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.

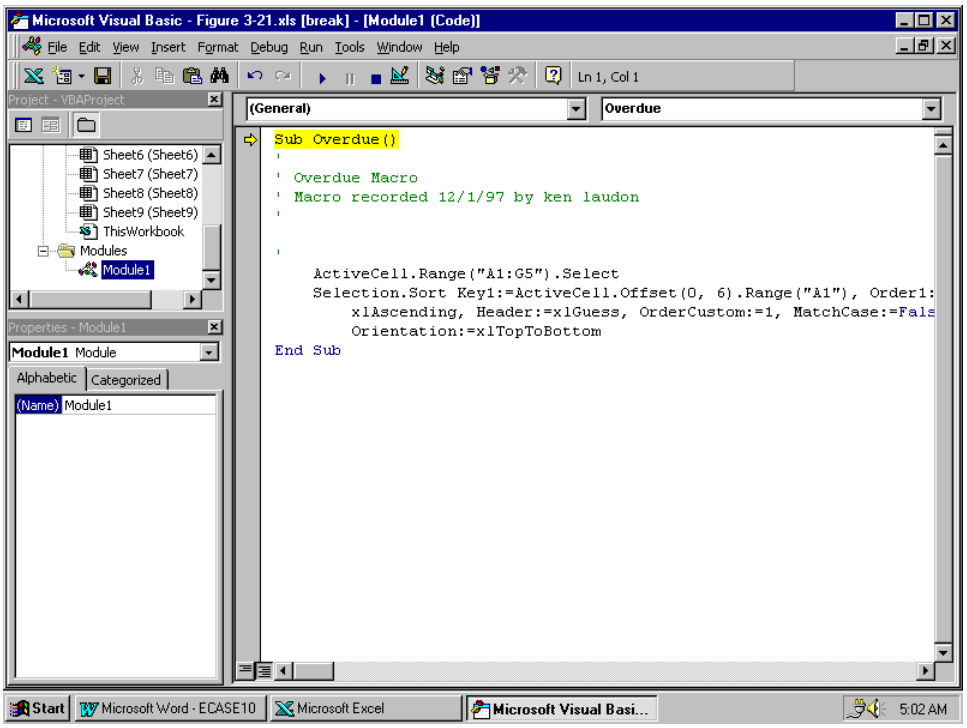
Viewing the Macro

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