WREP 108 / August 1988

75¢

Selecting a Database Management System *for Agricultural Record Keeping*

T.L. Cross and R.J. Lane

Many agricultural producers are using computerized database management systems to maintain and analyze their records. Very powerful database management systems now available for desktop computers are used in a wide variety of agricultural applications. Machinery, crop, livestock, and payroll records are all commonly maintained with database management systems.

The success (or failure!) of your computerized agricultural record-keeping activities may depend in part on the database management system you use to keep your records. Choosing the system most appropriate for your needs can be quite a challenge, given the number of programs currently on the market.

If you're not familiar with database management systems, you should read WREP 107, *Introduction to Database Management Systems for Agricultural Record Keeping*. It explains the basic terms and concepts associated with using a database management system. We assume you have mastered this introductory material—and are now ready to select a system.

This publication will present several hardware and software factors to consider in selecting a database management system for your agricultural record-keeping needs. After examining each of these factors, use the checklist (page 7) to evaluate and select a database management system.

Throughout this publication we'll use the acronym *DBMS* (*D*ata*B*ase *M*anagement *S*ystem) to refer to a microcomputer-based database management system. We classify DBMS's according to the number of files they can open at one time. File management systems are limited to one open file, while relational database managers may open two or more files concurrently.

At this point you may think, "I already own a DBMS. Why should I read about selecting a new one?" There are two reasons why this publication will be useful to every reader, even if you own a DBMS already.

First of all, the DBMS you own now is probably not the same one you'll use forever. At some time you'll want to buy a new one. We can help you decide which DBMS is right for your situation.

Second, you may choose to upgrade your DBMS to a new release. We'll help you decide if it's worthwhile to upgrade—or if you need a new DBMS.

No matter what your situation, the time to select a DBMS is *before* you invest a great deal of time and effort collecting your records. Ideally, you'd choose a DBMS before buying any computer hardware. In this way, you're sure to buy compatible software and hardware. When you select a DBMS, consider the computer hardware factors in the next five sections.

Hardware factors to consider

If you've purchased computer software, you know that it can be a frustrating experience. Different brands of computers and disk operating systems make choosing software a risky activity. While this publication is not a guide to buying hardware and software, we'll present a few general factors to consider in selecting a DBMS.

Operating systems

Above all, make sure that the DBMS you're considering will run on your operating system. Most DBMS's will state which operating systems they support (for example, MS-DOS version 3.1 or OS/2 version 1.0).

Timothy L. Cross, Extension economist, Oregon State University, and *Ronald J. Lane,* doctor of veterinary medicine candidate, University of Pennsylvania.

A Western Regional Extension Publication

The version of your operating system can be as important as the operating system brand. Software designed to run on MS-DOS version 3.1 may not work on MS-DOS version 2.1.

In addition, make sure the program is distributed on the same disk format as your computer uses. An MS-DOS-based DBMS distributed on a 5¼-inch floppy disk won't be worth much if you have 3½-inch disk drives, even if you have the correct MS-DOS operating system.

Memory capacity

The next hardware factor to consider is the number of kilobytes (KB) of Random Access Memory (RAM) in your system. One kilobyte of RAM will store approximately 1,000 characters. The majority of today's farm computers have 256 KB or more of RAM. As you work with your computer and software, most of the activity takes place in RAM.

Every software package requires a minimum amount of RAM to operate properly, so you must make sure that you have the minimum RAM the program requires. More RAM in a computer is always better than less—so don't worry about having too much.

Disk storage capacity

We briefly mentioned the importance of disk drives in regard to disk format, but other aspects of disk drives influence the choice of a DBMS. The most common microcomputer disk drive uses a 5¼-inch floppy disk and stores 100 to 400 KB of data.

Fixed (or hard) disk drives are available that store 5 to 70 megabytes (MB) or more of data (1 megabyte is about 1 million characters). Floppy disk drives tend to be slower and store less data than fixed disks, but they're less expensive.

Typical computer disk drive configurations include one fixed disk and one floppy disk, or two floppy disks. If you have a fixed disk, make sure the DBMS will operate on it. If you have a single floppy disk, check to see if the DBMS and your data will fit on *one* disk.

To estimate the amount of disk storage you'll need each year, multiply the estimated characters per record you'll keep by the number of records you'll maintain in 1 year. Next, multiply the result by the number of years of records you'd like to keep on-line or accessible. (Production of a 7-year summary report may require 7 years of records.)

This value provides an estimate of the amount of data disk storage you'll need over time. If the final calculated value is large (say, greater than 400 KB), you'll need a fixed disk drive.

For example, suppose you collect a maximum of 500 characters of data per cow and you own 100 cows. This results in 50,000 characters or about 50 KB of data collected per year. If the same amount of data is collected for each calf and every cow has a calf, then 100 KB of data would be stored per year.

Now suppose you'd like to examine your herd's performance over a 5-year period. This would require storing 5 years of records at 100 KB per year for a total of about 500 KB of data. This presents no problems to users with fixed disk drives, but 500 KB of data may be too large for users of computers with a single floppy disk drive or even dual floppy disk drives.

Printer support

Your printer is another important consideration in selecting a DBMS. Printers vary by width, speed, type of printing, and paper feed. DBMS's vary in their support of the many printers available. Make certain that the DBMS you select supports as many of your printer's features as possible.

Agricultural records tend to generate long, multicolumn reports that can be demanding on your printer. An ideal printer for agricultural records prints rapidly on 15-inch-wide paper, using a tractor feed mechanism to maintain a smooth paper flow.

It should also offer condensed print, bold, and underline features. The DBMS you select should take advantage of this printer's features by allowing you to build wide, custom reports with emphasized text as desired.

Monitor support

A final hardware factor to consider is the monitor and monitor adapter you'll use with your system. If you have a color monitor, make sure the DBMS you select will support it. It's frustrating to view dull, drab livestock records on your top-dollar, high-resolution color monitor. And if the DBMS you choose uses any type of high-resolution graphic displays, make certain your monitor adapter is compatible with it.

Your computer's operating system, RAM, disk drives, printer, and monitor will help to narrow the list of DBMS's you have to choose from. However, you'll likely find several programs compatible with your system. In order to narrow the choices to one, consider the following software factors.

Software factors to consider

The importance of any particular software feature (or lack of a feature) can be determined only on an individual basis. This section will point out desirable software features from an agricultural record-keeping standpoint, and explain why they're desirable.

Technical specifications

The technical specifications of a DBMS determine what constraints you face as you work with the program and your records. Ideally, a DBMS imposes as few constraints as possible without sacrificing speed of execution.

Over a period of years, your databases will grow. As they grow, the time required by your DBMS to add, edit, calculate, and report your records also will increase. Thus, speed of execution is a concern in selecting a DBMS.

Almost all DBMS's place a limit on the maximum number of fields you can include in one database. This can be a problem in working with file management DBMS's, since they allow only one file to be accessed at a time. All the data you collect must fit into the fields of one database file, which limits the total amount of data you can collect for each record.

This constraint is less important in a relational DBMS, which can access multiple files at the same time. If you run out of available fields in one database, you can simply create a second database linked on a key field with the first database and add the additional fields you need.

A good rule of thumb is to select a DBMS that allows at least 30 fields per database. Many systems allow 100 or more fields, which is far more than most agricultural record keeping will need.

You should note, however, that the number of data fields you store will grow as your record keeping becomes more sophisticated. Once again, more is better than less when it comes to maximum numbers of fields.

DBMS's also impose a limit on the maximum number of characters that can be stored in each field. Agricultural record keeping is concerned with identification codes, weights, values, names, and dates. This data will usually be from 1 to 30 characters in length, which is far less than the limit imposed by most DBMS's on the maximum characters per field.

Some data you'll collect is rather lengthy, though, and you should consider this when selecting a DBMS. Textual information, such as comments about animal health or soil characteristics, may require fields that store 80 characters of data or more.

If a DBMS has a remark or memo field, you can ignore the maximum number of characters limit. You can store long strings of characters in these remark fields, and you can plug your remaining data into appropriate numeric, character, and date fields.

Without a remark field, you'll have to use long character fields to store text entries. For additional points on data field types, see "Flexibility" (below).

Our next technical specification is the maximum number of characters per record. A DBMS may allow a large number of fields and characters per field—but then impose a limit on the maximum number of characters stored with each record. A limit of less than 1,000 characters per record could cause problems in storing your agricultural records.

Determine the maximum number of characters per record you require by estimating the number of fields you'll use and the length of each. Add the lengths of each field to obtain the number of characters you require for each record, and allow for future expansion of your record keeping.

Some DBMS's limit the total number of records stored in one database file. Ideally, the number of records in a file is limited only by your disk storage capacity.

If the DBMS you're considering imposes a lower limit, then you must estimate the number of records in each database file you maintain and compare this to the DBMS limit.

A final technical consideration is the maximum number of files that you can access simultaneously. This concerns only relational DBMS's, since file managers will open only one file at a time. A DBMS capable of working with three or more files will be adequate for even complex livestock or crop reports. The ability to open three files at once increases the ease with which you can generate reports.

Flexibility

The applications you currently plan for your DBMS are likely to be quite different in 1 or 2 years. The more flexible your DBMS, the longer it will serve your current needs and the more readily it will adapt to your future needs. We can apply several measures of flexibility to DBMS alternatives, each of which is important for agricultural record keeping.

Types of data. One measure is the number of types of data it supports in its fields. Agricultural records are most easily kept on a DBMS with five data types: character, date, numeric, logic, and remark data.

Character fields are useful for storing data involving letters and numbers. Identification numbers, management codes, and names are examples of character data. Keep any data involving dates in date fields. Examples include birth dates, weaning dates, and breeding dates.

Any numeric data that may be used in calculations is stored in numeric fields. Some examples are birth weights, fertilizer rates, sale weights, and yearling weights. Numeric data combined with date-type data produces most of the reports you'll need as an agricultural manager.

Logic fields are used to record true/false or yes/no data. These fields are used to store answers to questions such as: Has this cow been bred? Did this ewe have twins? Has this field been soil-tested? As you can see, logic data is useful for generating lists of all unbred cows, or all ewes that had twins, or all soil-tested fields.

Remark fields will prove to be useful in keeping any data that doesn't fit into the four types above. Remark fields are used for comments, opinions, and miscellaneous information that may be useful.

The absence of one or more of these five field types in your DBMS will limit your agricultural record-keeping capabilities. Either you'll be forced to maintain less data than you'd like, or you must substitute one field type for another in storing your records.

For instance, dates may be entered into character or memo fields, but then it's difficult (perhaps impossible) to use them in calculations. Comments may be stored in character fields, but they'll take up much more storage space than they would in a remark field, and this may limit the amount of data you can store per record.

Ability to change structure. Another measure of flexibility in a DBMS is the ability to change the structure of its database files. The structure is determined by its field names, field types, and field lengths. No matter how much you plan the design of a database structure, at some point you'll want to change it.

Not all DBMS's allow you to change the structure of a file after you define it. This can be frustrating, especially for new DBMS users. Select a DBMS that allows you to add or delete fields, increase or decrease field lengths, or change field names. Otherwise, you may find yourself defining a database and entering records several times before it yields the information you desire.

Ability to share data. A third measure of flexibility is the ability to share data with other programs. The importance of this aspect of DBMS selection is related to the computer programs you use now and plan to use in the future. If you work with computerized accounting systems or electronic spreadsheets, insist that your DBMS be able to share data with these programs.

The term *share* in this case is used in a bidirectional sense—your DBMS should be able to read data in from other programs and output data in a manner that other programs can read. The most popular way of exchanging data is through ASCII files.

ASCII, an acronym for *A*merican *S*tandard *C*ode for *I*nformation *I*nterchange, is a standard for representing computer data. If the DBMS you select is capable of reading and writing ASCII files, it will have the ability to share data with many software programs.

Suppose you maintain a beef database file that contains calving dates and actual weaning weights. You'd like to calculate adjusted weaning weights, but your DBMS doesn't support calculated entries in reports.

You could use your DBMS to create an ASCII file version of your database, then read that file into an electronic spreadsheet to perform the desired calculations. The calculated values could then be read back into the DBMS to produce your report. This illustrates that you can use the flexibility of sharing data to overcome the inability to calculate results in your DBMS reports.

Ability to custom-design reports. The final measure of flexibility is the ability to construct custom-designed reports. Reports are the most important part of any agricultural record-keeping activity. If your DBMS does not generate the reports you need for managing your firm, then entering the records into the program provides little advantage over hand-written records.

The reporting capabilities of DBMS's vary greatly. Some programs allow reports to be constructed that are applicable only to a limited number of situations; others will produce virtually any type of report.

The greater the ability to construct custom reports, the better the chances the DBMS will function for a wide range of agricultural record-keeping applications. Select a DBMS that has a high degree of custom-reporting capabilities.

Programmability

Some DBMS's have programming capabilities that can automate repetitive processes. Each DBMS applies different terms to these capabilities. *Command* files, *procedure* files, and *format* files all describe the files that store DBMS programs or instructions.

This publication isn't a text on DBMS programming, so we won't do an in-depth analysis of programming language features desirable in a DBMS. Instead, we'll present the general programming features useful in agricultural record keeping.

To program a DBMS, it must have a programming language. The language can be used to build custom menus, input screens, and reports. A programming language is a very powerful tool for experienced DBMS users, but beginners may not use it for several years. Like the flexibility measures (page 3), a DBMS with a programming language is an investment in the future.

Another type of programmability in a DBMS is a query language. A query language allows you to issue English-like commands to retrieve the information you desire. Some examples of queries are:

```
LIST ALL FIELDS WITH CORN AND YIELD
PRINT HEALTH REPORT FOR MARES IN PASTURE
  12
AVERAGE BIRTH WEIGHT FOR ALL CALVES
```

The exact format of queries is determined by the DBMS you're using. A query language provides a familiar communication link between you, your computer, and your DBMS, which can be especially useful for new DBMS users. English commands are more familiar, and easier to remember, than programming language commands.

An additional programmability factor in a DBMS is the ability to perform calculations in fields and reports. Many agricultural record-keeping applications require data based on calculated values. It may be most convenient to do the calculations and store the results within a database, especially if the calculated values are used in other calculations or appear in several reports.

Calculated values may be stored in fields in two different ways. First, some DBMS's allow a formula to be entered as the contents of a field, and the field type is then defined to be a calculated field. For example, suppose you're developing a livestock inventory database that stores livestock type, number of head, average weight per animal, and current market value in dollars per pound.

You could add a calculated total value field whose contents is:

Number of head × average weight × market value

Changing the market value per pound would automatically recalculate the value of your livestock, as would changes in number of head and average weight.

A second way calculated values are stored in fields is through the programming language. Programming commands are used to calculate the desired results and then enter them into blank fields in a database. A simple routine to accomplish this might be:

```
FIND FIRST TYPE OF LIVESTOCK
CALCULATE NUMBER OF HEAD × AVERAGE
    WEIGHT × MARKET VALUE
STORE THE RESULT IN THE TOTAL VALUE FIELD
LOOP BACK TO THE TOP OF THE PROGRAM AND
    CONTINUE UNTIL ALL LIVESTOCK TYPES
    HAVE BEEN CALCULATED
```

The exact commands used to write this routine vary for each DBMS.

Performing calculations in a report is highly desirable in any DBMS used for agricultural record keeping. If a DBMS has good calculation capabilities in reports, then calculated field entries are minimized. Instead of calculating values and storing them in your database (which wastes valuable disk space), you can do the calculations only when you need to see them in reports.

We've discussed four programmability factors related to selecting a DBMS. We presented these factors because they're useful for agricultural record keeping, not because we believe agricultural producers need to become programmers to use DBMS's. The power and capabilities of many DBMS's will allow you to store and retrieve any information you want without writing a single line of programming code.

Miscellaneous

Consider these final factors in selecting your DBMS, no matter what type of software you're purchasing.

The first and most important of these for a beginning DBMS user relates to the ease of learning the DBMS. Is the program menu-driven, command-driven, or a combination of both? Menu-driven programs tend to be easy to learn, while the command-driven systems tend to have the greatest number of features.

A beginning DBMS user will enjoy the menus and on-screen prompts associated with menu-driven systems. Experienced DBMS users quickly tire of wading through three or four menus to perform operations that could be accomplished with one command.

Don't expect to learn how to use your DBMS overnight. The number of features contained in a DBMS requires time to learn how to use them effectively. However, many users find that just knowing how to perform a handful of operations (creating databases, adding and editing data, and constructing reports) fulfills most of their needs, especially in their first year of use. They learn other program capabilities as needed or as time permits.

A related consideration is the documentation or written instructions and supporting materials that accompany the DBMS. Good documentation includes a step-by-step description of how to use the program and a detailed description of each feature. Many systems also include disk tutorials, programs to teach you how to use a DBMS.

Well-designed documentation can make even the most powerful DBMS easier and faster to learn and use. Before you purchase a DBMS, look at its documentation:

- Are the instructions easy to read and understand?
- Is there a detailed index to locate a discussion of any part of the program?
- Does a tutorial program or manual accompany the DBMS?

The answers to these questions will help to determine the quality of the documentation associated with your DBMS.

You should also determine the type and source of software support provided with the DBMS you're considering. Types of software support include toll-free phone lines, newsletters, and training workshops. One or more types of support are invaluable while you're learning and working with a DBMS. In general, the more software support available, the easier a program is to learn.

Support may come from one or more sources. Software publishers typically provide hot lines for questions about the use of their products. Dealers may provide training workshops and ongoing seminars. User groups often provide newsletters with tips from experienced users. Make sure you know who's responsible for providing you with timely and accurate software support.

Finally, consider the upgrade policy of the publisher. As program bugs are corrected and new features are added to programs, the publisher will sell new releases of the program.

Some publishers allow current program users to move up to a new version of the program at little or no cost. At the other extreme, a publisher may offer upgrades to current users only at the full sale price of the program. Obviously, low-cost upgrades are preferred.

DBMS checklist

In this publication, we've presented a number of hardware and software factors to consider in selecting a DBMS. We presented them to assist you in making an educated software purchase decision. These factors represent those that we've found to be the most important for agricultural record keeping.

Figure 1 contains a checklist summarizing each of the factors presented. This checklist shows the factors to consider and the desired attributes of each. You can use it to compare two or more DBMS's for selecting the program most appropriate for your agricultural record-keeping application.

While this checklist will help summarize the capabilities and limitations of alternative DBMS's, it can't help with your more subjective evaluations in selecting a DBMS. You may be faced with many trade-offs, such as ease of learning versus power, or customer support versus ease of use. It's up to you to rank these DBMS attributes—and select your software accordingly.

For further reading

This is a Western Regional Extension publication. Single copies are available from the Extension Service at cooperating universities, or from Agricultural Communications, Publications Orders, Administrative Services Bldg. 422, Oregon State University, Corvallis, OR 97331-2119. If you're ordering from Oregon, please include 25¢ for orders up to $2.50. If you're ordering in quantity, please call (503) 754-2513 for a price quote.

Cross, Timothy L., and Ronald J. Lane, *Introduction to Database Management Systems for Agricultural Record Keeping,* Western Regional Extension Publication WREP 107 (Oregon State University, Corvallis, 1988). $1.00

Factors to consider	Desired	Actual
Hardware		
Operating system supported	Version owned	
Minimum RAM required by program	Amount owned	
Operates on a:		
Floppy disk system	Yes	
Hard disk system	Yes	
Supports printer width and typestyle	Yes	
Supports monitor and display options	Color graphics	
Software		
Maximum number of fields	Minimum of 30	
Maximum characters per field	Minimum of 80	
Maximum characters per record	Minimum of 1,000	
Maximum number of records per file	Size of disk	
Maximum number of open files	Minimum of 3	
Supports many types of data		
Numeric	Yes	
Character	Yes	
Date	Yes	
Remark	Yes	
Logic	Yes	
File structure may be changed	Yes	
Data may be input from, and output to, other programs	Yes	
Custom reports may be constructed	Yes	
Contains a programming or query language	Yes	
Will store calculated fields	Yes	
Reports may contain calculations	Yes	
Easy to learn	Yes	
Good documentation provided	Yes	
Customer support available	Yes	
Upgrades and new releases available at low or no cost	Yes	

Figure 1.—DBMS selection checklist

Trade-name products and services are mentioned as illustrations only. This does not mean that the participating Extension Services endorse these products and services nor that they intend to discriminate against products and services not mentioned.

Issued in furtherance of Cooperative Extension work, Acts of May 8 and June 30, 1914, in cooperation with the U.S. Department of Agriculture, O.E. Smith, director, Oregon State University Extension Service. Other western State Extension directors include: Irvin Skelton, University of Alaska; Roy Rauschkolb, University of Arizona; Lowell Lewis, University of California; Kenneth Bolen, Colorado State University; Yukio Kitagawa, University of Hawaii; Harry Guenthner, University of Idaho; LeRoy Luft, Montana State University; Bernard M. Jones, University of Nevada/Reno; Robert Gilliland, New Mexico State University; R. Paul Larsen, Utah State University; Fred L. Poston, Washington State University; and Jim DeBree, University of Wyoming. The University of Guam Extension Service, W.P. Leon Guerrero, director, also participates. Extension invites participation in its programs and offers them to all people without discrimination.