Spreadsheet or relational database software: which to choose?

by James D. Linn

The modern clinical lab is in a constant state of change. In my laboratory, the direction of these changes is clear and consistent: do more with less.

A major source of savings for a clinical service is to move as much of the information management as possible into the lab and do it with lab people instead of high-paid programmers.

Getting acquainted. Databases are among the most important software tools for gaining access to and control of corporate and lab information. They also afford lab workers a way to use their computer expertise and to do tasks formerly done by LIS staff. But people are frequently reluctant to work with them because of their reputation for requiring programming. Fortunately, several powerful yet easy-to-use database programs have appeared recently.

First of all, a database is simply a collection of data organized so it can be easily manipulated. A dictionary is an example of a database. The words and their definitions are organized in alphabetical order to make locating them relatively easy.

Databases can be classified into two general types: flat file or relational. For a discussion of the differences, see Computer Dialog in the November 1994 issue of MLO.

What to consider. If you are contemplating acquiring a database tool, consider the following:

- Size of your information database. For small lists, having the information alphabetized on a sheet of paper is sufficient, but as your information mass grows, having the data in a spreadsheet or database becomes indispensable.
- Multiple ways to access information. If an information bank is organized in only one way and you don't have the information used in the organizing scheme, the mass of data becomes useless. How many times have you opened your dictionary to find out how to spell a word and stared at the book in dull resentment because you couldn't spell the word well enough to find it? Ideally you should be able to search a database and find what you want with whatever information you have.

Spreadsheet or relational database? Your choice of database software depends on the job at hand. Spreadsheet programs such as Excel (Microsoft Corp., Redmond, Wash.), Lotus 1-2-3 (Lotus Development Corp. Cambridge, Mass.), and Quattro Pro (Borland international Inc., Scotts Valley, Calif.) come out of the box with an extensive array of database tools. They are powerful and easy to use and allow you to see all your data lined up neatly in front of you in rows and columns. Data navigation and modification is usually much easier in a spreadsheet than in a database. For small databases that will be used by few people, they should be considered.

Spreadsheet programs, however, were designed primarily for number manipulation and calculations. If your needs exceed this, consider a full-fledged, relational database program. The three relational database packages with which I am familiar are FoxPro (Microsoft Corp.),

1 Medical Laboratory Observer, Feb 1995
<http://findarticles.com/p/articles/mi_m3230/is_n2_v27/ai_16581759/?tag=content;col1>
Microsoft Access (Microsoft Corp.), and Paradox (Borland International Inc., Scotts Valley, Calif.). All allow you to point and click your way through without programming.

**Data entry:** With any database tool (spreadsheet or relational database), most of the time and effort is invested in data entry. Once the data are entered, you will be reluctant to start over. So, when buying a database package, make sure it can export its tables in dBase format. dBase is the industry standard; all major database, spreadsheet, and word processing software can read it. If you become dissatisfied with your software, you can export your valuable information directly into a better product.

Unlike spreadsheet programs, database software is designed for one purpose: collecting, organizing, and storing data. So data entry into a database is generally easier and more controlled than with a spreadsheet program. And relational database software allows you to establish controls that can help prevent data entry errors.

For example, one problem with typing in data is that people can use different words for the same result. A negative result can be entered as "Neg," "N," "Negative," and "-." The computer doesn't understand such anarchy; when you ask it to find all "negative" results, all it shows you are those entered as "negative."

Databases circumvent this problem with list boxes. These boxes pop up on the screen and contain a list of entries that the data enterer can choose from, assuring, for example, that all negative responses are entered identically.

Databases also allow you to set up default entries for a result. Suppose the most common test result is "negative." When the data enterer reaches the test result field, "negative" pops up highlighted. If the patient is negative, the data enterer just tabs over to the next field; no entry is needed. If the patient is positive, the data enterer changes the default choice appropriately.

Furthermore, you can format the field so that information appears in the database in a consistent manner no matter how it is entered. For example, a "Last name" field can be formatted with the first letter capitalized and the rest lowercase. "smith," "sMITH," "SmiTH," and "SMITH" all go into the database as "Smith."

Lastly, you can set up checks on entered information to ensure that it is reasonable. For example, you can program the software to refuse letters where there should be numbers, draw dates in the future, and values less than the lower limit that the test can measure.

**Sorting and reporting data.** Once the information has been entered, it's time to sort, query, report, and number crunch. Now spreadsheets and relational databases become collaborators, not competitors. Data can flow easily back and forth from spreadsheet to database program. Database software can import spreadsheets and export data as a spreadsheet file.

For sorting and querying, it is usually much faster to use a relational database program due to their ability to index. For example, you will probably want to index patient data by last name so that everyone appears in alphabetical order. Other useful indices are draw and ordering dates. You can index on as many fields as you desire.
My prostate-specific antigen (PSA) database contains some 10,000 records with four fields per record. When my PSA data were in spreadsheet form in Excel, it took 1 to 45 seconds to find a name, depending on how far the name was from the starting point. Now, if I index on the name field and do the search in Microsoft Access, it takes less than 1 second no matter how far down a name is in the database. If you need to ask "what if" questions or apply high-level statistics or financial calculations to the numbers, however, you'll want to use spreadsheet software.

**Manipulating data.** When it comes to manipulating data and looking at information in different ways, relational databases have a tremendous advantage over spreadsheets because manipulation does not involve rearranging the data in the underlying table. When you sort a spreadsheet, the cells in the spreadsheet are actually shuffled, which can have unpleasant effects on the information. For example, if you sort a spreadsheet by name but forget to include the ranges with the birth dates before you issue the sort command, the spreadsheet will alphabetize the names and leave the birth dates untouched. The result: John Smith is lined up with Nancy Jones's birth date.

Relational databases, on the other hand, organize files in memory and present them to you in the order you have specified while not disturbing the underlying table. This makes the process faster and maintains the integrity of your data.

In addition, relational databases can quickly extract data from tables. For example, you suddenly wonder if there is a correlation between high PSA levels and immunosuppression therapy, so you request the PSA level on everyone who has had cyclosporin levels measured (i.e., the people in the intersection of the PSA and cyclosporin tables). Then you can look at PSA levels on men who have not had a cyclosporin measurement (the difference between the two tables).

My PSA reports are now produced on relational database software and look as good as if they were produced on a word processing program. Instead of manually typing each patient name, the database retrieves the name from the name field and puts it in the proper place on the report. Same with date of birth, latest draw date, doctor, and clinic. The program pulls the previous specimen dates and PSA results from wherever they reside in the database, organizes them on the report in order of draw date, graphs the results, puts the graph in the right place, prints the report, and goes on to the next patient. The whole process takes less than a minute and happens unattended and automatically.