

1

The design philosophy of QueryCalc

An Introduction

A Simple but Powerful Idea

At the heart of every HP3000 lies a very simple idea. The HP3000's database, IMAGE, was designed to be no more than an electronic filing cabinet, and it was originally meant to be just as easy to use. Somehow, over the years, this simple idea has been confused by a river of jargon. From this point on, we would like you to forget the stack of manuals you have on the shelf describing IMAGE and the intricacies of DBGETs, DBFINDs and DBPUTs. None of that information is particularly necessary to understanding why you would want to use IMAGE databases, nor how you might use existing databases to your advantage.

You bought your HP3000 to replace your steel filing cabinets with electronic ones. Now you should be able to get the information out at least as easily as you would have from the old filing cabinets. QueryCalc was designed to allow you to create management reports with ease, without a great deal of fuss or memorization. QueryCalc combines two very basic office ideas, spreadsheets and filing cabinets, into a powerful tool. All that's changed is that now the spreadsheets and filing cabinets are electronic.

This ease of use doesn't mean that anything's hidden from you. Quite the opposite, in fact. If you're ever really going to understand what's going on, you must be able to see what's in your databases. QueryCalc opens IMAGE up and makes your information visible to you.

The success of spreadsheets on computers has been historic, especially on personal computers. A spreadsheet encourages a "What if..." form of analysis. But this is not the primary manner by which you are going to use QueryCalc. QueryCalc allows you to extract information directly from the databases resident on your HP3000 and manipulate that information in the ways you require. If the computer is ever going to pay for itself, at least in the way you imagined it would before you bought it, this is where it's going to do it.

The single feature that made the electronic spreadsheet a success was *visible calculation*. You can see the relationships between the numbers on the page and you can change them. This feature alone accounts for much of the popularity of spreadsheets on personal computers.

QueryCalc is a 26-page, true three-dimensional spreadsheet. And QueryCalc can be used simply as a spreadsheet. Operations may occur over individual cells scattered over multiple pages, columns, rows, vectors through pages or as complete cubes. QueryCalc has all of the features and behaviors you would expect from the very best of the personal computer spreadsheets. But that's only the surface; QueryCalc is much more than that.

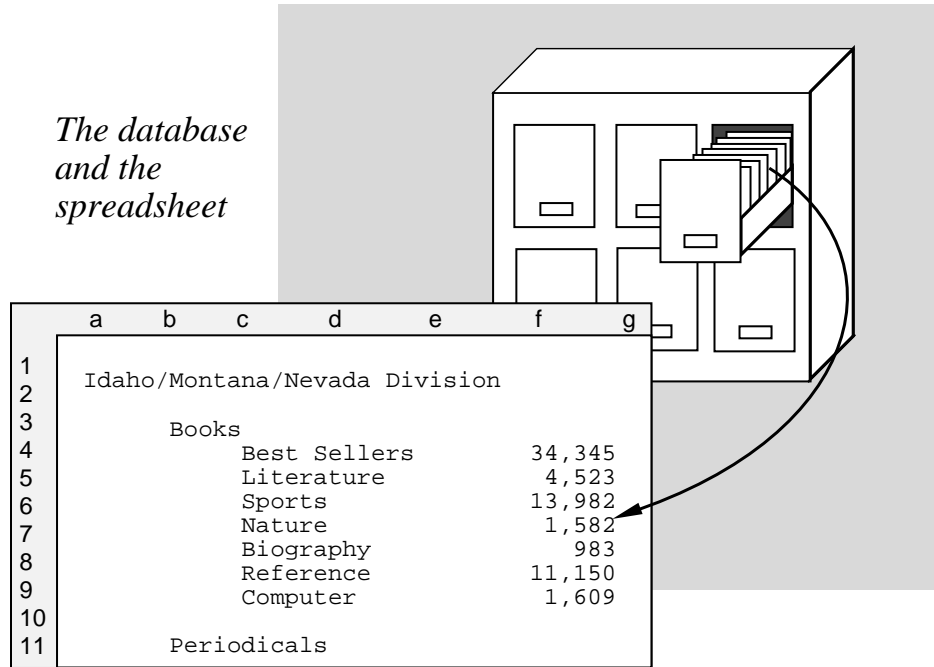
The Spreadsheet as a Report Writer

A normal spreadsheet is a large, flat plane of rows and columns. If you wanted to enter the sales figures for January, February and March, you would have to go off to the side, add the sales receipts yourself and type the answers in. Or you might use some form of a program to get that same information out of your IMAGE databases and transfer it into your PC by down-loading an HP3000 file to a PC. Doing either procedure is cumbersome, tedious work where nothing is immediately intuitive or transparent. And it will be just as much trouble to do it next month when a newly calculated, updated report is needed again.

Nevertheless, it's obvious that there is a strong pressure to do something exactly like that. Probably someone in your organization is already using one or the other procedures, trying to get information out of your IMAGE databases into a form that he or she can understand and is familiar with on a personal computer.

Power Software on the HP3000

QueryCalc is different. Where a normal spreadsheet is thin and flat, QueryCalc has depth because of its links into IMAGE. Every cell in QueryCalc's 26 pages can be a database inquiry question into any one of 10 IMAGE databases. Defining sums and relationships between the information extracted from the databases is simple because of the spreadsheet nature of the report. Text formatting and reorganization take on word processing-like attributes. QueryCalc is a "what-you-see-is-what-you-get" (*WYSIWYG*) report writer. No trial formats or trial compilations are necessary. You know what the output will look like before you print it because it's right there on the screen.



QueryCalc was designed so that everything is visible while you are putting a report together. The report you create, while you are creating it, is like putty. If you don't like what you've done, you can change it immediately. The data you retrieve from the IMAGE databases is completely open for your inspection. Reports can often be assembled 5 to 20 times faster than they could be using any other method. And yet the entire process is simple enough that you can learn to use it by mimicking someone else's report. QueryCalc can then be scheduled to run in the middle of the night, gathering and building your standard reports, week after week, without any further operator intervention.

How a Typical Report is Built

Consider the report on the following page. QueryCalc has a variety of calendar functions so that date and time equations may be placed in cells, as they have been in the upper left hand corner of this report. QueryCalc readily allows you to transmit any escape sequences necessary for your particular printer to print in bold, italics or enlarged print, as was done for the title headings shown.

The text that fills the left hand column of the report is nothing more complicated than text labels, typed in by the person composing the report as he or she wished them to be.

But it is the right hand column of numbers that defines the power of QueryCalc. The numbers in the right hand column were not simply typed in, as would normally be the case, but are the results of database query questions which reside in their respective cells. The displayed results came directly out of IMAGE database(s) into the spreadsheet, were summed and formatted and are now ready to be printed.

Creating the query questions is easily done. A uniform but powerful English-like syntax was created to allow you to ask just about anything imaginable. A question for one cell in this report might be "*(Show me the) sum of sales+receivables when date is between 19880611,19880617 and division is NW and category is 521*".

The next query question down the right hand column is quite likely to be very similar to the one above it, except that now perhaps the category is 522 instead of 521. Because of the spreadsheet nature of QueryCalc, a single query question may be replicated down a column. The query questions may then be modified using QueryCalc's on-line cell editor. More sweeping changes can be as easily accommodated with QueryCalc's search and replace function.

That's all there is to getting data out of IMAGE, KSAM or MPE databases. If you don't specify the dataset or database, QueryCalc will determine the proper database and dataset to get this information on its own. If multiple chained paths are available, QueryCalc will optimize the question to search down the shortest path. A variety of techniques were employed in the design of QueryCalc to make its database searches as fully self-optimizing as possible.

To complete the report on the right, only the column totals remain. Subtotals are created as column sums of the cells directly above [e.g., "SUM(E15:E21)"]. Displaying a grand total is no more difficult than composing a statement summing the subtotal cells, such as "E23+E34".

The report is now done. Putting an actual QueryCalc report together is no more difficult than it appears to be here.

Date: Monday, June 19, 1995
 Time: 9:24 AM

Page 1

The Guttenburg Book Dispensary Incorporated
 Western Regional Divisions
 Weekly Sales Report
 for the week ending June 17, 1995

Idaho/Montana/Nevada Division

Books	34,345.98
Best Sellers	4,523.78
Literature	13,982.02
Sports	1,582.45
Nature	982.09
Biography	11,150.23
Reference	1,509.53
Computers	
Periodicals	
Magazines	4,562.90
Newspapers	2,567.23
Journals	459.04

Total sales: Idaho/Montana/Nevada

83,430.82

Oregon/Washington Division

Books	44,745.98
Best Sellers	9,523.23
Literature	6,942.02
Sports	3,382.25
Nature	2,242.79
Biography	16,130.33
Reference	
Computer	
Periodicals	
Magazines	
Newspapers	
Journals	

Total sales: Oregon/Washington

The time to assemble this report may be as long as a few hours if it is your first report. But, by your third or fourth report, you will easily be able to put together a similar report in 30 minutes or less. The majority of that time will be consumed in nothing more elaborate than just typing the report in the way you wish it to appear.

The Design Criteria of QueryCalc

It doesn't take a good manager long to realize that the information being accumulated in the corporation's databases can tell him much about his company, especially about what's making money and what's not. The reports that are needed are often spur-of-the-moment questions, such as shipping cost analyses, inventory turn-around times, profit and loss sheets for the various subdivisions. These reports weren't initially planned for, but now they could be eminently profitable.

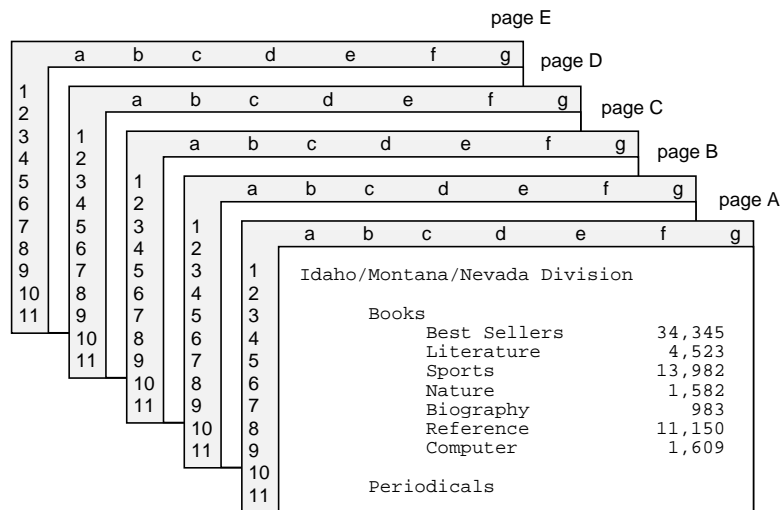
QueryCalc was designed from its inception to provide a mechanism so that members of the management staff could go to lunch, talk about what information they need, go back to the office and have the reports ready, with graphics, by about three in the afternoon. The report may be written by either a member of the programming staff or, even more likely, by one of the people at lunch. QueryCalc reports can be assembled so quickly that the final report can be put together while it is still being discussed.

Quite likely, better than 70% of the *ad hoc* reports that were designed on the spur of the moment have lasting value when regularly updated with new data, and will be used again. QueryCalc was designed to produce reports which are efficient, self-optimizing and capable of being run as regularly scheduled production reports.

How QueryCalc Works

Each of QueryCalc's 26 pages is composed of 26 columns by 90 rows (2340 cells) for a total of 60,840 cells. If you've had any previous spreadsheet experience, you already have a very good idea how to use QueryCalc. Although QueryCalc performs all normal spreadsheet functions (indeed many functions like depreciation or cashflow are far better implemented than you've ever seen them before), QueryCalc was meant first and foremost to be an HP3000 IMAGE database report writer.

To be the most plastic report writer possible, QueryCalc was designed around the concept that characterizes the best of the word processors: "what-you-see-is-what-you-get". If you make a change in a column's width, move text, or insert rows, what you see on the screen will be what you will see on the printed page. The output from QueryCalc can be directed to any printer on the HP3000, your terminal's printer or to an MPE(flat) ASCII file where it can be picked up by HPWORD, DSG, or other such software for further processing or incorporation into another, longer document.



The 3-dimensional nature of QueryCalc

The column widths for each page are independently configurable. Thus each page can be formatted differently. However, QueryCalc is fully three dimensional. A normal QueryCalc cell reference might be Ag15, where the first letter specifies the page (A), the second the column (g) and the number the row (15) indicating where the data is to come from. A sum [e.g., "SUM(BG15:EN25)"] may range over a column, a row, a rectangle, a vector through the spreadsheet, or a cube as in this example. Any cell that exists anywhere on any page is referenceable, even if its width has been collapsed to zero so that it will not print.

How to Use The Applications Guide

This *Applications Guide* is constructed in much the same fashion as QueryCalc. QueryCalc is a language-like program constructed of *primitives*. There are only four types of cells in QueryCalc (*numeric equations*, *text labels*, *text equations* and *query questions*) and only one command line. Everything in QueryCalc is built around these few primitives. What this means is that if you know how to do all of the simple procedures in QueryCalc, you'll have a good idea how to do something more complex.

Each chapter in the *Applications Guide* is directed towards a specific subject. Most of the chapters contain examples of actual reports, many of which reference the reports found in the demonstration account supplied with QueryCalc.

For the first several chapters, the keystrokes necessary to create each report

are listed, either in whole or in part. Practice entering and executing these reports. You will learn best by doing, not just by simply reading.

Advice for the New User

QueryCalc was designed so that one useful report may be mimicked many times over. And that, by the way, is one of the three basic "tricks" known to every successful programmer. At the risk of giving away the sorcerer's secrets, there are just these three items worth remembering:



1. ***Be aggressive.*** This is the most important trick of programming you'll ever learn. QueryCalc was designed precisely so that you can be aggressive. QueryCalc opens your IMAGE, KSAM or MPE databases in a read-only mode without locking the datafiles. Because you can't modify, create or delete any information in the database, you can do no harm to the database. QueryCalc opens your databases with such a light touch that system backups can proceed while you are executing you reports.

The worst harm that you can do is destroy your own work. But if sufficient time has elapsed since the report was first created, quite likely an earlier (if not identical) version of the report has been stored on a backup tape. If you work in a large organization and do not know your system managers well, such times are excellent opportunities to get to know them better.

QueryCalc was designed to automatically generate efficiently executed reports and will guide you towards the creation of a well thought-out construction. How can you tell if your report is efficient? Generally, just by the time it takes to execute. If you have done something that is quite slow (which may imply some sort of inefficiency), you will especially notice it in session mode. Execution that is taking some time can always be stopped in mid-process by pressing the CNTL-Y keys. You may then examine the cells that are taking so much time. By the time you are ready to job-stream your reports, they are almost always quite well-constructed and efficient.

The bottom line moral remains: *be aggressive*. You can do no harm to the databases, to the HP3000, or generally do anything that is all that inefficient. Being aggressive is the only way you will learn.



2. **Program by imitation.** The second great trick of programming is called "*ditto programming*". Every good programmer knows the trick. When you look at someone else's reports, think abstractly. The report you see won't be precisely the same one you need, but it is probably more similar than different. You already know a great deal of information about your own databases. Even if you don't know what the databases are called or where they are, you have a very good idea of what's in them and how important that information is to you. And you know what information you need to see. Search through the *Applications Guide* and find reports which generate report structures similar to those which would be useful to you. You should be able to write comparable reports using your own databases in 2 or 3 days.



3. **Know your databases.** This third bit of advice is critical. There is a growing tendency among many companies which manufacture report writing programs to isolate the user from the database, to relieve him of the requirement of intimately knowing the databases from which he is extracting his information. If that is not an easy recipe for disaster, it is at least a quick recipe for confusion and error.

A database is meant to be the *image* of a steel filing cabinet, filled with paper records. Nothing more. If you could have found the information you needed in a standard filing cabinet, you can find it in your databases. This is especially true of IMAGE on the HP3000, as you'll see in the following chapter. Almost all of the difficulty you'll have in using databases will come from determining what someone else called the items and finding out where they're located. But that's much the same problem you'd have when first using someone else's standard filing cabinets. The easiest solution is the obvious one: ask whomever is in charge of your databases for an explanation.