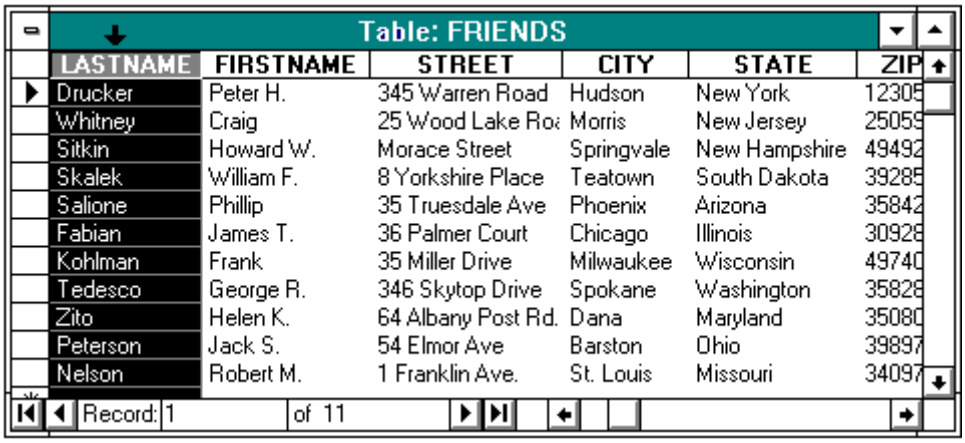
To sort on a single field:

1. From the FRIENDS Database Window click on the *Table* object, and then double click on the FRIENDS table to open in datasheet view. Let's say you would like to sort the records alphabetically by Last Name. To do this, move the mouse to the Last Name field selector until the ↓ arrow appears. Click to select the Last Name column (see Figure 5-45).

2. Click the Sort Ascending toolbar button, or select RECORDS/QUICK SORT and ASCENDING from the menu, and Access will sort the records into alphabetical order.

(Clicking on the *Sort Descending* toolbar button will sort in reverse order).

Figure 5-45



Warning: this is a dynamic sort and cannot be saved.

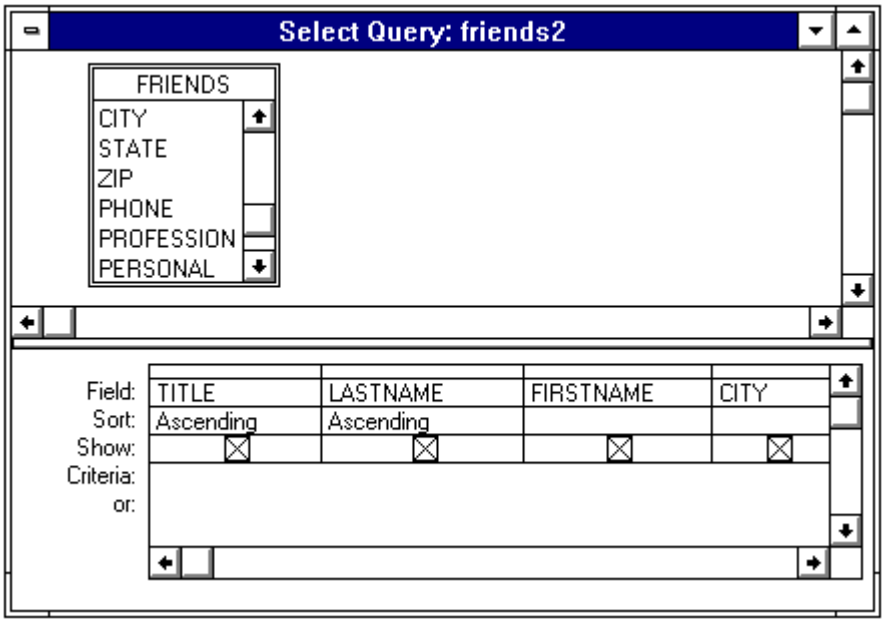
Sorts-within-Sorts

You can sort on multiple fields in a query to achieve a sort within a sort. This is analogous to a telephone directory where entries are firstly alphabeticised by Last Name, and then by First Name.

1. Press F11 to return to the Database Window. Click on the Query object and then double click on the *Tute2 Query2* query we created in Tutorial 2.

2. Let's sort the Title field and then within this, the Last Name field in ascending order. Before we do this, first delete "Dr" from the Criteria: bar in the title field. Then click on the Sort: bar in the Title field and select Ascending order. Repeat this action for the Last Name field. Your screen should now look like the one shown in Figure 5-46.

Figure 5-46



3. Click on the Run toolbar button, or select QUERY/RUN from the menu to see the results of your query. The records should be sorted firstly by Title (ie: Dr before Mr, Ms or Prof), and then within each Title grouping, in ascending Last Name order.

4. Save your query with a new name by selecting FILE/SAVE AS from the menu, and typing in a new name (eg: *Tute3 Query2*). Press F11 to return to the Database Window.

Sorting Data in Reports

When you print a report, you usually want to order the records in a particular way. For example, if you were printing out a list of suppliers, you may wish to sort the records alphabetically by company name. When you are setting up the parameters for a new report with Report Wizards, Access gives you the opportunity to specify a field(s) sort order. If you change your mind after the report has been created, use the Sorting and Grouping tool in report design view. This is a very powerful feature of Access.

The Sorting and Grouping tool allows you to sort on up to 10 fields and expressions, and you can sort on the same field or expression more than once. For example in a five character field, you could sort on ascending order on the first three characters, and indescending order on the last two characters.

Let's repeat the query sort we have just performed in report mode. (We could of course simply rename the record source for the report using the query we have just created to achieve the same results).

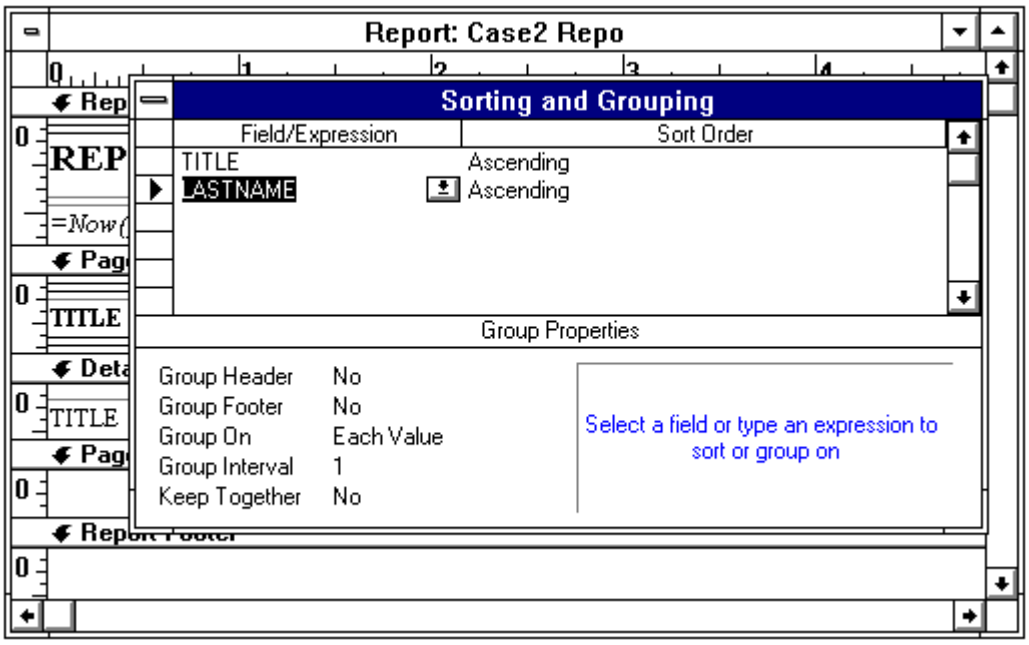
1. From the Database Window, click on the Report object, and click to highlight the *Tute2 Repo1* report created in the Tutorial for Case 2. Click on the Design button to go to design view for this report.

2. Click on the Sorting and Grouping toolbar button, or select VIEW/SORTING AND

GROUPING from the menu.

3. Click on City in the Field/Expression column, and change the field to Title. Click on the blank row under this to display a list of fields and select Last Name. Note that the default Sort Order is ascending which is perfect for our needs. Your screen should now look similar to Figure 5-47.

Figure 5-47



4. Click on the Print Preview button or select FILE/PRINT PREVIEW to see the effect that imposing a sort order has had on the report. Send your report to print.

5. Save the report with a new name (eg: *Tutorial3 Repo1*) by selecting FILE/SAVE AS from the menu. Press F11 to return to the Database Window, and exit Access by choosing FILE/EXIT from the menu.

Database Case 4

Hadad's Pharmacy

Problem:	Summarize and analyze data from a transaction processing system for government reporting requirements
Management Skills:	Organize Control
Access Skills:	Queries Sorting Data Reduction Reporting

Data Table: HADAD

Andrew Hadad, the owner and pharmacist of Hadad's Pharmacy in Portland, Oregon, was going through his daily mail and came across a letter he had been expecting from the state government. The letter informed him of new regulations and reporting requirements for certain categories of drugs.

The Oregon state government had begun an initiative to restrict the growing use of addictive and dangerous drugs. One of the first steps involved identifying the type and amount of these drugs sold by each pharmacy in the state. This data would then be analyzed and any regional trends identified.

The new regulations had been discussed at a recent meeting of Andrew's local chapter of the Pharmacy Guild - the state professional body for pharmacists. Most pharmacists agreed that the new controls were a good idea, but the thought of yet another record keeping chore was not attractive.

Three years ago the members of Andrew's Pharmacy Guild had each put in a personal computer with software to manage the pharmacy prescription process. When a customer wanted a prescription filled, the pharmacist entered their personal details (if a new customer) and the drug and dosage requirements prescribed by the doctor. The system produced three outputs: a sticky label to affix to the medication, a repeat prescription if required, and a detailed receipt for the customer for insurance purposes. For regular customers, the software also had the ability to check whether new medication would interact adversely with any existing medication.

The letter from the government specified that usage information was required for only six classes of drugs: sedatives, analgesics, anti-hypertensives, diuretic, anti-inflammatories and anti-depressives. At any given time, Andrew's pharmacy stocked around 100 drugs in these target groups.

The government required each pharmacy to submit a quarterly summary report indicating the usage of each drug issued in the 6 target classes during this period. Total usage of these 6 classes was also required. The government would conduct random cross checks with drug company delivery records to check the accuracy of the reporting. Severe penalties were specified for non-compliance.

The table below shows a sampling of the codes assigned to drug classes used by Andrew's pharmacy prescription program. Use this to identify the drug codes of interest to the government.

DRUG_CODE	DRUG_CLASS
B1	Steroids
B2	Diuretic
B3	Anti-hypertensives
B4	Analgesics
G7	Anti-diarrhoea
G8	Birth Control
G9	Sedatives
Q1	Anti-inflammatories
Q2	Anti-malaria
Q3	Antibiotic
Q4	Anti-depressives

ChemSoft, the firm who had provided the computerized prescription system had also attended the Pharmacy Guild's meeting. They indicated it would be a major job to change the current system to meet the government's requirements. They quoted a very high fee, and a 3 month delay before conversion would be possible. They could however immediately deliver an add-on program that would convert each quarter's prescriptions into a generic data format which could be readily used by a variety of Windows-based database packages.

For a low cost solution, ChemSoft recommended each pharmacist purchase a Windows-based database package and use it in tandem with the conversion program to perform the government's usage analysis.

The database package could also be used for other purposes. Many of the pharmacies stocked a range of gift lines, vitamins, and fashion accessories, and used a manual card system to keep track of their inventory and suppliers in these areas. Using a database to also automate this side of the business was an attractive proposition.

Andrew was sure it would not be difficult to design a database program to meet state government needs. The job could be performed on the pharmacy's personal computer on weekends when the store was closed. Andrew estimated that he only used 2 or 3 of the drugs on the target list on any given day. He also believed that some of the information required by the government would be useful for his business. For example, it would be handy to know how much he sold of each company's drugs.

One problem still bothered Andrew. Each drug company supplied medication in different tablet sizes. The strength of each tablet also varied in terms of milligram dosage. The government needed its usage statistics to be reported in milligrams (mgs). Fortunately, the pharmacy prescription system recorded both the number of tablets and their strength (in mgs) used for each prescription. To satisfy the government's requirements, it would simply be a matter of multiplying the number of tablets by strength to get total milligram usage.

Part of the data table produced by the conversion program for a quarter's of prescriptions issued by Hadad Pharmacy has been saved as an Access table called HADAD on SOLVEIT.MDB. Create a new empty database, and import this table now.

Tasks: There are six tasks in this case:

1. Design a query to filter out all of the drugs not of interest to the state government.

2. Design a summary report to present to the state government each quarter. Include only the total drug usage and the sub-total usage for each Drug Code. The report should be sorted by Drug Code. Do not include the individual prescription records.

3. Print this report.

4. Andrew would like to discover which drug companies produce the drugs he sells under each Drug Code. Design and produce a report that presents all the quarter's transactions sorted by Drug Code. Within each Drug Code sort alphabetically by Supplier. Include all appropriate fields including Drug Name and Supplier.

5. Andrew has been frustrated by the slow delivery of supplies by Roche. The Roche sales representative is coming to visit next week and Andrew will raise the subject. Andrew would like to prepare for the visit by finding out how much of Roche's products he actually uses. Create and print out a report of all the prescriptions Hadad Pharmacy dispensed where the drug was provided by Roche. He would like this list sorted by Drug Code with sub-total usages included for each class.

*6. ChemSoft has extended their pharmacy system. The new system is compatible with a number of Windows-based database packages including Access. The system is totally menu driven and the pharmacist just selects which operation is required. Menu options include, "Enter a New Customer" and "Fill Prescription". The system has four main data files: customer, drug information, inventory, and transaction records.

   The customer enters the store and gives the prescription to the pharmacist or shop assistant. The pharmacist goes to the computer and checks whether the customer is on file. If not, customer details are entered, otherwise the customer's record is located by the system. Prescription details are then entered. A check is automatically made to ensure the new drug does not interact with any medication currently being used by the customer.

   The prescription is then filled by the pharmacist, the inventory level for that drug is adjusted downwards, a sticky label is produced and a record is added to the transaction database. The transaction database is just a log of every prescription issued. A repeat script is printed if required, and a detailed customer receipt printed for insurance purposes.

Draw a data flow diagram (first-level only) describing ChemSoft's new system. Include the major processes, data files, data flows, system outputs and people involved in the system. Identify all the fields in each of the data files.

At the end of each week a number of management reports are produced. Suggest what you think these reports will be, and what fields they are likely to contain.

Information on producing data flow diagrams is contained in most information systems texts. Our reference is: *"Essentials of Management Information Systems: Transforming Business and Management", K.C. Laudon & Jane P. Laudon, 1999, Prentice-Hall.*

Time Estimates (excluding task marked *)

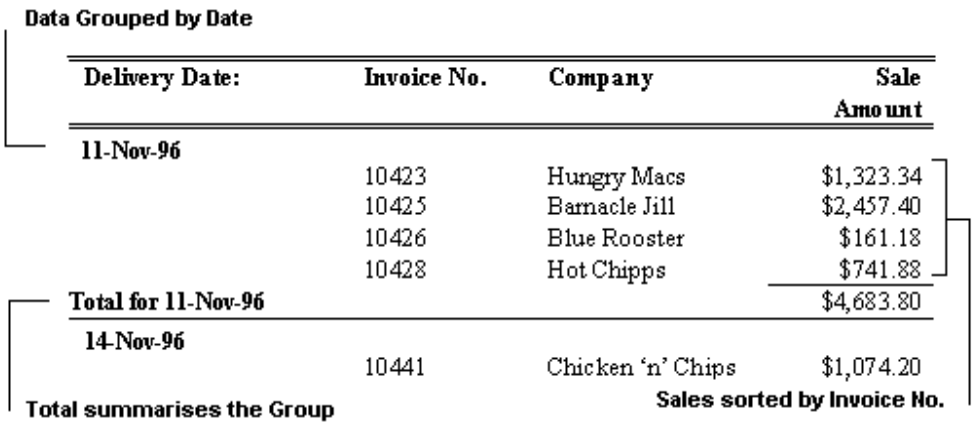
Expert: 30 minutes
Intermediate: 1.5 hours
Novice: 2.5 hours

Tutorial For Database Case 4 Using Access 97

How to Create Record Groups in Reports

For many reports, sorting the records isn't enough - you may also want to divide them into groups. A group is a collection of records that share a common characteristic such as the same Product Number or the same Zip code. In Access, a group consists of a group header, a series of detail records, and a group footer.

Grouping allows you to separate records of groups visually, and to display introductory and summary data for each group. For example, the report extract shown below groups sales by date and calculates the total amount of sales for each day.



Let's produce a report with groups using the FRIENDS database and the report (*Tute2 Repo1*) we created in the Tutorial for Case 2. We want to group the report by State to produce a listing of friends sorted alphabetically (ascending order) by state. For friends in the same state, we want to sort these alphabetically by Last Name. Before we can do this, we need to make some modifications to the existing FRIENDS table. Load the practice database FRIENDS.MDB.

1. From the FRIENDS Database Window, click on the Table object, and double click on the FRIENDS table. Add four or five new records to the table, making sure that some of the state names you enter in the State field duplicate those of existing records (see Figure 5-47). To speed up this procedure, just enter new records for the Title, Last Name, First Name, Address, City and State fields.

Figure 5-47

Table: FRIENDS

TITLE	LASTNAME	FIRSTNAME	STREET	CITY	STATE
Dr	Drucker	Peter H.	345 Warren Road	Hudson	New York
Mr	Whitney	Craig	25 Wood Lake Roa	Morris	New Jersey
Mr	Sitkin	Howard W.	Morace Street	Springvale	New Hampshire
Dr	Skalek	William F.	8 Yorkshire Place	Teatown	South Dakota
Prof	Salione	Phillip	35 Truesdale Ave	Phoenix	Arizona
Mr	Fabian	James T.	36 Palmer Court	Chicago	Illinois
Mr	Kohlman	Frank	35 Miller Drive	Milwaukee	Wisconsin
Mr	Tedesco	George R.	346 Skytop Drive	Spokane	Washington
Ms	Zito	Helen K.	64 Albany Post Rd.	Dana	Maryland
Dr	Peterson	Jack S.	54 Elmor Ave	Barston	Ohio
Dr	Nelson	Robert M.	1 Franklin Ave.	St. Louis	Missouri
Mr	Eddy	Steven	12 Hedge Street	New York	New York
Ms	Satchel	Simone	111 Eagle Ave	Green Bay	Wisconsin
Mr	Sorini	Roberto	4 Capital Court	Winslow	Arizona
Ms	Nation	Carry	90 Buzzard Rd	Tuscon	Arizona
Mr	Henderson	Henry	Seaspray Drive	Manchester	New Hampshire

Record: 12 of 16

2. Press F11 to return to the Database Window.

3. Click on the Report object, and then click and highlight the report you created in Tutorial 2 (eg: Tutorial2 Repo). Click on the Design button, to enter design view for this report.

3. Change the Record Source for the report to the FRIENDS table. To do this, choose SELECT REPORT from the EDIT menu, and then click the *Properties* toolbar button, or choose VIEW/PROPERTIES from the menu. This activates the report properties sheet.

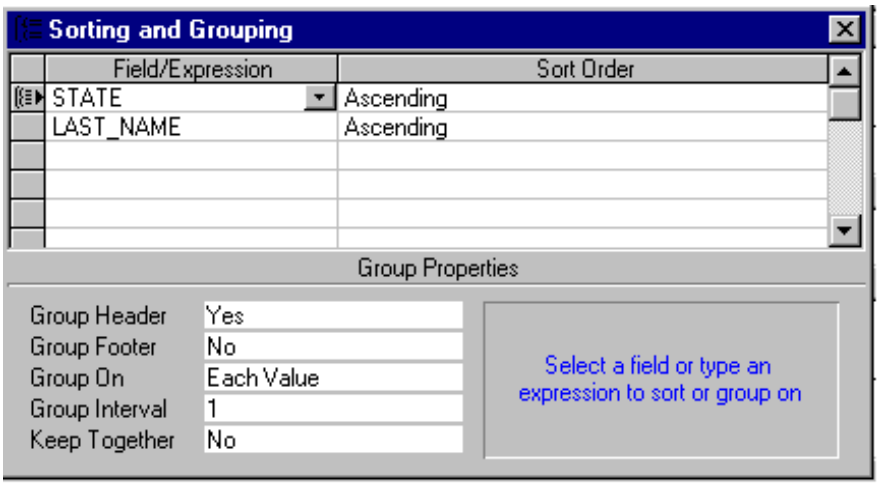
4. On the properties sheet, click the ▼ button to the right of the Record Source bar. This activates a drop down list of all tables and queries associated with the FRIENDS database. Click on the FRIENDS table to change the record source for the report.

5. Close the properties sheet by clicking on the ☒ box in the top right hand corner.

6. Click on the Sorting and Grouping toolbar button, or select VIEW/SORTING AND GROUPING from the menu.

7. Click on City in the Field/Expression column, and change the field to State, and then click on the Group Header bar and change the default to Yes. Click on the blank row in the Field/Expression column immediately under State to display a list of fields and select Last Name. Leave the Group Header for Last Name set to "No". Accept the default ascending sort order for both fields. Your screen should look similar to Figure 5-48.

Figure 5-48



8. Exit the Sorting and Grouping box by clicking on the box in the top right hand corner, and return to report design view.

You should notice that a blank *State Header* area has been added to the report design.

9. Select VIEW/FIELDLIST from the menu or click on the *FieldList* toolbar button to activate a scrollable field list for the FRIENDS table. Click and highlight the State field, and then holding down your left mouse button, drag the State field into the blank area of the State Header, and position against the left margin.

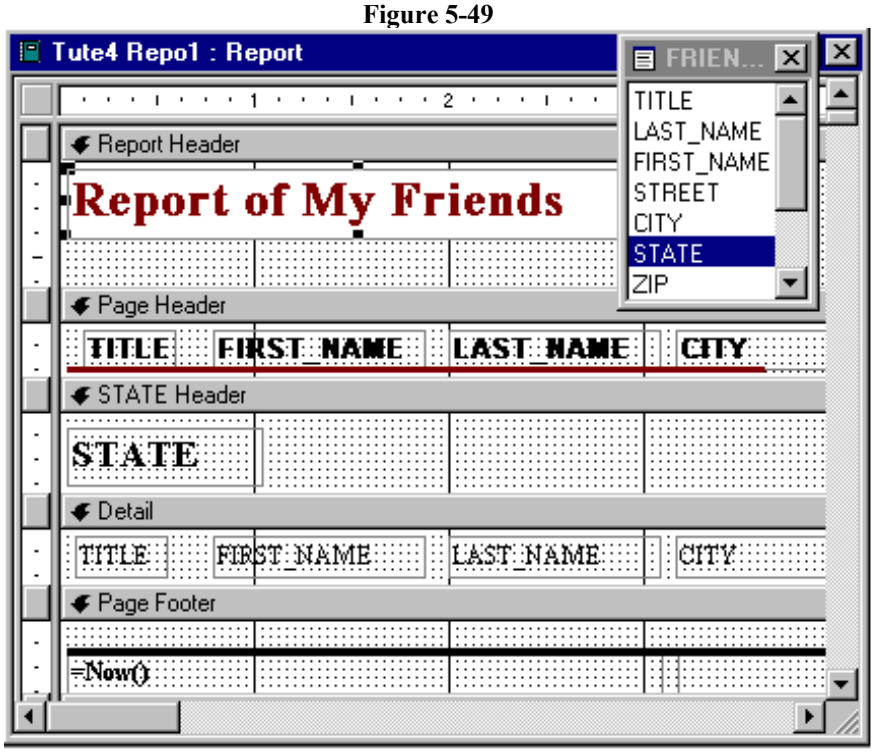
10. Double click on the State field to activate its *Properties* sheet. Scroll down the list and change the *Font Size* to 14, and the *Font Weight* to Bold type. Click on the box in the top right hand corner, and return to report design view. Your screen should look similar to Figure 5-49.

11. Select FILE/PRINT PREVIEW to see the effect that imposing a sort order has had on the report. If your report appears in Landscape orientation, select FILE/PRINT SETUP from the menu to change the orientation to Portrait mode. Send your report to print.

Your report should display the FRIENDS records grouped alphabetically by State, and within each group, ordered alphabetically by Last Name.

12. Save the report with a new name (eg: *Tute4 Repo1*) by selecting FILE/SAVE AS from the menu. Click on the *Close Window* toolbar button Close to return to report design view.

Figure 5-49



Calculating Group Summary Statistics

You may often want to ask questions about groups of data such as "How many orders did we receive this month?" or "What's the average price of all the products in our Toothpaste range?". You can perform calculations on groups of records in reports or queries. The following table shows some of the types of *functions* (calculations) you can use with Access (refer also Access Tutorial for Case 2)

Use this type of calculation	**To Find**
Sum	the total of values in a field
Avg	the average of values in a field
Min	the lowest value in a field
Max	the highest value in a field
Count	the number of values in a field (not including null values)

Let's use the Count function to count the number of friends we have in each State, and display that number at the end of each group in our report.

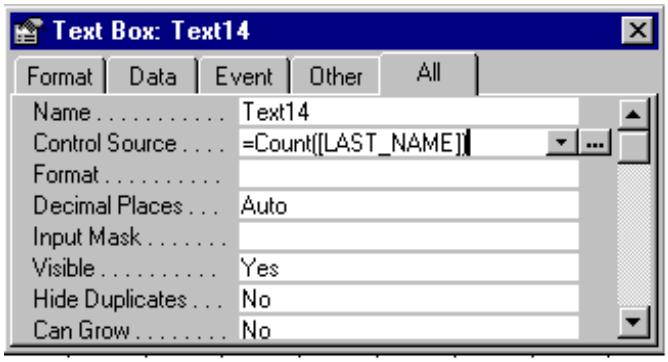
1. In report design view, click on the Sorting and Grouping toolbar button, or select VIEW/SORTING AND GROUPING from the menu.

2. Click on State in the Field/Expression column, and then click on the Footer bar and change the default to Yes.

3. Exit the Sorting and Grouping box by clicking on the box in the top right hand corner, and return to report design view. You should notice that a blank *State Footer* area has been added to the report design.

4. Select VIEW/FIELDLIST from the menu or click on the *FieldList* toolbar button to activate a scrollable field list for the FRIENDS table. Click and highlight the LastName field, and then holding down your left mouse button, drag the LastName field into the blank area of the State Footer, and position against the right margin.

Figure 5-50



5. Double click on the LastName field in the State Footer to activate its *Properties* sheet. Edit the *Control Source* bar to include the Count calculation (see Figure 5-50). Be particularly careful with the syntax here, and with the entering of round and square brackets.

Click on the box in the top right hand corner, and return to report design view.

6. Click on the Print Preview button or select FILE/PRINT PREVIEW to see the effect of this change on your report. Results should be similar to Figure 5-51. Send your report to print.

Figure 5-51

Arizona

Ms	Carry	Nation	Tuscon	
Prof	Phillip	Salione	Phoenix	
Mr	Roberto	Sorini	Winslow	
				3

Illinois

Mr	James T.	Fabian	Chicago	
				1

Maryland

Ms	Helen K.	Zito	Dana	
				1

7. Save the report with a new name (eg: *Tute4 Repo2*) by selecting FILE/SAVE AS from the menu. Press F11 to return to the Database Window.

Deleting Records In Access

To delete a record in an Access table, highlight the record row intended for deletion, and either press the *Del* key or select EDIT/DELETE. Access will ask you to confirm your action. Click OK to proceed with deletion of the record.

Tutorial For Database Case 4 Using Access 2.0

How to Create Record Groups in Reports

For many reports, sorting the records isn't enough - you may also want to divide them into groups. A group is a collection of records that share a common characteristic such as the same Product Number or the same Zip code. In Access, a group consists of a group header, a series of detail records, and a group footer.

Grouping allows you to separate records of groups visually, and to display introductory and summary data for each group. For example, the report extract shown below groups sales by date and calculates the total amount of sales for each day.

Data Grouped by Date

Delivery Date:	Invoice No.	Company	Sale Amount
11-Nov-94			
	10423	Hungry Macs	$1,323.34
	10425	Barnacle Jill	$2,457.40
	10426	Blue Rooster	$161.18
	10428	Hot Chipps	$741.88
Total for 11-Nov-94			$4,683.80
14-Nov-94			
	10441	Chicken 'n' Chips	$1,074.20

Total summarises the Group

Sales sorted by Invoice No.

Figure 5-52

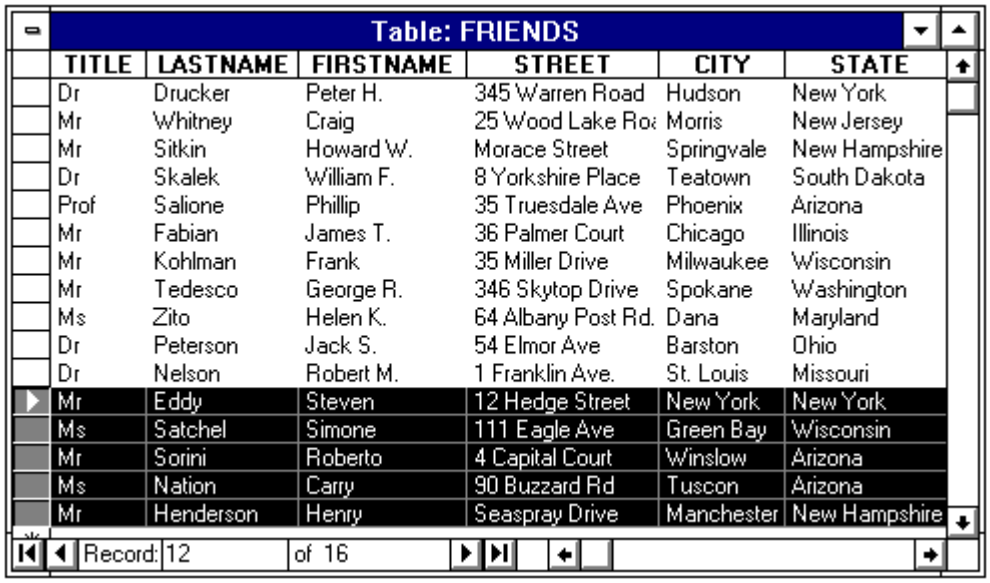


Table: FRIENDS

TITLE	LASTNAME	FIRSTNAME	STREET	CITY	STATE
Dr	Drucker	Peter H.	345 Warren Road	Hudson	New York
Mr	Whitney	Craig	25 Wood Lake Roa	Morris	New Jersey
Mr	Sitkin	Howard W.	Morace Street	Springvale	New Hampshire
Dr	Skalek	William F.	8 Yorkshire Place	Teatown	South Dakota
Prof	Salione	Phillip	35 Truesdale Ave	Phoenix	Arizona
Mr	Fabian	James T.	36 Palmer Court	Chicago	Illinois
Mr	Kohlman	Frank	35 Miller Drive	Milwaukee	Wisconsin
Mr	Tedesco	George R.	346 Skytop Drive	Spokane	Washington
Ms	Zito	Helen K.	64 Albany Post Rd.	Dana	Maryland
Dr	Peterson	Jack S.	54 Elmor Ave	Barston	Ohio
Dr	Nelson	Robert M.	1 Franklin Ave.	St. Louis	Missouri
Mr	Eddy	Steven	12 Hedge Street	New York	New York
Ms	Satchel	Simone	111 Eagle Ave	Green Bay	Wisconsin
Mr	Sorini	Roberto	4 Capital Court	Winslow	Arizona
Ms	Nation	Carry	90 Buzzard Rd	Tuscon	Arizona
Mr	Henderson	Henry	Seaspray Drive	Manchester	New Hampshire

Record: 12 of 16

Let's produce a report with groups using the FRIENDS database and the report (*Tute2 Repo1*) we created in the Tutorial for Case 2. We want to group the report by State to produce a listing of friends sorted alphabetically (ascending order) by state. For friends in the same state, we want to sort these alphabetically by Last Name. Before we can do this, we need to make some modifications to the existing FRIENDS table.

Load the practice database FRIENDS.MDB. From the FRIENDS Database Window, click on the Table object, and double click on the FRIENDS table. Add four or five new records to the table, making sure that some of the state names you enter in the State field duplicate those of existing records (see Figure 5-52). To speed up this procedure, just enter new records for the Title, Last Name, First Name, Address, City and State fields.

2. Press F11 to return to the Database Window.

3. Click on the Report object, and then click and highlight the report your created in Tutorial 2 (eg: Tutorial2 Repo). Click on the Design button, to enter design view for this report.

4. Change the Record Source for the report to the FRIENDS table. To do this, choose SELECT REPORT from the EDIT menu, and then click the *Properties* toolbar button, or choose VIEW/PROPERTIES from the menu. This activates the report properties sheet.

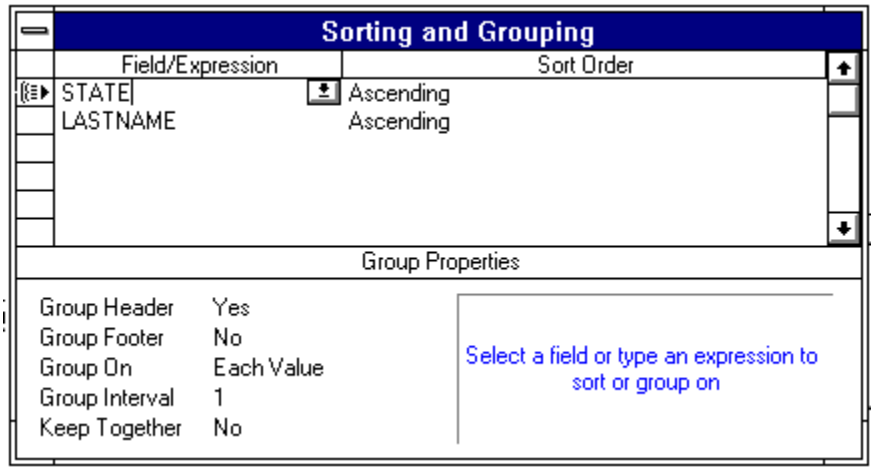
5. On the properties sheet, click the ▼ button to the right of the Record Source bar. This activates a drop down list of all tables and queries associated with the FRIENDS database. Click on the FRIENDS table to change the record source for the report.

6. Close the properties sheet by double clicking on the  box in the top left hand corner.

7. Click on the Sorting and Grouping toolbar button, or select VIEW/SORTING AND GROUPING from the menu. 

8. Click on City in the Field/Expression column, and change the field to State, and then click on the Group Header bar and change the default to Yes. Click on the blank row in the Field/Expression column immediately under State to display a list of fields and select Last Name. Leave the Group Header for Last Name set to "No". Accept the default ascending sort order for both fields. Your screen should look similar to Figure 5-53.

Figure 5-53



8. Exit the Sorting and Grouping box by double clicking on the box in the top left hand corner, and return to report design view.

You should notice that a blank *State Header* area has been added to the report design.

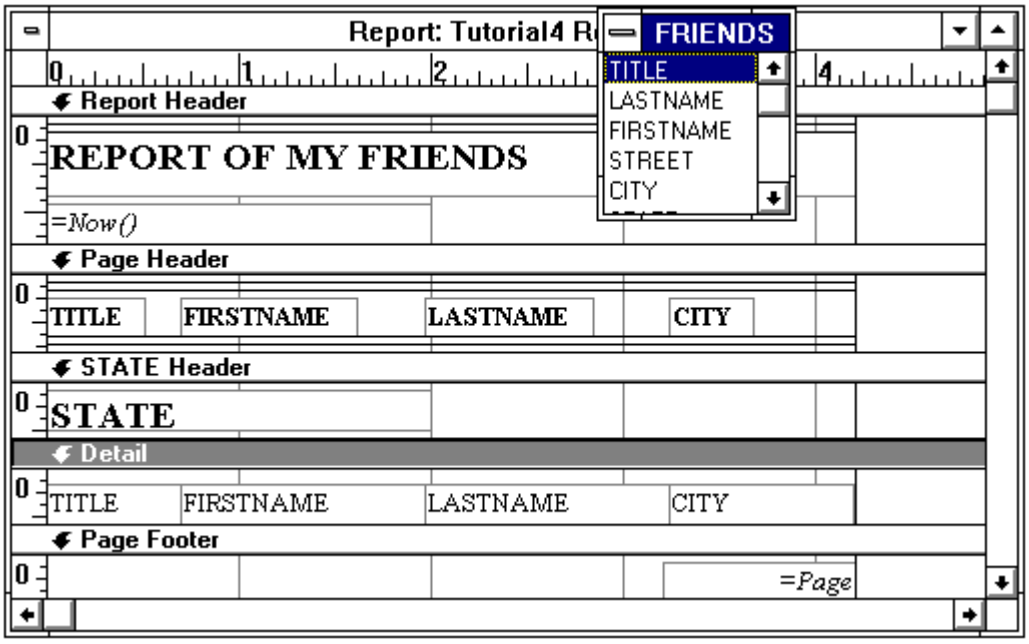
9. Select VIEW/FIELDLIST from the menu or click on the *FieldList* toolbar button to activate a scrollable field list for the FRIENDS table. Click and highlight the State field, and then holding down your left mouse button, drag the State field into the blank area of the State Header, and position against the left margin.

10. Double click on the State field to activate its *Properties* sheet. Scroll down the list and change the *Font Size* to 14, and the *Font Weight* to Bold type. Double click on the box in the top left hand corner, and return to report design view. Your screen should look similar to Figure 5-56.

11. Click on the Print Preview button or select FILE/PRINT PREVIEW to see the effect that imposing a sort order has had on the report. If your report appears in Landscape orientation, click on the Print Setup toolbar button, or select FILE/PRINT SETUP from the menu to change the orientation to Portrait mode. Send your report to print.

Your report should display the FRIENDS records grouped alphabetically by State, and within each group, ordered alphabetically by Last Name.

Figure 5-54



12. Save the report with a new name (eg: *Tute4 Repo1*) by selecting FILE/SAVE AS from the menu. Click on the *Close Window* toolbar button to return to report design view.

Calculating Group Summary Statistics

You may often want to ask questions about groups of data such as "How many orders did we receive this month?" or "What's the average price of all the products in our Toothpaste range?". You can perform calculations on groups of records in reports or queries. The following table shows some of the types of *functions* (calculations) you can use with Access (refer also Access Tutorial for Case 2)

Use this type of calculation	**To Find**
Sum	the total of values in a field
Avg	the average of values in a field
Min	the lowest value in a field
Max	the highest value in a field
Count	the number of values in a field (not including null values)

Let's use the Count function to count the number of friends we have in each State, and display that number at the end of each group in our report.

1. In report design view, click on the Sorting and Grouping toolbar button, or select VIEW/SORTING AND GROUPING from the menu.

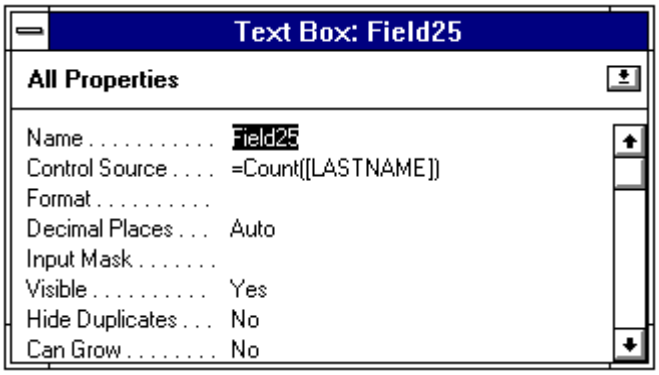
2. Click on State in the Field/Expression column, and then click on the Footer bar and change the default to Yes.

3. Exit the Sorting and Grouping box by double clicking on the box in the top left hand corner, and return to report design view. You should notice that a blank *State Footer* area has been added to the report design.

4. Select VIEW/FIELDLIST from the menu or click on the *FieldList* toolbar button to activate a scrollable field list for the FRIENDS table. Click and highlight the LastName field, and then holding down your left mouse button, drag the LastName field into the blank area of the State Footer, and position against the right margin.

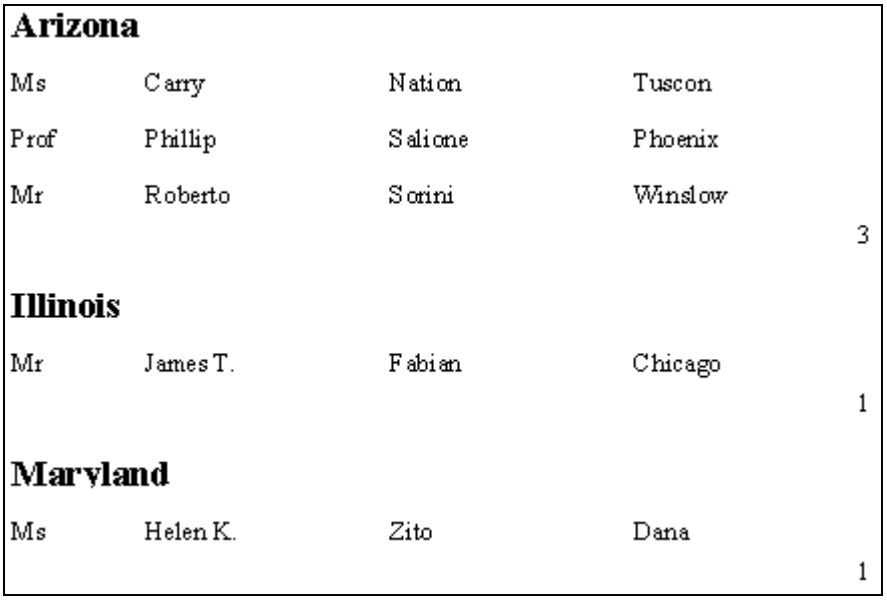
5. Double click on the LastName field in the State Footer to activate its *Properties* sheet. Edit the *Control Source* bar to include the Count calculation (see Figure 5-55). Be particularly careful with the syntax here, and with the entering of round and square brackets. Double click on the box in the top left hand corner, and return to report design view.

Figure 5-55



6. Click on the Print Preview button or select FILE/PRINT PREVIEW to see the effect of this change on your report. Your report results should be similar to Figure 5-56. Send your report to print.

Figure 5-56



Arizona

Ms	Carry	Nation	Tuscon
Prof	Phillip	Salione	Phoenix
Mr	Roberto	Sorini	Winslow

3

Illinois

Mr	James T.	Fabian	Chicago

1

Maryland

Ms	Helen K.	Zito	Dana

1

7. Save the report with a new name (eg: *Tute4 Repo2*) by selecting FILE/SAVE AS from the menu. Press F11 to return to the Database Window.

Deleting Records In Access

To delete a record in an Access table, highlight the record row intended for deletion, and either press the *Del* key or select EDIT/DELETE. Access will ask you to confirm your action. Click OK to proceed with deletion of the record.

Database Case 5

Highton University

Problem:	Develop an admissions database
Management Skills:	Coordinate Decide
Access Skills:	Queries Finding Records Sorting and Grouping Reporting

Data Table: HIGHTON

The Graduate School of Business (GSB) at Highton University began operations the mid-1960s. Since that time it has grown to become one of the most competitive and well respected schools of management in the United States. Highton has particularly strong faculty in the Marketing and Information Systems areas.

Drew Seddon has been Dean of Admissions at the GSB for over thirty years and he feels it is time for a change. The business of running the admissions program has become so politicized, argumentative, and chaotic that he is seriously considering resignation. Before he resigns he has promised to himself and his boss Tom Champy, the President of Highton, that he would build a new admissions system. He could use your help.

The Office of Admissions gathers applications and supporting documentation from a number of sources. Each student submits a completed application form and an essay. Faculty who have worked with the students at their undergraduate institution submit references. The undergraduate university also submits transcript verification forms. In addition, the College Board submits data on GMAT scores. Miscellaneous information on student ethnicity, age and sex is also collected to meet various legal requirements and reporting standards.

The Alumni group has made it very clear that they want admissions to treat the children (though not other more distant relatives) of alumni with special care.

All of this information is collected by the Admissions Office staff and summarized on an Admissions Sheet. The Admissions Sheet is the first piece of paper in the bulging folders kept on each student. The essays and outside faculty references are graded on a 1-5 scale (with 5 being the highest score) by faculty committees.

This procedure was developed in the mid 1970's when the school had 600 or so applicants for 200 positions. Today the school has 5,000 applicants for about 400 positions. Although the Admissions Office (AO) staff has doubled in size (to 8 full time people), the workload is still heavy, and the process is breaking down as older, experienced personnel retire.

The procedure for processing the Admissions Sheet is one source of the problems. Because the data arrive from different sources at different times, the AO clerks have to go back to each student folder many times to enter and update the information.

Students, parents, faculty, institutions, and alumni call in frequently to see if certain information has been received. Sometimes clerks pull the sheet to answer queries and then the sheet is lost. New sheets have to be coded up in this case.

A number of data quality problems have emerged. For instance, more and more students are claiming that their GMAT scores are wrongly recorded. Changes from the University Board are submitted with growing frequency, but AO staff are often so pressed for time that sometimes these record changes are not made. Other data quality problems involve the undergraduate grade point average, address errors, and majors. Students move, change majors, and their GPAs change. The current system should be able to keep track of these changes.

The existing system does not appear to support the group decision making process which is at the very heart of admissions. A small group of ten faculty and three Admissions Officers make all the decisions on the Admissions Committee. Each member of the Admissions Committee takes a different cut at the data, wants the data organized differently, and feels frustrated when this is impossible. One result is that members of the committee spend too much time arguing over the decision criteria and weightings, and too little time searching for appropriate candidates.

For instance, some faculty are interested only in GPA and GMAT scores and want to see the applicant pool sorted first by GPA, and then by GMAT. Other faculty are more oriented towards GPA and references and do not even want to see the GMAT.

Administrators want to make sure that within accepted academic criteria, the alumni's children get a special hearing. There is little opportunity to find those candidates who do well on all the criteria.

The manual resorting of lists and list compilation is a very tedious process characterized by long delays, mistakes, and glitches.

All of these problems have produced a political dimension: no one is happy with the existing system and everyone blames the Admissions Office. As Dean of Admissions, Drew Seddon is now under attack from many sides. The AO staff are also unsatisfied, even a bit surly at times, because of the long hours they must spend each Spring compiling lists, up-dating files, and meeting last minute deadlines.

Rather than rely on the School’s Computing Center (which is already extended beyond capacity), Drew has decided to build an PC-based admissions database.

A sample of the admissions database (HIGHTON) has been created for you on SOLVEIT.MDB as an Access table object. Create a new Access database and import the HIGHTON table now.

Tasks: There are six tasks in this case:

1. There are two typical faculty requests which the system must provide: a list of applicants sorted by GPA and GMAT scores, and a different list for other professors who want the students listed by GPA and references. Only include the essential fields in each report. For example, for the report sorted by GPA and GMAT scores, the professors only really need the student's name, school and scores. Create queries for the following:

   (a) A listing of all the applicants for professors on the admissions committee who need to see the applicants listed by decreasing GPA scores. Students with the same GPA should be sorted by decreasing GMAT scores.

   (b) A second listing showing the applicants sorted by decreasing GPA. Students with the same GPA should be sorted by decreasing reference scores. Now Print out the results of both listings.

2. The AO clerks are absolutely insistent on having rapid access to students' complete records on the basis of Last Name only. That is, they want to type the last name of the student into a PC and have the complete record pop up on the screen.

   Because so many people call in checking on student applicant files, this capability would save an enormous amount of time. Using the HIGHTON table, identify the Access sequence of commands that would enable the AO staff to do this.

3. Seddon would like a report or listing of just the applicants who are children of alumni. He will use this list as a crib sheet in committee deliberations. Whenever a candidate is settled on, he will check his list of alumni applicants to see if that person is on the list. If not, he will suggest an equally well qualified alumni applicant. Thus the alumni applicants should be sorted by decreasing GPA. Students with the same GPA should be sorted by decreasing GMAT scores.

   It is advisable given that 5,000 applicants will be in the final database to make two lists here. One list is alphabetical permitting quick look ups; the second list would arrange students by GPA and GMAT. (a) Create a query to display this information, and (b) print out the results at report level.

4. A small group of professors are concerned that the business school is loading up with Science majors. They would like to ensure that students from Liberal Arts and Social Science backgrounds are considered as well. The non-science students tend to have better GPA scores than GMAT scores (because of an alleged quantitative bias in the GMAT test) and they tend to have better recommendations.

   This group of faculty would like a report which plays to the strengths of the non-science majors. Create a report which lists applicants by college major, showing name, GPA, and reference scores. Students of the same major should be sorted by GPA.

*5. There is an on-going dispute amongst Admissions Committee members about the comparative strengths of the Social Science (Business and Psychology), Liberal Arts (English and History), and Science majors (Math, Physics and Engineering). Filter out these three groups and calculate their average scores on all quantitative variables. Then compare the groups in a one paragraph statement.

*6. There are number of ways this database system could be improved to better meet the needs of Highton. List the ways the system could be improved. Pick one of these improvements and implement it in the database.

Time Estimates (excluding tasks marked with *):

Expert: 1 hour
Intermediate: 1.5 hours
Novice: 3.5 hours

Tutorial For Database Case 5 Using Access 97

Indexing in Access

If you often search a table, or frequently need to sort records by a certain field, you can speed up these operations by assigning indexes to your fields. Access uses indexes in a table in the same way as you would use an index in a book: to find data, it looks up the location of the data in the index.

In Access, you create an index on a single field by setting the Index property. The table below lists the possible settings for Indexed properties.

Index property setting	Definition
No	Do not create an index for this field (the default)
Yes (duplicates OK)	Create an index for this field
Yes (no duplicates)	Create a unique index for this field

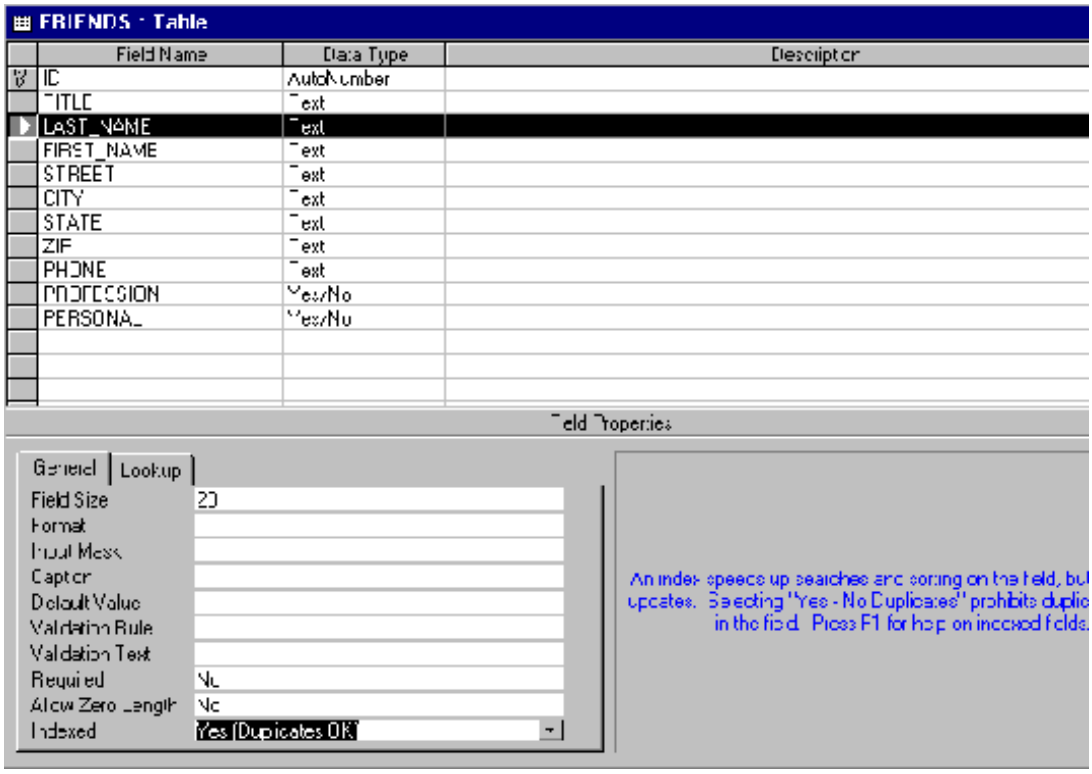
If you create a unique index, Access will not allow a value that already exists in that field, to be entered in the same field for another record. Primary key fields (refer Tutorial in Chapter 4) are automatically indexed by Access, to help speed up the execution of queries and other operations. Up to 32 indexes can be assigned to a single table or query; each index can contain up to 10 fields. Fields with Memo or Counter data types cannot be indexed.

You want an index only if it speeds up the execution of queries, searching or sorting. A Last Name field is a good candidate for an index since the values stored in it vary greatly, and it is often used to find specific records.

1. Load the practice database FRIENDS.MDB. From the FRIENDS Database Window, click on the Table object, and double click on the FRIENDS table. Switch to table design view by clicking the *Design View* toolbar button, or selecting VIEW/TABLE DESIGN from the menu.

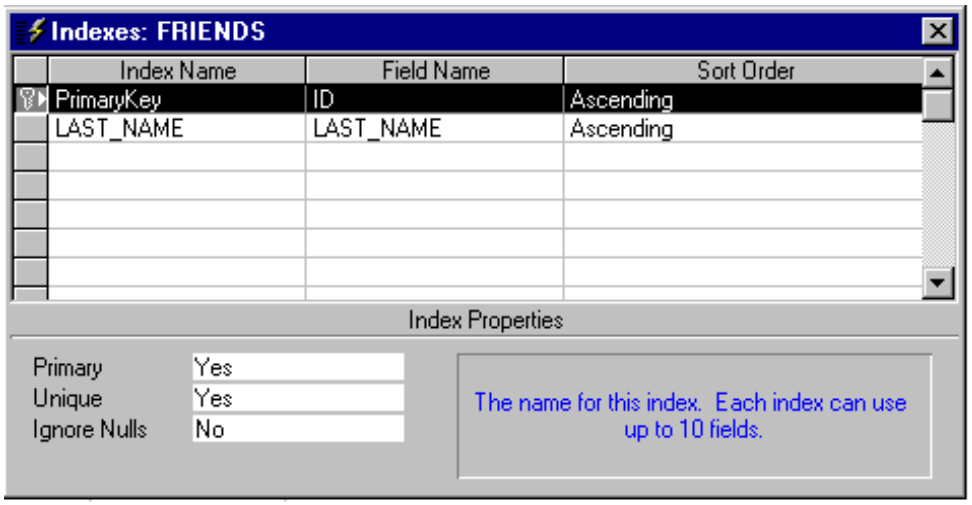
2. In the FieldName column, click on the LastName field, and then click on the Indexed bar in the Field Properties box. Select option *Yes (Duplicates OK)* to set the index for the LastName field. Click on the Save toolbar button or select FILE/SAVE TABLE from the menu. Your table design window should look similar to to Figure 5-43.

Figure 5-57



3. To view and/or edit existing indexes, open the Indexes window by clicking on the Indexes toolbar button or selecting VIEW/INDEXES from the menu (see Figure 5.58).

Figure 5-58



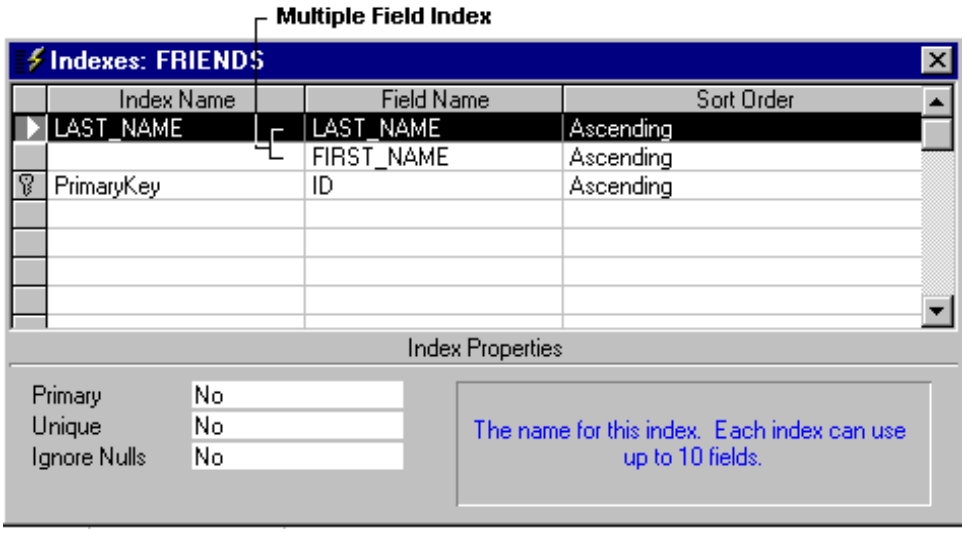
4. Change indexes or index properties as needed. To delete an index, highlight the intended row in the Indexes window and press the *Delete* key. This removes only the index, not the field itself. For Help on index properties, press F1 from the indexes window.
5. Close the Indexes window by clicking on the ☒ box in the top right hand corner.

Creating Multiple-Field Indexes

If you often search or sort by two or more specific fields at the same time, you may need to create a multiple field index. For instance, if you often set criteria in the same query for LastName and FirstName fields, it makes sense to index on both fields. When you sort a table by a multiple field index, Access sorts firstly by the first field listed in the Indexes window. If there are records with duplicate values in this field, Access then sorts by the second field listed.

Multiple field indexes are created in the Indexes Window by including a row for each field in the index, but including the index name in the first row only (see Figure 5-59). Access treats all rows as part of the same index until it reaches a different Index Name. To insert a new row between existing indexes, click the row *below* the location of the row to be inserted. Then press the *Insert* key.

Figure 5-59



Save your changes, and press F11 to return to the Database Window.

Forward and Backward Searching in Access

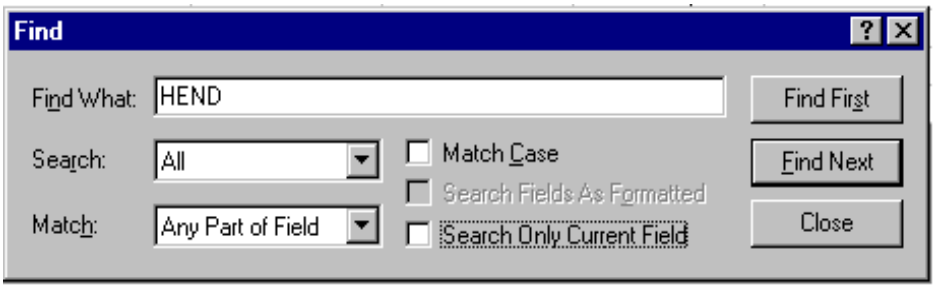
When you want to find a specific record or certain values within fields, you can use the *Find* command to go directly to a record. Find can be used with most Access objects. To use Find, select EDIT/FIND from the menu. This displays the Find dialog box (see Figure 5-60).

Type the wanted text into the Find What bar. For added flexibility, *Find* enables you to enter word stems, and wildcard symbols. Similar to DOS, a question marks (?) stands for any single character in the same position as the question mark. The asterisk (*) stands for any number

of characters in the same position as the asterisk. The hash (#) stands for a single numeric digit in the same position as the hash.

Choose between seaching across fields or a single field, and whether to search backwards, forwards, or the whole file. Practice using *Find* with your FRIENDS table.

Figure 5-60



Also explore the Access *Replace* command (select EDIT/REPLACE from the menu) which enables you to search and replace text within a field.

Tutorial For Database Case 5 Using Access 2.0

Indexing in Access

If you often search a table, or frequently need to sort records by a certain field, you can speed up these operations by assigning indexes to your fields. Access uses indexes in a table in the same way as you would use an index in a book: to find data, it looks up the location of the data in the index.

In Access, you create an index on a single field by setting the Index property. The table below lists the possible settings for Indexed properties.

Index property setting	Definition
No	Do not create an index for this field (the default)
Yes (duplicates OK)	Create an index for this field
Yes (no duplicates)	Create a unique index for this field

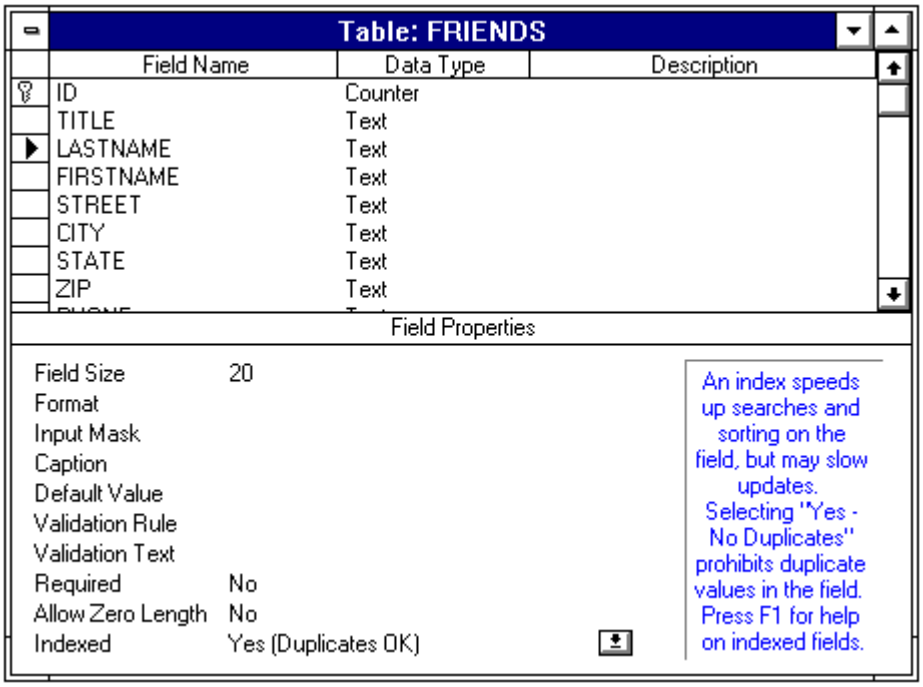
If you create a unique index, Access will not allow a value that already exists in that field, to be entered in the same field for another record. Primary key fields (refer Tutorial in Chapter 4) are automatically indexed by Access, to help speed up the execution of queries and other operations. Up to 32 indexes can be assigned to a single table or query; each index can contain up to 10 fields. Fields with Memo or Counter data types cannot be indexed.

You want an index only if it speeds up the execution of queries, searching or sorting. A Last Name field is a good candidate for an index since the values stored in it vary greatly, and it is often used to find specific records.

1. Load the practice database FRIENDS.MDB. From the FRIENDS Database Window, click on the Table object, and double click on the FRIENDS table. Switch to table design view by clicking the *Design View* toolbar button, or selecting VIEW/TABLE DESIGN from the menu.

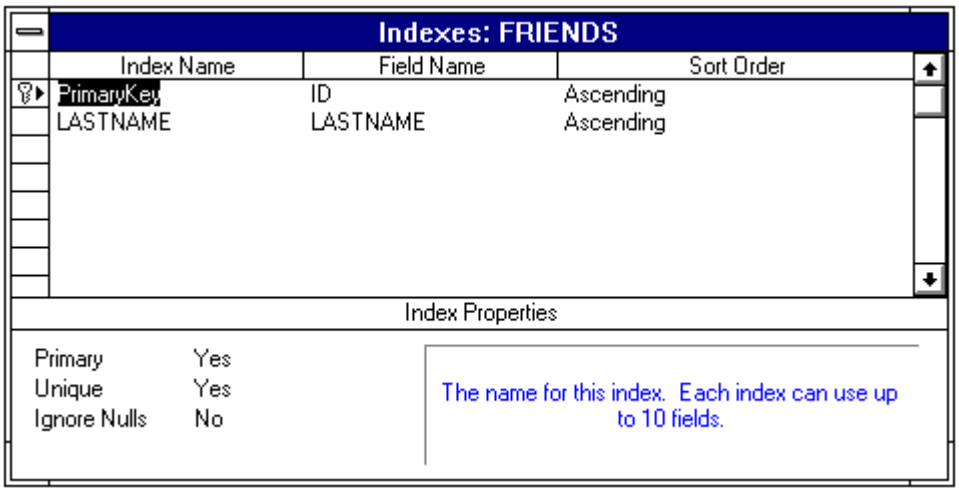
2. In the FieldName column, click on the LastName field, and then click on the Indexed bar in the Field Properties box. Select option *Yes (Duplicates OK)* to set the index for the LastName field. Click on the Save toolbar button or select FILE/SAVE TABLE from the menu. Your table design window should look similar to to Figure 5-61.

Figure 5-61



3. To view and/or edit existing indexes, open the Indexes window by clicking on the Indexes toolbar button or selecting VIEW/INDEXES from the menu (see Figure 5-62).

Figure 5-62



4. Change indexes or index properties as needed. To delete an index, highlight the intended row in the Indexes window and press the *Del* key. This removes only the index, not the field itself. For Help on index properties, press F1 from the indexes window.

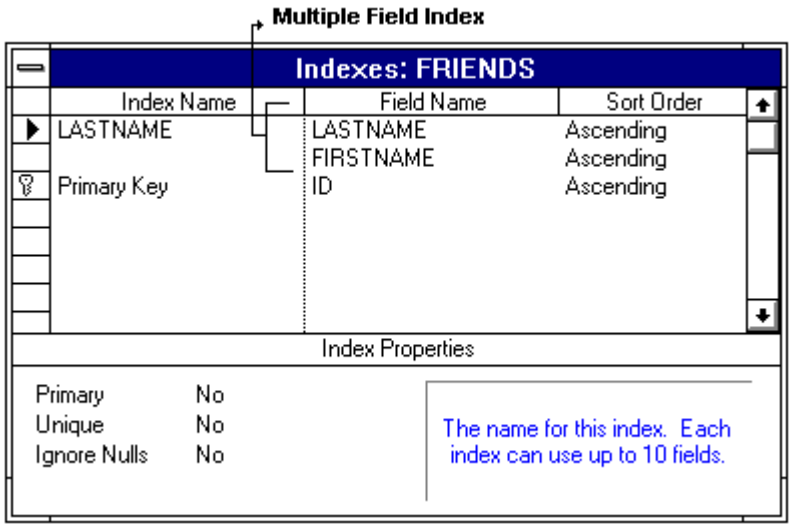
5. Close the Indexes window by double clicking on the box in the top left hand corner.

Creating Multiple-Field Indexes

If you often search or sort by two or more specific fields at the same time, you may need to create a multiple field index. For instance, if you often set criteria in the same query for LastName and FirstName fields, it makes sense to index on both fields. When you sort a table by a multiple field index, Access sorts firstly by the first field listed in the Indexes window. If there are records with duplicate values in this field, Access then sorts by the second field listed.

Multiple field indexes are created in the Indexes Window by including a row for each field in the index, but including the index name in the first row only (see Figure 5-63). Access treats all rows as part of the same index until it reaches a different Index Name. To insert a new row between existing indexes, click the row *above* the location of the row to be inserted. Then select EDIT/INSERT ROW from the menu.

Figure 5-63



Save your changes, and press F11 to return to the Database Window.

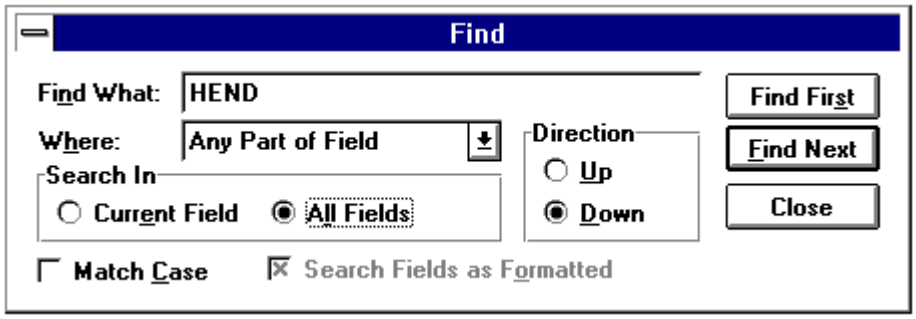
Forward and Backward Searching in Access

When you want to find a specific record or certain values within fields, you can use the *Find* command to go directly to a record. Find can be used with most Access objects. To use Find, select EDIT/FIND from the menu. This displays the Find dialog box (see Figure 5-64).

Type the wanted text into the Find What bar. For added flexibility, *Find* enables you to enter word stems, and wildcard symbols. Similar to DOS, a question marks (?) stands for any single character in the same position as the question mark. The asterisk (*) stands for any number of characters in the same position as the asterisk. The hash (#) stands for a single numeric digit in the same position as the hash.

Choose between seaching across fields or a single field, and whether to search backwards or forwards. Practice using *Find* with your FRIENDS table.

Figure 5-64



Also explore the Access *Replace* command (select EDIT/REPLACE from the menu) which enables you to search and replace text within a field.

Database Case 6

Mak Audio 1

Problem:	Develop a mailing list database
Management skills:	Organize Control
Access skills:	Database Design Queries Reports Linking Data Tables
Data Tables:	MAK_A MAK_B

Mak Audio is a medium-sized company based in Nashville, Tennessee. The business was established in the early 1970s by Allan and Rick Mak, and grew out of a hobby the two brothers shared in music and electronic circuitry. The company has two distinct business divisions. The Mak Design side of the business designs, manufactures and retails a range of audio equipment at the high end of the market. This includes custom built sound systems, the famous Mak valve amplifier which is eagerly sought by audiophile buffs around the world, and high performance ribbon speakers. This side of the business is high margin and very successful, and is managed by Allan Mak.

Mak Design's main customers are audio retailers, and music production houses such as EMI, Warner Music, and Polygram Records. Every six months, Mak Design distributes a product catalogue to its 3,500 customers.

Rick Mak manages Anabasis, the other division in the company. Anabasis is a lucrative mail order business which locates and purchases rare and hard-to-find sound recordings through contact agents around the world. Many of these items are sourced from the deceased estates of musical performers. Music areas include classical, opera, jazz, rock, country and western, folk, soul, reggae, and ethnic (Asian and African). The business currently carries around 15,000 titles in this area, which are held in audio CD, vinyl record, tape or video format. These are sold to audio retailers and music production houses throughout the world. Currently Anabasis has around 12,000 customers on its mailing list, and like Mak Design, distributes a catalogue of new and current offerings on a biannual basis.

The existing Anabasis mailing list presents a difficulty for Rick Mak. While the marketing side of Mak Design has been handled by a PC-based database package for some years, Mak has been unsure how to go about integrating the Anabasis mailing list into this system, although he knows this could make good business sense. Mak suspects that at least 25% of the customers on his mailing list are also customers who regularly purchase from Mak Design.

The Anabasis mailing list has been steadily growing since the mid 1970's, when this side of the business was formed. Originally the mailing list was kept on a series of cards which recorded customer mailing and music interest details. This method soon became unwieldy as the list grew. In the late 1970's, Anabasis outsourced the data management of its list to Good Code Computer Services. Good Code provides data entry and maintenance services, and produces mailing label printouts for Anabasis whenever a catalogue mailout is required.

Mak has a number of problems with the bureau method. He has recently received an invoice from Good Code requesting its usual quarterly payment of $2,500 for maintenance of his mailing list. Mak has been concerned by the escalating costs of using Good Code for some time and is no longer sure their services are worthwhile. He also feels that the bureau method does not give him control of his data.

The current mailing list is not a database. In the 1970s, Good Code created the original list as a long sequential file organised alphabetically by customer last name, and has continued to maintain it in this format. Mak is not happy with this structure which he feels is inflexible, and cannot provide information vital to the management of his business. For example, he cannot easily obtain information about the music interests of his customers for targeted mailouts. A simple query like *"print a list of all Australian customers with an interest in Funk music"*, requires a special programming task by Good Code, a lengthy delay before Mak receives the information, and incurs an additional charge for Anabasis. More complex queries are simply impossible.

The structure of the current mailing list also makes it difficult to update existing data or to check whether a customer is already on the list. Mak knows that a large proportion of his mailing list is probably out of date. He has noticed that the number of return-to-sender envelopes Anabasis receives has rapidly increased as customers move addresses or contact names change. The present system makes it difficult to incorporate these changes. In the meantime, Anabasis is incurring a lot of unnecessary mailout associated expense. Mak has also noticed that some of these returns have been sent to the same customer several times. This means that there is a level of data duplication in the mailing list. From current indications, Mak suspects it could be as high as 20%.

Another inherent problem with the current system is that if there are 45 customers at a large music company like Warner Music, the name and address of the company is repeated 45 times! If the name of the company were to change (as is common in the music business) the new name would need re-entered 45 times!

Mak is frustrated by the deficiencies of his current mailing list. Although Anabasis is profitable, Mak suspects it could be more so if he had the right information when he wanted it, and was able to do targeted mailouts with ease. He has decided that the only way to get what he wants is to build a parallel mailing list system from the ground up. After consultation with his brother, Mak has purchased a Pentium-based microcomputer, and the same relational data base package used by Mak Design. Once the new mailing system is working satisfactorily, he plans to dispense with Good Code's services. At a later stage, he will consider integrating his system with that of Mak Design.

The existing mailing system has the right information, but it is not stored efficiently, and it is not easy to interrogate. The existing system has the customer's name, primary music interest (MusIntA), secondary music interest (MusIntB), organisation name, address, city, state, country and zip code. The music interests of customers are coded using an internally developed schema.

To make storage and updating of the mailing list more efficient, Mak wants the data stored in two tables with one containing the names of the customers, and the other the name and addressing details of the company they work for. Whenever a mailing list is required, the two tables could be joined and the results printed. Mak needs your assistance in developing this project.

The Access data tables on your *SolveIt!* diskette MAK_A and MAK_B provide the overall design and sample data for the new mailing system. These tables form the basis of this case.

Tasks: There are six tasks in this case:

1. Link the two tables MAK_A and MAK_B on the Org field. It may be helpful to print out the structure of these tables first. Then print out the results of this join.

2. Create a report for marketing purposes that lists all customers separated into alphabetical organisational groups. The fields describing reading interests and address are not needed. Each group should contain one organization with the records sorted alphabetically by last name within these groups.

   Design this report as a professional document. Include a title, group headings, and any other useful features you think of.

3. Print the report.

4. Design a custom label with four lines of information that can be used as a mailing label.

5. Anabasis has recently obtained some rare Soul and Jazz sound recordings. Print a report of all USA customers with one of two music interests (i.e. 64 or 77) in these areas, as either their primary or secondary music interest. Be careful with the Boolean logic in this problem.

*6. The current system uses codes (e.g. 64, 77) for data in the music interest fields. The user of the system and recipients of its reports must know what these codes mean. Devise a way to include the meaning for these codes into the database. You will need to make up meanings for each of the music interest codes (e.g. 64 is Jazz Music and 77 is Soul Music).

   Make sure that the method you devise does not include data redundancy. It would be poor database design to have the meanings for these codes repeated (redundantly) throughout the database. From your new table structure, produce a report of customers, their organisation and music interests which does not include codes.

Time Estimates (excluding task marked with *):

Expert: 45 minutes
Intermediate: 1.5 hours
Novice: 3 hours

Tutorial For Database Case 6 Using Access 97

In this case, you will learn two important new skills: joining table files (creating relationships), and creating labels in reports.

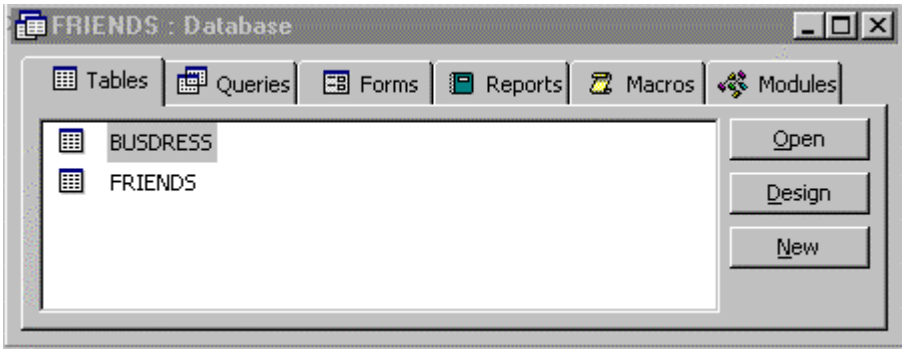
Joining Tables

In many earlier database applications, the joining of tables is a procedure that would require you to generate programming code. Access allows you to do this simply via the menu or toolbar, and to run the results of the join in a query (or a report). Access will also generate the SQL programming code for the procedure, so that you can see the effect of your actions.

Before you can join two tables, the related field must be present in both tables. While the field names do not have to be the same, for the join to work, the two fields must contain matching data within records. For this procedure, we will be using the FRIENDS database, and a second practice table BUSDRESS. The latter contains the last names and business addresses of persons in the FRIENDS database.

1. Load the practice database FRIENDS.MDB. Click on the *Table* object and then import the second practice database file BUSDRESS.DBF using FILE/GET EXTERNAL DATA and then IMPORT from the menu or by clicking on the *Import* toolbar button. Recap the tutorial in Chapter 4, if you are unsure about importing external files. BUSDRESS is a table contined within the SOLVEIT master database, so make sure that Access is slected as the format during the import procedure. FRIENDS.MDB should now have two tables, and look similar to Figure 5-65.

Figure 5-65



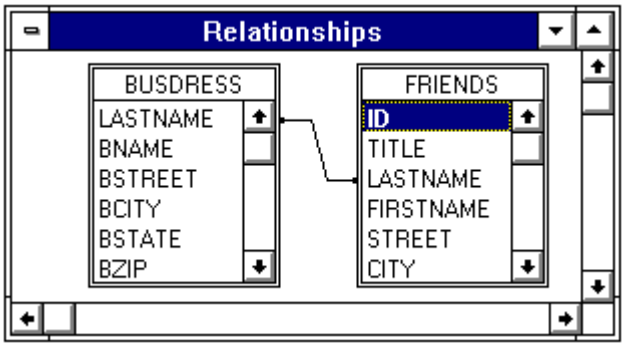
2. From the Database Window, highlight and open the FRIENDS table, and then repeat this procedure for the BUSDRESS table. Compare the two tables, noting that both LastName fields contain identical data. Remember that the field names themselves do not have to be the same. Close the tables in turn by double clicking on the ▭ box in the top left hand corner of each.

3. From the Database Window, create a relationship between the two tables. This will speed up the execution of the query we will create in a moment. Table relationships were last discussed in the tutorial for Chapter 4.

4. To create a relationship: click the *Relationships* button on the toolbar. From the Show Table dialog box, highlight the BUSDRESS table and click Add. Repeat this action for the FRIENDS table.

5. Highlight the LastName field in the BUSDRESS table and while holding down your left mouse button, drag this over to the LastName field in the FRIENDS table. Release the mouse button. A Relationships dialog box will appear. Now click on the *Create* button. Access generates a dynamic link (line) between the two known as an *equi-join*. An equi-join selects all the records from both tables that have the same value in the field selected as being common to both. Your Relationships window should look similar to Figure 5-66.

Figure 5-66



6. Save the relationship by choosing FILE/SAVE or clicking on the *Save* toolbar button.

7. Press F11 to return to the Database Window.

You are now ready to create a query to generate the effect of the joining action.

8. From the Database Window, click on the *Query* object, and click New. Now either select the *Simple Query Wizard* or the *Design View* option from the New Query dialog box. This tutorial assumes that you have selected the *Design View* option.

9. From the *Add Table*, highlight the BUSDRESS table and click Add. Repeat this action for the FRIENDS table. Click Close. Notice that Access has remembered the table relationship, and has immediately established a join between the two LastName fields.

10. Select VIEW/SQL VIEW from the menu or click on the SQL toolbar button 

to see the equivalent of this action in the Access Basic programming code:

```
SELECT
FROM BUSDRESS INNER JOIN FRIENDS ON BUSDRESS.LAST_NAME = FRIENDS.LAST_NAME;
```

Click on the design view button  to return to the Query Design window or select VIEW/DESIGN VIEW from the menu.

11. From the query window, select and drag the following fields down into the Field bar of the QBE Grid.

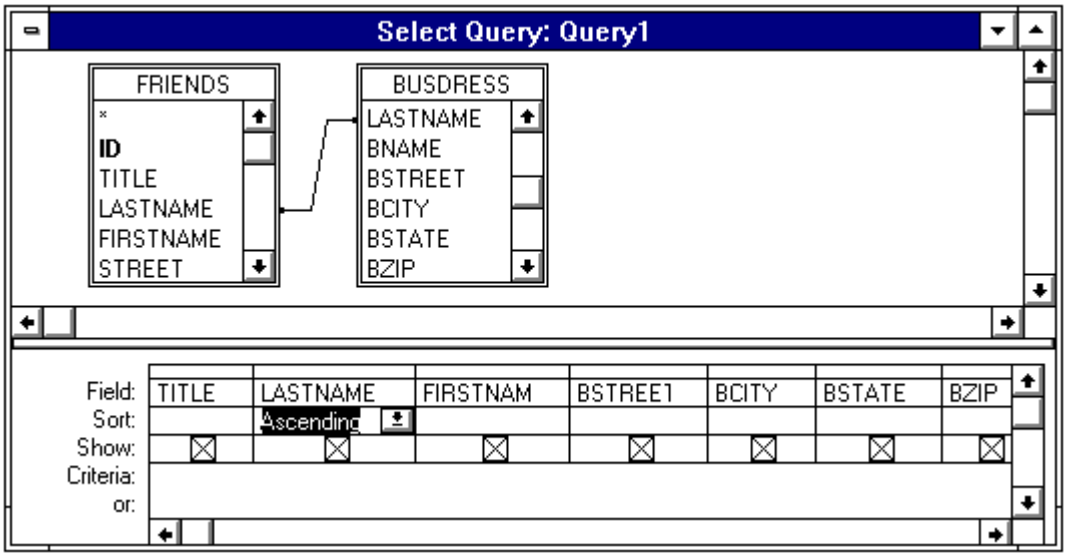
From FRIENDS	From BUSDRESS
Title	BStreet
FirstName	BCity
LastName	BState
	BZip

Lastly, click on the *Sort:* bar in the LastName field, and choose Ascending order. Your query window should now look similar to Figure 5-67.

12. Click on the SQL button again, to see the effect this action has had on the Access Basic programming code:

```
SELECT FRIENDS.TITLE, FRIENDS.LAST_NAME, FRIENDS.FIRST_NAME, BUSDRESS.BSTREET,
BUSDRESS.BCITY, BUSDRESS.BSTATE, BUSDRESS.BZIP
FROM BUSDRESS INNER JOIN FRIENDS ON BUSDRESS.LAST_NAME = FRIENDS.LAST_NAME
ORDER BY FRIENDS.LAST_NAME;
```

Figure 5-67

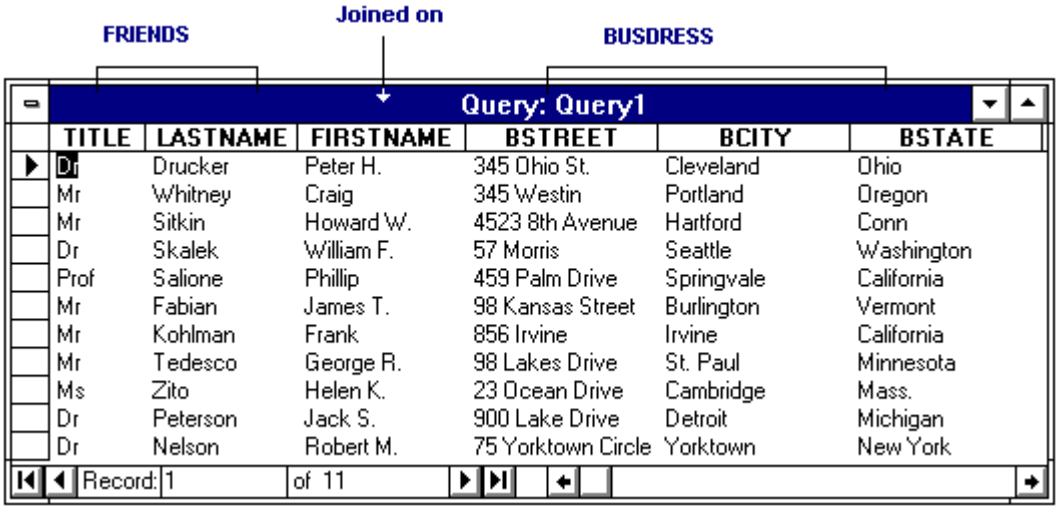


13. Click on the *Run*  query toolbar button, and view the resulting dynaset. Access has matched and merged the business address data in the BUSDRESS table with the name data in the FRIENDS table to create a composite display containing non-duplicative data from both (see Figure 5-xx).

14. Print the query if you wish to, or use as the basis for a new report.

15. Save the query (eg: *Tute6 Query1*), and press F11 to return to the Database Window.

Figure 5-68



How to Create Labels in Access

A report designed to print names and addresses on labels is a common feature of most database applications. The report feature in Access includes a powerful Mailing Label wizard, with sizes to match most common labels. Let's use the sample FRIENDS database to create a simple three line address label.

1. Load FRIENDS.MDB. Click on the Report object, and click New. From the New Reports box, click on the ▼ button to the right of the *Choose the Table or Query Where an Object's Data Comes From*, and click and highlight the *FRIENDS* table.

2. Now click and highlight the *Label Wizard* option, and then click *OK*.

3. On the first wizard screen, accept the default label type and dimensions. Click the Next> button.

4. On the second screen, select the font, weight and colour you want for the label text. Click the Next> button.

5. Choose fields for your label:

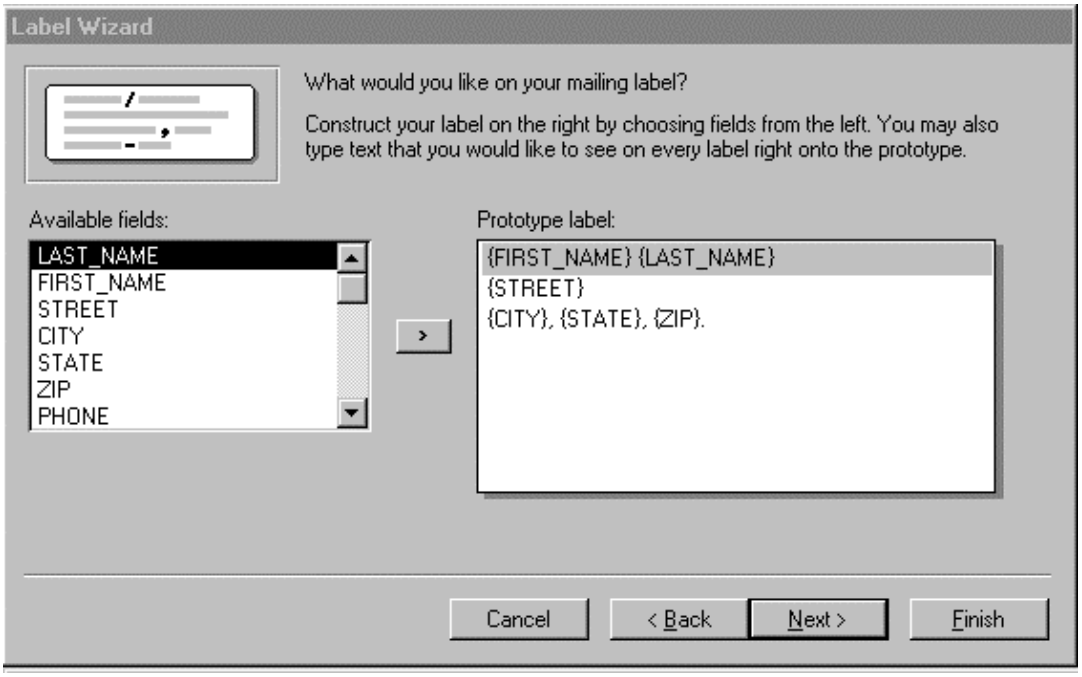
   First Line
   Select *Title*, and then click on the > button to lodge onto the label. Press Space on your keyboard.
   Repeat this action with the *FirstName* field.
   Select *LastName*, and then click >. Now press Return on your keyboard.

   Second Line
   Select *Street*, and then click >.. Now press Return.

Third Line
Select *City*, and then click . Add a comma (,) and then press Space.
Repeat this action with the *State* field.
Select *Zip*, and then click .

Your screen should now look like the one in Figure 5-69. Click the Next> button.

Figure 5-69



6. On the next report wizard screen, select the LastName field to sort by. Click the Next> button.

7. On the last wizard window, choose a name for your report (eg: *Tute6 Repo1*). Click the Finish button, and Access will generate your labels and display them in Print Preview mode.

8. Send your labels to print.

9. Press F11 to return to the Database Window, and then exit Access.

Tutorial For Database Case 6 Using Access 2.0

In this case, you will learn two important new skills: joining table files (creating relationships), and creating labels in reports.

Joining Tables

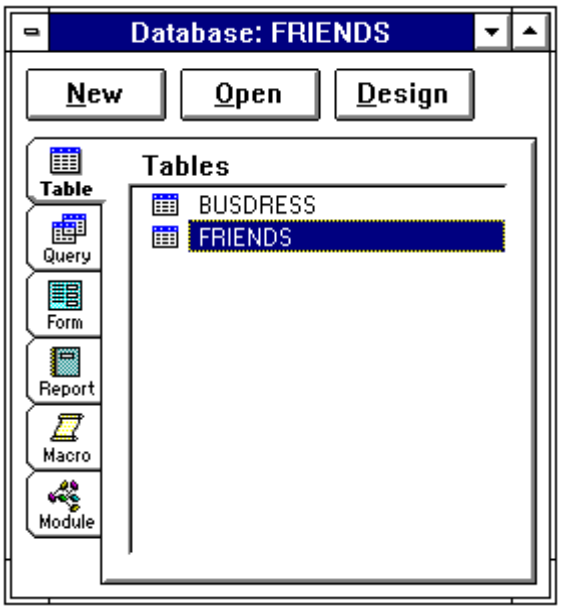
In many database applications, the joining of tables is a procedure that would require you to generate programming code. Access allows you to do this simply via the menu or toolbar, and to run the results of the join in a query (or a report). Access will also generate the SQL programming code for the procedure, so that you can see the effect of your actions.

Before you can join two tables, the related field must be present in both tables. While the field names do not have to be the same, for the join to work, the two fields must contain matching data within records. For this procedure, we will be using the FRIENDS database, and a second practice file BUSDRESS.DBF. The latter contains the last names and business addresses of persons in the FRIENDS database.

1. Load the practice database FRIENDS.MDB. Click on the *Table* object and then import the second practice database file BUSDRESS.DBF using FILE/IMPORT from the menu or by clicking on the *Import* toolbar button. Recap the tutorial in Chapter 4, if you are unsure about importing external files. BUSDRESS.DBF is a dBase file, so choose dBase III as the format during the import procedure. FRIENDS.MDB should now have two tables, and look similar to Figure 5-70.

Import 

Figure 5-70



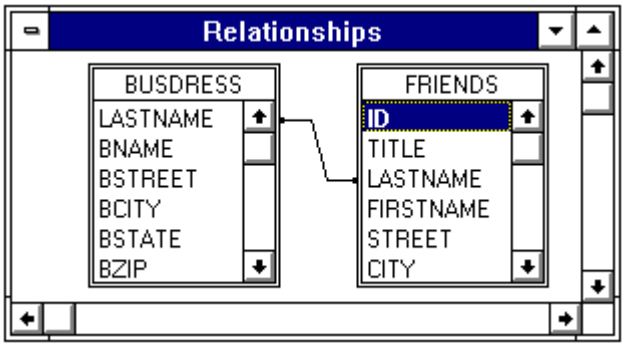
2. From the Database Window, highlight and open the FRIENDS table, and then repeat this procedure for the BUSDRESS table. Compare the two tables, noting that both LastName fields contain identical data. Remember that the field names temselves do not have to be the same. Close the tables in turn by double clicking on the box in the top left hand corner of each.

3. From the Database Window, create a relationship between the two tables. This will speed up the execution of the query we will create in a moment. Table relationships were last discussed in the tutorial for Chapter 4.

4. To create a relationship: click the *Relationships* button on the  toolbar. From the Add Table, highlight the BUSDRESS table and click Add. Repeat this action for the FRIENDS table.

5. Highlight the LastName field in the BUSDRESS table and while holding down your left mouse button, drag this over to the LastName field in the FRIENDS table. Release the mouse button. Access generates a dynamic link (line) between the two known as an *equi-join*. An equi-join selects all the records from both tables that have the same value in the field selected as being common to both. Your Relationships window should look similar to Figure 5-71.

Figure 5-71



6. Click OK, and save the relationship by choosing FILE/SAVE LAYOUT or clicking on the *Save* toolbar button.

7. Press F11 to return to the Database Window.

You are now ready to create a query to generate the effect of the joining action.

8. From the Database Window, click on the *Query* object, and click New. Now click on the New Query button.

9. From the *Add Table*, highlight the BUSDRESS table and click Add. Repeat this action for the FRIENDS table. Click Close. Notice that Access has remembered the table relationship, and has immediately established a join between the two LastName fields.

10. Select VIEW/SQL from the menu or click on the SQL toolbar button 

to see the equivalent of this action in the Access Basic programming code:

```
SELECT DISTINCTROW
FROM BUSDRESS INNER JOIN FRIENDS ON BUSDRESS.LASTNAME =
FRIENDS.LASTNAME;
```

Click on the design button  to return to the query design window.

11. From the query window, select and drag the following fields down into the Field bar of the QBE Grid.

From FRIENDS	From BUSDRESS
Title	BStreet
FirstName	BCity
LastName	BState
	BZip

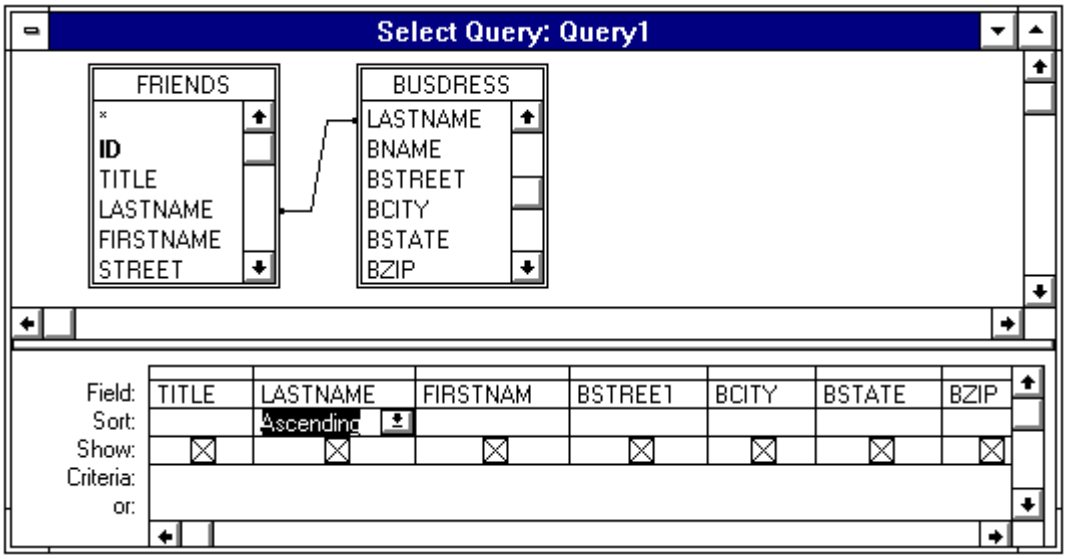
Lastly, click on the *Sort:* bar in the LastName field, and choose Ascending order. Your query window should now look similar to Figure 5-72.

12. Click on the SQL button again, to see the effect this action has had on the Access Basic programming code:

```
SELECT DISTINCTROW FRIENDS.TITLE, FRIENDS.LASTNAME, FRIENDS.FIRSTNAME,
BUSDRESS.BSTREET, BUSDRESS.BCITY, BUSDRESS.BSTATE, BUSDRESS.BZIP
```

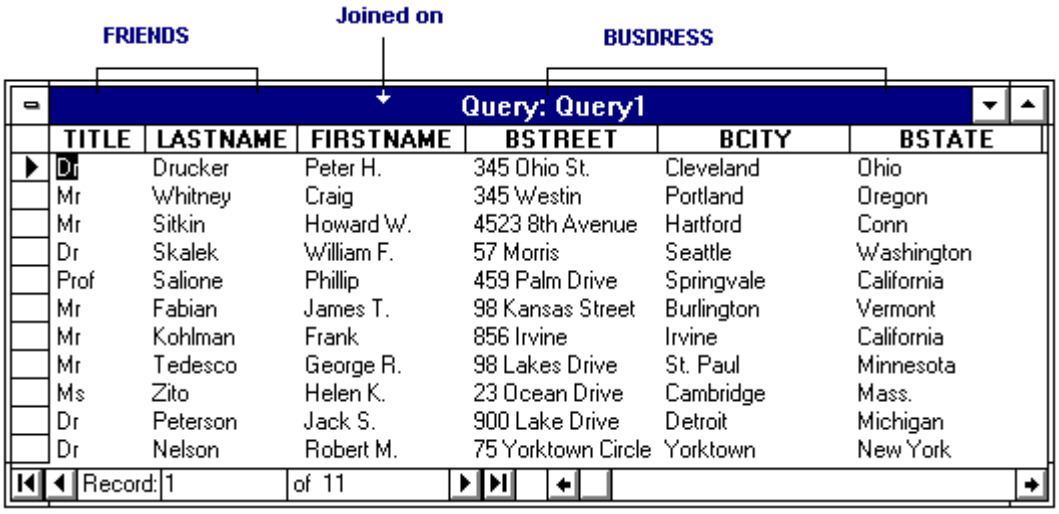
FROM BUSDRESS INNER JOIN FRIENDS ON BUSDRESS.LASTNAME =
FRIENDS.LASTNAME ORDER BY FRIENDS.LASTNAME;

Figure 5-72



13. Click on the *Run*  query toolbar button, and view the resulting dynaset. Access has matched and merged the business address data in the BUSDRESS table with the name data in the FRIENDS table to create a composite display containing non-duplicative data from both (see Figure 5-73).

Figure 5-73



14. Print the query if you wish to, or use as the basis for a new report.

15. Save the query (eg: *Tute6 Query1*), and press F11 to return to the Database Window.

How to Create Labels in Access

A report designed to print names and addresses on labels is a common feature of most database applications. The report feature in Access includes a powerful Mailing Label wizard, with sizes to match most common labels. Let's use the sample FRIENDS database to create a simple three line address label.

1. Load FRIENDS.MDB. Click on the Report object, and click New. From the New Reports box, click on the ▼ button to the right of the *Select a Table/Query* Bar, and click and highlight the *FRIENDS* table.

2. Click on the *Report Wizards* button. Click on *Mailing Label*, and then click *OK*.

3. Choose fields for your label. Use the buttons below the field list to add spaces and punctuation to your label and to start each new line:

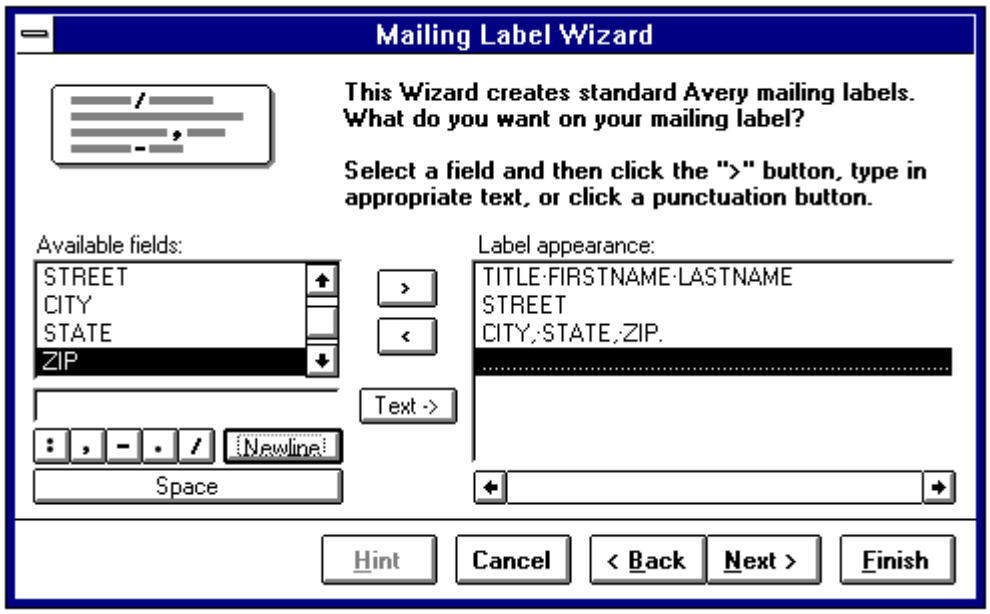
First Line
Select *Title*, and then click on the > button to lodge onto the report list. Click Space
Repeat this action with the *FirstName* field.
Select *LastName*, and then click >. Click Newline.

Second Line
Select *Street*, and then click >.. Click Newline.

Third Line
Select *City*, and then click >. Click comma (,) and then click Space.
Repeat this action with the *State* field.
Select *Zip*, and then click >.

Your screen should now look like the one in Figure 5-74. Click the Next> button.

Figure 5-74



4. On the next report wizard screen, select the LastName field to sort by. Click the Next> button.

5. On the third wizard screen, accept the default label type and dimensions. Click the Next> button.

6. On the final screen, select the font, weight and colour you want for the label text. Click the Finish button, and Access will generate your labels and display them in Print Preview mode.

7. Send your labels to print.

8. Save your new report (eg: *Tutel6 Repo1*). Press F11 to return to the Database Window, and exit Access.

Database Case 7

Caulfield Contractors

Problem:	Develop a payroll system
Management Skill:	Control
Access skills:	Select Queries (calculated fields) Append Queries Reports (wizards)

Data Tables: CONTRACT
1_MONTH
2_MONTH
3_MONTH

After twelve years working for a general contractor, Martin Caulfield struck out on his own in 1991 to start his own road construction firm, Caulfield Contractors. For a start-up firm, Caulfield had done extremely well: in 1998 they showed gross revenues of $15 million and a net profit of $950,000.

Before starting his own company, Martin had gained a sound understanding of the road building and public sector market working as an estimator for a general contractor in Boston, Massachusetts. He built up a network of contacts among local government officials, and learned the procedures and pitfalls of responding to government contract solicitations. Fortunately, the local economy grew very rapidly as major corporations, government agencies and related housing created a boom market for road construction.

The key to Martin's success has been in keeping permanent employee numbers low, relying on sub-contractors, and keeping his capital costs very low by renting or leasing heavy equipment. Automated tools have also played an important role.

About 80% of Caulfield's business is in road construction. Caulfield responds to government agency solicitations for bids on construction projects. Caulfield, and other firms, prepare competitive proposals and tender bids. A large part of the bid preparation process, especially the design, cost estimation and technical specification, is aided using a proprietary decision support system (DSS) written for the road construction industry. The DSS runs on a Pentium PC in Caulfield's office.

Payroll and personnel--along with other office administrative tasks--are somewhat more chaotic. There are only 5 full time employees (Martin, two engineers, and two office staff). However, about 40 sub-contractors work for the firm during a typical monthly pay-roll period. More than 400 individuals will work for the firm in a year.

Currently, one of the office staff spends 80% of her time keeping track of the firm’s part time and full time employees. Originally Martin had considered outsourcing his payroll, but this

proved far too expensive an option. Martin is convinced the answer lies in building a simple in-house personnel system to handle employee wages, and related payroll information.

Last year, the situation became serious when the Internal Revenue Service (IRS) audited the firm's books. The IRS wondered why Caulfield had more than 400 employees but showed very low tax withholdings. Martin explained that most of the employees were short term, part time workers and/or sub-contractors. In recent times the IRS has taken a dim view of employees being declared as "contract workers" because federal tax is not withheld for contract workers and it is difficult for the IRS to collect from them later.

While Caulfield's contract workers appeared to be legitimate, examination of the records found a number of errors in withholding statements for regular employees who were part time workers. For 35 part time workers, no withholding was deducted or submitted to the IRS because of clerical errors. The IRS has given Martin one year to straighten out the records.

Martin has decided that a PC-based database system would be ideal for a small payroll system. He would like you to design such a system for a sample quarter. Employees are paid monthly.

A sample of the employee data file showing the name, address, ID number, and age can be seen by loading the CONTRACT data table from your *SolveIt!* diskette. For each monthly payroll period a monthly file is created. Three hand calculated files for the first quarter have been built for you in Access: 1_MONTH, 2_MONTH, and 3_MONTH.

Tasks There are five tasks in this case:

1. Combine the three individual monthly reports (ie: 1_MONTH, 2_MONTH, 3_MONTH) into a single data table representing the first quarter.

   Hint: Use Append and Action queries to do this. You should end up with 26 records in a single, new data table.

2. (a) Develop a query which will calculate the gross pay, federal and state withholdings, FICA, and resulting net pay for all employees in the first quarter.

   Set federal withholdings at 17.4% of gross, FICA at 2.7% of gross, state withholding at 8% of gross, and calculate net pay as gross pay minus all deductions.

   (b) Now use this query to create a report which shows the gross pay, all deductions, and net pay for all employees for the quarter.

3. Develop a payroll report which shows just the firm's grand totals for the quarter for gross pay, all the deductions and net pay.

4. Develop a report which shows for each employee the earnings and deductions for the quarter, giving subtotals by employee. At the end, the report should also show firm totals.

*5. Produce a report for the IRS for the first quarter which shows the employees' name, address, all deductions and net pay.

You will need the data table CONTRACT on the *Solve it!* diskette to do this task. This table contains the names and addresses of Caulfield's employees.

Time Estimates (excluding task marked with *):

Expert: 1.5 hour
Intermediate: 2.5 hours
Novice: 4+ hours

Tutorial For Database Case 7 Using Access 97

Copying and Pasting Table Structures

From the Database window, you can create an empty copy of an existing table (ie: field structure only), as a first step in merging records from a number of different tables to a single table.

Let's practice this by creating an empty copy of the FRIENDS table:

1. Load FRIENDS.MDB. From the Database window, click to highlight the FRIENDS table.
2. Select EDIT/COPY from the menu.
3. Select EDIT/PASTE from the menu.
4. From the *Paste Tables As* window, give the table a new name (eg: Tute7 CopyTable)
5. Under the Paste Options section, click on *Structure Only* and then click OK. Access creates an empty table with an identical field structure to the FRIENDS table.

Action Queries

When you create a new query via the Simple Query Wizard or Design View options, Access generates a Select query by default. *Select* queries simply retrieve and display data from tables according to a specified user criteria. We have used Select queries for *Solve it!* database cases 1 to 6. In contrast, *Action* queries which are constructed within the Select query window, actually alter data in tables. Action queries can be used to add, delete, or change data, and to create new tables from existing records. For example, a Delete action query removes obsolete records from a table. Access uses four types of action queries. A brief description of each type and the graphical objects that identify them within the database envelope, is shown in Figure 5-75. In this tutorial, we will concentrate on *Append* action queries. Other Action query types will be covered in later tutorials.

Figure 5-75

Types of Queries

Access query types include:

Select queries select a group of records from one or more tables.

Action Queries

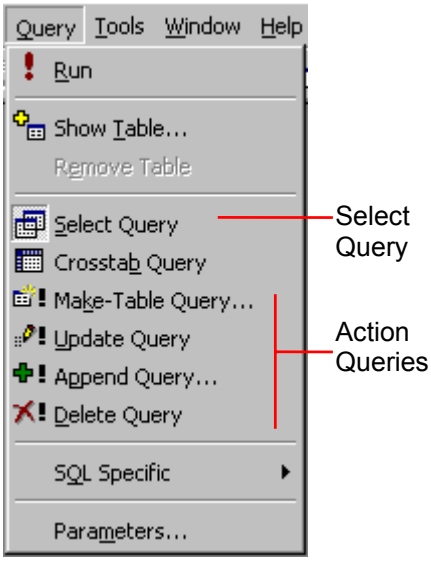
The following action queries change the data in your tables:

Make-table queries create a new table from all or parts of other tables.

Delete queries delete records from one or more tables.

Append queries add a group of records to a table.

Update queries make changes to data in a group of records.



Append Queries

Append queries copy all or some records from one table to the bottom of another existing table. This is especially useful if you use separate tables to manage certain data (eg: you keep payroll data on a month by month basis, and need to merge it every financial quarter). Append queries are also handy for storing historical data. For instance in an Orders table, completed orders could be separated from active, uncompleted orders to prevent the Orders table from becoming too large, and to provide a useful backup for future reference.

When data is appended, the tables involved in the query do not need to have the same structure, but the data types of the appended fields must match. (The exception is Counter data types, which may be appended to Long Integer data types). When records are appended to a table, the records in the original table remain intact. Access does not delete the original records.

Note:	if two tables involved in an Append query have identical field structures, the * in the table field list can be used instead of the individual field names in the Field: bar of the QBE Grid

Let's use the FRIENDS table and the empty copy of the FRIENDS table (eg: TUTE7 COPYTABLE) we created earlier in this tutorial to generate an Append action query.

1. From the Database window, create a new query (using the Design View option) based on the FRIENDS table.

2. From the query design window, select QUERY/APPEND from the menu or click on the Append query toolbar button. The results of this action will turn the Select query into an Append query, and cause Access to add an *Append to:* line to the QBE Grid.

3. In the *Append* dialog window, select and set the *Append to Table Name* box to the TUTE7 COPYTABLE we created earlier, and click OK. This means that we are going to append records from the FRIENDS table to the TUTE7 COPYTABLE.

4. From the Append query design window, select and drag the ✳ symbol in the FRIENDS field list down onto the Field: bar of the QBE Grid. This tells Access two things: that all FRIENDS fields should be included in the query, and that the field structure of TUTE7 COPYTABLE is identical to FRIENDS. Access immediately adds notation to this effect in the *Append to:* cell.

5. Test the Append query before committing to its execution. One of the things to note about any Action query is that it changes table data in some way. For this reason, it is good practice to trial the query before running it. Trialling also allows you to check for errors in the query setup.

To trial your Append query, click on the  button. The resulting dynaset should display all the records in the FRIENDS table. Return to query design view, and run the query. Access should append all FRIENDS table records to TUTE7 COPYTABLE.

6. Save your query (eg: TUTE7 APPEND QUERY), and return to the Database window. Open the TUTE7 COPYTABLE and check the contents.

Hint:	You will need to use both the Copy/Paste table structure sequence and Append queries to complete Task 2 of Case 7.

Tutorial For Database Case 7 Using Access 2.0

Copying and Pasting Table Structures

From the Database window, you can create an empty copy of an existing table (ie: field structure only), as a first step in merging records from a number of different tables to a single table.

Let's practice this by creating an empty copy of the FRIENDS table:

1. Load FRIENDS.MDB. From the Database window, click to highlight the FRIENDS table.
2. Select EDIT/COPY from the menu.
3. Select EDIT/PASTE from the menu.
4. From the *Paste Tables As* window, give the table a new name (eg: TUTE7 COPYTABLE)
5. Under the Paste Option section, click on *Structure Only* and then click OK. Access creates an empty table with an identical field structure to the FRIENDS table.

Action Queries

When you create a new query, Access generates a Select query by default (unless you click on the Query Wizards button). *Select* queries simply retrieve and display data from tables according to a specified user criteria. We have used Select queries for *Solve it!* database cases 1 to 6. In contrast, *Action* queries which are constructed within the Select query window, actually alter data in tables. Action queries can be used to add, delete, or change data, and to create new tables from existing records. For example, a Delete action query removes obsolete records from a table. Access uses four types of action queries. A brief description of each type and the graphical objects that identify them within the database envelope, is shown in Figure 5-76. In this tutorial, we will concentrate on *Append* action queries. Other action query types will be covered in subsequent tutorials.

Figure 5-76

Types of Queries

Access query types include:

Select queries select a group of records from one or more tables.

Action Queries

The following action queries change the data in your tables:

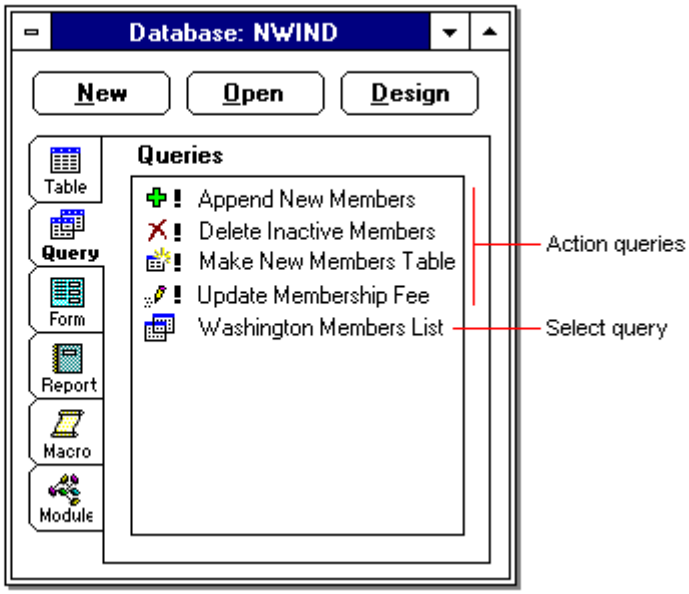
Make-table queries create a new table from all or parts of other tables.

Delete queries delete records from one or more tables.

Append queries add a group of records to a table.

Update queries make changes to data in a group of records.

Source: Access Cue Cards



Append Queries

Append queries copy all or some records from one table to the bottom of another existing table. This is especially useful if you use separate tables to manage certain data (eg: you keep payroll data on a month by month basis, and need to merge it every financial quarter). Append queries are also handy for storing historical data. For instance in an Orders table, completed orders could be separated from active, uncompleted orders to prevent the Orders table from becoming too large, and to provide a useful backup for future reference.

When data is appended, the tables involved in the query do not need to have the same structure, but the data types of the appended fields must match. (The exception is Counter data types, which may be appended to Long Integer data types). When records are appended to a table, the records in the original table remain intact. Access does not delete the original records.

Note:	If two tables involved in an Append query have identical field structures, the * in the table field list can be used instead of the individual field names in the Field: bar of the QBE Grid

Let's use the FRIENDS table and the empty copy of the FRIENDS table (eg: TUTE7 COPYTABLE) we created earlier in this tutorial to generate an Append action query.

1. From the Database window, create a new query based on the FRIENDS table.

2. From the query design window, select QUERY/APPEND from the menu or click on the Append query toolbar button. This action turns the Select query into an Append query, and causes Access to add an *Append to:* line to the QBE Grid.

3. In the *Query Properties* window, select and set the *Append to Table Name* box to the TUTE7 COPYTABLE we created earlier, and click OK. This means that we are going to append records from the FRIENDS table to the TUTE7 COPYTABLE.

4. From the Append query design window, select and drag the ✳ symbol in the FRIENDS field list down onto the Field: bar of the QBE Grid. This tells Access two things: that all FRIENDS fields should be included in the query, and that the field structure of TUTE7 COPYTABLE is identical to FRIENDS. Access immediately adds notation to this effect in the *Append to:* cell.

5. Test the Append query before committing to its execution. One of the things to note about any Action query is that it changes table data in some way. For this reason, it is good practice to trial the query before running it. Trialling also allows you to check for errors in the query setup.

To trial your Append query, click on the  button. The resulting dynaset should display all the records in the FRIENDS table. Return to query design view, and run the query. Access should append all FRIENDS table records to TUTE7 COPYTABLE.

6. Save your query (eg: TUTE7 APPEND QUERY), and return to the Database window. Open the TUTE7 COPYTABLE and check the contents.

Hint:	You will need to use both the Copy/Paste table structure sequence and Append queries to complete Task 2 of Case 7.

Database Case 8

Leedy Rentals Inc.

Problem:	Develop a transaction and fee checking system
Management skills:	Control Decide
Access Skills:	Forms Action Queries Macros

Data Table: LEEDY

Doug Leedy punched the buttons on his hand-held calculator. Three months ago Leedy had been an independent rental car dealer, based in Fargo, North Dakota, when Best Price Rentals, a medium sized operator, offered him a sizeable sum for majority ownership of Leedy Rentals. Best Price had recognized that the recent boom in the Fargo area was merely the start of sustained growth. Best Price also realised that the three existing rental agencies (two of them franchises of national dealerships), would provide intense competition for any start-up enterprise in the area. Best Price decided to approach the single independent dealer with a partnership offer.

Doug was surprised and relieved when the Best Price offer arrived. His 17 year old dealership was straining under the pressure from the newly established national franchises which were able to offer some cut price deals, one-way rentals, and arrangements with frequent flyer schemes which he couldn't. Best Price had a cooperative deal with one of the key domestic airlines and Leedy realised that when rental dealers prices were very similar, customers made decisions based on issues like frequent flyer points. Best Price could also assist during temporary price wars. Faced with deciding between hard times as an independent and receiving cash for losing control, Leedy accepted Best Price's offer.

Leedy soon realized that the extra security offered by Best Price came with additional obligations such as monthly reporting duties. These reports involved the calculations he was currently performing manually on his calculator. Leedy already had a computerized reservation and charging system, a necessity in his industry. However, his system did not have the capability to provide the required reports. Although Best Price were willing to supply their own software, Leedy was happy with his system. He knew that his system did offer the feature of writing details of transactions to an exportable file. With this knowledge, Leedy asked Best Price to send one of their system programmers. Best Price obliged by dispatching Helen Linden.

Helen examined a typical file on Leedy's system and found that it could be readily imported into any common database management system. Although not ideal, this solution would certainly prevent the need for Leedy to manually calculate his reports. Helen advised him to purchase a Windows-based database package, and use it to meet Best Price's monthly reporting requirements.

The management of Best Price needs quarterly information in order to make informed decisions. They require the total revenue derived from each of the car model sizes over each month. Each transaction record contains certain information: a sequential code generated by the computer, date of transaction, model code (S=small, M=medium, L=large, X=sports, W=wagon), car registration, miles travelled, number of rental days, rental fee and a logical field signifying whether the rental was "limited miles" or not. The customer usually has the choice of having unlimited miles or a limit of 100 free miles with a lower base rate. For a limited mile transaction, each mile above 100 miles would incur a per mile cost.

The LEEDY data table supplied on the *Solve it!* disk contains a partial list of transactions downloaded from Leedy Rentals reservation system.

Tasks: There are three tasks in this case:

1. Doug Leedy wants Helen to develop a simple solution for him so that he can check the accuracy of the transactions in the sample list against the paper records he has in his office. He wishes the solution to include an entry screen where the user may enter the transaction code (e.g. S24155). The procedure will then display the record corresponding to the code if it is valid. If the code is not valid the user should be informed the entry is invalid and be required to re-enter the code. The user should be able to exit the system by entering a certain code (e.g. "X"). The entry screen should have ample instructions for the user to use the system.

   Hint: use Forms and Form Control Wizards to complete this task.

2. A frustrating aspect of the existing package is that the output files do not contain the rental fee. To meet Leedy's reporting commitments, he must manually calculate the revenue from each transaction. Thus, he is keen for Helen to develop a procedure to calculate the fee for each transaction and place the result in the database. As Best Price requires the total revenue by model type, Leedy wants Helen to include this facility as well. The charges for each model type are listed in the table below:

	Small	Medium	Large	Sports	Wagon
Limited Rate ($)	45	50	55	60	60
Additional cost per mile ($)	0.55	0.65	0.75	0.85	0.80
Unlimited Rate ($)	50	55	60	65	65
Depreciation Rate (%)	3.3	3.2	3.0	3.5	4.4
Purchase Price ($)	9200	11650	17800	19250	17100

   Hint: use a series of Update Action Queries to complete this task.

*3. Best Price is becoming more demanding in its reporting requirements, adding to Leedy's workload. They wish to know the depreciation of any car in Leedy's yard.

   The method of depreciation in the car rental industry is dictated by the Internal Revenue Service. The depreciation amount for each vehicle depends on the total distance travelled, read from the odometer, the rate of depreciation for the model type (e.g. small) and the purchase price of the vehicle. The IRS defines the depreciation amount as a fixed

percentage (i.e. the depreciation rate) of the purchase price, for every 6,000 miles the vehicle has travelled. Develop a procedure to calculate the amount of depreciation for each car based on the depreciation rates and purchase prices listed in the table above. Allow the user to enter the registration of a particular vehicle and then calculate the depreciation on that car.

Hint: use Parameter statements within Queries to complete this task.

Time Estimates (excluding task marked with *):

Expert: 1.5 hour
Intermediate: 2.5 hours
Novice: 4 hours

Tutorial For Database Case 8 Using Access 97

Creating Forms in Access 

Forms are used to view, enter or edit data in a database, and are often the most convenient way of performing these operations. Forms can be based on tables or queries, and display data on a record by record basis. As with reports, the forms feature includes a series of wizard dialog boxes, which enable the user to customise the way data will be displayed on a form. With one click of the mouse, you can switch from Form to Datasheet view, which is a tabular view of the same set of records.

With an Access form, you can include lists of values to choose from, use colours to highlight important data, include graphics, and display messages to indicate when an incorrect value has been entered. You can also set up your form so that Access automatically inserts data for you, or prints data on the click of a command button, and/or displays the results of calculations. Let's use the FRIENDS database to create a simple form.

1. Load the practice database FRIENDS.MDB. From the Database window, click on the *Form* object and then click New. From the *New Form* dialog box, click on the ↓ button to the right of the *Choose the Table or Query Where an Object's Data Comes From*, and select the FRIENDS table. Now click and highlight the *Autoform:Columnar* form option which will automatically generate a form that displays fields in a single column. Click OK. Access creates the form and opens it for you. The form shows the first record in your table (see Figure 5-77). Save and name your form (eg: FRIENDS SHOW). In a matter of minutes, you have created a form thatcan be used for viewing, editing, adding and deleting records in the FRIENDS table.

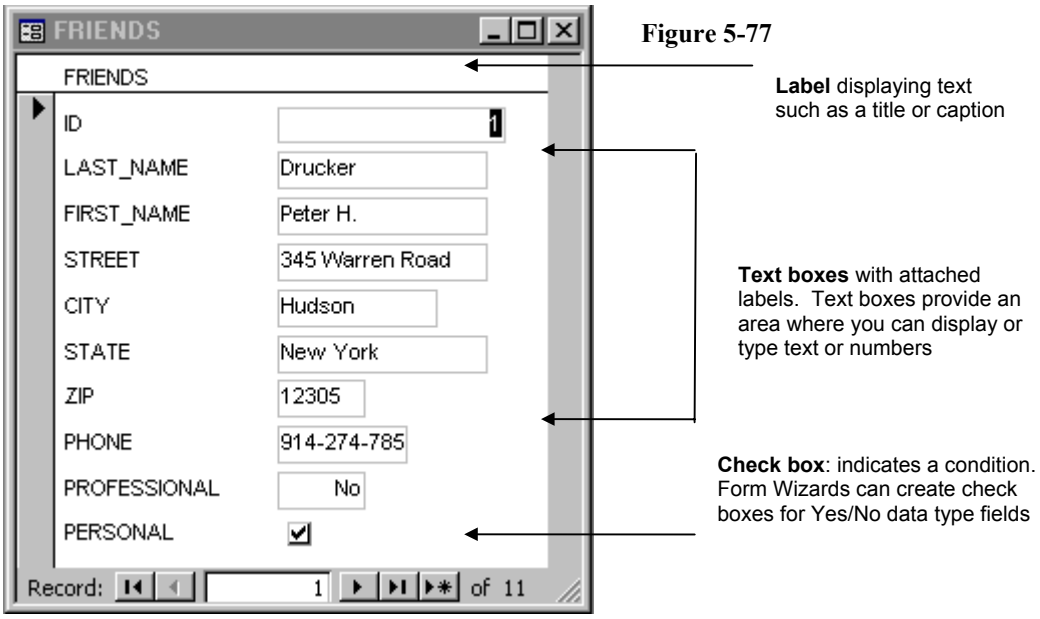


Figure 5-77

Access offers a number of other form wizard types:

Autoform:Datasheet - which displays fields in datasheet format

Autoform:Tabular - which displays each record as a row of fields.

Chart Wizard - creates a form that displays graphs.

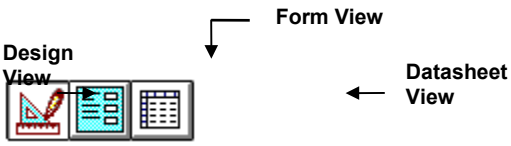
Form Wizard - creates a form based on user-selected fields. This wizard supports a variety of form display formats.

Pivot Table Wizard - creates a form with a Microsoft Excel pivot table - a feature which enables analysis and summarization of data in lists and tables.

or you can create your own form from scratch using the *Design View* option. Forms can be also be used with Macros which are discussed later in this tutorial. You may find this useful to know when considering how to tackle some of the tasks associated with the development of the Leedy Rentals system.

There are a number of different control objects which you can use when designing a form. The *Autoform:Columnar* example in Figure 5-xx displays three of these - labels, text boxes and check boxes. Forms use the same set of Toolbox buttons as Reports (recap Access Case 2 tutorial for a listing and short explanation of these buttons).

Three Ways of Viewing a Form



Access offers three quick and easy ways of viewing a form. You will need to use each of these when designing and using forms. To switch views, simply click on the appropriate toolbar button.

Design View	use for customising the appearance, or changing the structure of forms, and for adding toolbox controls
Form View	use for viewing, entering and editing record data
Datasheet View	displays the underlying table or query dynaset on which the form is based

You can also use the *Print Preview* button to see how a form will look when printed.

Remember:	*right click*, means to click the right hand button of your mouse once. *double click top left*, means to click the control menu button on a open window

Creating a Form from Scratch (using the Design View option)

As the name suggests, you start off with a blank form (which is generated by selecting Design View from the New Form dialog box), and use the toolbox buttons in form design view to add all the text boxes, labels and other controls required for your form.. For this reason, blank forms often have no initial connection to an underlying query or table. *Hint: you will need to use the blank form format for Task 1 of Case 8.*

To create a blank form:

1. From the Database window, click the Form object and then click New. From the New Form Dialog box, highlight the Design View button. Access immediately opens the form in design view. The initial form will only contain a Detail section. Forms use the same type of sectioning as Reports. Recap Access tutorial for Case 2 for an explanation of report sections.

Let's create a simple customized form using the FRIENDS table. The form is designed to lookup and display specific FRIENDS records based on their ID numbers (Note: Your FRIENDS table will need to contain an ID field set to Autonumber field type). *Hint: the criteria used could just as easily be product numbers or transaction codes.*

1. Select VIEW/FORM HEADER/FOOTER from the menu to add a Header to your new form. Next, click on the Label toolbox button, and then click in the Header section of the form. Type a heading for the form (eg: *Friends Lookup*). Change box and font size to suit.

2. Click on the Text Box toolbox button, and then click in the **Detail** section of the form to create an unbound text box and label for the form. Click on the label,

and then *right click* and select Properties from the menu. In the Properties window, change the Caption bar to: *Search for a Friend*. Resize your label so that it displays all of the text. *Double click top left* to save and exit back to form design view.

3. Click on the unbound text box, and then *right click* and select Properties (within Properties, select the All Propoerties tab). Scroll down, click on the Validation Rule bar and enter >=1 And <=16 (eg: where 16 is the last record in your FRIENDS table) to set up a valid ID number range. (Right click and select Zoom if you need more room to type). Click on the Validation Text bar and enter in some infringement text (eg: You don't have that many Friends !! Click OK and ReEnter). *Double click top left* to save and exit back to form design view.

4. Save your form (eg: FRIENDS LOOKUP). Next, we will create two *Command Buttons*. One that links the current form through an event procedure to the FRIENDS SHOW form and displays the relevant records, and one that Exits from FRIENDS SHOW and returns to FRIENDS LOOKUP.

Using Control Wizards in Forms

Creating an *event procedure* in a Report or Form without having to directly do any programming is done through the Control Wizards function. An *event* is a particular action that triggers a procedure such as clicking on a command button in a form or report. A *procedure* is a unit of programming code designed to accomplish a specific task. Access Control wizards automate many frequently used activities associated with forms and reports, such and opening and closing objects, linking fields between objects, and going to specific records in tables or forms.

1. Let's create the first of our command buttons. Click on the Control Wizard toolbox button. Click the *Command Button* tool in the toolbox, and then click on an empty section of the **Detail** section in the FRIENDS LOOKUP form.

The first of several Command Button wizard dialog boxes appears. Make these choices as you proceed through the dialog boxes:

- Under *Categories*, select Form Operations
- Under *Actions*, select Open Form
- Click Next
- Select FRIENDS SHOW as the Form to Open
- Click Next
- Choose to *Open the Form and Find Specific Data to Display*
- Click Next
- From FRIENDS LOOKUP, highlight the Text1 control
- From FRIENDS SHOW, highlight the the ID field and click the <-> button to match the fields
- Click Next
- Select *Text*, and change the button caption to Show Record
- Click Finish to generate the Command Button

2. Save the FRIENDS LOOKUP form again.

3. Right click on the Show Record button and select Properties. Scroll down the list to the On Click: bar. Notice that Access has appended an Event Procedure to this line. Click in the On

Click: bar, and then click on the **...** Build button. Access displays the event procedure code behind the Show Record button (see Figure 5-78). *Double click top left* to return to form design view.

Figure 5-78

```
Private Sub Command3_Click()
On Error GoTo Err_Command3_Click

   Dim stDocName As String
   Dim stLinkCriteria As String

   stDocName = "Friends Show"

   stLinkCriteria = "[ID]=" & Me![Text1]
   DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_Command3_Click:
   Exit Sub

Err_Command3_Click:
   MsgBox Err.Description
   Resume Exit_Command3_Click

End Sub
```

4. **Time to Test:** Switch to Form View. Enter an ID number between 1 and 16 (or whatever the last record number in your FRIENDS table is), and click the Show Record Button. The sequence works, but there is no easy or immediate way to run the procedure again, or to exit from FRIENDS SHOW. We need to add an Exit and ReRun button to FRIENDS SHOW.

5. Change to design view in FRIENDS SHOW. Create another command button in the **Detail** section of the form using the control wizards to Exit FRIENDS SHOW and ReRun the ID sequence. This procedure will be based around closing a form. Save the changes to FRIENDS SHOW, return to form view, and test the new button. It should close FRIENDS SHOW and return to FRIENDS LOOKUP.

6. **Extra Practice:** To complete this exercise, we really need a graceful way to exit the FRIENDS LOOKUP procedure and return to the Database window. Create another command button using the control wizard. (*Hint:* it's identical to the one just created in FRIENDS SHOW). Save and test the button, by clicking and exiting to the database window.

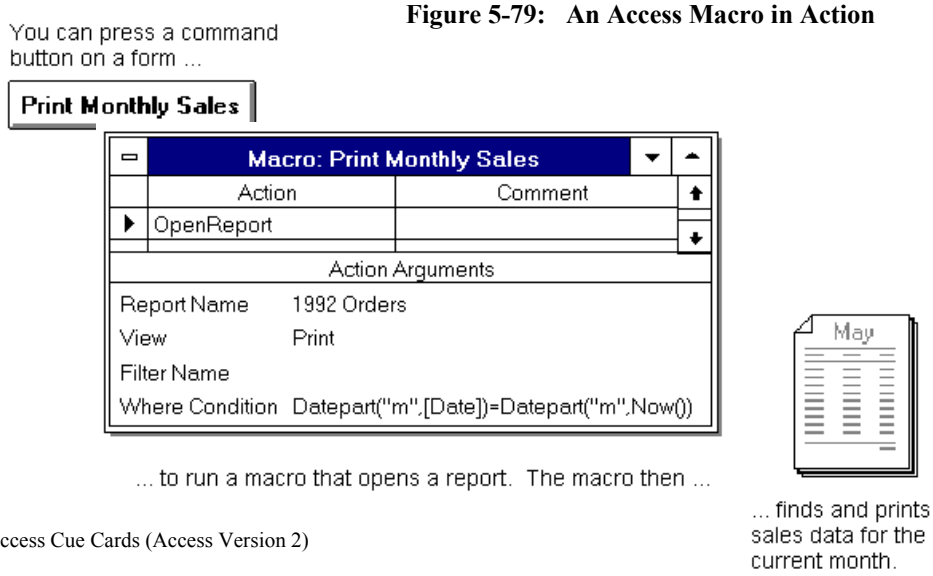
7. From the Database window, click Form and FRIENDS LOOKUP and Open to test and run the procedure several times from the beginning. Also enter an ID not in the valid range to make sure the validation procedure is working. With two simple forms, and three easy to generate command buttons, we have created a robust, easy to use FRIENDS lookup program.

Macros in Access

A *macro* in Access is a set of actions that automate common tasks such as opening a table, or printing a report. Macros help you to work smarter and save time without having to learn programming. Macros are simple to create, and Access offers a choice of around 50 different macro actions which include (see also Figure 5-79):

- opening and closing table, forms and reports
- opening a form, and finding records related to another form
- automatically printing reports upon opening a particular database
- checking or improving data validity

Figure 5-79: An Access Macro in Action



Source: Access Cue Cards (Access Version 2)

The great thing about macros is that they can handle tasks that would often require extensive programming skills in other database packages, and the more you use them, the more you learn about the underlying principles of programming. Macro actions often involve *arguments*. These are simply parameters which govern how an action is executed. If you have problems choosing macro actions or specifying arguments, press the F1 key for online help.

Let's create a simple macro using the practice database FRIENDS.MDB. This macro will open the FRIENDS table, then display only records whose Zip code is greater than 35000, and then within this, filter out all records where the City is not Arizona, before returning to the full FRIENDS table record set.

1. Load FRIENDS.MDB. Click on the *Table* object in the Database Window, and double click on the FRIENDS table. Change to table design view, and make sure the Zip field is set to a numeric data type. Save necessary changes, and close the FRIENDS table.

2. From the Database Window, click the *Macro* object, and then click *New*. This will display the Macro Builder window. Click in the first blank row of the *Action* column, and then click the ⬇ button to display a drop down list of all Macro actions. Using your mouse, scroll down this list, and select the *OpenTable* action. (Read the information box at bottom right of the

window, for an explanation of what this action does). In the *Comments* column, type a short description of the action (eg: Opens the FRIENDS table).

3. Click on the *Table Name* bar in the *Action Arguments* section of the window, and then click on the ⬇ button to the right of this and select the FRIENDS table.

4. Click on the next empty macro row, and select *ApplyFilter* from the *Action* list. In the *Comments* column, type a short description of the action (eg: Only display records where the Zip code is greater than 35000).

5 Click on the *Where Condition* bar in the Action Arguments section and type:

Where Condition	[FRIENDS]![ZIP]>35000

Alternatively, click on the expression builder button to the right of the Where Condition bar to do this.

6. Click on the next empty macro row,and select *ApplyFilter* from the *Action* list. In the *Comments* column, type a short description of the action (eg: Within this reduced record set show records only from Arizona).

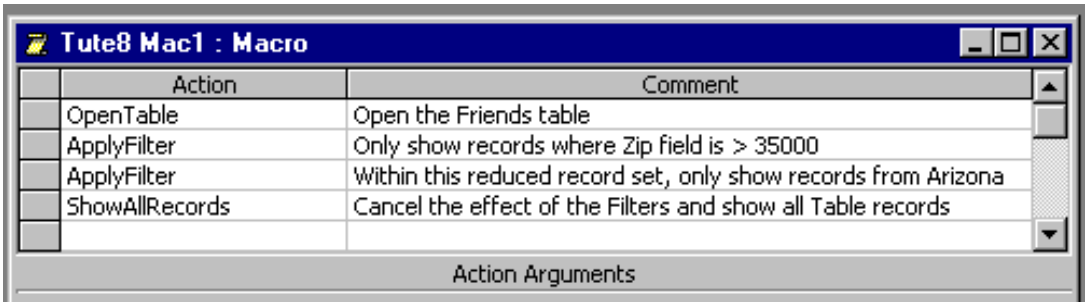
7. Click on the *Where Condition* bar in the *Action Arguments* section and type:

Where Condition	[FRIENDS]![STATE]="Arizona"

or use the expression builder to do this. (Additional records were added to the FRIENDS table in the Tutorial for Case 4. If you did not do this tutorial, you will need to add 4 or 5 new records to the FRIENDS table, making sure that some of the state names you enter in the State field duplicate those of existing records. Choose a State name (eg: Arizona) for your macro line that has duplicates).

8. Click on the next empty macro row,and select *ShowAllRecords* from the *Action* list. In the *Comments* column, type a short description of the action (eg: Cancel effect of all Filters). Your macro window should now look similar to Figure 5-80.

Figure 5-80



9. Save your macro by clicking on the *Save* toolbar button or selecting FILE/SAVE from the menu. In the *Save As* window give your macro a name (eg: TUTE8 MAC1), and click *OK*.

10. Execute the macro, by clicking on the *Run* toolbar button, or selecting MACRO/RUN from the menu. As the macro runs, it firstly opens and displays all the records in the FRIENDS table. The first filter is then applied, and only records where the Zip code is greater than 35000 are shown. The second filter then is then applied, and the record set is further reduced to records

where the Zip code is greater than 35000 and the State is Arizona. Finally, the ShowAllRecords action is applied, and the effect of the filters are cancelled. The full record set of the FRIENDS table now appears.

11. Return to the Database window.

Use Macros When:	• performing tasks such as opening and closing tables or forms, or running reports • your application involves custom menus and submenus for forms • your application is basically simple and uncomplicated, and does not require debugging procedures

Update Queries in Access

Update queries are a type of action query which modifies data according to a specific criteria such as increasing all car rental rates or certain product prices by 15%. Update queries are created in much the same way as other queries except that a new value is specified for a particular field. Like all other types of action queries, update queries save time and effort, but must be used carefully because they actually change the data in the underlying table. Let's use the sample data table HARDWARE to create a simple update query. HARDWARE is included in SOLVE98.MDB.

1. Load the HARDWARE table, and browse the records of the table, particularly noting the data in the Unitcost field (see Figure 5-81).

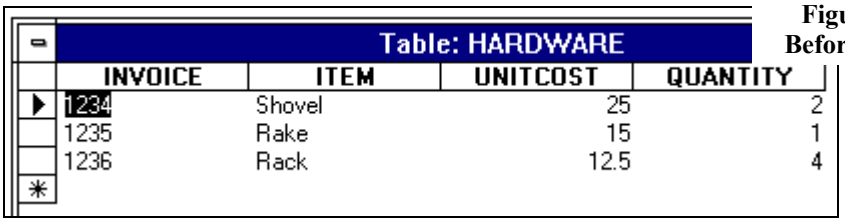


Table: HARDWARE

INVOICE	ITEM	UNITCOST	QUANTITY
1234	Shovel	25	2
1235	Rake	15	1
1236	Rack	12.5	4

Figure 5-81 Before Update

Create a new query based on the HARDWARE table. From the query design window, select QUERY/UPDATE from the menu or click on the Update query toolbar button. This action turns the Select query into an Update query, and causes Access to add an *Update to:* line to the QBE Grid.

3. From the Update query design window, select and drag the Item and Unitcost fields in the HARDWARE field list down onto the Field: bar of the QBE Grid. In the *Update to:* cell under the Unitcost field enter the expression [Unitcost]*1.15. This tells Access that we want to increase prices of all items in the HARDWARE table by 15%.

4. Save (eg: TUTE8 QUERY1) and run the query, and then press F11 to return to the Database window. Open the HARDWARE table and notice the changes made to the Unitcost field (see Figure 5-82). Access has updated all prices by 15%.

Table: HARDWARE

INVOICE	ITEM	UNITCOST	QUANTITY
1234	Shovel	28.75	2
1235	Rake	17.25	1
1236	Rack	14.375	4

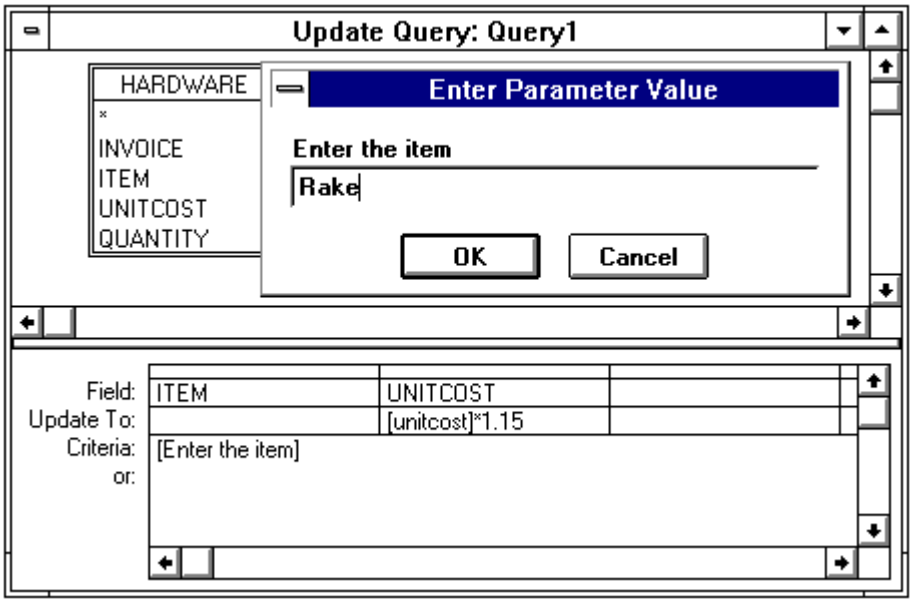
Update queries can also be limited to certain records or groups of records. Let's say we only wanted to update the unitcost of Rakes. A simple criteria or *parameter* must be added to the query to impose a filter on the table records. Access offers two ways of doing this:

a) In the Criteria: cell under the Item field of the update query window, enter Rake. This tells Access to limit update of the Unitcost field to only those records which contain the word Rake in the Item field.

alternatively:

b) In the Criteria: cell under the Item field of the update query window, enter a parameter statement: [Enter an Item]. Syntax is exactly as shown. When the query is run, Access will display the query Parameter Value window, prompting the user to enter text that will limit the update action (refer Figure 5-83). Entering Rake will produce the same result as method (a).

Figure 5-83



Parameter statements can be used with all types of queries - select and action. They provide an added degree of control by minimising user intervention with the query design window.

Tutorial For Database Case 8 Using Access 2.0

Creating Forms in Access

Forms are used to view, enter or edit data in a database, and are often the most convenient way of performing these operations. Forms can be based on tables or queries, and display data on a record by record basis. As with reports, the forms feature includes a series of wizard dialog boxes, which enable the user to customise the way data will be displayed on a form. With one click of the mouse, you can switch from Form to Datasheet view, which is a tabular view of the same set of records.

With an Access form, you can include lists of values to choose from, use colours to highlight important data, include graphics, and display messages to indicate when an incorrect value has been entered. You can also set up your form so that Access automatically inserts data for you, or prints data on the click of a command button, and/or displays the results of calculations. Let's use the FRIENDS database to create a simple form.

1. Load the practice database FRIENDS.MDB. From the Database window, click on the *Form* object and then click New. From the *New Form* dialog box, click on the ⬇ button to the right of *the Select a Table/Query* bar, and select the FRIENDS table. Click on the *Form Wizards* button, and select *Autoform*. Click OK. Autoform creates the form and opens it for you. The form shows the first record in your table (see Figure 5-84). Save and name your form (eg: FRIENDS SHOW)

Note: The same result can be achieved by highlighting the FRIENDS table in the Database Window, and then clicking on the *Autoform* button in the toolbar. In a matter of minutes, you have created a form thatcan be used for viewing, editing, adding and deleting records in the FRIENDS table.

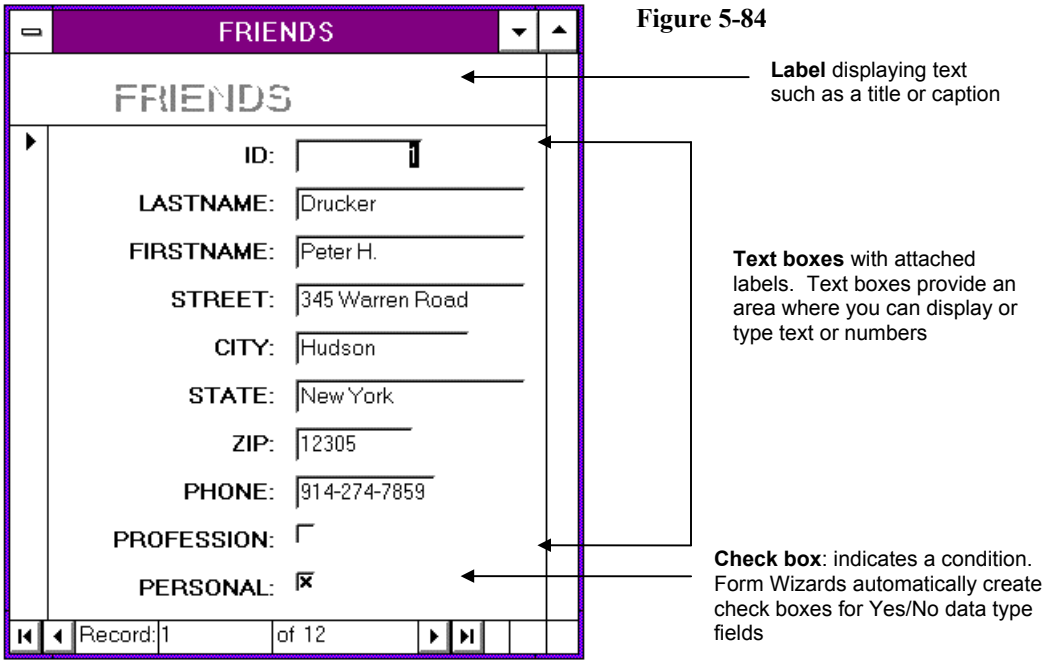


Figure 5-84

Autoform bypasses the Form Wizard dialog boxes by automatically creating a form based on a selected table or query. There are four other form wizard types:

Single Column - which displays fields in a single column.

Tabular - which displays each record as a row of fields.

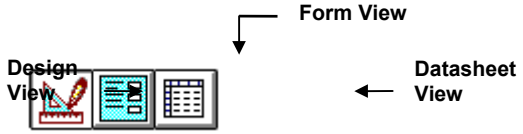
Graph - creates a form that displays graphs.

Main/Subform - creates a form that contains another form. This format displays a one to many relationship between data in the main form (one), and data in the subform (many). Subform data is displayed in datasheet view.

or you can create your own form from scratch. Forms can be also be used with Macros which are discussed later in this tutorial. You may find this useful to know when considering how to tackle some of the tasks associated with the development of the Dickerson Rentals system.

There are a number of different control objects which you can use when designing a form. The Autoform in Figure 5-86 displays three of these - labels, text boxes and check boxes. Forms use the same set of Toolbox buttons as Reports (recap Access Case 2 tutorial for a listing and short explanation of these buttons).

Three Ways of Viewing a Form



Access offers three quick and easy ways of viewing a form. You will need to use each of these when designing and using forms. To switch views, simply click on the appropriate toolbar button.

Design View	use for customising the appearance, or changing the structure of forms, and for adding toolbox controls
Form View	use for viewing, entering and editing record data
Datasheet View	displays the underlying table or query dynaset on which the form is based

You can also use the *Print Preview* button to see how a form will look when printed.

Remember:	*right click*, means to click the right hand button of your mouse once. *double click top left*, means to click the control menu button on a open window

Creating a Form from Scratch

As the name suggests, you start off with a blank form, and use the toolbox buttons in form design view to add all the text boxes, labels and other controls required for your form.. For this reason, blank forms usually have no initial connection to an underlying query or table. *Hint: you will need to use the blank form format for Task 1 of Case 8.*

To create a blank form:

1. From the Database window, click the Form object and then click New. From the New Form Dialog box, click on the Blank Form button. Access immediately opens the form in design view. The initial form will only contain a Detail section. Forms use the same type of sectioning as Reports. Recap Access tutorial for Case 2 for an explanation of report sections.

Let's create a simple customised form using the FRIENDS table. The form is designed to lookup and display specific FRIENDS records based on their ID numbers. *Hint: the criteria used could just as easily be product numbers or transaction codes.*

1. In Form design view, select VIEW/PROPERTIES from the menu to open the form properties window. Click on the Record Source bar and set to the FRIENDS table by clicking on the ▼ button and selecting from the drop down list. *Double click top left* in this window to save and return to design view.

2. Select FORMAT/FORM HEADER/FOOTER from the menu to add a Header to your new form. Next, click on the Label toolbox button, and then click in the Header section of the form. Type a heading for the form (eg: *Friends Lookup*). Change box and font size to suit.

3. Click on the Text Box toolbox button, and then click in the Detail section of the form to create an unbound text box and label for the form. Click on the label, and then *right click* and select Properties from the menu. In the Properties window, change the Caption bar to: *Search for a Friend*. Resize your label so that it displays all of the text. *Double click top left* to save and exit back to form design view.

4. Click on the unbound text box, and then *right click* and select Properties. Scroll down, click on the Validation Rule bar and enter >="1" And <="16" (where 16 is the last record in your FRIENDS table) to set up a valid ID number range. (Right click and select Zoom if you need

more room to type). Click on the Validation Text bar and enter in some infringement text (eg: You don't have that many Friends !! Click OK and ReEnter).

5. Save your form (eg: FRIENDS LOOKUP). Next, we will create two *Command Buttons*. One that links the current form through an event procedure to the FRIENDS SHOW form and displays the relevant records, and one that Exits from FRIENDS SHOW and returns to FRIENDS LOOKUP.

Using Control Wizards in  **Forms** Control Wizards Button

Creating an *event procedure* in a Report or Form without having to directly do any programming is done through the Control Wizards function. An *event* is a particular action that triggers a procedure such as clicking on a command button in a form or report. A *procedure* is a unit of programming code designed to accomplish a specific task. Access Control wizards automate many frequently used activities associated with forms and reports, such and opening and closing objects, linking fields between objects, and going to specific records in tables or forms.

1. Let's create the first of our command buttons. Click on the Control Wizard toolbox button, or select VIEW/CONTROL WIZARDS from the menu.
Click the *Command Button* tool in the toolbox, and then click on an empty section of the detail section in the FRIENDS LOOKUP form. 

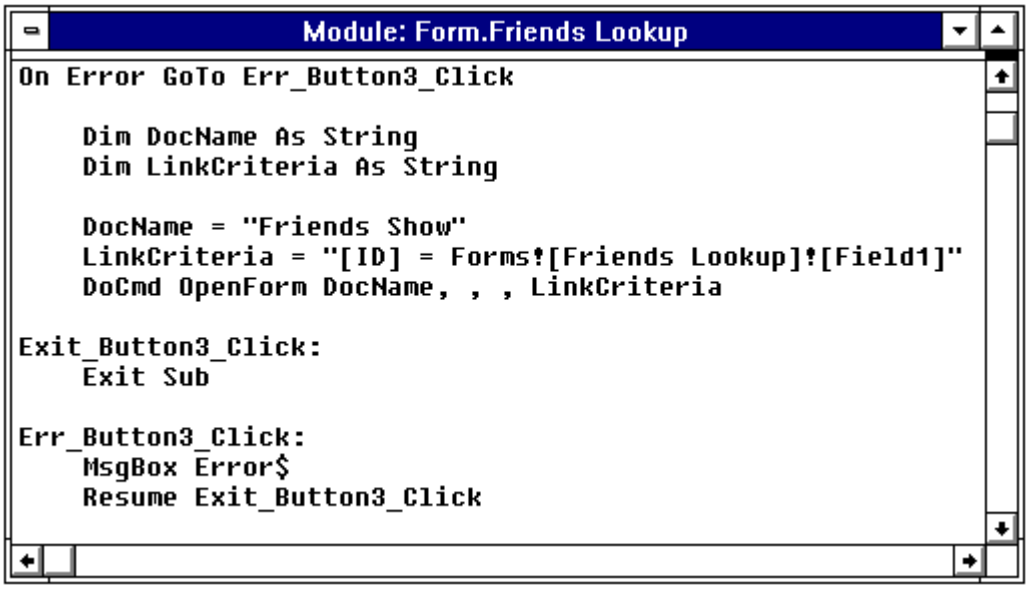
The first of several Command Button wizard dialog boxes appears. Make these choices as you proceed through the dialog boxes:

- Under *Categories*, select Form Operations
- Under *When Button is Pressed*, select Open Form
- Click Next
- Select FRIENDS SHOW as the Form to Open
- Click Next
- Choose to *Open the Form and Find Specific Data*
- Click Next
- From FRIENDS LOOKUP, highlight the Field1 control
- From FRIENDS SHOW, highlight the the ID field and click the <-> button to match the fields
- Click Next
- Select *Text*, and change the button caption to Show Record
- Click Finish to generate the Command Button

2. Save the FRIENDS LOOKUP form again.

3. Click on the Show Record button, and then *right click* and select Properties. Scroll down the list to the On Click: bar. Notice that Access has appended an Event Procedure to this line. Click in the On Click: bar, and then click on the **...** Build button. Access displays the event procedure code behind the Show Record button (see Figure 5-84). *Double click top left* to return to form design view.

Figure 5-84



4. **Time to Test:** Switch to Form View. Enter an ID number between 1 and 16 (or whatever the last record number in your FRIENDS table is), and click the Show Record Button. The sequence works, but there is no easy or immediate way to run the procedure again, or to exit from FRIENDS SHOW. We need to add an Exit and ReRun button to FRIENDS SHOW.

5. Change to design view in FRIENDS SHOW. Create another command button using the control wizards to Exit FRIENDS SHOW and ReRun the ID sequence. This procedure will be based around closing a form. Save the changes to FRIENDS SHOW, return to form view, and test the new button. It should close FRIENDS SHOW and return to FRIENDS LOOKUP.

6. **Extra Practice:** To complete this exercise, we really need a graceful way to exit the FRIENDS LOOKUP procedure and return to the Database window. Create another command button using the control wizard. (*Hint:* it's identical to the one just created in FRIENDS SHOW). Save and test the button, by clicking and exiting to the database window.

7. From the Database window, click Form and FRIENDS LOOKUP and Open to test and run the procedure several times from the beginning. Also enter an ID not in the valid range to make sure the validation procedure is working. With two simple forms, and three easy to generate command buttons, we have created a robust, easy to use FRIENDS lookup program.

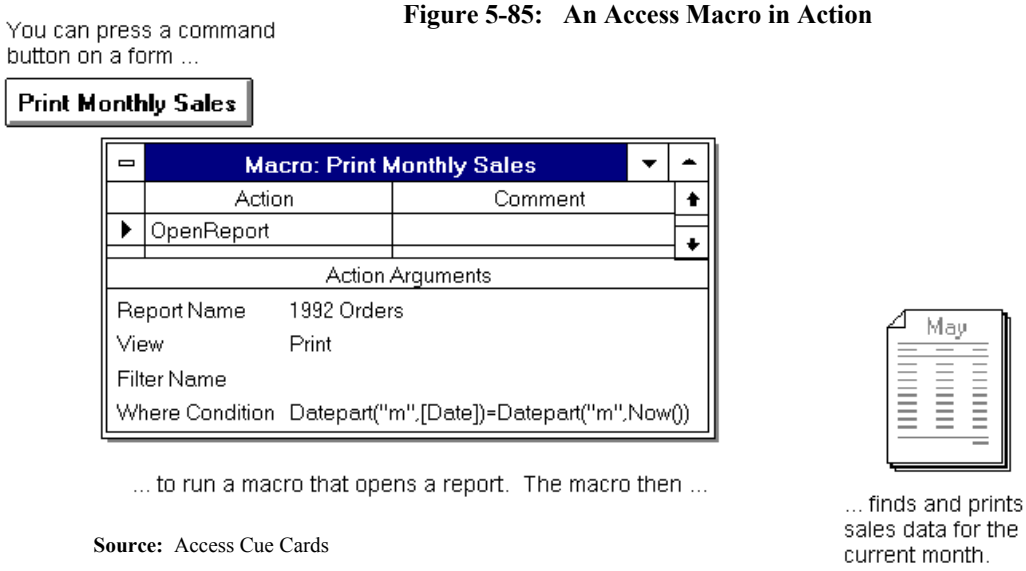
Macros in Access

A *macro* in Access is a set of actions that automate common tasks such as opening a table, or printing a report. Macros help you to work smarter and save time without having to learn programming. Macros are simple to create in Access, and this operation is made even simpler if

you use Cue Cards as an interactive assistant. Access offers a choice of 47 different macro actions which include (see also Figure 5-85):

- opening and closing table, forms and reports
- opening a form, and finding records related to another form
- automatically printing reports upon opening a particular database
- checking or improving data validity

Figure 5-85: An Access Macro in Action



Source: Access Cue Cards

The great thing about macros is that they can handle tasks that would often require extensive programming skills in other database packages, and the more you use them, the more you learn about the underlying principles of programming. Macro actions often involve *arguments*. These are simply parameters which govern how an action is executed. If you have problems choosing macro actions or specifying arguments, press the F1 key for online help and examples screens.

Let's create a simple macro using the practice database FRIENDS.MDB. This macro will open the FRIENDS table, then display only records whose Zip code is greater than 35000, and then within this, filter out all records where the City is not Arizona, before returning to the full FRIENDS table record set.

1. Load FRIENDS.MDB. Click on the *Table* object in the Database Window, and double click on the FRIENDS table. Change to table design view, and make sure the Zip field is set to a numeric data type. Save necessary changes, and close the FRIENDS table.

2. From the Database Window, click the *Macro* object, and then click *New*. This will display the Macro Builder window. Click in the first blank row of the *Action* column, and then click the ⬇ button to display a drop down list of all Macro actions. Using your mouse, scroll down this list, and select the *OpenTable* action. (Read the information box at bottom right of the window, for an explanation of what this action does). In the *Comments* column, type a short description of the action (eg: Opens the FRIENDS table).

3. Click on the *Table Name* bar in the *Action Arguments* section of the window, and then click on the ⬇ button to the right of this and select the FRIENDS table.

4. Click on the next empty macro row, and select *ApplyFilter* from the *Action* list. In the *Comments* column, type a short description of the action (eg: Only display records where the Zip code is greater than 35000).

5 Click on the *Where Condition* bar in the Action Arguments section and type in the expression:

Where Condition	[FRIENDS]![ZIP]>35000

Alternatively, click on the expression builder button to the right of the Where Condition bar to do this.

6. Click on the next empty macro row,and select *ApplyFilter* from the *Action* list. In the *Comments* column, type a short description of the action (eg: Within this reduced record set show records only from Arizona).

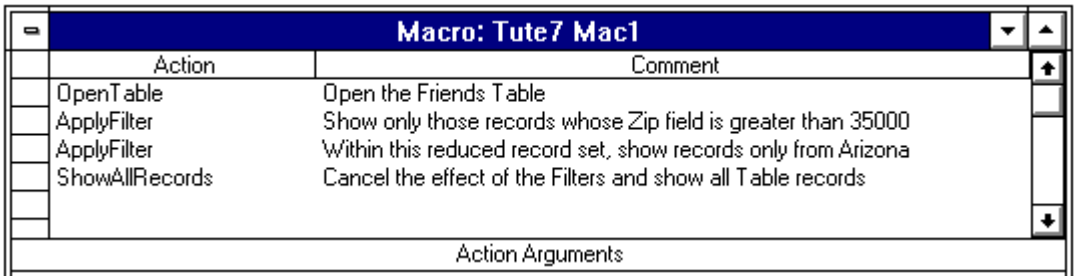
7 Click on the *Where Condition* bar in the *Action Arguments* section and type in the expression:

Where Condition	[FRIENDS]![STATE]="Arizona"

or use the expression builder to do this. (Additional records were added to the FRIENDS table in the Tutorial for Case 4. If you did not do this tutorial, you will need to add 4 or 5 new records to the FRIENDS table, making sure that some of the state names you enter in the State field duplicate those of existing records. Choose a State name (eg: Arizona) for your macro line that has duplicates).

8. Click on the next empty macro row,and select *ShowAllRecords* from the *Action* list. In the *Comments* column, type a short description of the action (eg: Cancel effect of all Filters). Your macro window should now look similar to Figure 5-86.

Figure 5-86



9. Save your macro by clicking on the *Save* toolbar button or selecting FILE/SAVE from the menu. In the *Save As* window give your macro a name (eg: TUTE8 MAC1), and click *OK*.

10. Execute the macro, by clicking on the *Run* toolbar button, or selecting MACRO/RUN from the menu. As the macro runs, it firstly opens and displays all the records in the FRIENDS table. The first filter is then applied, and only records where the Zip code is greater than 35000 are shown. The second filter then is then applied, and the record set is further reduced to records where the Zip code is greater than 35000 and the State is Arizona. Finally, the ShowAllRecords action is applied, and the effect of the filters are cancelled. The full record set of the FRIENDS table now appears.

11. Return to the Database window.

Use Macros When:	• performing tasks such as opening and closing tables or forms, or running reports • your application involves custom menus and submenus for forms • your application is basically simple and uncomplicated, and does not require debugging procedures

Update Queries in Access

Update queries are a type of action query which modifies data according to a specific criteria such as increasing all car rental rates or certain product prices by 15%. Update queries are created in much the same way as other queries except that a new value is specified for a particular field. Like all other types of action queries, update queries save time and effort, but must be used carefully because they actually change the data in the underlying table. Let's use the sample database HARDWARE.MDB to create a simple update query.

1. Load HARDWARE.MDB, and browse the records of the table, particularly noting the data in the Unitcost field (see Figure 5-87).

Figure 5-87 Before Update

Table: HARDWARE

INVOICE	ITEM	UNITCOST	QUANTITY
1234	Shovel	25	2
1235	Rake	15	1
1236	Rack	12.5	4

2. Create a new query based on the HARDWARE table. From the query design window, select QUERY/UPDATE from the menu or click on the Update query toolbar button. This action turns the Select query into an Update query, and causes Access to add an *Update to:* line to the QBE Grid.

3. From the Update query design window, select and drag the Item and Unitcost fields in the HARDWARE field list down onto the Field: bar of the QBE Grid. In the *Update to:* cell under the Unitcost field enter the expression [Unitcost]*1.15. This tells Access that we want to increase prices of all items in the HARDWARE table by 15%.

4. Save (eg: TUTE8 QUERY1) and run the query, and then press F11 to return to the Database window. Open the HARDWARE table and notice the changes made to the Unitcost field (see Figure 5-88). Access has updated all prices by 15%.

Figure 5-88 After Update

Table: HARDWARE

INVOICE	ITEM	UNITCOST	QUANTITY
1234	Shovel	28.75	2
1235	Rake	17.25	1
1236	Rack	14.375	4

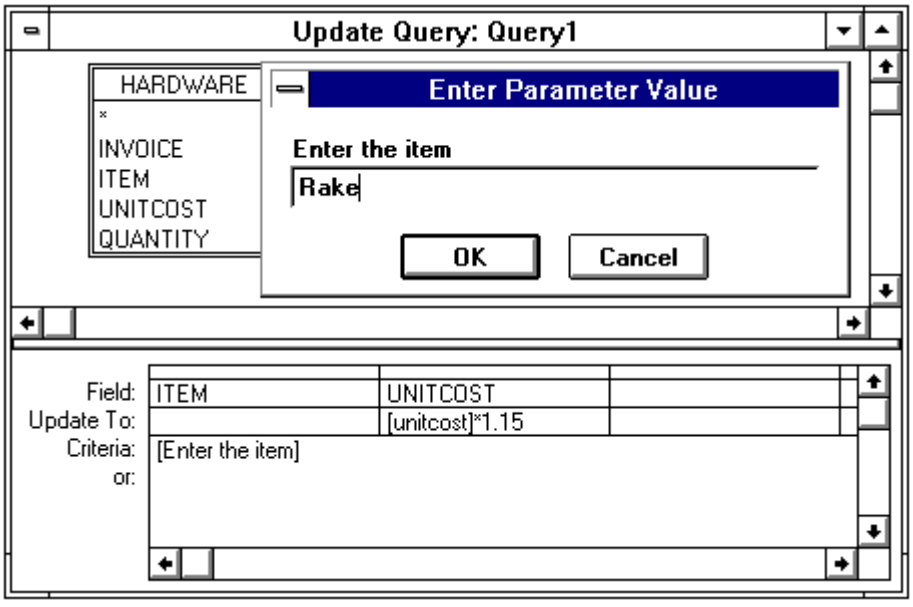
Update queries can also be limited to certain records or groups of records. Let's say we only wanted to update the unit cost of Rakes. A simple criteria or *parameter* must be added to the query to impose a filter on the table records. Access offers two ways of doing this:

a) In the Criteria: cell under the Item field of the update query window, enter Rake. This tells Access to limit update of the Unitcost field to only those records which contain the word Rake in the Item field.

alternatively:

b) In the Criteria: cell under the Item field of the update query window, enter a parameter statement: [Enter an Item]. Syntax is exactly as shown. When the query is run, Access will display the query Parameter Value window, prompting the user to enter text that will limit the update action (refer Figure 5-89). Entering Rake will produce the same result as method (a).

Figure 5-89



Parameter statements can be used with all types of queries - select and action. They provide an added degree of control by minimizing user intervention with

Database Case 10

Mak Audio II

Problem:	Create a complete system
Management skill:	Control Organize
Access Skills:	Forms Macros Advanced Queries (Union)
Data Tables:	MAK_A MAK_B MAK_C

This case is a continuation of Solve it! database Case 6. Refer to this case for background information relating to Case 10.

Rick Mak was feeling very pleased with himself as he gazed out of his office window at the busy Nashville streetscape. He had just finished his perusal of Anabasis' financial performance figures for the 1997-98 year. His mail order music company had performed stunningly last year, with sales improving by nearly 19%. Mak had also noticed that the costs associated with postage and packaging of his mail order catalogs, and the maintenance of his mailing list had dropped significantly.

Mak believed that these results were largely due to the ability of the PC-based database mailing system his company had developed to provide carefully targeted mail outs for his various customer segments. Prevously Mak had relied on blanket mailouts to all customers on his mailing list whenever he wanted to advertise new product offerings. Thanks to his new mailing system, his company was now able to tailor its mailouts so that catalogs were sent out only to customers with interests in the music areas which were relevant to his new products.

Mak now wanted to complete the development of his new system by adding some enhancements and new features. He began to jot down his main requirements.

a) A system that is easy to use.

The new system worked well, but it required a thorough understanding of the database package it had been created in to operate effectively. Mak was the only one who knew how the system worked and what it did, and he currently maintained the system himself. He did not regard this as a productive way to spend his time, and wanted to pass over the operation and day-to-day maintenance of the system to his clerical staff.

Mak had to remember which query did what, and which report gave him the result formats he wanted. The system was efficient but it was not the kind of product you would want in the hands of untrained people. What was needed was a more robust and user-friendly version of the system.

Mak's objective is to produce a self-contained mailing system that could be operated by someone with little or no database knowledge. The system must be menu driven and modular in design. The menu should also execute automatically when the user opens the database which contains the system (ie: the .MDB file). When a task option is selected from the menu, the menu will open the required object, and then execute that option. Each option should contain a facility to exit back to the menu.

b) Integrate the Anabasis mailing list with that of Mak Design.

Mak suspects that at least 25% of the customers on his mailing list are also customers who regularly purchase from Mak Design, his brother's side of the business (refer back to *Case 6*). While the marketing side of Mak Design has been handled by a PC-based database package for some time, the system is not menu driven and requires someone with a good understanding of databases to operate it. Mak and his brother would like to combine their mailing lists under one menu based system. They have agreed that while information unique to each of the businesses will be maintained separately, the names and addresses will be shared. Although the two divisions do business with many of the same organizations, Anabasis' customer contact names may often be different to those used by Mak Design.

Rick Mak would like to test how well this procedure will work. His brother Allan has provided him with a sample listing of Mak Design customer names. This is provided on the *SolveIt!* data table MAK_C. Rick wants to combine this file with his own listing (MAK_A). The result of this procedure should show no duplicate records. The combined listings will then be incorporated into a menu option on the new system.

Mak needs your help. He does not have the time or expertise needed to create the menu driven mailing system required, and would like you to develop a prototype for him. Part of the job was completed in *SolveIt!* database Case 6, and you will need to use the database used for this case to complete Case 10.

Since this case is an extension of *SolveIt*! Case 6, it is recommended that you make a copy of the Case 6 files and save as Case 10, or to simply rename the Case 6 files as Case 10 before starting this case. As a first step, copy and save your Case 6 database with a new name.

Tasks

There are 6 tasks in this case.

1. Create a Main Menu form and appropriate command buttons which allow the user the following executable options. Call this form MAINMENU.

    1. Add a new customer.

    2. Add a new company.

    3. Produce a complete mailing list.

    4. Exit the system.

Hint: Use the command button toolbox tool on the menu form for the first three options: These options will not initially be active. Use the form control wizards and the command button toolbox tool to set up an active Exit procedure for option four.

2. Create data entry forms for Options 1 and 2. Include Exit to Menu and Add New Record command buttons and test them thoroughly. Call these forms OPTION1 and OPTION2.

3. Devise a procedure that will execute the Main Menu automatically whenever the database is opened.

4. Create a procedure that will combine the customer name tables of Anabasis (MAK_A) and Mak Design (MAK_C). There are a number of names which appear in both tables. Your aim is to produce a listing with no duplicate records. The combined file should then be joined to the MAK_B table to give a full listing of customers and their organisational details.

5. Create a report to display results for Task 4. The report should be sorted firstly by organisation, and then by last name. The Country field should also be included. Incorporate this report as Option 3 on the Main Menu form

   Hint: Use the existing report created for Task2/3 of Case 6, and simply reassign the report record source to reflect the changes.

*6. Extend the system even further to include the sales and number of sound recordings sold to each customer. Add the necessary fields and some sample data to the appropriate table. Then add a new option to the menu which will calculate and display the total sales for Anabasis and the average price per recording.

Time Estimates (excluding task marked with *):

Expert: 2 hours
Intermediate: 3 hours
Novice: 5 hours

Tutorial For Database Case 10 Using Access 97

Advanced Queries - the SQL-Specific Query

SQL (Structured Query Language) is a simple programming language used for querying, updating and managing relational databases. When you create a select or action query, Access automatically generates the equivalent SQL statement in background. You can view or edit this SQL statement by choosing SQL from the View menu in the Query window or clicking the SQL View toolbar button.

Some Access query types however, can only be created by writing an SQL statement. These queries are known as SQL-Specific queries. This group includes:

Union	*queries which combine fields from two or more tables or queries*
Pass-through	*queries which send commands directly to a database server*
Data-definition	*queries used to create or alter database tables in the current database*
Subqueries	*an SQL SELECT statement inside another select or action query*

SQL statements refer to expressions that define SQL commands such as SELECT, UPDATE or DELETE, and include clauses such as WHERE, GROUP BY and ORDER BY. SQL statements are typically used in the construct of queries and aggregate funtions.

The Union Query 

You will need to use the Union SQL-Specific query type as part of the requirement for completing Task 1 of Case 10.

Using simple *SELECT ... FROM* SQL statements, *Union Queries* enable you to combine fields from two or more tables into one listing. In contrast, *Select queries* which are based on a join, creates a dynaset only from those records whose related fields meet a specified condition. Commands and clauses commonly used in the creation of Union queries include:

SELECT *fieldlist*
FROM *tablenames* **IN** *database name*
WHERE *search conditions*
GROUP BY *fieldlist*
HAVING *search conditions*
ORDER BY *fieldlist*

To create a Union Query:

1. From the Database Window, click on the Query object, and then click on the New button. This will display the New Query dialog box. Highlight the Design View option and then click OK.

2. Choose the Close button from the Add Table dialog box. Union queries do not use the Query Window/QBE Grid for their construction.

3. Choose QUERY/SQL SPECIFIC/UNION from the menu. Access displays the Union Query window.

4. Enter the appropriate SQL *SELECT....xx* statements needed for your Union Query. For example:

SELECT [Supplier Name], [City] FROM Suppliers UNION SELECT [Customer Name], [City] FROM Customers;	*Retrieves the names and cities of suppliers and customers from the Suppliers and Customers tables*
SELECT [Supplier Name], [City] FROM Suppliers WHERE [Country] = "Brazil" UNION SELECT [Customer Name], [City] FROM Customers WHERE [Country] = "Brazil" ORDER BY [City];	*Retrieves the names and cities of suppliers and customers located in Brazil, sorted by the City field*

5. Be sure to include a semi colon at the end of each statement (refer examples above) to complete the required Access syntax. Note also the use of square brackets to enclose field names. This is particularly important if you are using field names which are composed of two or more words separated by spaces.

6. Click on the Run Query toolbar button to execute and test your Union query. Save the query, and press the F11 key to exit back to the Database Window

Note: Unless you specify otherwise, a Union Query automatically removes duplicate records from the resulting listing. If you want to show all records, including duplicates, add the word ALL after the word UNION in your statement. For example:

SELECT [Supplier Name], [City] FROM Suppliers **UNION ALL** SELECT [Customer Name], [City] FROM Customers;	*Retrieves the names and cities of suppliers and customers from the Suppliers and Customers tables, and shows all records including duplicates*

Task 1 of Case 10 requires you to create a merged file of the customer records of Anabasis and Mak Design. Contrast the difference between omitting and and including ALL in the Union Query you need to create to complete this task.

> Caution: Don't convert an SQL-specific query to another type of query, such as a select query. If you do, you'll lose the SQL statement that you entered. *You can however use an SQL-specific query as part of a select query.*

Creating an AutoExec Macro

You can create a special macro that runs automatically whenever you open a Microsoft Access database. For example, you may wish to open certain tables and forms every time you open a database.

To create a macro that runs whenever you open a database

1. Create a macro.
2. Add the actions that you want this macro to perform.
3. Save the macro. Its name must be AutoExec.

AutoExec macros can be used to create a custom workspace, import data from other databases, or to execute tasks that you want to perform every time the database is opened. To prevent the AutoExec macro from running, hold down the Shift key while opening the database.

Tutorial For Database Case 10 Using Access 2.0 

Advanced Queries - the SQL-Specific Query

SQL (Structured Query Language) is a simple programming language used for querying, updating and managing relational databases. When you create a select or action query, Access automatically generates the equivalent SQL statement in background. You can view or edit this SQL statement by choosing SQL from the View menu in the Query window or clicking the SQL View toolbar button.

Some Access query types however, can only be created by writing an SQL statement. These queries are known as SQL-Specific queries. This group includes:

Union	*queries which combine fields from two or more tables or queries*
Pass-through	*queries which send commands directly to a database server*
Data-definition	*queries used to create or alter database tables in the current database*
Subqueries	*an SQL SELECT statement inside another select or action query*

SQL statements refer to expressions that define SQL commands such as SELECT, UPDATE or DELETE, and include clauses such as WHERE, GROUP BY and ORDER BY. SQL statements are typically used in the construct of queries and aggregate funtions.

The Union Query 

You will need to use the Union SQL-Specific query type as part of the requirement for completing Task 1 of Case 10.

Using simple *SELECT ... FROM* SQL statements, *Union Queries* enable you to combine fields from two or more tables into one listing. In contrast, *Select queries* which are based on a join, creates a dynaset only from those records whose related fields meet a specified condition. Commands and clauses commonly used in the creation of Union queries include:

SELECT *fieldlist*
FROM *tablenames* **IN** *database name*
WHERE *search conditions*
GROUP BY *fieldlist*
HAVING *search conditions*
ORDER BY *fieldlist*

To create a Union Query:

1. From the Database Window, click on the Query object, and then click on the New button. This will display the New Query dialog box. Click on the New button.

2. Choose the Close button from the Add Table dialog box. Union queries do not use the Query Window/QBE Grid for their construction.

3. Choose QUERY/SQL SPECIFIC/UNION from the menu. Access displays the Union Query window.

4. Enter the appropriate SQL *SELECT....xx* statements needed for your Union Query. For example:

SELECT [Supplier Name], [City] FROM Suppliers UNION SELECT [Customer Name], [City] FROM Customers;	*Retrieves the names and cities of suppliers and customers from the Suppliers and Customers tables*
SELECT [Supplier Name], [City] FROM Suppliers WHERE [Country] = "Brazil" UNION SELECT [Customer Name], [City] FROM Customers WHERE [Country] = "Brazil" ORDER BY [City];	*Retrieves the names and cities of suppliers and customers located in Brazil, sorted by the City field*

5. Be sure to include a semi colon at the end of each statement (refer examples above) to complete the required Access syntax. Note also the use of square brackets to enclose field names. This is particularly important if you are using field names which are composed of two or more words separated by spaces.

6. Click on the Run Query toolbar button to execute and test your Union query. Save the query, and press the F11 key to exit back to the Database Window

Note: Unless you specify otherwise, a Union Query automatically removes duplicate records from the resulting listing. If you want to show all records, including duplicates, add the word ALL after the word UNION in your statement. For example:

SELECT [Supplier Name], [City] FROM Suppliers **UNION ALL** SELECT [Customer Name], [City] FROM Customers;	*Retrieves the names and cities of suppliers and customers from the Suppliers and Customers tables, and shows all records including duplicates*

Task 1 of Case 10 requires you to create a merged file of the customer records of Anabasis and Anaheim Design. Contrast the difference between omitting and and including ALL in the Union Query you need to create to complete this task.

Caution: Don't convert an SQL-specific query to another type of query, such as a select query. If you do, you'll lose the SQL statement that you entered. *You can however use an SQL-specific query as part of a select query.*

Creating an AutoExec Macro

You can create a special macro that runs automatically whenever you open a Microsoft Access database. For example, you may wish to open certain tables and forms every time you open a database.

To create a macro that runs whenever you open a database

1. Create a macro.
2. Add the actions that you want this macro to perform.
3. Save the macro. Its name must be AutoExec.

AutoExec macros can be used to create a custom workspace, import data from other databases, or to execute tasks that you want to perform every time the database is opened. To prevent the AutoExec macro from running, hold down the Shift key while opening the database.

6

Internet Cases and Exercises

World Wide Web Case 1

Southern Papers

Problem: Locating information and investigating the feasibility of electronic commerce on the Internet's World Wide Web

PC skills: Web Searching

Southern Papers is a medium-sized paper manufacturer located in Tasmania, Australia. The company produces a range of paper products which includes newsprint, corrugated card boards, and a number of high margin, specialist stationery lines.

In line with paper recycling trends worldwide, around 40% of Southern Papers' products involve repulping and reusing paper. Tree-based fibers break down quickly, and each time recycled paper is used, this substantially weakens the strength of the resultant product. To produce a product which is sufficiently robust, Southern Papers needs to import around 5million tons of softwood wood chips each year. This is shredded and mixed with the recycled paper pulp to put strength back into the product. This is a costly and escalating expense for the company; around US$2.5m per year.

The supply of woodchips must be sourced from overseas as the demand for timber products cannot be met by Australia's own forest resources. To offset depletion of its natural forests, the Australian government requires all paper producers to set aside areas for reforestation. Southern Papers has many coniferous softwood plantations under various stages of regrowth. However, one of the main problems with plantation timbers is the length of time needed to grow the trees to a sufficient level of maturity where they can be harvested. This can take between 10-20 years depending on the type of timber used. In the meantime, Southern Papers must rely on imported softwoods to continue business operations.

Sam Cagot, the owner of Southern Papers is constantly on the lookout for opportunities that will give his company an edge over the many competitors in its marketplace. He reads with interest an article in the latest issue of *Paper Producers* exploring the many uses of industrial hemp.

Industrial hemp refers to those parts of the *Cannabis sativa* plant which contain less than 1% tetahydrocannabinols (THC), the psychoactive chemical found in some strains of the *Cannabis* species. Industrial hemp is not to be confused with marijuana. It has no psychoactive properties, and can be grown as a profitable, high quality fiber crop without producing marijuana.

Hemp has been valued throughout history as an important and versatile raw material for many products including textiles, cordage and paper. It is the strongest natural fiber in the world, and was used extensively until the 1950's when a prohibition was imposed because of its narcotic associations.

Cagot would like to find out more about industrial hemp. He has recently attended a demonstration of the Internet's World Wide Web, but does not really understand how it works or what it is. He wonders whether the Internet can provide the information he requires. He also wonders whether the Internet could be used as an alternative channel of distribution for the company's products.

Over a cup of coffee with Chris Wang, the company's Marketing Manager, Cagot outlines his needs. He gives Wang the task of preparing a feasibility report on the possibility of using hemp for reducing raw material and production costs. He would also like Wang to investigate what business opportunities the Internet has to offer Southern Papers, and the type of costs that would be involved for an investment in this technology.

Tasks:

You are employed in the Marketing Department of Southern Papers as a business analyst. Wang has asked you to prepare a report that will satisfy Cagot's requirements. Your report should include answers for at least the following:

1. What is the Internet and where did it come from? Provide a short description.

2. What advantages could the use of industrial hemp offer Southern Papers ?

3. How the Internet could be used to conduct, promote and possibly increase business. Is anyone else doing this in Southern Paper's business areas - in Australia or elsewhere ?

4. Whether it is worthwhile for Southern Papers to establish a Web site on the Internet. If so, what sort of presence should it be ?

5. What are the options and costs involved with establishing a Web presence ?

Your report must explain how you located useful information (or where you searched if you did not find useful information) in a way in which Southern Papers could check these sources for themselves. Note: You need to explain reasons for your recommendations, and it is quite legitimate to suggest that Southern Papers take no action re involvement in the Internet.

Additional Problems:

*6. Chris Wang is currently undertaking some market research to explore the benefits Southern Papers' might gain from global expansion. Locate and download from the Web statistics on USA paper use and/or production, to assist him in this task.
*7. What other types of information available on the Internet would be of interest and useful to a business such as Southern Papers ?

Tutorial For Web Case 1

What Is The Internet ?

The Internet is a massive, worldwide network of computers that rides on the shoulders of the phone system. The term "internet" literally means "network of networks". It is the largest information resource in the world and is the ultimate experiment in distributed computing. Over 400 million people use the Web worldwide, and in the US about 170 million people use the Web.

Although the Internet appears to be a single network, it is actually an interconnected "web" of thousands of privately owned networks, that have agreed to co-operate and transport information of every conceivable type. Each network on the Internet is connected to one or more other networks, usually over high-speed telephone lines. Each network is therefore able to access every other network. The Internet doesn't own the many telecommunications networks and systems it passes through, and no single entity owns or manages the Internet.

A profile of the Internet (circa 2001)

Key Users	Individuals, business firms, government agencies, academic institutions, research organizations, Community Groups
No. of Users	400 mllion world-wide, 170 million in US
Coverage Area	100 countries have full internet services. 176 countries have at least email access.
No. of Computers (1/96)	over 85 million host computers
No. of Networks	over 300,00 individual networks
Topology	Packet Switching, Routing
Network Protocols	TCP/IP
Main Services	email, WWW, FTP, Telnet NewsGroups, Chat, etc.

Making the Connection

The Internet uses TCP/IP (Transmission Control Protocol/Internet Protocol) as the mechanism for allowing the sharing of resources between computers which co-operate across its many networks. A protocol is simply a set of rules which determines the method by which information may be exchanged between different systems. TCP/IP is a routed, connectionless, packet protocol. What this means is that Internet traffic is divided into unequally sized, individually addressed chunks of data which are then routed through the network over a dynamically assigned path. The Internet uses a number of different algorithms to determine the best route at any time. This method is analogous to sending a friend a postcard every day for a month. While the cards may arrive at their destination out of order or take different routes to get there, the friend (ie: the destination host computer) can sort them out at the other end.

TCP/IP uses two types of addressing protocols. For instance, the address of the Melbourne Business School at the University of Melbourne in Australia can be expressed as either:

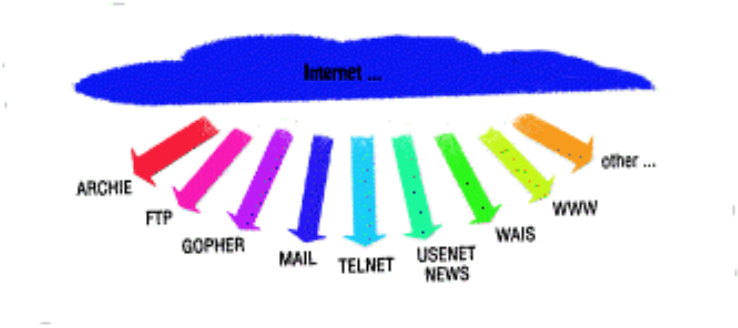
a) Domain Address: **probe.mbs.unimelb.edu.au** or as
b) IP Address: **128.250.180.90**

The domain address identifies **who**, **what** and **where**. The IP address is simply a 32 bit conversion or machine address equivalent of the domain address.

Internet Services

As shown in Figure 6-1, there are a number of different services available through the Internet. These services include electronic mail, interactive conferences, access to information resources, access to data archives, network news, and the ability to transfer files.

Figure 6-1: Internet Services



Source: http://www.srl.rmit.edu.au/pd/surfing/defint.htm

The four most commonly used application services of the Internet are:

Email (electronic mail) - the ability to send and receive messages via computer. When a user sends an email, the file containing the message is forwarded from one computer to another until it reaches its destination. Email comprises around 60% of all Internet traffic, and email only access is still very common with commercial users. Some 45 million email messages are sent via the Internet each day.

FTP (**f**ile **t**ransfer **p**rogram). An Internet feature which is used for the downloading of files stored at remote data archive locations. For FTP to be successful, the user requires a working knowledge of basic Unix commands. The remote archive also requires a valid login name and password before allowing entry to its data store. In many instances, it is possible to download files using an anonymous FTP connection, where login=*anonymous* and password=*guest*.

World Wide Web a client-server based, *graphical interface* to the Internet which supports linked multimedia (color, images, text, video and voice) documents

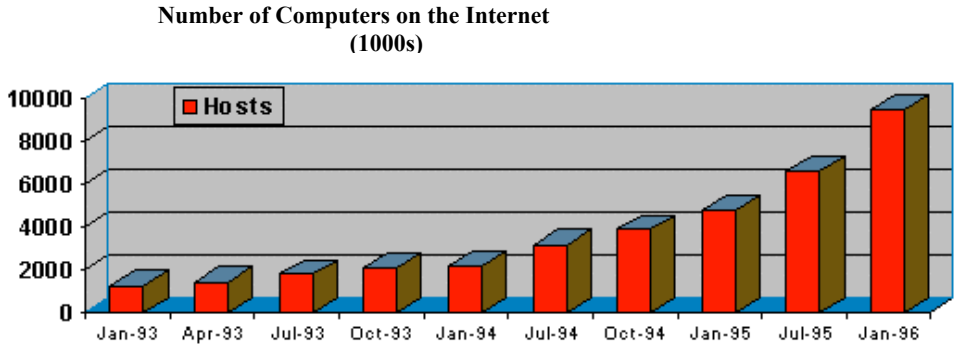
Telnet an Internet feature which enables a user to logon and interrogate databases situated at remote locations. Despite the advent of the World Wide Web, many databases on the Internet are still only accessible through a telnet request.

Growth of the Internet

The Internet was originally created as a research and information exchange tool for academic, government and research organizations. The first electronic mail (email) message was sent in 1969. In the early 1990s, access was extended to commercial organizations and individuals. Now anyone with a computer and a modem can connect to the Internet. Interest in the Internet by commercial firms and private individuals also coincided with the development and release of the Internet's Windows and hypertext based World Wide Web (WWW, W^3 or Web) in late 1992, which provided an easy to use and familiar format. For the first time, users could access Internet resources which displayed color, graphics, sound, video and even animation.

Since 1993, the Internet's growth rate has been explosive (refer Figure 6-2). Current estimates are that the Internet doubles in size every 12 months. The business sector forms one of the strongest growth areas. There are over 450,000 commercial firms currently connected to the Internet, with 1,500 new firms connecting up each day.

Figure 6-2: Growth of the Internet (1993-1996)



The World Wide Web

The World Wide Web and the Internet are not interchangeable terms. The Web is a distributed client-server based service available via the Internet. It refers to a body of information - an abstract space of knowledge - while the Internet refers to the physical side of the network - a giant mass of cables and computers. The Web uses the Internet as a mechanism to transmit hypermedia documents between computer users with a Web connection.

WWW Basic Terms

BROWSER: a software tool which enables users to **view** the resources of the Web. Common browsers are Netscape and Mosaic.

HTML: (HyperText Markup Language) a simple tagging language which is used to generate documents for the World Wide Web. Documents are known as **pages**. These are viewed via a **browser**.

HTTP: (Hyper Text Transport Protocol) the TCP/IP access protocol used for the Web.

SEARCH ENGINE: a software tool used to **search** the resources of the Web **via keywords**. Search engines include Yahoo and AltaVista.

WEB SITE (Web Address, Home Page or Domain Address): the **public entry point** of an organization with a presence on the Web
e.g.: http://www.mbs.unimelb.edu.au

The Web is believed to be largely responsible for the sudden growth of the Internet. Current estimates of Web growth exceed 500 gigabytes of information per month. There are a number of reasons for this. The Web is:

1. *easy to use*. The simple Windows-based approach enables any *newbie* to become a productive user of the Internet within 30 minutes.

2. *Multi protocol*. It also supports other Internet applications such as email, Telnet and FTP.

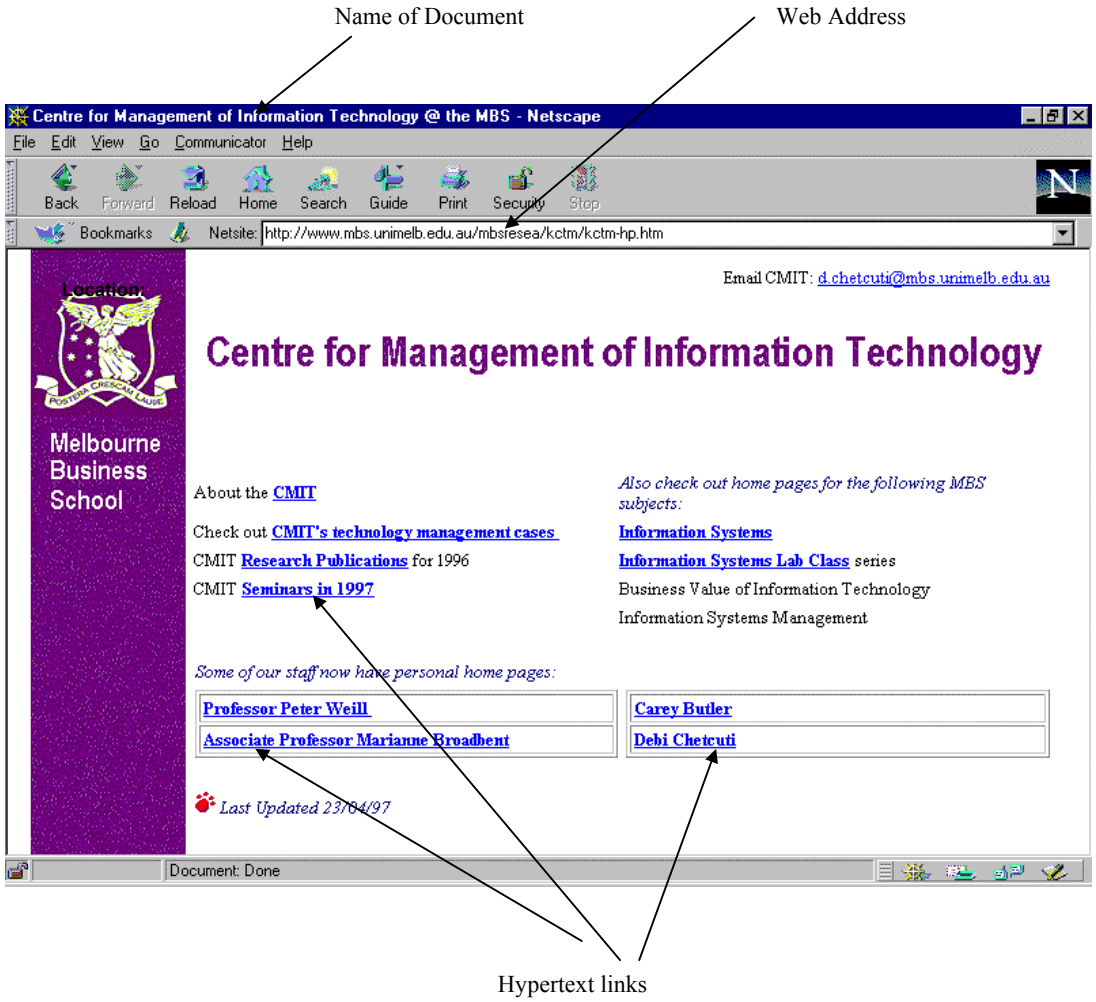
3. *integrative*. It supports hypertext links to other locations within the same Web document, other documents at the same site, or documents at another site entirely.

4. *interactive*. It has *multimedia* presentation (combination of video, audio, still images, text, graphics, animation, etc) capabilities.

Using the World Wide Web

There are literally millions of documents which may be accessed through the World Wide Web. These documents or pages are viewed via a *browser*, and are stored on a *server*. A browser is a piece of software that enables a user to view the resources of the Web by sending requests for documents to a server. For this tutorial we have used the Netscape Communicator browser Version 4.01. A server is simply a computer (or allocated part of a computer) that is dedicated to performing one function - in this instance, storing and retrieving Web documents. Figure 6-3 below uses the Netscape browser to view the contents of a Web document called the Centre for Management of Technology @ the MBS. Let's have a brief look at the salient points of this screen.

Figure 6-3



As with any Windows-based application you will notice the familiar pull down menus, toolbar buttons, and scrollable bars. Take some time to examine and familiarize yourself with these features as they relate to your browser.

The first thing of interest to note on Figure 6-3 is the Net site bar (also known as a Location or URL - Uniform Resource Locator in earlier versions of Netscape) which displays the Web Address of the document: *http://www.mbs.unimelb.edu.au/mbsresea/kctm/kctm-hp.htm*. This tells us where on the Internet the KCTM Home Page is located. The segments comprising this address are decoded as follows:

http://	This is a Web address (as indicated by *http* – TCP/IP's hypertext transfer protocol)
www	located on a World Wide Web server
mbs	of the Melbourne Business School
unimelb	which is part of the University of Melbourne

edu	which is an educational institution
au	in Australia.
mbsresea	This is a directory on the MBS web site
kctm	and this is a subdirectory within the *mbsresea* directory
kctm-hp.htm	which contains a document called *kctm-hp.htm*

All web addresses follow a similar construct. Let's look some more at just the institutional part of the address.

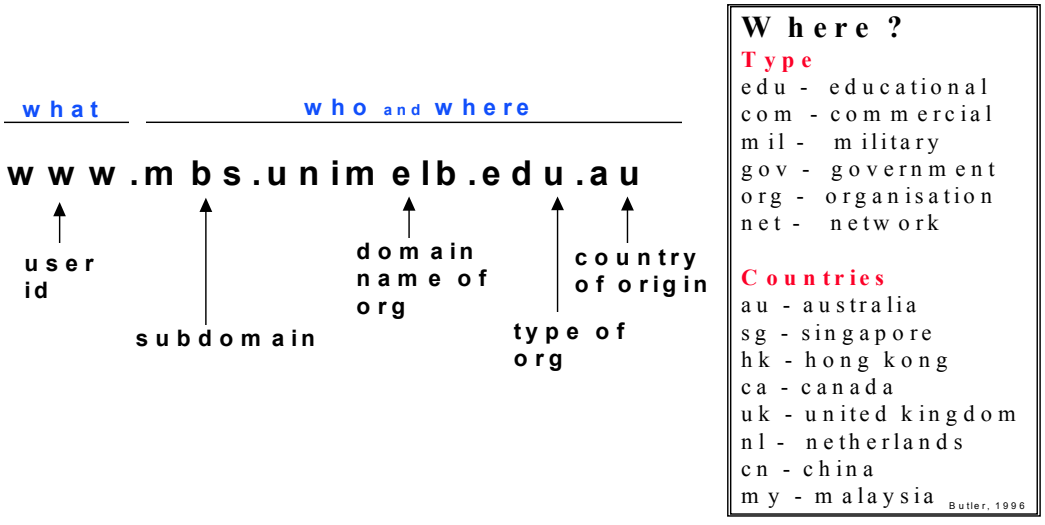
www.mbs.unimelb.edu.au

Just like any address you may write on an envelope, the address moves (left to right) from the specific to the general.

www.mbs	The WWW server at the MBS	(*most specific*)
unimelb	The University of Melbourne has many web addresses	
edu	There are many Australian educational institutions on the Web	
au	There are many Web addresses in Australia	(*most general*)

This is sometimes known as domain addressing. *Figure 4* provides further illustration of how this works. The least specific parts of an address are the last two segments - the organisational type and the country. Figure 6-4 lists a sampling of these. Similarly, most web addresses need a country code. The general exception is the United States. Web addresses for the USA usually omit the country code.

Figure 6-4: Web Addressing



Exercise 1

Try decoding the following addresses for their organizational type and country of origin:

a) *http://www.harvard.edu*
b) *http://www.vicnet.net.au*
c) *http://www.oup.co.uk*
d) *http://www.wellsfargo.com*

Navigating the Web

Now let's go back to Figure 6-3. The second thing to note on this screen is the reference *to Hypertext Links*. Hypertext is the same as regular text (it can be viewed, searched, saved or edited) with one important exception: hypertext contains embedded connections within the text to other documents. These are also sometimes known as *Hot Links* or *HyperLinks*, and will generally appear on your Netscape browser screen as blue or red underlined text, or form a blue or red frame around a graphic. Either indicates an active item. Clicking with your mouse on the active item of your choice, will take you to either another location within the same Web document, another Web document at the same site, or a document at another site entirely.

Hyperlinks are the basic navigation tool of the Web. Generally each document which appears on your screen will itself have links and connections to other documents again. Continually selecting hyperlinks takes you on a free-associative tour of information. In this way hyperlinks can create a complex virtual web of connections.

Exercise 2

Let's visit three of the Web sites shown in Exercise 1:

a)	Harvard University	*http://www.harvard.edu*
b)	VicNet	*http://www.vicnet.net.au*
c)	Oxford University Press	*http://www.oup.co.uk*

Type each address in turn into the Netsite bar (refer *Figure 3*), of your browser and then press *Enter* to go to that address. Browse each site using the HyperLinks provided.

Exercise 3

A more recent development for navigating within Web documents is the *image map*. This refers to another category of active item graphics. This approach is well illustrated by the Security First Network Bank (SFNB) web site. Type the address into the Location bar of your browser and then press *Enter.*

d)	Security First Network Bank	*http://www.sfnb.com*

The SFNB home page contains an interactive graphical image. Move your mouse over different areas of the graphic. Notice how the mouse pointer changes to a pointing hand symbol - ☝. This indicates a clickable item which works in exactly the same way as a HyperLink. Notice also that you are given a short description of the function for each of these clickable items as you move your mouse around the graphic.

Search Engines on the Web

Search Engines are software programs that enable users of the Web to locate documents of interest through keyword searching. Most of them also allow the searcher to use Boolean logic (AND, OR, NOT operators) to refine searches It is estimated that there are around 900 search engines available for use on the Web. *Free-to-user* search engines are accessed by simply typing in the relevant Web address, entering text into the search bar, and then clicking on the search button. Some of the most popular *free-to-user* search engines include:

WORD-ORIENTED SEARCH ENGINES

AltaVista
http://www.altavista.com

Web Crawler
http://www.webcrawler.com

Lycos
http://www.lycos.com

Excite
http://www.excite.com

SUBJECT-ORIENTED SEARCH ENGINES

Yahoo
http://yahoo.com

Each of these search tools displays results in very different ways. *Alta Vista* is true keyword searching of document titles and text within documents, and is possibly the most comprehensive search engine of its type. *Yahoo!* is thesaurus-based, and presents results sorted into subject categories. *Web Crawler, Excite* and *Lycos* present results as lists of documents in ranked in order of keyword relevancy.

***Note**: Netscape Communicator4.01 users also can select EDIT/SEARCH INTERNET from the pull down menu to access a selection of the most popular search engines. Users of earlier versions of the Netscape browser (i.e.: Versions 2.x and 3.x) can select DIRECTORY/INTERNET SEARCH from the pull down menu to achieve the same function.*

Exercise 4

With so many search engines available, it can be difficult to decide which is the best one to use. One of the ways you can do this is to use a number of different ones to search for the same text, and then evaluate your results.

The Australian city of Sydney will be hosting the Olympic Games in the year 2000.
Evaluate each of the five search engines listed above using the words **Sydney Olympic Games.** Browse through the results and compare the differences.

Sites of Interest on the Web

The following web addresses provide further information on the Internet and the World Wide Web. Also included are some introductory business-oriented sites.

John December's Guide to the Internet	http://www.december.com/web/text/index.html
WhatIs ? - a guide to Internet terminology	http://www.whatis.com/
Internet Society	http://info.isoc.org/
Internet Statistics	http://lcweb.loc.gov/global/internet/inet-stats.html
Internet Tutorial	http://www.waisman.wisc.edu/~colantonio/aauap/int01.htm
A Business Researcher's Interests	http://www.brint.com
The MBA Page	http://www.cob.ohio-state.edu/dept/fin/mba.htm
Yahoo's Directory of BSchools	http://www.yahoo.com/Business/Business_Schools/

World Wide Web Case 2

Grapelli Grapes

Problem:	Locating information on the Web
Web Skills:	Advanced Web Searching Downloading graphics from the Web Creating bookmark files

Steven Grapelli looked anxiously at the darkening sky as he tended the well-ordered rows of grapevines of his vineyard. Grapelli Grapes is a medium-sized vineyard of some 1000 acres, located in the Fresno region of California. The Fresno area is California's premier agricultural producing county, with more than 200 commercially produced crops from grapes (for wine, raisins and the fresh market) and cotton to kiwis, pomegranates and a wide variety of specialty vegetables. Each year, Grapelli produced a small amount of premium white wine (primarily Riesling and Chardonnay), which was highly valued by wine buffs around the world. Grapelli's main business however, came from growing and offering for sale: fresh grapes for the table, partially dried grape products such as sultanas and raisins, and grapes for the purpose of producing grape juice.

Grapelli hurried back to his office and sat down at his personal computer. He logged on to the Internet's World Wide Web (the Web), and then using the bookmarks file of his Web browser, quickly connected to a database on the University of California's (Davis) Web site to look at weather reports for his area. Concerned about an approaching rainstorm, he again consulted his bookmarks file and this time connected to a national weather database for the Western region of the United States. Grapelli then printed off a weather map generated by a satellite less than an hour earlier. He was relieved to see that the rain would pass to the North.

Two hundred miles away in Sacramento, Enrico Santos, manager of the All Pure fresh produce store, consulted a different database on a different Web site at U.C. Davis that tracked the availability of table grapes from California, Arizona, and Mexico. Another option on the database supplied prices for different varieties of grapes. As he drank from his carton of Grapelli grape juice, he noted that Arizona supplies were very low, and that the price difference between the imported Mexican and the California grapes wasn't enough to justify the longer shipment time. Using his email application, Santos then sent an order for Californian grapes to his wholesale supplier.

Back at work among his lines of grapevines, Steven Grapelli frowned in concern at the appearance of a white powdery substance on the leaves of some of his plants. In his 15 years as a grape grower, his vines had been remarkably disease free. He wondered if the affected vines were suffering from powdery mildew – an ectoparasitic fungal disease known to attack grapes. When he returned to his office for lunch, Grapelli again connected to the Web. He wanted to locate information on powdery mildew as it affected grapes, and how he should treat this condition. Grapelli also wanted to locate and download a graphic of an affected leaf so he could see whether it was the disease he suspected.

In an industrial park in San Francisco, a molecular biologist at a small biotechnology company was putting the finishing touches to documentation for a patent application on a process for the obtention of must on bunches of grapes. She then sent it via electronic mail to the company's patent attorney in Washington DC. She sat back to relax, pulled a bunch of fresh Grapelli table grapes from her lunch sack, and checked the weather maps on the Web's national weather database: *"Hey, it looks like it's going to rain"*, she remarked to no one in particular. A co-worker threw open the door of the windowless room. Across the hallway, heavy raindrops splattered against the window.

Seconds later the patent application documentation arrived in the attorney's online mailbox. He skimmed the text, forwarded a copy to his clerk, and saved the message for his own use. Something in the application sparked a recollection in his memory. As he munched on a handful of Grapelli raisins, he connected to a US Patents database on the Web. Seconds later he was keyword searching through a list of recent US Patents.

Later that day Grapelli once more sat down in front of his computer and logged on to to the Web. As he sipped on a chilled glass of Grapelli Chardonnay, he connected to his favorite search engine. He was due to leave for Australia next week on a business trip to investigate the vineyards of South Australia's Barossa Valley, Australia's best-known wine-growing district. Grapelli was hoping to locate a list of the vineyards in that region, and possibly also a map of the area.

Tasks

This case has six multi part questions. As you locate answers to each of the questions, you should save the Web addresses in the bookmarks file you will create in Question 1.

1. Create a bookmarks file called GRAPELLI.HTM and save this file to floppy disk. Save all your answers (ie: web addresses and graphics) to the following questions in this bookmarks file.

2. Using the Web, locate the weather databases in California and Illinois. Now download and save a weather map graphic onto floppy disk.

3. Locate a database on the Web that allows the tracking of grape prices and their availability.

4. Using the Web, locate information on powdery mildew as it affects grapes. Now locate, download and save to disk, a graphic of powdery mildew on grapes.

5. Locate a US Patents database on the Web, and search it to find a recent patent on *obtention of must on bunches of grapes*. See if you can also find a full description of the patent and a diagram of the invention.

6. Using the Web, locate and print out a listing of wineries in Australia's Barossa Valley region. Now locate, download, and save to disk, a map of the Barossa Valley area.

Time Estimates (excluding task marked with *):

Expert: 2 hours
Intermediate: 3 hours
Novice: 4+ hours

Tutorial For Web Case 2

Creating your own Bookmarks/Favorites File for use on the Web

Bookmarks are simply a shortcut way of accessing frequently used Web addresses. Bookmarks (used by Netscape and many other browsers) or Favorites (used by Microsoft Internet Explorer) files are a feature of most Web browsers. This feature enables users to :

- save frequently used or interesting Web addresses to a specific HTM/HTML file by simply selecting an *Add Bookmark* (or similar command) option from the browser menu. The browser will then save and record the path to the selected address;

- rapidly return to a chosen location without having to retype the Web address each time, by firstly selecting the *Go to Bookmarks* (or similar command) option from the browser menu, and then selecting the title of the saved address

Most browsers will also allow deletion and editing of book marked addresses. Here's how to create and use your own Bookmarks file:

1. Insert a floppy disk into drive A:\
2. Now open your word processing application.

If you are using **Word for Office 1997**, do the following:

- create a new document using FILE/NEW and then click OK

- save the document according to the following steps:
  - from the menu, select FILE/SAVE AS HTML. HTML is the language which underlies most documents on the Web, and is readily understood by your browser
  - give the file a name e.g.: GRAPELLI.HTM and make sure the drive path is set to A:\., and then click on the Save button.
- Now exit from Word

If you are using an **earlier version of Word or another word processor** which does not have an automatic HTML saving option, do the following:
- create a new document using FILE/NEW and then click OK
- save the document according to the following steps:
  - from the menu, select FILE/SAVE AS and within the Save As dialog box, change the File Type to **Text**
  - give the new file a name e.g.: GRAPELLI.HTM (the extension must be .htm (not .doc or .txt) in order for your browser to recognize it). Make sure the drive path is set to A:\, and then Save the file.

- Now exit from your word processor

You have now created an empty bookmark file.

Using your Bookmark File

These instructions are based on the Netscape browser. If you are using a different browser, your Instructor will assist:

1. Open your Web browser
2. Select Bookmarks/Go to Bookmarks from the menu
3. From the File menu, select Open
4. Change the drive path to A:\. Click and highlight the file you have just created and click OK (this will make your file the active default for bookmarks).
5. Click outside the Bookmarks window to return to the main Netscape browser screen.

   Browse/surf the Web in the usual way. As you come across hyperlinks that you would like to save to your bookmarks file:

6. Click on the link to highlight, and then select Bookmarks/Add Bookmark from the menu to append the link to your bookmarks file.
7. Netscape should automatically save the added link.

Accessing Saved Bookmarks

1. Follow steps 1-4 in the previous section.
2. Double click on the wanted bookmark to go to that location.

   Happy Surfing !!!!

Downloading Image Files (*.gif, *.jpeg, etc) from the Web

The Web contains a huge and wonderful range of graphical images most of which can be very simply captured, saved to disk, and then used to enhance your own Web pages or even PowerPoint presentations or Word documents. If you have downloaded a graphic from a commercial Web site, please note that it may be proprietary artwork, and you should therefore acknowledge the source of the graphic (e.g.: the Web address you located the graphic at) in a footnote or something similar.

Here's how to do it:

1. Using your Web browser, locate a Web page that contains a graphic you wish to capture to disk. Select your graphic by:

   - pointing to the graphic with your mouse; and then
   - clicking the **right mouse** button

2. This will bring up a short menu allowing you to save the graphic to disk. from the menu. Select the *Save This Image As* .... option, choose your drive path (e.g.: A:\), and then click OK to save.

3. The graphic can now be used directly to enhance Web pages, PowerPoint presentations or Word documents.

Using Graphics Files In Microsoft PowerPoint

1. Open the PowerPoint application

2. Open a new, blank presentation

3. Select *Insert/Picture* from the menu:
   - choose your drive path (e.g.: A:\)
   - double click on the filename of the graphic

4. PowerPoint inserts the graphic into the presentation. The graphic can now be used as part of a PowerPoint presentation or slideshow (select *View/Slideshow* from the menu to active this latter option)

Using Graphics Files In Microsoft Word

1. Open the Microsoft Word application

2. (a) Open A New Document (e.g.: File/New from the menu) or (b) an existing document that would be enhanced by inclusion of the graphic (in the case of (b), you will need to choose the point of insertion before proceeding to Step 3)

3. Select Insert/Picture from the menu:
   - choose your drive path (e.g.: A:\)
   - double click on the filename of the graphic

4. Word inserts the graphic into the document.

World Wide Web Case 3

Green Thumb Gardening Supplies

Problem: Locating information and investigating the feasibility of electronic commerce using the Internet's World Wide Web and other Internet-based computing models

PC skills: Advanced Web Searching

Green Thumb is one of Australia's largest gardening supply wholesalers, and plant mail order firms. The company supplies and distributes a range of fertilizers, potting mixes, gardening tools, trees, and plant seedlings to some 2,000 commercial gardening outlets, landscaping firms, hardware stores, and supermarkets around Australia. It also has around 500 mail order customers who regularly purchase plants (roses and spring bulbs) and seeds (herbs and vegetables). Green Thumb has established exclusive agreements and strategic partnerships with 45 of Australia's best plant growers and garden products suppliers. These relationships allow Green Thumb access to a very large inventory of quality products, and has gained the company a solid reputation for being able to supply rare or unusual plants. The phrase: *"Get it at Green Thumb"* is frequently used by Green Thumb's many customers.

Green Thumb is owned and managed by Declan Corgi. The gardening supply industry is not renowned for being high-tech, but Corgi is proud that his company has used technology to automate the firm's ordering, inventory, and financial processes. To run his business, Corgi relies on his eight Pentium PC's which are linked in a local area network (LAN). Spreadsheet, database, and word processing software is used to do accounts, correspondence, produce product catalogues, and keep track of stock and orders. Orders are usually received by fax, phone or post. Corgi advertises his business through occasional line advertisements in gardening journals such as the *Nurseryman* and *Your Garden*.

Corgi recently attended a seminar at the Melbourne Business School, on *Business Use of the Internet*, and he is interested in the possibilities it could offer for his type of business, particularly as an alternative selling and distribution channel. He also wondered whether the Internet could be used as an avenue for gathering intelligence about his market and competitors.

At the seminar, Corgi learned that there were four main types of Web sites commonly used by businesses with a presence on the Internet's World Wide Web:

- promotional or billboard. This type of site was usually static in nature. It often included large graphical images, but had little information content or value. Often this type of site represented a company's first *entre* on the Web.
- informational. This site type often contained a rich variety of information on the company and its various products or services, and allowed some interactivity with the user. It did not have transactional capabilities.
- transactional. These sites allowed the purchase and payment of goods and/or services. Usually a variety of payment mechanisms were offered.

- intelligent agent. This type of site contained programs that supported limited independence of action. Typically an agent program permitted the user to personalize a profile of their interests. Other types of agents included site watchers that informed the user when a specific Web site or a variety of sites had been updated. This latter capability is sometimes referred to as push technology.

These categories were not mutually exclusive (ie: it was possible for a company's Web site to have concurrent informational, transactional and intelligent agent capabilities).

Corgi wondered whether a Green Thumb Web site could be used to further enhance relationships with his customers and suppliers, and possibly to increase business. Several of his larger *customers* had already inquired whether his company is on the Web, and he knew that a number of his *suppliers* already were. In addition, Green Thumb had a sales force of 25 staff. These individuals were largely field-based, and spent their time visiting customer sites around Australia, taking orders for Green Thumb products. Corgi noticed that there was often a considerable delay between when an order was placed by a customer, and when a sales rep actually phoned or faxed it in. Typically, sales reps tended to accumulate orders over a two or three day period, before phoning or faxing them through to Green Thumb in a single batch. Corgi wondered whether involvement with the Internet might assist in reducing order receipt delays.

At the seminar he attended, Corgi also heard about Intranets and Extranets. He knows that a number of companies are using them, but he is not really sure about what the terms actually mean, and how they are actually used, how they fit in with Web sites, and what opportunities and/or advantages they might offer for his business.

Tasks

You have been contracted by Corgi to advise on the suitability and appropriateness of using the Internet in the Green Thumb business. Your task is to prepare a report for Green Thumb recommending a future course of action for involvement with the Internet. Your report should include answers for at least the following:

1. (a) How do businesses use the Web and why? (b) Is anyone else doing this in Green Thumb's particular business area (i.e. the Australian gardening supply industry)?

2. Use the Web to locate an example of each of the four types of Web sites (i.e. promotional, informational, transactional, and intelligent agent) that businesses may use.

3. Now locate an example of each of the four types of Web sites in Green Thumb's industry. Note: the sites you locate should be Australian sites (i.e. xx.com.au). Thinking back to Question 1b, do any of these companies represent threats to Green Thumb ? Why or why not ?

4. Should Green Thumb have a Web site ? If so, what type of Web site would you recommend, and what benefit would the company derive ? If not, explain why.

5. What is an Intranet, and what is an Extranet ? Provide a brief explanation of what these terms mean, how they relate to each other, and how they relate to a Web site. Now locate on the Web and download to disk a graphical depiction of (a) an Intranet and (b) an Extranet. Paste these into your report.

6. Would you recommend an Intranet and/or Extranet solution for Green Thumb ? If so, what benefit would the company derive ? If not, explain why.

Your report must explain how you located useful information (or where/how you searched if you did not find useful information) in a way in which Corgi could check these sources for himself. This should be done in a bookmark/favorite file. Note: You need to explain reasons for your recommendations, and it is quite legitimate to suggest Corgi take no action re involvement in the Internet.