

11 Databases

If you're still keeping track of important names and addresses, recipes, or a CD collection using a word processor, you're a prime candidate for a database program.

In this chapter, we'll show you how databases organize information and explain the differences between flat-file and relational database programs. We'll also give you a rundown on the most popular software available so that you can decide which database program is right for you. Finally, at the end of the chapter, you'll find a collection of tips so you can get the most from your database program.

Contributors

Steve Schwartz (SS) is the chapter editor.

Jay Lee (JL) spent many years in technical support and is a veteran Mac user. He is currently working as a computer consultant and multimedia designer/producer in Silicon Valley.

Don Crabb (DC) is a well-known writer and *MacWEEK* columnist.

Elizabeth Castro (EC), Arthur Naiman (AN), Robert Lauriston (RL), John Kadyk (JK), Sharon Zardetto Aker (SZA), and **Susan McCallister (SM)** were all contributors to earlier versions of *The Macintosh Bible*, from which parts of this chapter were taken.

Contents

What Is a Database?	311
What Can You Do With a Database Program?	312
Parts of a Database File	312
Flat-File versus Relational.....	313
Choosing a Database Program	313
FileMaker Pro	313
4D First	315
4th Dimension	315
Helix Express.....	317
Phyla.....	318
Panorama	319
Visual FoxPro	320
Working With a Database Manager.....	321
Creating a Database	321
Using a Database Program.....	323
Database Tips	325
Database Design	325
FileMaker Pro Tips.....	326

What Is a Database?

(SS/EC)

A database organizes information by dividing it up into small, discrete pieces called *fields*. An address book might be divided into name, address, and phone number fields; a checkbook into check number, payee, description, and amount fields. A computer can use these fields to help it quickly sift through huge amounts of data, perhaps in order to find a certain name (or check number or entry word) or to arrange a client list by ZIP code (or other criteria).

Spreadsheets versus Databases

(EC)

Spreadsheet and database programs provide similar ways of looking at information and, in fact, can perform many of the same functions. Each row of a spreadsheet contains a collection of pieces of information, each with its own label (the column label). You can think of the columns as *fields* and the rows as *records*. (If you don't know what fields and records are, don't worry, we explain them in "Parts of a Database File".)

And although you can create a database file with a spreadsheet program, and can search, sort, and perform calculations, you will be severely limited with respect to reports, entry forms, and other types of specialized layouts.

On the other hand, you can use a database as a spreadsheet, keeping track of monthly and yearly totals, for example, but you will have limited means of charting and creating what-if scenarios—a spreadsheet's most powerful feature.

The address book and checkbook mentioned above are all *database files* (or simply, *databases*). You create, view, and manipulate databases with a *database program* (also called a *database manager*), such as FileMaker Pro, Visual FoxPro, 4D, Helix Express, or Panorama. The tricky part is that many people use the term *database* when speaking about both the files and the programs. (To make things easy for you, we'll just refer to the files as *databases* and the applications as *database programs*.)

All of the programs discussed in this chapter can be used to create custom databases, enabling you to organize and present your information in any way you like. Some examples of home-grown databases are recipe files and videocassette catalogs, but many people also create contact managers, checkbook registers, and bookkeeping databases. If you'd rather just concentrate

on entering data—leaving the design work to others—you'll be pleased to learn that most database programs include a variety of *templates* (preformatted databases that you can immediately put to use in your home or business).

Some stand-alone programs such as Now Contact and Act! are actually single-purpose database programs, usually marketed as "personal information managers." Using one of these programs may save you some time, effort, and money when compared to buying and working with a full-fledged database program.



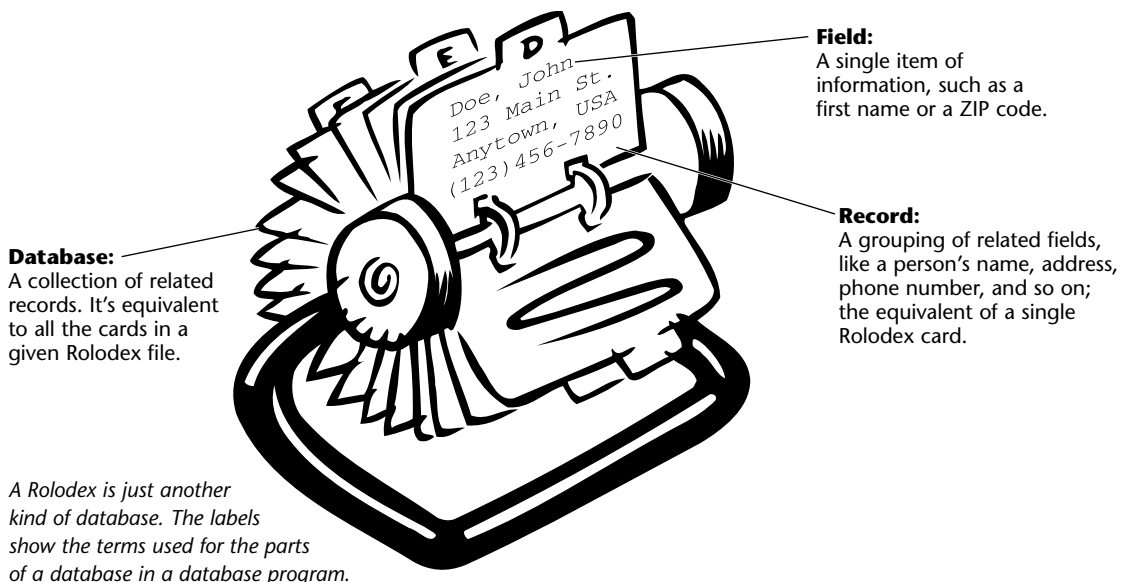
What Can You Do With a Database Program? (SS/EC)

Word processors can also hold information such as names and addresses, recipes, and the contents of your CD collection. So why would you want to use a database program to organize that information? The answer lies in the way information is stored in a database. The information for each person, place, or thing within the database is stored as a discrete piece of information, such as a first name, check number, or recipe ingredient. This data segmentation allows the database program to access and manipulate the information quickly and easily, which in turn allows you to consult individual parts of your data, order it, and then output portions of it (or the whole thing).

Storing information in a database lets you view the same information in many ways, selected and ordered according to your current needs. For example, suppose you have a list of names and addresses. A database will allow you to order (that is, *sort*) the list by last name in order to create a printed phone directory, and later by ZIP code, so you can print labels for a bulk mailing. You can also quickly find the portions of your data you want to work with—say, all of your clients in the northeast, or just those who haven't placed an order in the last six months. These tasks would be much more difficult with a list stored in a word processing document.

Parts of a Database File (SS/SZA/AN/SM/JK)

Here are some basic terms you need to know when using a database program. To help make them clearer, we'll compare each with a concrete, real-world example that you're undoubtedly familiar with—a Rolodex file.



Flat-File versus Relational

(SS/EC)

The main difference between the two major kinds of database programs—*flat-file* and *relational*—is in their information-sharing capabilities. Suppose you have a database that you use to do invoicing. In order for the name and address of your client to appear on an invoice, the database must already contain fields for recording this information or there must be a link to another database that stores client addresses.

In a flat-file database program (such as Panorama), you normally include all necessary fields in one database. Thus, the address information would be an integral part of the invoicing database. In a relational database, a link (based on a key field such as an ID number) can be made to information in a Client database in which the address data is stored. Whenever you look at or print an invoice, the program consults the Client file and displays the latest address information.

The primary advantage of a flat-file database is ease of learning. Understanding how relations work, on the other hand, can be conceptually difficult. Advantages of relational databases include speed and the avoidance of duplicate data. Instead of copying or retyping address information into every database where it's needed, you can place it all in a single file and then simply *refer* to it.

Now that FileMaker Pro has become a full-fledged relational database program, the distinction between flat-file and relational programs has blurred considerably. Keep the following in mind, though: You can create flat-file databases in *any* database program. Just because a program has relational capabilities doesn't mean that you have to use them.

Choosing a Database Program

In this section, we review several of the leading database programs.

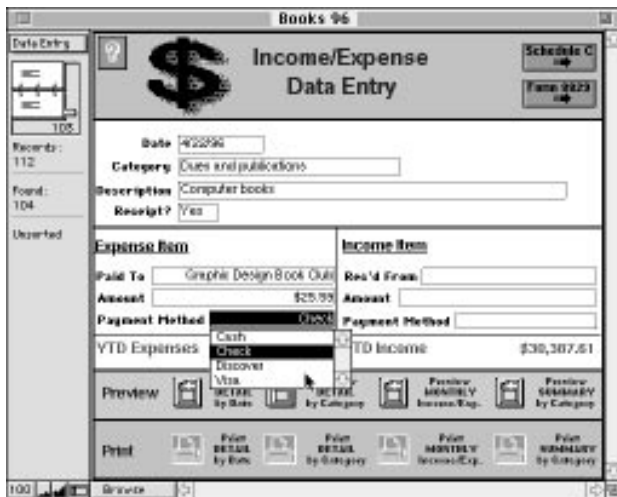
FileMaker Pro

(JL)

Claris has finally released the long-awaited upgrade to **FileMaker Pro** (\$200), the popular database program that many users favor for its ease of use and power. FileMaker Pro 3 adds many important new features, such as relational capabilities, more scripting options, improved text handling, and integrated mail merges (a real time- and sanity-saver for anyone who has to do this on a regular basis). Forty templates are provided for business, education, and home use that can either be used as is or modified to suit your needs. And for users who have made the jump to the Power PC, version 3 is accelerated for the Power Macintosh.

While some of the new capabilities increase the program's complexity, the basic features still work the same way. This means that it remains easy to use and is a good first program for database novices, but also gives previous users the ability to go relational without the steep learning curve associated with switching to a new program.

FileMaker Pro 3 can open files created by earlier versions and automatically convert them to the current format. Text, ClarisWorks 2 and 3, Excel 3 and 4, SYLK, DBF, and other file types can also be



FileMaker Pro layouts can include graphics, pop-up lists, and buttons that execute a series of actions, such as printing a report or displaying it on screen.

opened directly, bypassing the old method of creating a new database and then importing the old data.

Making the leap to relational databases is straightforward. You create a relational database by defining *relationships* between files—two fields (one in each file) which contain a matching value (a customer ID number, for example). When a matching value is entered in a record, the data from one database is displayed in the other. Any changes made to that data are reflected in both files. In other words, when a customer number is

entered in an invoice, information (such as an address and credit card number) from the related database is shown. If this information is changed in either database, the record in the other database can automatically be updated.

Earlier versions of FileMaker Pro accomplished file linking by means of *lookups*, where the data was actually pasted from one database into another. Not only was the data duplicated, but changes made to the data were not automatically updated. This meant that if you weren't careful, you could have two different addresses for the same customer, for instance.



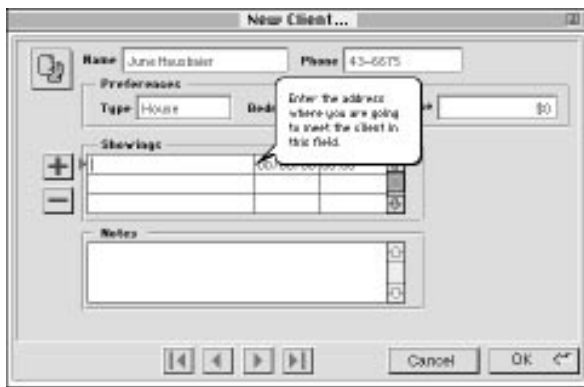
FileMaker Pro's scripting capabilities have also been beefed up. Conditional tests (If..Then, If..Then..Else) and loops are now supported. A new Dial Phone script step lets you dial a telephone through your modem or the Mac's speaker (an extremely handy feature for contact databases). The new Button tool makes it easy to create 3-D buttons to which scripts can be attached, enabling you to select a particular subset of records, sort, and then print a report with just a single mouse click, for example.

To sum up, FileMaker Pro 3 still provides an easy-to-use solution for the beginning database user, while offering plenty of opportunity for growth. For the experienced FileMaker user, the new features solve many issues associated with earlier versions and provide enough new tools to take their databases to new levels. And dedicated users finally have what they've craved: true relational capabilities.

4D First

(EC/SS)

For a relational database at a bargain price, try **4D First** (\$100, ACI US). Designed as an entry-level database with many automatic features, 4D First is a pared-down version



You can add your own Balloon Help to any 4D First layout to help remind you or data entry operators what and how information should be entered in each field.

of 4th Dimension (4D) that can stand on its own or be used as a run-time version to open 4D files. This means that you can not only use the hundreds of 4D templates that are already available, but you can also create new databases as you need them.

Although 4D First is slightly more difficult to learn than a flat-file database program, the relational strength and automated features it offers may be worth it. For example, 4D First automatically creates input and output



layouts, and its QuickReport and Graph features make analyzing your information easier than with FileMaker Pro.

4D First has two nice features that help you prepare your database for other users. First, you can add balloon help to the fields in a layout. If you (or your data entry operator) forget what goes in a field, simply activate balloon help and point to the field with the mouse. Second, you can create a whole set of custom menus for your users so that they have access to all the commands (and only those commands) that you want them to use.

Finally, if you ever need a really powerful database, you can upgrade directly to 4th Dimension. And every database you've developed will still work.

4th Dimension

(JL)

4th Dimension (\$895, ACI US) is a full-featured relational database that is well suited for even the most demanding applications. Users can start by creating basic databases with limited functionality and still have plenty of room for change as their needs and proficiency grow. For experienced users and developers, there is a wealth

of tools and features—including a high-powered programming language—for creating sophisticated custom applications. 4th Dimension (4D) databases can also be compiled into run-time versions for distribution.

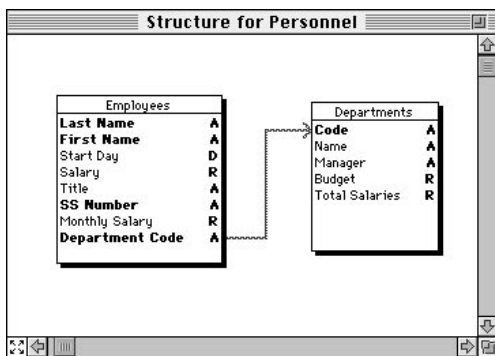
4D has many features to solve problems that database administrators face, including data validation (making sure that items such as phone numbers are formatted correctly), a robust search editor, and password protection. Scripting can be done in the traditional text mode or you can use a graphical flowchart-style interface. Procedures (scripts) can be global or associated with a particular layout or file, adding to 4D's flexibility.



HOT TIP

Generating reports is another task that can be troublesome for a database administrator, but 4D handles it with aplomb. The QuickReport feature helps you set up meaningful reports with breakdowns and subtotals by particular fields. You can also chart your data in 4D rather than having to export it to another program.

4D has all of the relational capabilities you could want, including the ability to do one-to-many, many-to-one, and many-to-many joins. Relationships are drawn (using a connect-the-dots metaphor) and displayed graphically, making them easily understood. You can also create *subfiles* (a separate file that is associated with a particular



You can create and display relationships graphically in 4th Dimension.

record). This means you can keep track of performance reviews, revisions, or other data that may be present on some records but absent on others.

One of the most significant changes to occur in the latest release (version 3.5) is the addition of cross-platform compatibility. This is especially important in corporate settings where acceptance of a Macintosh-only solution is fast becoming a rarity.

4D is part of a comprehensive product family including 4D Server, a software development kit, and a unique (for a database product) suite of utilities that enable you to integrate word processing, spreadsheet, drawing, and charting capabilities into your database applications.

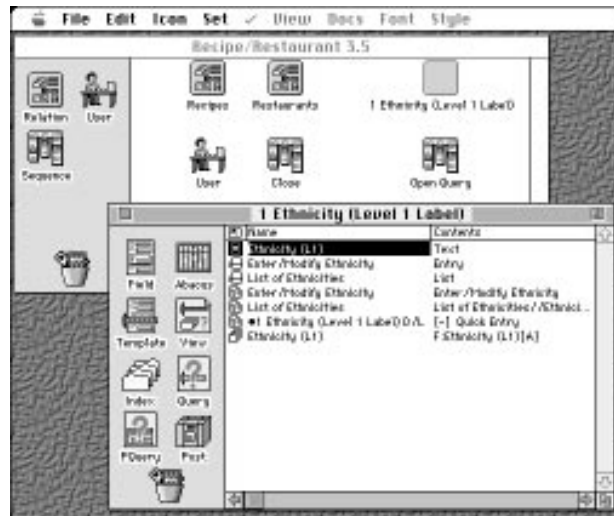
4th Dimension is not a program for novices, but that isn't necessarily bad. Any program with the power to handle complex data requirements sometimes suffers in ease of use. There's no mistaking that 4D requires an investment on the part of the user, but—given its rich feature set—this investment is sure to pay off in the long run.

Helix Express

(DC)

In the world of Mac databases, **Helix Express** (\$590, Helix Technologies) lives about as far from the mainstream as can be. While many folks worry about procedural data languages, SQL, ODBC, and other relational and client/server technologies, Helix Express users command an icon-laden, object-oriented database unlike any other on the Mac (or on Windows, for that matter). Although Helix Express is fully relational and supports its own client/server multiuser architecture, learning to use it is nothing like learning other Mac relational database programs.

When you fire up Helix Express, you're faced with a mini-desktop holding four icons (Trash, Relation, Sequence, and User) and desktop space in which to place them—which is the first step in building an application. Relation icons define fields and form templates, views, and queries. Sequence icons allow you to set up simple procedures to interact with Relations. The User icon defines passwords, permissions, and related access parameters. You build your Helix application by placing these icons on the desktop and by opening them and using the tools and commands they provide.



In Helix, you can define and modify databases by manipulating icons.

Like its competitors, Helix Express provides a programming language to tie together actions and results. Unlike its competitors, the language of Helix is less richly defined, lacking basic looping and branching structures. Still, once you understand how to use the basic and subsidiary icons (Field, Abacus, Template, View, Index, Query, PQuery, and Post), you can build Helix Express applications that work well—making Helix Express a reasonable choice for building small business applications.

To help with its rather underpowered development environment, Helix Express now fully supports a color development environment, as well as conditional macros, conditional buttons, a fixed-point data format, faster query technology, and improved performance. The latest release also adds support for Publish and Subscribe, and Apple Events, although it does not support AppleScript, QuickDraw GX, or PowerTalk. Helix Express has improved its operating speed so that it's now in the middle of the pack when it comes to sorting, searching, writing to disk, and related tasks. And its multiuser performance remains strong.

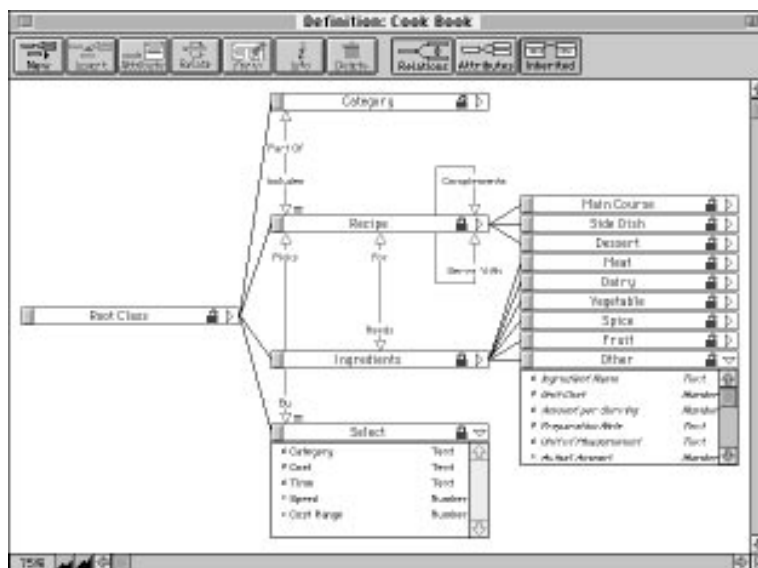
What Helix Express lacks is mainstream connectivity with other Mac databases and development environments, a procedurally driven interface that most users are familiar with, and a large enough user base to make it a comfortable choice for many potential users. If you need a Mac relational database and don't mind living outside the mainstream, Helix Express is competent and worth a closer look.

Phyla

(DC)

Mainstay calls **Phyla** (\$300) an object-oriented database—and for good reason. Rather than come at you with flat files or tables related via common fields, Phyla uses an object-based architecture. Object classes substitute for flat files or for relational tables; attributes do the work of fields. But in many ways, this is simply a nomenclature issue. Once you are using Phyla to build mailing lists, invoice files, customer reports, and inventories, it won't really matter to you that these are Phyla objects, except that—because of its object-orientation—it takes less effort to manipulate your data than is sometimes the case with procedurally driven programs.

If you've cut your teeth doing C++ programming with Metrowerks CodeWarrior, then you'll be right at home with Phyla's class definitions. Phyla makes it easy to create an object—simply give it a name and a set of attributes. You could create an object called My Clients and give it these attributes: name, address, phone number, credit rating, and so on. Then later you could add other attributes, as needed. Unlike C++, Phyla can also relate different objects easily—just by drawing a relational arrow connecting them and then typing the relational definition.



In Phyla, each application has its own definition window in which you can graphically create and display relations and attributes.

Because of Phyla's object orientation, drag-and-drop database creation and editing is possible. You can also bring non-Phyla files into Phyla, or export file types that other database programs can use. Phyla updates all its objects as a background activity. This means speed, speed, speed when it comes to searching, sorting, reporting, and other tasks.

Although you don't *have* to know anything about C++ or other OOP languages to use Phyla, knowing such things helps you get the hang of the system quicker, since the program bears little resemblance to others for the Mac. And although Phyla is fast and easy to use once you get the hang of its object orientation, getting that "hang" is likely to be harder than with traditional programs like FileMaker Pro. Phyla also suffers from being on the periphery of Mac databases, making it harder to find Phyla gurus to help you. But having said all that, Phyla is still a solid effort and worth checking out.

Panorama

(DC)

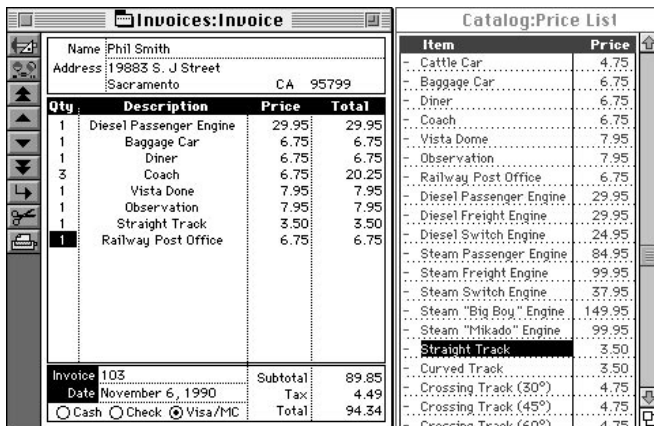
Like Helix Express, **Panorama** (\$395, ProVue Development) marches to the beat of a different database drummer. And good for them. The latest version of Panorama takes its spreadsheet-style interface and adds full multiuser capabilities, an integrated word processor, a snappy forms generator, real security, AppleScript awareness, and an improved programming language (including a much-needed interactive debugger).

Panorama's long-standing claim to fame is its speed. It loads the entire database into RAM, even though it's a fully relational program. The new version's multiuser capabilities come as a result of licensing a version of EveryWare's Butler SQL server. This gives individual users the ability to work on a part of a large database while on the road or at home, and then automatically synchronize those changes with the host copy whenever you reconnect to the server. The program also supports record locking when used in client/server mode. And whenever you want to convert an existing single-user Panorama database into a multiuser version, the program will create the necessary SQL code automatically.

Using Panorama's new client-server model also helps fix a problem that the previous versions suffered from—you couldn't build a database larger than available RAM. Now, subsets of databases can be loaded into RAM on a local Mac, while the rest of the database lives on the server.

The new security scheme is very "AppleShare-ish" in implementation. You get users and groups, 255 different access levels, and data-access restrictions on a record-by-record and field-by-field basis. While this security is not as complete as what 4th Dimension offers, it's pretty good for a product in this class.





While not a relational database program, Panorama can link databases via procedures. Clicking an item in the price list (right) automatically generates a new line item in the current invoice in the Invoices database (left).

Each database field can be edited with the program's word processor. You can format and style text to your heart's content. This makes it easier to build meaningful forms and reports, as well as fancier interfaces. Interfaces are further improved thanks to an interface engine that ProVue calls SuperObject. You can use it to create lists, menus, scalable text and pictures, buttons, and matrix objects (like schedules, multiple column lists, or calendars) without having to program. Just select the

kind of SuperObject you want (or create your own), and then plug in the particulars. Panorama will resize, redraw, reposition, and color the objects and windows on your forms to match the available color depth and size of your screen. This makes it very handy for exporting databases to Macs whose characteristics are unknown.

Panorama's spreadsheet-like interface, coupled with its new client-server features, make it a powerful tool for the SOHO crowd, as well as for mid-sized companies.

Visual FoxPro (SS)

Long known as the speed demon of Mac database programs, FoxPro is currently undergoing a major transformation. The resulting cross-platform program will be **Visual FoxPro 3.0** (Microsoft), a 32-bit application-development tool with full object-orientation and integrated client-server capabilities.

Visual FoxPro allows users to build reusable *components* (objects). Visual design tools and wizards can be used to design forms, queries, and reports, as well as create classes and subclasses. The Visual Class Designer enables users to create classes without having to learn the syntax for the new object model. You can also build full-featured, distributable applications.

Among the new and improved visual design tools included in Visual FoxPro are the Project Manager, Database Designer, and Form Designer. The new version promises blinding speed, reusability of components, and enhanced development tools. If you're considering any of the other high-end programmable databases, Visual FoxPro should certainly be on your list.

Working With a Database Manager (SS/EC)

Work with a database program is generally divided into two parts: designing databases and entering and viewing the information.

Creating a Database

When you create your own database, it is up to you to decide how to divide up the information, how it will be entered, and how it will be output. If this is done with a little thought and planning, entering and using the information should be easy.

Defining fields. The fields in a database serve to divide the information into smaller pieces so the program can sift through the information more efficiently. You should create a field for each category of information; that is, each piece of information that will be similar among many records but different in some way from the rest of the information within a single record.

Should You Do It Yourself, Buy a Canned App, or Hire a Programmer? (RL)

For simple tasks like address books and recipe collections, it's fairly easy to make your own databases using FileMaker Pro or Panorama. For more complex tasks, like inventory tracking or accounts receivable, however, the do-it-yourself approach may be crazy, since even professional database developers may need weeks or months to create such complex applications.

When a job's too big to do yourself, the main decision is whether to buy an off-the-shelf application for the task (an accounting program, for instance) or hire a developer to create a custom database. If you can find a canned application that meets your needs, it's likely to be a lot less expensive than a custom job. On the other hand, it may not do exactly what you want, and you may end up having to figure out workarounds for those limitations. Ask other people in your line of business what programs they use and how satisfied they are, or check trade publications for ads. Check out the vendors' on-line support forums on CompuServe and America Online to get an idea of the problems their users might be facing.

If you *can't* find an off-the-shelf solution, check the same sources for referrals to database programmers. You want to find people who understand not just programming, but also the needs of your particular business. Before hiring anyone, check their references carefully. It's best to choose a programmer or company with a long history of customer satisfaction. Make sure your developers fully document their work, including extensive "comments" in the database code so that you are not dependent on them for future modifications.

The classic example is a Last Name field. In an address database, all your records will have one, and the last name is clearly distinguishable from the other information in the record (Street, ZIP code, Telephone number, and so on).

Another example might be the Category field in a Recipe database. Each recipe will be of a certain type (Dessert, Main Course, Appetizer, and so on), and this information is clearly different from the other fields (Ingredients, Directions, Prep time, Cooking time, and so on).

Database Templates

(EC/SS)

If your heart doesn't skip a beat at the thought of calculated fields, procedures, and subtotals, but you'd still like to use a database, consider buying a *database template*—a ready-to-use database that someone has already created with a particular database program.

Sometimes the database already holds information you can use or add to (a list of U.S. ZIP codes and matching cities, for example); other times you simply enter your own data (keeping track of items in your stamp collection, for example). In many cases, you can modify the database's structure, adding fields and layouts when necessary.

Even if you're looking for a specialized database with which you can run your business, such as a dental office or video rental store, you don't have to rush out and hire a database programmer. Vertical applications of this sort are commonly available—although they're often quite expensive.

If you have a limited budget, you can start by looking at the many freeware and shareware templates that are available online. Claris, for example, provides a FileMaker Pro forum on America Online (Keyword: Claris) that is chock full of database templates. Software publishers' Web sites are often another good source of templates. *[Beware, though. The quality of shareware and freeware database templates ranges from extraordinary to pure garbage. And you'll find plenty of examples of the latter. Plan on lots of fruitless downloads before discovering the gems.—SS]*

There are developer associations for almost every database program, where you can find out about templates for your program. For FileMaker database templates, call the Claris Solutions Alliance (408/727-8227). For Panorama templates, contact ProVue (714/892-8199). There are more than 1,200 4D developers; call ACI US for a catalog (408/252-4444).

Database programs often come with several templates. In addition to helping you better understand how to create your own databases, some of them are actually quite useful. *[For the last several years, for example, I've done all my invoicing in a database that came with FileMaker Pro.—SS]*

Date	Ck #	Pay To	GL Category	Debit	Cr
01/01/90		OPENING BALANCE			35,978
01/08/90	1907	Northern Illinois Mold	Equipment Rental	96.05	
01/08/90	1908	U S Postmaster	Postage	75.00	
01/08/90	1909	Advertiser's Mailing Service, I	Advertising	390.80	
01/16/90	1910	Coudert Brothers, Attorney's /	Legal Fees	223.52	
01/16/90	1911	Paramount Stationers	Office Supplies	105.84	
01/17/90	1912	California Capitol	Insurance	36.00	
01/17/90	1913	California Capitol	Insurance	28.00	
01/17/90	1914	U S Postmaster	Postage	75.00	
01/17/90	1915	Sacramento Bee	Advertising	795.00	
01/18/90		DEPOSIT			3,846
01/22/90	1916	Walthers	Purchases	12,463.00	
01/22/90	1917	Blue Cross Of Calif	Insurance	279.03	
01/22/90	1918	Sherman Douglas Ins	Insurance	418.60	
01/22/90	1919	Cannon Astro	Office Supplies	145.72	

Different layouts present the same information (or parts of the same information) in different ways. Information viewed in a check register layout (left) looks completely different when viewed as an individual check (below).

Creating layouts. The second step in creating your own database is designing layouts. You should create different layouts for different purposes: one for data entry, one for viewing lists on screen, one for mailing labels, one for the annual sales report, and so on.

1915
January 17th, 1990
795.00

THE CHOO-CHOO DEPOT
123 Railroad Crossing Way
Caboose, CA 90000

PAY TO THE ORDER OF Sacramento Bee

Seven hundred ninety five dollars and 00 cents

National Bank
Santa Fe Springs, CA 90000

FOR Advertising

You'll have to decide which fields to put in each layout and where to place them. You will often put field labels next to the fields so that you can remember what information is contained (or needs to be contained) in each field.

Depending on the database program, creating layouts is often similar to using a graphics program. It involves dragging elements around the screen, changing their size, drawing lines, and aligning objects.

Using a Database Program

Once you've defined your fields and created the layouts, you can begin to enter information, search or sort the information that the database contains, and output it in practically any way you need.

Entering information. Of all the tasks involved with databases, entering the information is by far the easiest—and the duller. You simply create a new record and start typing. When you have filled in the information for one record, you create a new record and start again. Most databases save your data automatically whenever you create a new record.

Many database programs allow you to import existing information from word processors, spreadsheets, and other databases so that you can keep typing to a minimum.

The original information must be divided into fields (usually by commas or tabs) and records (generally with returns) so that the database knows where to put each piece of information.

Sorting. All databases allow you to put your records in order, according to your criteria. For example, you can sort an address database by last name or by ZIP code (or by last name *and* ZIP code), or sort your recipe file by ingredients, prep time, or category. In addition to using any criterion (or combination of criteria), you can usually sort in ascending, descending, or in a custom order (such as high, medium, and low).

Searching. Searching lets you instantly find a client after entering his or her ID code, or find all of the recipes that contain salmon *and* heavy cream. Searching yields all the records that satisfy all or part of the given criteria, depending on the logical operators used.

Previewing and printing data. The simplest way to output data is to show it on screen (called *previewing* in FileMaker Pro). For example, you can examine an individual record or preview a multipage report. Databases are especially useful because they also allow you to *print* your data in almost any way you can imagine. You can create mailing labels, directories, monthly summaries, form letters, or fax templates, for example.

Summary reports are one of the more complicated but powerful features of a database. (Ain't that always the case?) For example, you can list your baseball cards by Card Type and then by Year, and have the database calculate the value of each year's collection.

Using macros or scripts. Most database programs let you create macros or scripts to speed up and automate your work. For example, to create your monthly mailing labels, suppose you normally search for all your active clients, sort them by ZIP code and last name, and then print them using a layout called Mailing Labels. You can create a macro or script named *Print Monthly Mailing* to perform all of these steps automatically. The next time you need to print mailing labels, you just run the script.



FileMaker Pro's Find request form looks exactly like the layout when viewed in Browse mode. To begin a search, you type the information you're looking for in the appropriate fields (in this case, a P.O. number) and click Find.

Database Tips



Database Design

How you organize your databases can mean the difference between having information, and having information at your fingertips. For the latter, read on.

Use all the data entry control available (EC). Almost all databases offer some help in keeping your data accurate and consistent. Use data entry control options to let you select values from a list, mark checkboxes or radio buttons, automatically enter default data, restrict data to a certain value or range, or insist that a field contain an entry.

Different layouts for different uses (EC). Even if your monitor is so big it touches both sides of your room, you shouldn't try to put every single field in each

layout. Instead, create a different layout for each use—one for data entry, one for creating a phone book printout, one for address labels, and so on—and show only the fields that are necessary.

By visually dividing up blocks of information, you can put more on a data entry layout without confusing your data entry operators.

If you do have many fields in a layout, use background color (or gray shades) to distinguish different areas.

Making menus (SS). If your database is so complex that it requires more than a couple of layouts, consider adding a main menu layout that lets you (or others) choose what to do next. At its simplest, a menu layout might contain buttons for data entry, reports, and help information, for example. A more complex menu system could incorporate several submenus, enabling you to select specific types of reports, execute searches, and so on.

Formatting ZIP code and telephone fields (SS/SZA/AN/EC). Make sure that ZIP code and telephone fields are defined as text rather than number fields. A numeric ZIP code field will strip the leading zero from ZIPs like 07461 and may perform a subtraction on 07461-8976. Since text fields can contain text *or* numbers, you don't have to worry that leading zeros will disappear.



Formatting ZIP codes and telephone numbers as text fields also makes it easier to find a group based on the first few numbers. If you search for 9 in a text ZIP code field, you'll get all those that *begin* with a 9. You'll get all those ZIP codes that are *equal* to 9 if you search for 9 in a *numeric* ZIP code field, which—in this country—wouldn't yield many.

Duplicate files before working on them (Steve Michel). It's good practice to keep at least two copies of any database file. But if you're too lazy to do that, at least do yourself the favor of duplicating a database file before launching the application to work on it. Most databases keep their files on disk and automatically save changes to them while you work. So the file you had on disk when you began to work is not the file you'll return to when you're done.

FileMaker Pro Tips

Getting around (EC/SS). You can use the keyboard shortcut **⌘(Command) Tab** to go to the next record, find request form, layout, or report page, or **⌘(Command) Shift Tab** to go to the previous one.

The number below the little book, in the upper-left part of the window, tells you how many records (or find requests, layouts, or preview pages) you have created. Go directly to the one you need by selecting the number and typing a replacement number. This is an especially good trick in Preview mode when you know what page you want to look at.

Selecting from a list of values (SZA/SM/EC). When a field is formatted to display a set of values, either in a list or with checkboxes or radio buttons, you can use the arrow keys to move through the values, or you can type the first few letters of the word you want. Then hit **Return** (not **Enter**) to accept the entry. The moral of the story? Set up your lists so that each value begins with a different letter and don't ever use pop-up menus—they can't be accessed from the keyboard but are identical to pop-up lists in every other way.

Choosing "None of the above" (EC). FileMaker Pro lets you add an "Other" option to pop-up menus, checkboxes, and radio buttons (select Field Format and then check Include "Other..." item). However, if you format your field as a pop-up list, you don't need an Other option. Simply press **Enter** or **Esc** instead of selecting a value from the list, or press **Delete** if you've already selected something, and type a new value or leave it blank.



Check the “Strict” option to prevent users from overriding the validation options you’ve set for a field.

“**Strict**” field validation (SS). In previous versions of FileMaker Pro, you could specify loads of field validation criteria, but the user could freely override them. While this was useful for dealing with exceptions (a Canadian Postal Code contains more characters than a U.S. ZIP, for example), it also meant that your data could be easily compromised. In FileMaker Pro 3, you can make sure that the right type of data is *always* entered into an important field. Just set validation options and then click in the “Strict: Do not allow user to override validation” checkbox.



Bypassing the Delete alert (SZA/JK). When you use Delete Record to remove a record from your file, FileMaker asks you if you’re sure you want to delete it. This is a normal Mac touch—giving you a chance to change your mind before an irreversible operation is performed. But if you’re *sure* you want to delete the record, **Option-Command-Enter** will avoid the dialog box. [*This command also works for deleting find requests.*—SS]

Looking for empties and nonempties (EC). The best way to find empty fields is by typing an equal sign (=) in the field in Find mode. To find nonempty fields, type an equal sign in the field, check the Omit box, and then click Find.

On the other hand, the fastest way to find empty *records* is to sort your file in order of a field that should never be empty—such as a client’s name. All the empty records will appear at the top of the list.

Eliminating duplicate records (EC). First and foremost, *don’t* use the duplicate operator to look for duplicates. Because of the way FileMaker creates its index, analyzing each individual word of a field, it would flag *Claris Corporation* and *Microsoft Corporation* as duplicates, but not *MS Corp.* and *Microsoft Corporation*.

A good way to find and eliminate duplicates in an address database is to sort by ZIP code, address, and name—in that order—and view them as a list. The real duplicates are those that are at the same address, with *almost* the same name but not quite. You’ll be able to identify and get rid of them quickly.

Ask for a little and you’ll get a lot (SS/EC). You’re Ronald Reagan and you’re looking up your old friend, the former leader of the Soviet Union, in your database. You know his name is Gorbachoff or Gorbichef, or something like that. Don’t tax yourself;



HOT TIPS

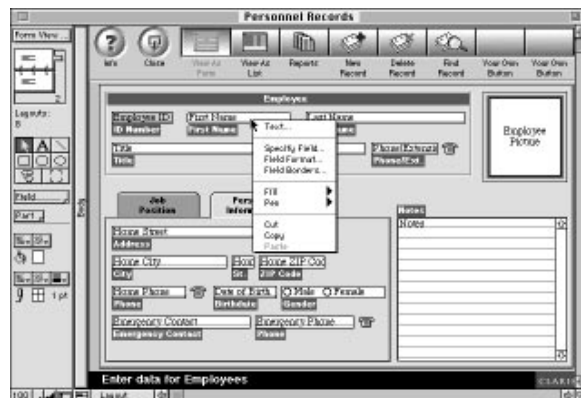
just search for “Gorb.” Any name (or word in a name) that begins with those letters will appear in the found set. Even if you know how to spell it, you should search for as little as possible. This saves you the trouble of entering more than is necessary and gives you a better chance of finding the information, even if it was entered incorrectly.

Holding on to a tool (SS/EC). Generally, after you use a tool in Layout mode, it automatically switches back to the Pointer tool. FileMaker Pro will let you hang on to a tool indefinitely by double-clicking to select it. If you want tools to be locked with a single click, check “Always lock layout tools” in the Layout Preferences dialog.

Fast sizing and styling in layouts (SZA/JK). Create a copy of any object on a layout by holding down **[Option]** and dragging. This is a fast way to create text labels with the same fonts, size, and style—just make several copies and then edit the text.

When you drag a single *field* this way, you get a dialog box that lets you choose which field you want to appear in the new spot—it doesn’t have to be the one you moved. This is an easy way to add new fields with the same size and attributes.

Magic menus (JL). FileMaker Pro 3 provides a new way to modify layout elements and field data. Holding down **[Control]** while clicking on a field or object displays a pop-up menu for changing attributes of data in that field or the selected object (such as font, style, fill color, and so on). This trick can be used in both Layout and Browse modes.



Control-clicking a field or other element when in Layout mode displays this handy menu of formatting options.

Dotted lines (EC/SS). You can create dotted lines by selecting a 2- or 3-point width line and then applying a zigzag line pattern to it. They’re great for separating subsummaries.

Selecting tricks (SS/EC). Press **[⌘-A]** while selecting objects and fields, and anything the selection marquee touches will fall into its grip. Without **[⌘-A]** you will have to completely surround the objects you wish to select.

Click any tool before choosing Select All to select all the objects created with that tool.

Select one field, press **[Option]**, and then choose Select All to select all fields.

Restraining yourself in one or more directions (EC). You probably know that if you hold down **[Shift]** while resizing an object or field, your movement is restricted to

just one direction, horizontal or vertical. If you hold down **[Option]**, your movement is restricted in *both* directions, and the object becomes a perfect square or circle.

A database with moving parts (SS/EC). You can delete any empty part (that's Part with a capital *P*, as in Body, Footer, and so on) in a layout by dragging the bottom border up to the next part or by selecting the part's tag and pressing **[Delete]**.

If the part contains objects or fields, you can still delete it by clicking its label and pressing **[Option][Delete]**. Its objects are automatically added to the part immediately below it. (Note that this tip cannot be applied to the bottom-most part on a layout.)

You can change a part's identity by double-clicking its tag and selecting a different part type. This won't affect its contents.

Get rid of the body (EC). When you're creating summary reports—say, to count your baseball cards to check how much the collection is worth—don't forget that you don't need a body. The body contains information about each card that may distract you from the totals that the report is designed to show (like the fact that your Topps collection is worth \$26,457).

Cheat sheets (EC/SS). Since FileMaker Pro lets you create scripts, you don't have to remember how to sort, search, or print your report. But how do you remember what a particular script was for? Write a short description at the top of the report, format it in red, and then select "Do not print the selected objects" from the Slide Objects dialog box (select Sliding/Printing in the Format menu). *[In FileMaker Pro 3, there's a way to do this that doesn't mess up your layouts. You can use the new Comment script step to document the script.—SS]*

Copying layouts (SS). FileMaker Pro 3's support for Drag-and-Drop allows you to copy a layout from one database to another by simply changing to Layout mode in both databases and then dragging fields, text, and graphics into the receiving database. Fields dragged in this manner however, will be undefined in the new database (unless you've already created fields by the same name).

Relookup of just some of the records (SZA/EC). If you've changed information in a file that's used as a lookup (say, you've updated names and addresses that are looked up by your invoicing file), the Relookup command in the Edit menu can update all, some, or just one of your records.

To update all your records, choose Find All in the Select menu, place the cursor in the lookup field, and select Relookup. If you want to update a group of records, select those records with the Find command before using the Relookup command.



HOT TIPS



HOT TIPS

To force an update for only the current record, move to the field that does the lookup. Make a minor change to it—say, add a space and then delete it. When you move out of the lookup field, the “new” information in it will force a lookup and the rest of the record will be updated.

Creating automatic serial numbers with letters (SS/EC).

If the serial numbers in your company happen to start with or include embedded letters, you can still get FileMaker to fill the information in automatically. Choose Define Fields from the File menu, select the serial number field (a Number field), and then click Options. In the Auto Enter section of the dialog box that appears, type the starting serial number, such as B17-6A1. When each new record is created, only the numeric portion of the number will be increased, producing B17-6A2, B17-6A3, and so on.



Creating a serial number that contains numbers and letters is easier than you'd think.

Checking for length (EC). When entering names and addresses into your database, your assistant often puts in the first three digits of the ZIP code but sometimes forgets to look up the rest. You want to check and make sure each record in your database has a five- or nine-digit ZIP code. Create a Calculation field (Ziplength) with the following formula:

$$\text{Ziplength} = \text{Length}(\text{Zip code})$$

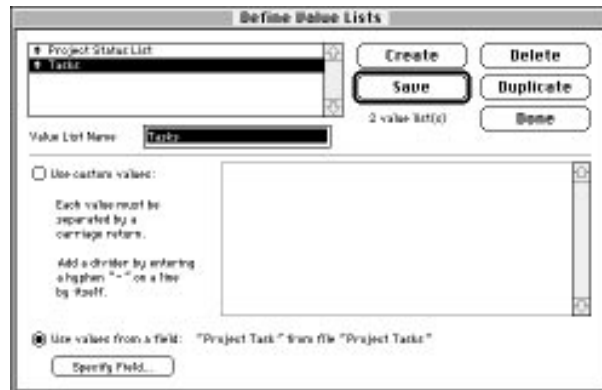
Now search the Ziplength field, omitting all those records with five or ten (nine digits plus a hyphen) characters. You'll immediately find which ZIP codes have been entered incompletely.

Creating relational value lists (JL). Once you've taken the plunge and created a relationship between files, the next issue is how do you display this information? One way is to have a field value list comprised of records from the related file. This is a good idea when the items in your value list change frequently. Examples of this include client names in your daily planner, product lists in an invoice, or tasks in a project database. To do this:

1. Create the field on your layout, and select Field Format from the Format menu.
2. Choose Pop-up list from the Style box and select Define Value Lists.
3. Give your value list a name, and click Create.

4. Click “Use values from a field,” and click Specify Field.
5. Choose the file and field you want to display as your value list, and click OK.
6. Click Done when you’re finished, set any other attributes you’d like, and click OK.

When you switch to Browse mode and tab into or click in this field, you’ll see a list comprised of records from the related file that you specified.



Although value lists are commonly created by just typing the words for the items, you can also base a value list on the contents of a field—in either the current database or a related one.



Creating portals (JL). Suppose you want to look at more than one field from a related file, such as item description, price, and stock number. That’s where *portals* come in. A portal is similar to the relational value list described in the last tip, but allows you to display multiple related fields per record. To set up a portal, follow these steps:

1. Switch to the Layout mode, and select the Portal Tool.
2. Click and drag a rectangle where you want the related fields to be displayed.
3. In the Portal Setup dialog box, choose the relationship you want to use. (You can create a new relationship if one doesn’t already exist.)
4. Choose Options from the Format section of the dialog box, and click OK. You’ll see a rectangle with the number of repetitions you specified.
5. Drag the fields you want to use into the top row of the portal. Select related fields in the Specify Field dialog box that appears. (Names of related fields are preceded by two colons, as in ::Last Name.)



You use the Portal Setup dialog to specify the relationship on which the portal will be based, as well as set display options for the portal.

When you switch to Browse mode, you’ll see fields from the records of the related file you specified.



HOT TIPS

Cross-platform printing (DC). Scripts that include Page Setup/Print Setup and Print steps are not totally cross-platform compatible—even if you're printing to the same printer from both platforms.

The Print steps in ScriptMaker rely on the current printer driver in order to find out what paper sizes, orientation, and so on are available. Even if Macintoshes and PCs are connected to the same printer, the drivers can be significantly different, so FileMaker is unable to totally restore page setup and print options cross-platform. To work around this, you have three options:

1. Create a conditional test in your scripts that uses Status (CurrentPlatform) to determine what computer is being used and Status (CurrentPrinterName) to select the needed print script for either Macintosh or Windows 95. In order for this to work, you must create the Macintosh script on the Mac and the Windows 95 script on the PC.

```
If ["Status(CurrentPlatform) = 1"]
  If ["PatternCount(Status(CurrentPrinterName), "LaserWriter")>0"]
    Go to Layout ["LaserWriter Layout"]
    Page Setup [Restore, No dialog]
  Else
    If ["PatternCount(Status(CurrentPrinterName), "ImageWriter") > 0"]
      Go to Layout ["ImageWriter Layout"]
      Page Setup [Restore, No dialog]
    Else
      If ["PatternCount(Status(CurrentPrinterName), "StyleWriter") >0"]
        Go to Layout ["StyleWriter Layout"]
        Page Setup [Restore, No dialog]
      End If
    End If
  End If
End If
```

2. Create separate scripts for Macintosh and Windows 95.
3. Do not select Perform Without Dialog. This will allow you to change the setup options before you print as part of the script.

