Patricia Seybold's

UNIX IN THE OFFICE

PRODUCTS • TRENDS • ISSUES • ANALYSIS

VOL. 2, NO. 6
ISSN: 0887-3054
JUNE, 1987

INSIDE

•FEATURE•

DBMS Comparison........Page 22
A three-page comparison chart highlights the features of the Ingres, Informix-SQL, and Unify databases.

•DEPARTMENTS•

The Vision..........................Page 2
What does the future hold for the office computing environment and for Unix in particular? This month's editorial looks at the basics of a well designed office computer system.

Edging Out the Competition..............Page 25
The Edge 2000 processors from Edge Computer offer OEMs a fast and inexpensive product which also supports Pick under Unix.

Facets of the Pyramid....................Page 27
Pyramid Technology Corporation makes a move to the commercial arena as well as introducing a new "minimainframe" hot box called the Series 9000.

•NEWS•

The Series 2000, Altos Computer's 386 system, • Arete Systems Corporation's low end Arete 800 multiprocessor system • Navy's choice is Santa Cruz Operation's Xenix System V • Three new products from Oracle Corporation........Page 29

The Ingres Relational DBMS

Riding the Wave

By Judith R. Davis

THE RELATIONAL DATABASE is clearly the rising star in the evolution of the database management system (DBMS) in the office. The relational database model has become the tool of choice. It allows great flexibility in developing ad hoc queries and reports, and it is much easier for the end user to learn and use than other types of DBMSs (such as the more traditional hierarchical and networked databases found on mainframes and minis). Two other *(continued on page 3)*

• EDITORIAL •

Blueprint for An Office System: 1987

This is the direction the industry is taking. How closely does your vision map to this model?

By Patricia B. Seybold

A CONSENSUS IS building among product planners about the way an integrated office system should be designed. Here are some of the generally agreed-upon elements:

BUILDING BLOCKS. Most users will use PCs as workstations. These might be IBM-compatibles or Macintoshes. If they are IBM's, we might as well assume we are talking about the newer PS/2 machines with graphics support. If they are not IBM's or equivalents, then we can assume they will be Macs.

These PCs will be networked together, creating a seamless environment for the users. Whether that networking is accomplished by linking PCs to a LAN server (such as a micro or minicomputer) or whether it is handled via software distributed among the PCs is not important.

USER INTERFACE. The user interface is fast, graphical, and supports a well-integrated multitasking environment. This means that a user can quickly flip from one application to another. And applications can be linked to one another so that conditions in one program trigger activities in another program.

Of course, the PC will function as the user's "window on the world." It will be possible for the user to transparently access information and applications from anywhere in the world.

USER PROGRAMMING. The user will be able to customize an environment both by tuning the user interface to his or her preference and by developing his or her own applications. End-user application development will be supported by a new-generation macro language (we need a better term) that extends across applications. The user will be able to "teach" a procedure to the system by executing a set of commands and operations once. Then he or she can call up and refine the procedure, and nest conditions and loops.

It is critical that the same calls and procedures work across applications no matter what the heritage of the applications. Therefore, a de facto standard is needed in the area of universal macros. Both Microsoft and Apple are hard at work defining their own object-oriented macro languages to sit on top of their respective operating systems.

UNDERLYING UTILITIES. The power that will be unleashed by this sort of end-user tool will be considerable because of the underlying utilities these "programs" can leverage. Utilities will include underlying databases, networking and communications capabilities, and directories for mail and peripherals. While the tools described above (workstations, user interface, and cross-application macros) are clearly derived from the PC world, these fundamental utilities are more likely to emerge from the system suppliers.

WHAT ROLE DOES UNIX PLAY? Unix can provide the backdrop for what is becoming a predominantly PC world. We see it as fertile ground for the creation and evolution of the underlying utilities required to deliver transparent power to the desktop. ◐

Patricia Seybold's
Office
Computing
Group

Editor-in-Chief PATRICIA B. SEYBOLD
Managing Editor RONNI T. MARSHAK

Senior Editor
MICHAEL D. MILLIKIN
News Editor
JUDITH S. HURWITZ
Marketing Director
BRUCE A. F. POLSKY
Subscriptions
RICHARD ALLSBROOK JR.
Customer Service Coordinator
DEBORAH A. HAY

Associate Editors
JUDITH R. DAVIS
11 Ellery Square, Cambridge, Massachusetts 02138
Telephone: (617) 876-4081
DAVID L. TERRIE
135 Vernon Road, Scituate, Massachusetts 02066
Telephone: (617) 545-7401

148 State Street, Suite 612, Boston, Massachusetts 02109 Telephone: (617) 742-5200

•INGRES•

(continued from page 1) factors are making relational DBMSs more attractive: the dramatically improved performance of the DBMS and the price/performance of the hardware. These developments mitigate the pain of heavy memory and storage requirements often imposed by relational databases and make them more viable candidates for core applications that have usually been in the domain of networked or hierarchical DBMSs. Examples here are traditional transaction-processing systems such as accounting/general ledger, payroll, employee records, inventory control, and sales order entry.

An interesting new development comes from Sybase, which has just announced commercial availability of its relational database. This product is aimed squarely at the transaction-processing environment, with its huge databases accessed by hundreds or even thousands of users. Performance and availability (i.e., uptime) of the database become critical issues in transaction processing applications.

> *One applications developer we spoke to described achieving a true distributed database system as a "five-year progressive effort." We are just beginning to see the first stages of distributed database products hit the market.*

MULTIPLE PLATFORMS. Organizations want the DBMS to span a network, running in any environment that happens to be part of that network. This can include mainframes, minis, and PCs. A DBMS that runs in all three provides many benefits. Two prominent benefits are flexibility to develop applications on one system and port that application unchanged to another system, and a consistent user interface and applications development methodology.

The PC has become the workstation of choice, and it must be able to participate in a database network. This demand has significant implications for vendors porting their mini/mainframe DBMS products to the PC. Generally, these products have been designed for the professional applications developer. If these vendors want to supplant existing PC DBMS software (dBase III Plus, Rbase System V, Paradox, et al.), they will have to compete successfully with the user interface, friendliness, and ease of use of these products. PC end users will not tolerate having to write SQL statements on a command line to retrieve data or to write a program (regardless of how nonprocedural it is) to generate a simple report. Easy to use, interactive tools for these functions and for end-user applications development are becoming required.

SQL. IBM's Structured Query Language, SQL, has become the standard query language for relational databases. Compatibility with SQL is now mandatory for players in the relational DBMS arena.

DISTRIBUTED PROCESSING. Vendors of the major relational DBMSs are busy modifying the architecture of their products to allow the front-end and back-end modules to run on different machines. The front end of a DBMS includes the user interface and the application; the back end includes the data manager and the database itself. This trend is a response to the increasing popularity of networked systems at all levels and the need to improve performance across a network. The major benefits of this architecture are a reduction in network traffic as users access the database and the opportunity to offload processing from the CPU to the workstation. Vendors are also developing a database server architecture in which a single back end serves multiple front ends.

DISTRIBUTED DATABASES. Another—and more complex—architectural modification to a DBMS is to allow the database itself to be distributed across multiple systems. In its simplest form, this means that a user on one system can perform a join across tables that reside on two different systems and retrieve data from both tables (systems) in a single query. The fact that the tables are on different systems is transparent to the user.

The retrieval process is the easy part. But when it comes to updating data across multiple systems in a single transaction, product designers begin to sweat. And the really tricky part is trying to update a view that is a combination of tables on multiple systems. The transaction management function becomes very complex.

One applications developer we spoke to described achieving a true distributed database system as a "five-year progressive effort." We are just beginning to see user organizations taking advantage of these distributed database products.

Where Does Ingres Fit?

Relational Technology Incorporated, is well on its way to providing all of these capabilities in its Ingres relational DBMS. In this issue, we take a look at Ingres from the perspectives of both the end user and the experienced applications developer. In the past, we reviewed two other relational DBMSs: Unify from Unify Corporation (Vol. 1, No. 3) and Informix-SQL from Informix Software (Vol. 1, No. 8). We have added Ingres to the comparison chart presented in the Informix article to give you a quick look at how each stacks up in terms of features. We also refer the reader to these two previous issues for additional background on relational DBMSs.

Company Background

The Ingres DBMS grew out of a research project called Ingres (which originally was an acronym for Integrated Graphics and Retrieval System) conducted at the University of California (UC) at Berkeley in the early 1970s. Lawrence Rowe, Michael Stonebraker, and Eugene Wong, three professors in electrical engineering and computer science, developed the Ingres relational DBMS at about the same time that Dr. E. F. Codd's research on relational database concepts was evolving at IBM. A prototype of Ingres was available in 1975 and was shipped as part of the Berkeley Unix operating system. Within a year, Ingres was installed in approximately 150 sites.

Relational Technology correctly anticipated the DEC VAX environment as an up-and-coming market with tremendous potential. As a result, Ingres was commercially introduced on the DEC VAX under VMS.

User demands for enhancements eventually outstripped the resources of the research project, and the developers decided to market the product commercially. Relational Technology was incorporated in 1980 by the three original architects of the product plus two other participants, one of whom was Gary Morganthaler, a former consultant with McKinsey and Company. Morganthaler is now chairman and CEO of Relational Technology. Rowe, Stonebraker, and Wong, still professors at UC Berkeley, maintain their ties with the company as consulting vice presidents. Stonebraker is now involved in another research project, Postgres, exploring the integration of relational DBMSs and artificial intelligence.

Relational Technology is a privately held company. Revenues have grown steadily from $900,000 in fiscal year 1982 to $28.1 million in 1986. While it is still smaller than its major rival, Oracle (over $50 million in revenues), it is considerably larger than Informix Software and Unify Corporation. The company does not report income, but indicates that it has been profitable for each of the past five years. Relational Technology now employs approximately 420 people.

Product Positioning

Relational Technology's focal point is its Ingres relational DBMS. The formal moniker for the product is "Ingres, the Distributed SQL Relational Database System." As you can see, the company is committed to adherence to the evolving SQL standard and to the concept of distributed databases. Development efforts over the past five years have been in the following areas:

- Porting the product to additional hardware and operating environments
- Enhancing both the functionality and performance of the product
- Providing both distributed processing and distributed database capabilities
- Providing better compatibility with other DBMSs

PRODUCT HISTORY. While the original Ingres prototype was developed in Unix, Relational Technology correctly anticipated the DEC VAX environment as an up-and-coming market with tremendous potential. As a result, Ingres was commercially introduced on the DEC VAX under VMS in 1981. At the time, the Unix market was still primarily academic. It started to move into the scientific world with AT&T's commitment to Unix, and it only recently has penetrated the corporate market. In 1983, Ingres became available in a number of Unix environments, including Amdahl, Pyramid, Computer Consoles Incorporated (CCI), and DEC VAX. Ingres for the IBM mainframe world (VM/CMS) was announced in late 1985, and the product shipped in early 1986. Since then, Relational Technology has introduced its distributed database product, Ingres/Star, and, most recently, Ingres for Personal Computers. Relational Technology is also committed to supporting the IBM MVS operating environment and will formally announce an MVS version of Ingres later this year.

Ingres Relational DBMS. The current release of Ingres is 5.0, announced in September 1986. In addition to the back-end relational database manager, the Ingres product includes:

- Ingres/Menu—a menu interface tying together all of the Ingres subsystems
- Two query languages—Structured Query Language (SQL) and QUEry Language (QUEL). (SQL was a new addition with Release 5.0.)
- Ingres/Query—a query-by-example facility also known as QBF, or Query-By-Forms
- Ingres/Reports—a nonprocedural report writer and a forms-driven report writer called RBF, or Report-By-Forms
- Ingres/Forms—the forms run-time system and Vifred, the Visual-Forms-Editor

Optional modules provide embedded SQL or QUEL, an appli-

The Ingres Family of Products

Product	Introduced	Currently Runs On	Cost
Ingres Distributed SQL Relational Database System, including: • SQL and QUEL query languages • Ingres/Menu • Ingres/Query (QBF or Query-By-Forms) • Ingres/Reports (Report Writer and RBF or Report-By-Forms) • Ingres/Forms (Forms Run-Time System and Vifred Visual-Forms-Editor)	1981 1983 1985	DEC VAX/VMS Unix—Over 24 environments IBM VM/CMS	$3,000 on Unix workstation up to $140,000 on IBM mainframe
ESQL/C and EQUEL/C (embedded SQL and QUEL preprocessor for C)	3/87	Same as Above DOS	10% of basic Ingres licensing fee $400 on PC
Ingres/Applications for SQL and QUEL (ABF or Applications-By-Forms which includes a 4GL)	early 1984 3/87	Same as Above DOS	25% of basic Ingres licensing fee $500 on PC
Ingres/Embedded SQL and QUEL preprocessors for Cobol, Basic, ADA, Pascal, Fortran, PL/1		DEC VAX/VMS Unix IBM VM/CMS	10% of basic Ingres licensing fee for each
Ingres/VAXcluster option		DEC VAX/VMS	Customer buys license for highest machine class in cluster and negotiates balance
Ingres/Graphics (Vigraph or Visual-Graphics-Editor)	1985	DEC VAX/VMS Unix	20% of basic Ingres licensing fee
Ingres/PC Link (Personal Computer Decision Support Connection)	1985	DEC VAX/VMS Unix IBM VM/CMS DOS	10% of basic Ingres licensing fee $200 on PC
Ingres/Net (database networking support system)	1983 5/87	DEC VAX/VMS Unix DOS	15% of basic Ingres licensing fee $200 on PC
Ingres/Star (distributed data manager)	3/87	DEC VAX/VMS	Percentage of Ingres licensing fee; varies depending on configuration
Ingres for Personal Computers, including: • SQL and QUEL query languages • Ingres/Query (QBF or Query-By-Forms) • Ingres/Reports (ReportWriter and RBF or Report-By-Forms • Ingres/Menu • Ingres/Forms (Forms Run-Time System and Vifred Visual-Forms-Editor) (RBF and Vigraph are not available for the PC product)	3/87	DOS	$950

cations developer and a fourth-generation programming language (4GL), and graphics.

Ingres/Net. Ingres/Net is Relational Technology's distributed processing product. Introduced in 1983, it allows an Ingres application running on one computer to access a single remote Ingres database. Initially limited to VMS, Ingres/Net now handles Unix as well as dissimilar operating systems (e.g., VMS <—> Unix, DOS —> Unix, DOS —> VMS). Ingres/Net only supports an asynch communications protocol on the PC, but support for DECnet/DOS in the VMS environment and TCP/IP for Unix is coming in the future. The PC can only act as the front end; an Ingres application running on Unix or VMS cannot yet access data on the PC.

Ingres/Star. There seems to be some controversy in the industry over whether anyone has a true distributed database product on the market yet. Ingres/Star was announced in June 1986 and formally went into production in March 1987. The company has shipped about 20 licenses for Ingres/Star and is waiting until its initial customers are far enough along in building applications to be able to talk about the product. Ingres/Star runs on VAX/VMS at present and will be extended to other operating environments. It is in beta test mode.

Ingres/Star is built upon Ingres/Net to allow the user to transparently access multiple Ingres databases in multiple locations. The first release provides the ability to retrieve data from multiple sites, limits updates within a transaction to one site, provides real-time updates of a primary table and deferred updates of copies of the table stored at other network locations, and allows query-optimizing in order to minimize communications traffic. Future releases will expand these capabilities to allow the update of multiple sites in a single transaction, concurrent updates of all copies of a table, support for IBM mainframes and the PC, and DBMS gateways to other SQL-based systems (IBM's DB2 and IMS first, followed, perhaps, by Oracle and DEC's Rdb).

Marketing Strategy

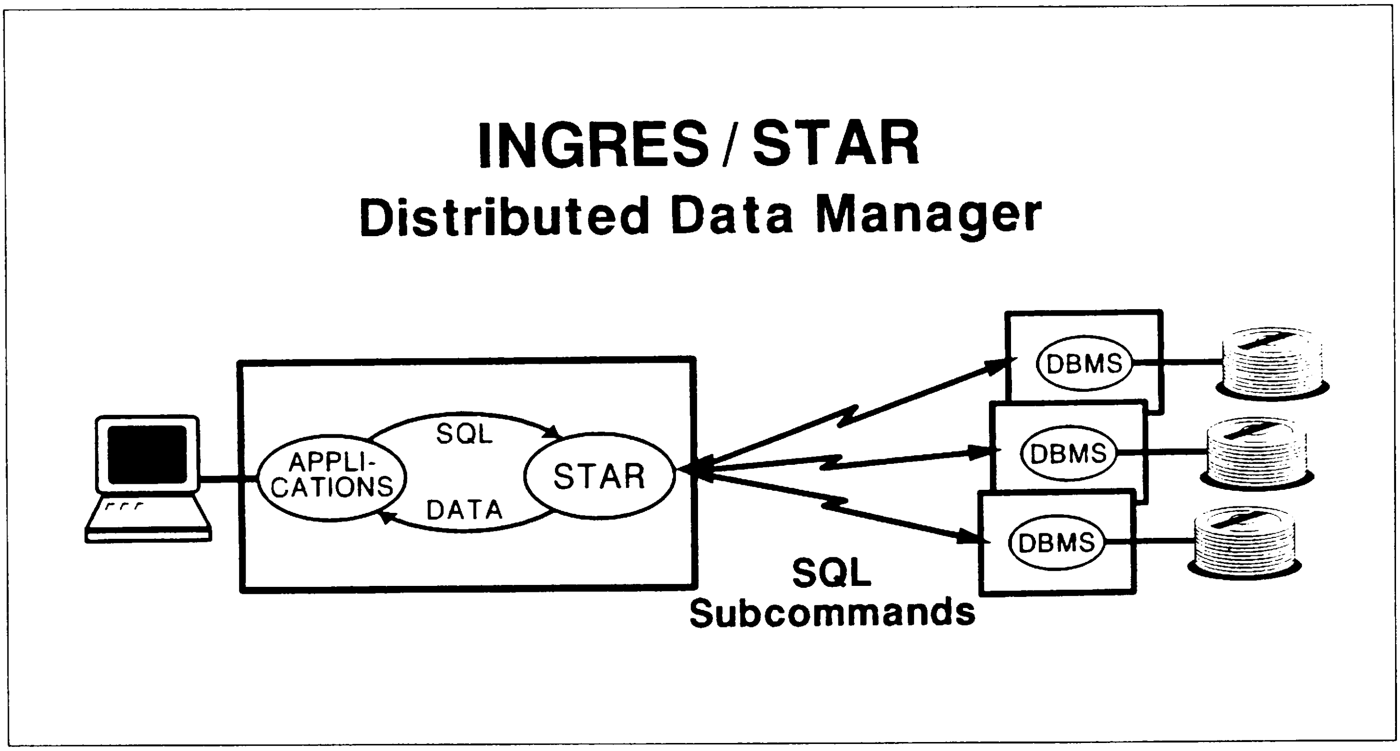
Relational Technology's marketing efforts have always been focused on Fortune 500 companies. Unlike some of its competitors, 80 to 90 percent of Relational Technology's sales are directly to customers, not through hardware manufacturers (OEMs) or value-added resellers (VARs). In addition, the product is clearly targeted for the applications developer rather than the end user.

OPERATING ENVIRONMENTS. Because of its emphasis on the Fortune 500, Relational Technology must be able to play successfully in the following computing environments:

- DEC VAX or a multivendor configuration with DEC VAX, IBM mainframes, and PCs.

- Unix, most likely on one of the larger systems (Pyramid, Alliant, VAX, Sequent, AT&T 3B, etc.) and/or workstations (Sun, Apollo, DEC VAXstation, IBM RT, HP 9000 series). Workstations are particularly prevalent in manufacturing and are often networked (a shared central database on a server, with the front end and applications running on remote nodes).

- Many IBM PCs.



Relational Technology believes that one of its major competitive advantages is the fact that Ingres runs in all of these environments. While a number of its competitors (Oracle, Informix, and Unify) are also following the same strategy, it gives the company a definite leg up on two other rather imposing competitors, DEC and IBM. The company's general approach is to port new releases of Ingres first to VAX/VMS, then to the appropriate Unix systems and to the IBM mainframe. Ingres does not run on several low-end Unix systems, such as Altos. The reason, according to the company, is that these systems are not prevalent in its target market. (It could also be because Ingres requires heavy commitments of memory and storage.) Relational Technology is strongly committed to the Unix workstation market.

BUSINESS SOLUTIONS. Relational Technology also describes its product development strategy as a commitment to providing business solutions in three major areas:

- Distributed solutions, the ability to access and integrate data across a wide variety of computers and operating systems. Products are Ingres/Net for access to a single remote database, Ingres/PCLink and Ingres/Star for the ability to distribute a database across multiple systems. The need for distributed solutions also means that the DBMS product must run at all three levels in an organization—mainframe, mini, and PC.

- Integrated tools for high productivity in applications development. These include the forms-based development tools Ingres provides and its 4GL.

- High-performance solutions to address transaction-oriented applications. According to Relational Technology, the cost per transaction per second using a relational database has improved by a factor of 50 in the last five years. The company sees the relational DBMS finally becoming competitive with other DBMS structures for transaction-based systems. The performance of Ingres has improved tenfold since its introduction, and the price/performance of the hardware itself has also improved dramatically.

SERVICE AND SUPPORT. Relational Technology has a strong commitment to customer service and support. Ingres maintenance releases are usually issued every quarter, and a major release is issued at least once a year. The company estimated that 25 percent of its staff is devoted to customer support functions.

MARKET SHARE. The company considers the following environments to be competition:

- DEC VAX/VMS. The main competitors are DEC's Rdb and Oracle. Relational Technology indicates that it now has over 3,000 Ingres licenses under VMS. Statistics released by Computer Intelligence Corporation indicate the following market shares for relational DBMSs on VAX/VMS:

Relational DBMSs		All DBMSs
37%	Ingres	19%
28%	DEC's Rdb	15%
22%	Oracle	12%
3%	Informix-SQL	
2%	Unify	
2%	Unipress	
2%	Boeing Computer Services	
5%	Others	14%
	DEC—nonrelational non-Rdb	21%
	In-house	9%
	Poise	4%
	Software House	3%
	Henco Software	3%

Source: Computer Intelligence Corp.

- Unix. The main competitors are Oracle, Informix, and Unify, with the primary competition coming from Oracle on the larger Unix systems.

- IBM mainframe. The main competitors are IBM's DB2 and Oracle.

- PC. The main competitors here are PC Focus and Oracle. Relational Technology's present approach is to position Ingres for PCs as a complementary product to DOS-based DBMSs such as dBase III, Rbase, and Paradox. The emphasis here is on Ingres/PC Link. This product, built on Network Innovations' Multiplex, allows the PC user to access and display Ingres databases in a familiar spreadsheet-type format, select specific data, and download it to the PC. The data can be converted to several popular PC formats and then accessed with PC applications (Lotus 1-2-3, dBase III, etc.).

PC Link is a key product because it allows the PC user access to networked Ingres databases and integration of Ingres data with familiar PC software. The company sees the relational database market growing so rapidly that there is room for everyone at the moment. The biggest challenge will be successfully managing the growth process.

VARS. Relational Technology has about 60 VARs at the present time (including Westinghouse, General Electric, and Pansophic). Although VARs have not been a principal marketing focus, Relational Technology sees an opportunity here and wants to expand its VAR market. And the company has grown to the point where it feels comfortable adding more elements to its business strategy. As a result, it has hired a VAR manager and launched a new VAR program. In a very recent development, Relational Technology signed a VAR agreement with Atex under which Atex will build its next generation publishing system on top of the Ingres relational DBMS.

OEMS. When Relational Technology first moved into the Unix market, sales were generally through OEMs, and most were exclusive agreements. Increasingly, Relational Technology is finding that it achieves the best long-term customer satisfaction when it handles product support directly. Thus, the current trend is away from exclusive agreements and more toward joint marketing agreements with the hardware manufacturer. Under this arrangement, Relational Technology sells and supports the Ingres product but works jointly with the hardware vendor in making the sale. So, while the company still signs exclusive OEM agreements, a greater balance exists among exclusive, nonexclusive, and joint marketing agreements. Relational Technology has OEM agreements with approximately 20 hardware manufacturers.

SALES ORGANIZATION. Relational Technology has 20 regional sales offices in North America. Outside North America, the company's international headquarters in London distributes Ingres through both direct sales offices and distributors.

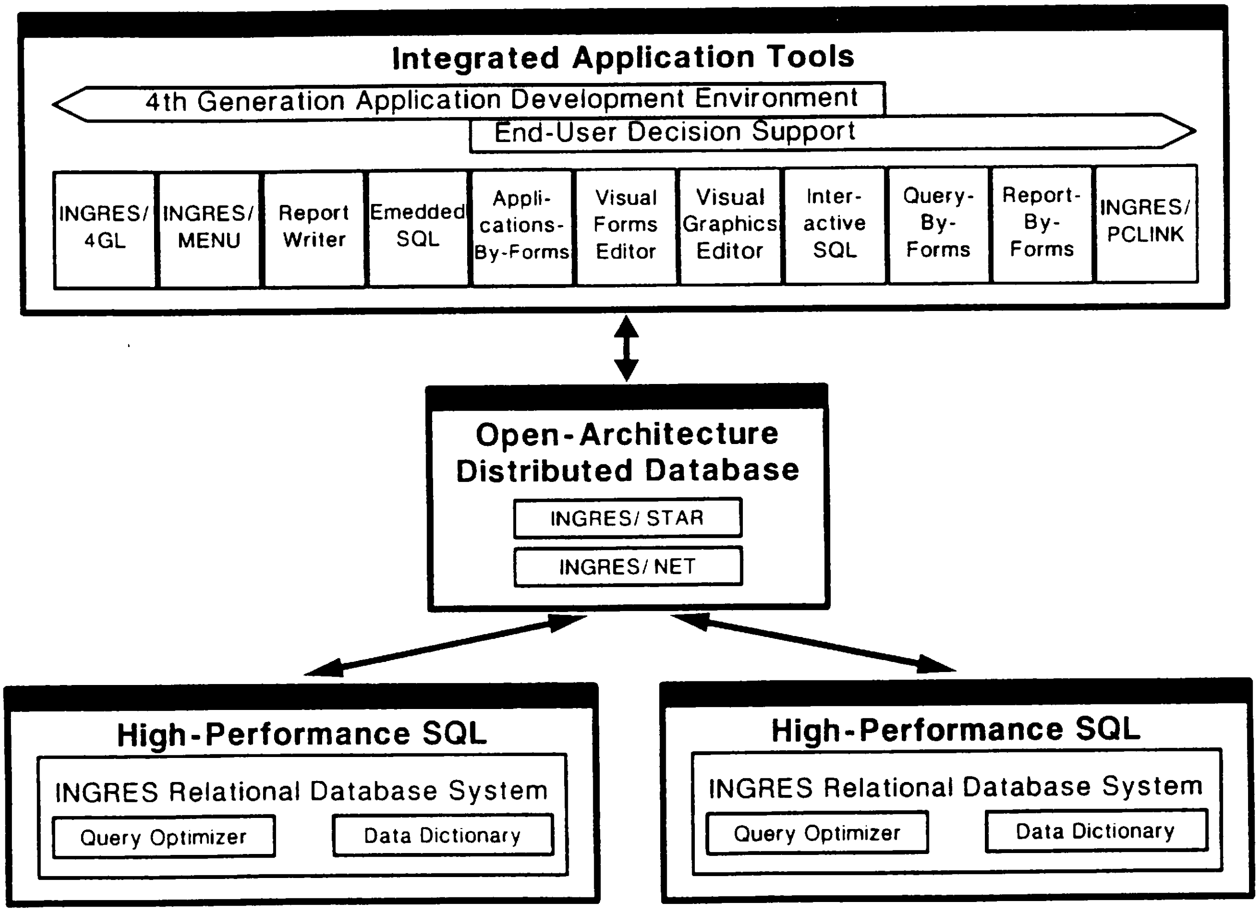
Relational Technology, Inc.
1080 Marina Village Parkway
Alameda, CA 94501
(415) 769-1400

USING INGRES: THE END USER'S PERSPECTIVE

We evaluated Ingres Release 5.0 on an NCR Tower 32 under Unix System V. We had no formal training, and we installed and learned the program on our own, using the manuals for reference.

DATABASE PARAMETERS. Ingres has no limit to the number of tables in a database, rows (records) in a table, or

The Distributed SQL Relational Database System



indexes for each table. It allows 127 columns (fields) in a table and 2,000 bytes per record, which is more limited than Informix-SQL or Unify. The company's philosophy is that for a relational database to adequately manipulate many tables, it is better to have shorter rows. In addition, the default page size in some systems is 2,000 characters, so this becomes a convenient amount of information to process as an entity. The company also pointed out that the user can create views of the database that are much larger than 2,000 characters per row.

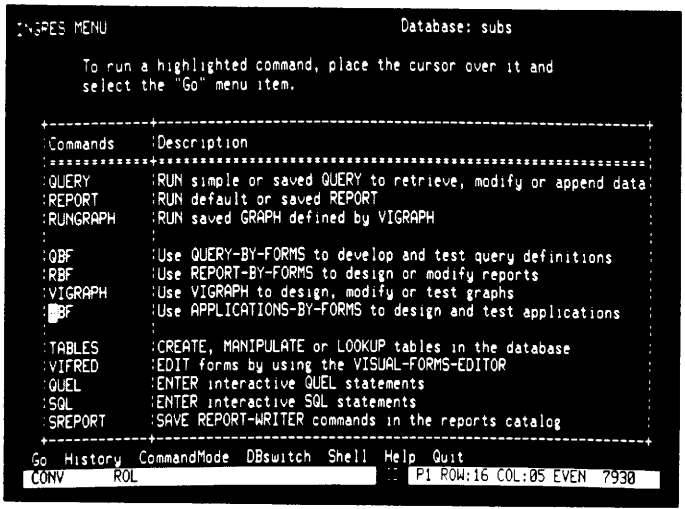
SYSTEM REQUIREMENTS. Memory and storage requirements are hefty. The minimum memory requirement is 4MB. This will support two users; for each additional user, you add .5MB. The Ingres software takes up 22MB of storage space.

User Interface

As we mentioned above, Ingres is composed of several modules that are integrated under a menu-driven interface called Ingres/Menu (see photo below). Each module can also be accessed directly from the Unix system prompt.

Ingres/Menu uses a frame concept as the user interface. Each frame is a single screen of information divided into two sections: a form and a single-line menu across the bottom of the screen. The user moves between the form and the menu with the menu key (either ESC or PF1, depending on the terminal).

MENUS. While the overall concept works fine, we encountered a number of frustrating annoyances. On most terminals, the menu items are mapped to function keys and can be invoked directly. If they are not mapped to function keys (ours weren't), the user enters the desired option at the menu line prompt. But if you make a mistake typing your option at the menu line, you cannot backspace and correct your error. While we could erase the entry on some screens and start over, on others, we were thrown out to the operating system when we tried to erase it.



```
INGRES MENU                                      Database: subs

      To run a highlighted command, place the cursor over it and
      select the "Go" menu item.

   +-----------+----------------------------------------------------+
   :Commands   :Description                                         :
   :===========+====================================================:
   :QUERY      :RUN simple or saved QUERY to retrieve, modify or append data:
   :REPORT     :RUN default or saved REPORT                         :
   :RUNGRAPH   :RUN saved GRAPH defined by VIGRAPH                  :
   :           :                                                    :
   :QBF        :Use QUERY-BY-FORMS to develop and test query definitions :
   :RBF        :Use REPORT-BY-FORMS to design or modify reports     :
   :VIGRAPH    :Use VIGRAPH to design, modify or test graphs        :
   :ABF        :Use APPLICATIONS-BY-FORMS to design and test applications :
   :           :                                                    :
   :TABLES     :CREATE, MANIPULATE or LOOKUP tables in the database :
   :VIFRED     :EDIT forms by using the VISUAL-FORMS-EDITOR         :
   :QUEL       :ENTER interactive QUEL statements                   :
   :SQL        :ENTER interactive SQL statements                    :
   :SREPORT    :SAVE REPORT-WRITER commands in the reports catalog  :
   +-----------+----------------------------------------------------+
  Go  History  CommandMode  DBswitch  Shell  Help  Quit
 CONV      ROL                                 P1 ROW:16 COL:05 EVEN  7930
```

On the Ingres main menu, you highlight on the form the module you want to invoke, move the cursor to the menu line, and select the appropriate action item. On most terminals, the menu items at the bottom of the screen are mapped to function keys which are displayed next to the option name.

You need to type in only enough characters to identify a unique item. But there's the rub: Most Ingres menus have two or more commands that begin with the same letter, such as QBFNames and Quit, History and Help. You would think the menu at the bottom of the screen would indicate the additional letters required to differentiate these commands (EXamine and ENd, for example), but not so. Guaranteed frustration. And, instead of just beeping, displaying an error message, and letting you edit the entry, Ingres gives you a message, tells you to press Enter to return to the menu line, and makes you start over.

Scrolling through the items on the main Ingres/Menu is not circular; when you reach the bottom, you must scroll back up to get to previous items.

Ingres always places you on the form, not at the menu prompt, so you must hit the menu key (ESC) to go to the menu line. In some cases, this doesn't make sense. We found that we sometimes needed two menu items in a row and would prefer to be left on the menu with the option to enter the form.

It would also be nice if Ingres would remember the last options you used on a particular screen and keep these as the defaults.

FUNCTION KEYS. The user can custom-map the menu options in Ingres to function or control keys on the keyboard and invoke these with one or two quick keystrokes. Ingres has a nice feature: Once you map an option to a key, that key is automatically displayed on the menu line as a reminder. Additional flexibility comes with the ability for custom software to dynamically change the key-mapping definitions.

HELP. Ingres provides context-sensitive "Help" in a format consistent with other Ingres screens, displaying Help text on a full-screen form with a menu across the bottom. Generally, the Help function provides good information.

However, in order to maintain context for the user, we would prefer to see Help text in a separate window. There is also no overall index of Help topics through which the user can browse, which would be valuable. In addition, we would like to see more Help information available on procedures at a conceptual level. Currently, Help focuses more on what to do on a particular screen (e.g., how to fill in the screen form).

The "Find" feature (which looks for the first occurrence of a character string in the Help text) is valuable. You can also get help on the Help function—a nice touch.

ERROR HANDLING. Errors are generally handled well. A few times, we hit a weird key combination, got strange messages, and were thrown back to the main menu. But we always got an error message, and the message was usually clear enough for us to understand where we went wrong.

DOCUMENTATION. The documentation is quite intimidating. Ingres comes with four large, heavy notebooks labeled Volumes 1 through 4. We started with 1. The very first section is entitled "The Ingres Terminal User's Guide," which sounded rather foreboding, not to mention technical. We were hoping to find something a little friendlier, like an introduction or a "Read Me First" pamphlet. Yet this is where the user starts with the overview of Ingres. Ingres needs a tutorial for end users.

The manuals are nicely and clearly organized into separate sections covering each module of the program. However, there is no overall index covering all four volumes. Each section has its own index, and you must know which modules to search for information. Some of the sections are quite large; tabbed subsections would be helpful. We would also prefer the manuals to be organized by database function rather than by module.

The documentation is not particularly inviting, it is not as complete and informative as it should be, and we found it difficult to use effectively.

The manuals are formatted as mostly text and would benefit greatly from the use of bulleted lists, graphics, and color. Packaging in a smaller notebook (like standard PC software manuals) would be handier and friendlier.

In terms of content, the manuals are too repetitive and too formal. They need to be simpler, lighter, and more straightforward. Too many big words and document titles are repeated throughout. We felt as if we had already read some points three times, and we weren't going anywhere.

In many cases, the documentation does not provide enough detailed information and illustrations. And information is often presented in the wrong order. For example, the documentation tells you how to run a query and a report before it tells you how to create one.

The bottom line here is that the documentation is not particularly inviting, it is not as complete and informative as it should be, and we found it difficult to use effectively.

INSTALLATION. Installing Ingres is much more difficult and time-consuming than installing either Informix-SQL or Unify. You must be very knowledgeable in both the Unix operating system and Ingres to successfully install the product. We do not recommend this be done by the end user.

Database Design

CREATING A DATABASE. Creating a database is not included within Ingres/Menu. When we questioned Relational Technology about this, the company's explanation was that the database is the umbrella structure; therefore, you must have already created a database in order to use Ingres/Menu. This relationship is the opposite of what we expected. It seems to us that the menu system should be the umbrella containing all of the Ingres functions.

Once you know how, creating a database is quite simple. You execute the "createdb" command at the Unix prompt and give a database name.

At this point, Ingres first enters information in the overall data dictionary ("dbdb" or database database) about the new database (name, owner, date, and its physical location). Ingres then builds a subdirectory for the database within the Ingres installation. Next, Ingres builds a data dictionary or system catalog for the database. This is where information about the specific database is stored: tables, fields, access permissions, etc. Every table in the database is a separate Unix file.

Once you have created a database, you enter Ingres/Menu in order to define tables, enter data, perform queries, and develop reports. Within Ingres/Menu, you can switch to a different database (but not create a new one) without exiting from the program.

The "accessdb" command, also entered at the Unix system prompt, accesses the Ingres Control Program and provides summary information about what databases have been created, where they are located, who the users are, and what permissions exist. You can also add, modify, and delete Ingres users.

TABLES DEFINITION. Creating a table is also quite straightforward. You specify the table name, field (column) names, and the type of data that goes in each column.

Ingres supports the following data types:

- Character—Maximum of 255 characters.
- Vchar—An extended set of ASCII control characters (up to 2,000 characters).
- Integer—Numeric data without decimal points (integer, small integer, or integer1). The difference among these three is the internal-storage representation: 4 bytes (+/-2,147,483,647), 2 bytes (+/-32,767), or 1 byte (+/-127), respectively.
- Floating point—Numeric data with decimal points; single precision (4 bytes, 7 decimal-digit precision) or double precision (8 bytes, 16 decimal-digit precision).

- Money—A special form of floating-point field with two decimal places, preceded by a dollar sign.

- Date—Ingres supports date and arithmetic manipulation as well as several different date formats.

Table names and field names are limited to 12 alphanumeric characters; the first character must be a letter. Field names may contain letters, digits, or underscore.

MODIFYING TABLE STRUCTURE. Much to our dismay, Ingres only allows you to modify the structure of a table (e.g., add, change, or delete fields) by using the query language, SQL or QUEL. And you cannot simply add or change a new field; you must rebuild the entire table. We want to simply edit the field information on a screen similar to the one we used to create the table (a la dBase III).

Creating and Modifying Screen Forms

You use the Ingres Visual-Forms-Editor, Vifred, to create and modify screen forms. This module is easy to learn and use, and it provides a great deal of flexibility. Once we got familiar with its capabilities, we found Vifred a very valuable and well-designed facility for creating customized forms.

Ingres will generate a default form for a table that can be used as is or can be edited. Editing is interactive; the menu at the bottom of the screen assists you in painting the screen in the desired layout.

Forms can contain trim (static text for form titles and explanatory statements), fields, and table fields (which allow you to display multiple rows on the screen rather than showing just one row at a time). You can easily insert blank lines and delete unneeded components from the form. And Vifred has an "Undo" function that comes in handy.

Within fields, the field title (label) and the data window (data format and number of characters) are edited independently. These two components can be moved together as a unit or separately. "Move" options include left, center, and right justification, and the ability to place the item at a specific location on the screen. Data formats supported on forms are somewhat different than data types supported when you create tables. This can be a little confusing. For example, a date is represented as a character format on a form but is a specific date data-type within a table.

For each field you can also:

- Assign attributes—such as boxing the field on the screen, making an entry mandatory, converting the entry to upper or lower case, and various display options (blinking, reverse video, etc.).

- Assign a default value.

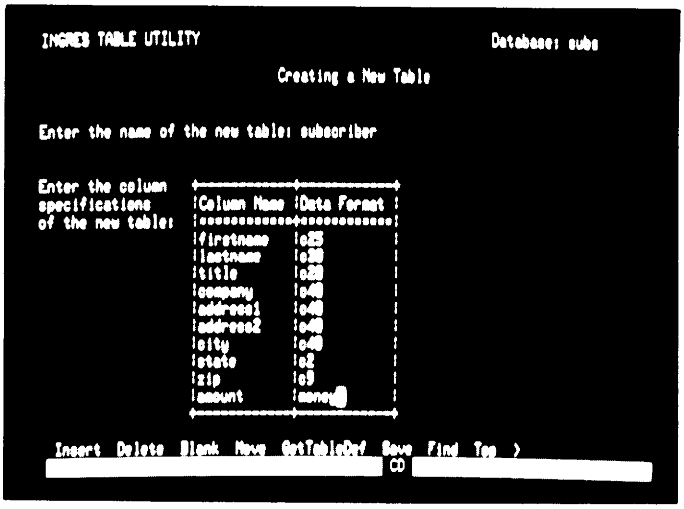
- Define a validation check, such as comparing an entry with another field, or comparing it to a constant, restricting entry to a specified list of values or to a set of values in another table. You can enter multiple validation checks using conjunctions (and, or, not).

- Specify an error message to display if the entry fails the validation check.

- Specify a color on a color terminal.

- Change the order in which the cursor moves from field to field on the form.

MULTIPLE TABLES. An essential feature for a relational database is the ability to define a form containing information from two or more tables. This is quite easy to do in Ingres, once you get the hang of it. (A typical application for this is displaying information from two tables that have a field in common, such as a customer master file and a customer billing file.) You first define a join definition ("JoinDef" in Ingres parlance), indicating what tables are related, how the tables are connected (i.e., what field(s) they have in common on which the join is performed), and whether the relationship is master/master or master/detail. Ingres is smart enough to try to figure out what field the tables have in common and will give you a default for this. The process of creating a join definition takes place within the Query-By-Forms module. Once the join is defined, you are free to create a default form and edit it as easily as a form generated from a single table. Ingres knows which fields belong to which table.



INGRES TABLE UTILITY Database: subs

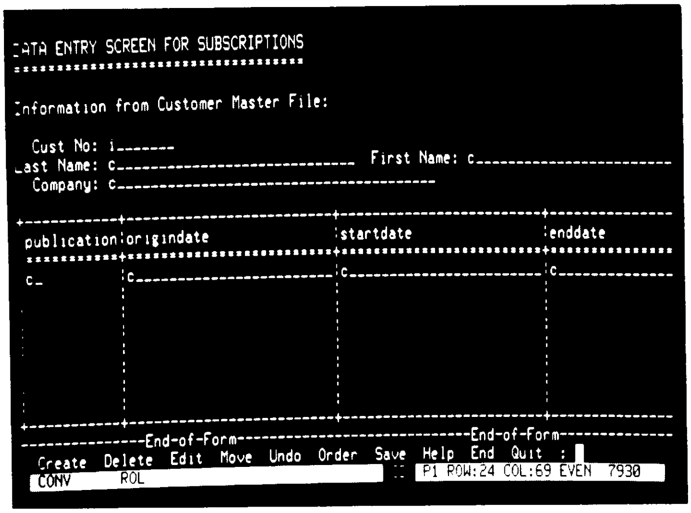
Creating a New Table

Enter the name of the new table: subscriber

Enter the column specifications of the new table:

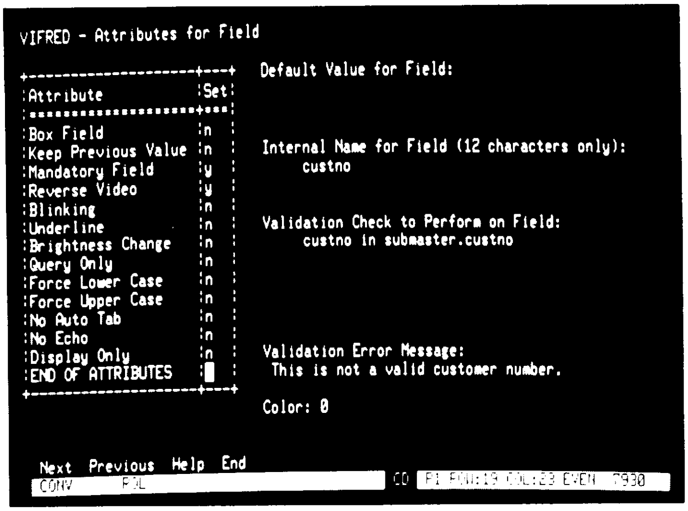
Column Name	Data Format
firstname	c25
lastname	c30
title	c20
company	c40
address1	c40
address2	c40
city	c40
state	c2
zip	c9
amount	money

On this frame, you define the fields in a table.



```
:ATA ENTRY SCREEN FOR SUBSCRIPTIONS
=================================

:nformation from Customer Master File:

  Cust No: i-------
_ast Name: c-------------------------  First Name: c----------------------
  Company: c-------------------------------------

+----------+---------------------+---------------------+-------------
publication:origindate           :startdate            :enddate
==========+=====================+=====================+=============
c_        :c---------------------:c---------------------:c------------
          :                     :                     :
          :                     :                     :
          :                     :                     :
+----------+---------------------+---------------------+-------------
--------------End-of-Form----------------------------End-of-Form-------------
  Create  Delete  Edit  Move  Undo  Order  Save  Help  End  Quit  :
 CONV      ROL                                   ::  P1 ROW:24 COL:69 EVEN  7930
```

An example of the Vifred forms-design environment.



```
VIFRED - Attributes for Field

+--------------------+---+  Default Value for Field:
:Attribute           :Set:
:====================+===:
:Box Field           :n  :
:Keep Previous Value :n  :  Internal Name for Field (12 characters only):
:Mandatory Field     :y  :       custno
:Reverse Video       :y  :
:Blinking            :n  :
:Underline           :n  :  Validation Check to Perform on Field:
:Brightness Change   :n  :       custno in submaster.custno
:Query Only          :n  :
:Force Lower Case    :n  :
:Force Upper Case    :n  :
:No Auto Tab         :n  :
:No Echo             :n  :
:Display Only        :n  :  Validation Error Message:
:END OF ATTRIBUTES   :   :   This is not a valid customer number.
+--------------------+---+
                            Color: 0


  Next  Previous  Help  End
 CONV      ROL                                 CD  P1 ROW:19 COL:23 EVEN  7930
```

On this screen, you define attributes for the field and validation rules for entering data.

Data Entry and Editing

Data is entered in a table and edited in the Query-By-Forms (QBF) module. After you select QBF and specify the table name or join definition name, QBF will give you a choice of Append, Retrieve, or Update. "Append" allows you to enter new rows into a table. "Update" enters a mode for editing existing records. QBF then displays the form associated with the table or join definition (it creates a default form if you don't have a customized one). In Update, you first execute a query to locate the record(s) you wish to edit. Any validation criteria defined on the form will be in effect when data is entered through QBF.

You can also enter and edit data in tables using either of the Ingres query languages, SQL or QUEL. Both include data manipulation statements.

Indexing

When you create a table, the data is stored in a default "heap" structure: The rows are random and unordered in the table, new rows are added at the bottom of the table, and duplicate rows are not removed from the table. When Ingres queries a heap table, it must look at all rows because the rows are random. (Heap structure corresponds to buffered sequential access.)

You can modify the storage structure of a file and create indexes using either SQL or QUEL. Options include:

- "Sorted Heap" is essentially a "one-time" ordering of the table and is not dynamically maintained; it only affects rows in the table at the time the modification takes place. New rows are still added to the bottom of the table.
- "Hash" is a storage structure that accelerates exact match queries.
- "ISAM" (indexed sequential access method) is appropriate for queries based on a range of values. Like the Sorted Heap index method, an ISAM index is not dynamically updated. You would use this type of indexing on a table that is not updated frequently but on which you perform frequent lookups and need fast performance.
- "B-tree" is similar to Isam, but a B-tree index is maintained as the table is updated. The method Ingres uses for B-tree indexing is to actually store the index with the data; pages of the index are interleaved with pages of data. The benefit of this method is that, as Ingres traverses the tree structure of the index, it is always within a page or so of the data. According to Relational Technology, performance using B-tree indexes is approaching that of Isam indexes. Therefore, B-tree may ultimately replace Isam as an indexing method, particularly since B-trees are dynamically updated and always reflect current data in the table.
- "Unique" is a specification that eliminates duplicate rows. If you have a table with a Unique index, it will not allow you to enter a record with a duplicate value in the indexed field.

A table can only be stored in one structure at a time, so the storage structure becomes a primary index for accessing the table. If necessary, you can create secondary indexes (the number is unlimited). Ingres has a powerful query optimizer that looks at the indexes available for a table and decides what access method is the most efficient for retrieving the requested data. You can also modify the storage structure of a table.

Indexing is not explained particularly well, and the fact that it is done through the query language makes it appear difficult and complex. We would like to have the option to specify index keys when a table is created and have the indexing done automatically.

Queries

Ingres offers you three ways to query a database: Query-By-Forms (QBF), QUEL, or SQL.

QUERY-BY-FORMS. QBF is a forms-based query facility that is also known as query-by-example. In typical fashion, a form is displayed on the screen, the user fills in search criteria in the appropriate fields, and the results of the query are returned on the same form. QBF is more flexible than many query-by-example facilities. It supports both "and" and "or" searches and the use of parentheses to specify the order for multiple criteria. And the criteria do not have to fit in the data window on the screen. For lengthy search criteria, the entry scrolls within the data window.

QBF works on what Ingres calls "query targets." A query target can be a single table, a join definition, or a QBFName. A QBFName is simply a form associated with a table or a join definition. A query target defines the set of information (tables and rows) from which the user wants to retrieve data. If a special form has not been customized for a table or join definition, QBF will generate a default form for entering the query.

You can select either the "Retrieve" or "Update" function in QBF to execute a query. The Retrieve option will display only the requested information and does not allow editing. Selected rows are displayed on the screen individually or in lists (called a "table field format"), depending on how the user has designed the form.

Initially, the differences and relationship between QBF and Vifred are confusing. QBF is used for data manipulation; Vifred is used for creating and editing screen forms. However, QBF, using Vifred, can create default forms when necessary, and you must create a join definition in QBF before you can create and edit an associated custom form in Vifred. Once the user grasps these rules and relationships, it becomes easier to use these two tools effectively.

QBF will sort retrieved data as defined by the user. You can specify up to 127 sort fields in sequence and can specify each as ascending or descending.

Some Loose Ends. If you leave the first screen blank when entering QBF, you can access a helpful Start-up menu that allows you to see a catalog of existing query objects (tables, queries saved as QBFNames, or join definitions). However, the first QBF screen does not tell you that; you must read this in the manual or select Help.

The QBF function does not tell you how many records it finds that satisfy the criteria.

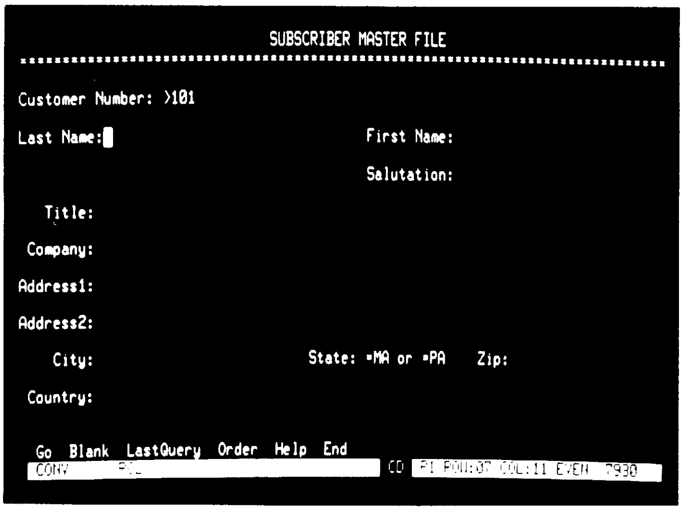
We entered an invalid query target name. QBF tried to execute the query, but couldn't find the object and dumped us back at the main menu. We would prefer to get an error message and have the option to reenter a valid object name.

SQL and QUEL. Structured Query Language (SQL) is the IBM-compatible language that is fast becoming the de facto standard for querying relational databases in both Unix and mainframe environments. It is even beginning to grab attention in the PC world as user organizations are articulating with more frequency and ardor the desire to run the same (or at least compatible) DBMSs on all systems in a network. SQL was only recently added to Ingres with Release 5.0 in late 1986. QUEL (QUEry Language), which is not IBM SQL-compatible, was originally developed with Ingres.

SQL and QUEL were built in parallel in the 1970s and are very close in syntax. Queries are performed in the same way with both. Selecting either one on the main menu invokes the appropriate interactive query editor. You then enter your query and use the menu to run, edit (via the Unix system editor), or store your query in a file. Neither wordwrap nor horizontal scrolling is in effect when you enter a query; you must hit Return to continue on the next line to avoid errors in the query. We would rather not worry about this; we're busy enough figuring out how to structure the query.

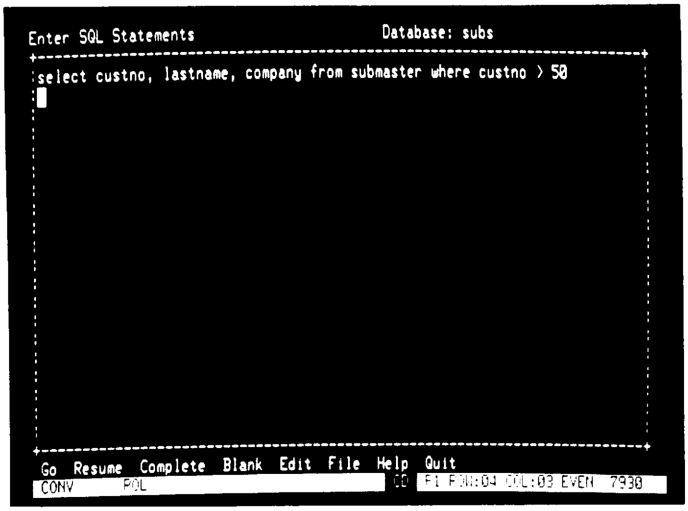
Both query languages include commands to create databases and tables; manipulate data; query the database; impose integrity constraints on data; specify access permission to the database by user, table, function, time of day, and location; create views; create and modify the way data is stored in the table; and much more.

In the current versions of Ingres, an SQL query is parsed, or "translated," to its QUEL equivalent before accessing the database. In the next release of Ingres, as the company continues its commitment to SQL compatibility, this process will be reversed: QUEL statements will be parsed to SQL.



```
                              SUBSCRIBER MASTER FILE
**************************************************************************

Customer Number: >101

Last Name:                               First Name:

                                         Salutation:

   Title:

 Company:

Address1:

Address2:

    City:                          State: =MA or =PA    Zip:

 Country:
```

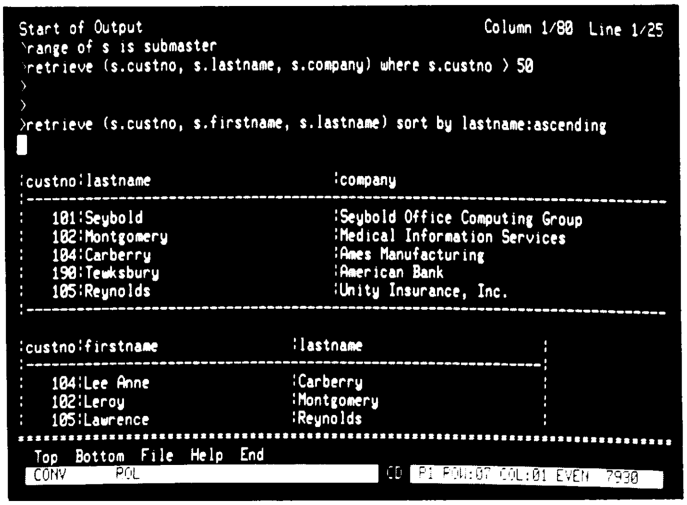
QBF provides flexible options for entering search criteria on a form.



```
Enter SQL Statements                            Database: subs
+-----------------------------------------------------------------+
:select custno, lastname, company from submaster where custno > 50 :
:
```

Go Resume Complete Blank Edit File Help Quit
CONV POL
CD
F1 POW:04 COL:03 EVEN 7930

Ingres provides an interactive query editor for both QUEL and SQL. This is an SQL query.



```
Start of Output                                    Column 1/80  Line 1/25
>range of s is submaster
>retrieve (s.custno, s.lastname, s.company) where s.custno > 50
>
>
>retrieve (s.custno, s.firstname, s.lastname) sort by lastname:ascending

:custno:lastname                    :company
:------------------------------------------------------------------
:   101:Seybold                     :Seybold Office Computing Group
:   102:Montgomery                  :Medical Information Services
:   104:Carberry                    :Ames Manufacturing
:   190:Tewksbury                   :American Bank
:   105:Reynolds                    :Unity Insurance, Inc.
:------------------------------------------------------------------

:custno:firstname              :lastname                    :
:-----------------------------------------------------------:
:   104:Lee Anne               :Carberry                    :
:   102:Leroy                  :Montgomery                  :
:   105:Lawrence               :Reynolds                    :
********************************************************************
```

Here are the results of two QUEL queries (which you can see displayed at the top of the screen). You can execute multiple queries in one "Go" command. Note that you only need to specify the "Range" command once in QUEL; it remains in effect for the entire query session.

Developers can also embed SQL or QUEL queries in programs to access Ingres databases.

Generating Reports

RBF. Report-By-Forms (RBF) is a forms-based report facility that enables the end user to generate simple reports by modifying a screen form. RBF operates very much like Vifred, and the two products have a consistent interface.

RBF will generate a default report format for you and decide whether to use a columnar or block format, depending on how the data in the table fits on the screen. Or you can specify which format you want. The default columnar report will automatically perform a primary sort on the contents of the first column and a secondary sort on the second column.

The three main sections on a report are the title, column headings, and detail lines. All of these can be edited. Some of the design capabilities allow you to:

- Insert or delete trim components, blank lines, or columns
- Move items on the report (column headings and detail line entries can be moved separately or together), including the report boundaries
- Define the sorting order for data and establish break columns, run-time parameters, and aggregates (count, sum, average, minimum, maximum)
- Set the page length
- Specify formatting templates (e.g., $xxx,xxx.nn)

Like Vifred, RBF has an "Undo" function. You cannot edit the headers for breaks and pages, page footers, or break footers.

You save the report when you are done editing; this stores the report in the database itself. RBF generates standard Report-Writer formatting commands for report definitions. A user can also develop a basic report in RBF and then make a text file copy of the report's formatting commands for further editing within the Report Writer.

Input Data. A report can be associated with a particular table or with a view. A view is created with a "Query" command (in either SQL or QUEL). Ingres only provides one pass through the data for a report. If you cannot create the desired set of data in a single table or view, you cannot generate the report in Ingres.

Output. You cannot run a report from within RBF. You must exit RBF and select the "Report" option on the main menu to actually run the report. This creates extra work if you want to view a report while designing it. The Report option allows you to run the report to the terminal screen or to a printer, but the option to print the report is not obvious.

Furthermore, the documentation does not cover output procedures in any depth. The manual states that, to print a report, you must output the report to a file first. The only way to do this is to use the Report command at the Unix prompt; this cannot be done from within RBF. The output file can then be displayed with the Unix "more" or "pg" commands or printed with the "lpr" command. Although this is not explicit in the manual, you can also enter "lpr" as the file name in the Report command, and Ingres will send the report to the printer.

REPORT WRITER. Ingres also provides a nonprocedural Report-Writer for more complex reports. In this product, you use the standard Unix editor (e.g., vi) to create the report and then compile it with "sreport", which checks for valid syntax and performs some debugging. However, sreport does miss some things. For example, it doesn't verify whether the tables and fields referred to on a report really exist. Once the report is compiled and saved, the Report option on the menu executes the report.

With the Report Writer, you are essentially writing a program to generate a report; this product is aimed primarily at the applications developer or sophisticated end user.

Customizing Applications

Ingres provides extensive tools for customizing applications. The Ingres Applications-By-Forms (ABF) module is a forms-based, object-oriented applications development environment. With ABF, the user can build an application without using a programming language. A fourth generation programming language (4GL) called Operations Specification Language (OSL) is also available within ABF for developing more complex applications.

One interesting note here is that the Ingres product is itself an Ingres application. The ABF module of Ingres was built using ABF. In fact, all of the forms modules are ABF applications.

ABF uses forms as the interface to an application. The concept of frames, mentioned earlier, is very important here. To refresh your memory, a frame is a single screen display; it consists of a form (fields and trim) at the top of the screen with an associated menu of operations displayed across the bottom of the screen. A frame can be used for either input or output. Thus, a forms application comprises a sequence of one or more frames in which the user enters and reads data. An ABF application cannot span two databases.

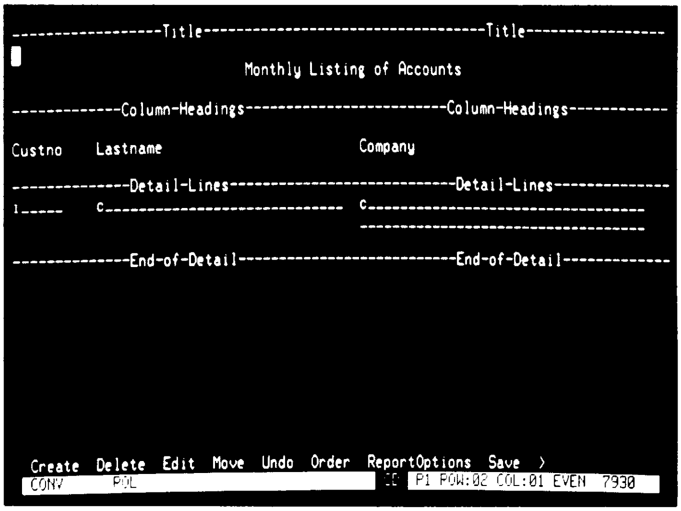
ABF takes care of compiling and linking the pieces of the application. It also allows the user to run the application while defining it. This interactive testing process is extremely helpful. In addition, ABF can access any of the Ingres form-based modules (QBF, RBF, Vifred, and Graphics) and thus can incorporate existing reports and forms in an application. ABF can also access code written with the Ingres 4GL or a non-Ingres procedure, such as one written in C.

ABF is designed for applications developers and sophisticated end users. While certain aspects of ABF are straightforward, such as developing a frame that calls up QBF and a particular query form, others are more complex and can be confusing to the less experienced user. Designing menu screens, for example, requires writing simple OSL statements, so it is difficult to escape having to write at least a basic program. Relational Technology offers extensive training programs for applications developers.

From the perspective of the experienced applications developer, however, the flexibility and power of the Ingres applications development environment is one of the product's major strengths.

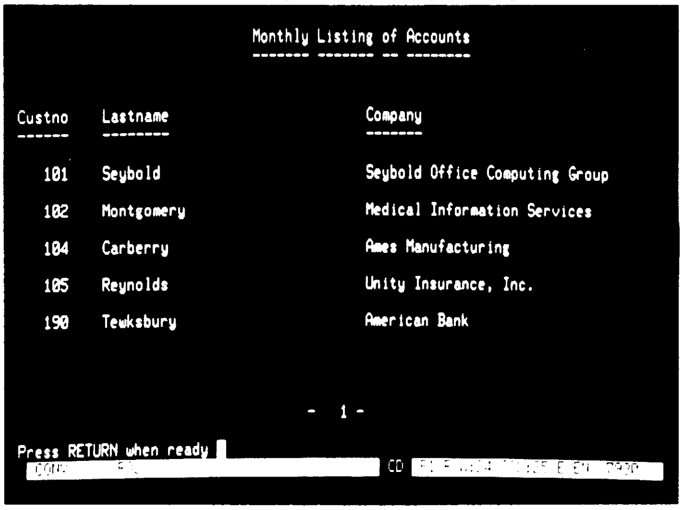
Database Security

An Ingres account manager maintains the list of users that can access the Ingres DBMS. Within Ingres, the creator of a database automatically owns the database. The owner can then grant permissions to other users to query, add, update, or delete information in a specific table. The permission can be limited to specific records and fields. Access can be further restricted to a specific terminal and day/time.



```
----------------Title----------------------------------Title----------------

                       Monthly Listing of Accounts

------------Column-Headings-----------------------Column-Headings-----------

Custno   Lastname                         Company

-------------Detail-Lines--------------------------Detail-Lines-------------
i-----   c---------------------------     c------------------------------
                                          -------------------------------

-------------End-of-Detail-------------------------End-of-Detail------------




  Create  Delete  Edit  Move  Undo  Order  ReportOptions  Save  >
  CONV     POL                              CD  P1 POW:02 COL:01 EVEN  7930
```

The RBF environment is much like Vifred's. We have modified a default report to eliminate many columns, add a descriptive title, and change the width of the report.



```
                      Monthly Listing of Accounts
                      ------- ------- -- --------

Custno   Lastname                       Company
------   --------                       -------

   101   Seybold                        Seybold Office Computing Group
   102   Montgomery                     Medical Information Services
   104   Carberry                       Ames Manufacturing
   105   Reynolds                       Unity Insurance, Inc.
   190   Tewksbury                      American Bank


                                 - 1 -

Press RETURN when ready
  CONV                                    CD  [illegible]
```

Here is the resulting report.

Database Integrity

The concept of database integrity is appropriate at two levels. Referential integrity prevents a database from becoming logically inconsistent. For example, referential integrity would not allow you to delete a customer master record for a customer with outstanding orders. In Ingres, this type of database integrity must be built in at the application level. There is no facility for including these types of integrity rules in the data dictionary.

The second major integrity issue, particularly important in large databases with frequent updates, is the concept of the transaction. This is the ability to define a series of operations on the database as a single transaction and to ensure that the transaction is implemented entirely or not at all. If the system crashes in the middle of a transaction, the database has the built-in facility to "rollback" the database to undo a partial transaction. Ingres has a very strong transaction-management function, providing for the definition of a transaction and automatic rollback capability.

In designing Ingres for PCs, Relational Technology has made extensive use of disk overlays to fit Ingres into the 640K DOS limit. One of the developers we interviewed is impressed with the performance of Ingres on the PC AT.

Ingres for PCs

Ingres for personal computers is a brand new product that has been available since March. We had the opportunity to look briefly at a beta test version (Version 5.0/02a) of this product.

SYSTEM REQUIREMENTS. Ingres for PCs runs on the IBM XT and AT and compatibles. It requires DOS 2.1 or later, 640K of memory, and at least a 10MB hard disk (6MB must be free to accept the full Ingres software code). A 20MB drive is probably more reasonable.

Relational Technology claims that Ingres can run in as little as 220K of memory, leaving the user with approximately 380K (allowing for the operating system) for memory-resident applications. However, we have been cautioned by users that the product is resource-intensive, that you need the full 640K of RAM, and that it cannot be used with memory-resident software. It will, in fact, run in 220K, but once you start up any of the Ingres subsystems (e.g., QBF, which requires 360K all by itself) from Ingres/Menu, 220K is not sufficient. Thus, the requirement for 640K.

Ingres for PCs comes on 20 diskettes, but installation is easy and very straightforward. The documentation is a vast improvement over what we have seen for the Unix product, and we hope to see the new documentation migrate back to the host versions of Ingres.

PERFORMANCE. In designing Ingres for PCs, Relational Technology has made extensive use of disk overlays to fit Ingres into the 640K DOS limit. One of the developers we interviewed indicated that his firm is impressed with the performance of Ingres on the PC AT, stating that Ingres is "very fast."

Another source for performance statistics is Palmer & Associates Incorporated, a consulting firm in Duluth, Georgia. Palmer & Associates recently conducted a benchmark comparison of three relational DBMSs for PCs—Informix-SQL (Version 2.00.00b), Ingres (the latest beta test version at the time was 5.0/02a), and Oracle (Version 4.1.4). The tests were run on a 640K IBM AT with no special hardware (such as floating point accelerators) to improve performance. The benchmarks consisted of 22 queries designed to test a wide range of query types and different index conditions on the tables (110 queries in all). Ingres was "the clear leader in performance," finishing first 87 times out of 110 (79.6 percent). Informix was second with 15.9 percent, and Oracle came in third with 4.4 percent. According to the study, Ingres was from 1.6 to 2.9 times faster than Informix and from 11 to 29 times faster than Oracle. (Oracle has since announced a new version of its PC product, Version 5.0, which requires 1.5MB of memory and a 286/386 PC.)

LOOK AND FEEL. What does Ingres look like? Is it different from its Unix counterpart? Relational Technology likes to say "Ingres is Ingres is Ingres" regardless of operating environment. As a result, we expected to see the same main Ingres/Menu screen on the PC and the same menu choices. Instead, the main menu has been streamlined and organized differently; some choices on the Unix system have been combined, and others are not available (the PC product does not include the Report-By-Forms, Vigraph, or Rungraph options or functionality).

In developing the PC product, Relational Technology has intensified its emphasis on "objects" rather than on the functions a user wants to perform. We noticed this orientation in the Unix product, but it is even more pronounced on the PC. For example, in Unix, to add records or enter data in a table, you are expected to know enough to select either QBF or the query languages. In the PC product, the option you must choose for these operations is "forms." And the word "query" does not appear on the main menu of the PC product, as it does in Unix. The focus here is more on the objects a developer would manipulate to create applications than on the functions that would concern an end user, an approach that will not help the end user feel comfortable with Ingres on the PC.

USER INTERFACE. Ingres for PCs provides two styles of user interface: the Unix style (full-screen forms with one-line menus) and Lotus-like ring menus, which are more state-of-the-art and more familiar to the PC user. We were hoping that our problems with the Unix user interface would be solved in this newer product, but we were disappointed. Some things we don't like about the Unix-style user interface are:

- As in the Unix product, you cannot see the entire menu across the screen. It irritated us to have to go to the menu line with the menu key and press the menu key again to see the rest of the menu.
- Relational Technologies has not eliminated the problem of menu items with non-unique first letters.
- PageUp and PageDown do not work when the user is browsing through forms retrieved through QBF.
- When you exit to a previous menu, you are not returned to the choice you originally selected (even Unix does this, for the most part).

The ring menus are much easier to work with. They provide a default choice and are displayed across the top of the screen as in Lotus.

Ingres for PCs could make much better use of graphics on the screen. Its tradition of accommodating the lowest common denominator (a dumb terminal) has crept into the PC product without taking much advantage of the PC's increased capabilities. For example, boxes could be used to highlight certain information, such as setting off the names of tables, etc.

One of Ingres's main strengths is its applications development environment. The developer can get Ingres applications up and running quickly, and the set of tools that surround the Ingres DBMS are comprehensive and very powerful.

We found Ingres slow when changing between functions, such as entering or exiting QBF.

FUNCTIONALITY. In general, the functionality of the PC product is the same as that on the Unix product. Once you get used to the interface differences, Ingres IS Ingres is Ingres. We were quite comfortable performing the same functions as we had learned them in the Unix environment. We are, however, disappointed that RBF is missing from the PC product. This leaves a large gap in the process of generating reports. The only available option for the user is writing programs with the Report-Writer.

RBF was not included because it is a very large program, and a major engineering effort would have been required to enable it to run in 640K. Relational Technology acknowledges the problem and plans to address it. In the meantime, some users are building reports with external report utilities (such as Concentric's 1-2-3 Report Writer; Ingres integrates with this through its Visual Query Language—VQL).

On the graphics front, Relational Technology decided that the graphics software already available for the PC was more than sufficient and therefore did not port Vigraph to DOS. The company decided that its customers would be better served by implementing tight integration with existing graphics; this is also done at the present time through the Visual Query Language (VQL). Relational Technology has no other plans here, but hinted that perhaps a popular graphics package would incorporate the ability to read Ingres files directly.

VISUAL QUERY LANGUAGE. One of the most attractive features of Ingres for PCs is the VQL. This is essentially the Ingres/PCLink product, which, in turn, is based on Network Innovations' Multiplex product (for more information on Multiplex, see Vol. 2, No. 1). VQL displays columns and rows from the database in a familiar spreadsheet format and allows the user to easily build very complex queries. We would like to see this implemented in all versions of Ingres.

INGRES: THE DEVELOPER'S PERSPECTIVE

To get additional perspective on the Ingres DBMS product, we interviewed experienced applications developers, just as we did for both our Unify (Vol. 1, No. 3) and Informix-SQL (Vol. 1, No. 8) reviews. These developers have worked with very large databases and have used Ingres in a custom-programming environment.

Strengths

APPLICATIONS DEVELOPMENT. One of Ingres's main strengths is its applications development environment. The developer can get Ingres applications up and running quickly, and the set of tools that surround the Ingres DBMS are comprehensive and very powerful. These include the forms development facility, the fourth generation programming language (4GL), embedded SQL and QUEL, graphics, and its PCLink product, among others. These are tools the user doesn't have to acquire from another vendor; they are an integral part of the Ingres family. In addition, because the Ingres product has matured over the past five years, these tools tend to be more fully developed and more sophisticated than those offered by competitors such as Informix Software and Unify.

With Applications-By-Forms (ABF), through menus and forms, the user can build simple applications without using the 4GL. ABF is particularly good for rapid prototyping, allowing the developer to get a good feel for what the screen will look like.

And the Ingres 4GL is accessible through ABF if it is needed for more complex applications. The 4GL provides for building screen and menus and incorporates all the facilities of the query language. It can also call other subsystems of Ingres.

The third level of applications development comprises the embedded SQL and QUEL products, allowing the developer to access an Ingres database by embedding SQL or QUEL statements in programs written in third-generation languages such as C or Cobol.

QUERY LANGUAGES. As we mentioned, the Ingres DBMS was built around QUEL as its query language, and SQL was added in late 1986 with Release 5.0. All of the developers we interviewed described QUEL as more powerful and more flexible than SQL. SQL can be "very fussy" about specific syntax, whereas QUEL presents fewer restrictions in developing queries. For example, in QUEL you can retrieve data into a table on the fly, which you cannot do in SQL; the QUEL range command applies for an entire query session and does not have to be repeated for every query. As one of our developers put it, "QUEL is so easy compared to SQL. The commands are not nearly as complex and the QUEL terminology makes more sense." Another developer said, "QUEL is a better language. There's just one small problem—all of IBM's products are based on SQL, and Relational Technology can't afford to buck that trend." The plus for Ingres is that it offers both languages; the developer can choose the most appropriate—QUEL for its power and flexibility, or SQL for its compatibility with other DBMSs.

> *The developers speak positively of Relational Technology's networking efforts. "With its networking, Ingres/Star, and the future for gateways to other DBMSs, Ingres will be a super product."*

Relational Technology describes its SQL as "broadly compatible" with IBM's SQL. Basically, this means that Ingres supports SQL and most of the SQL constructs. Some are not there yet but will be in the future. SQL commands with QUEL equivalents were implemented quickly, but the fact remains that the underlying Ingres DBMS was not modeled on DB2 nor built around SQL (as was Oracle).

For example, Ingres as yet supports only static SQL queries, not dynamic ones, and some applications require dynamic queries. A static query is one that is embedded when a program is written, is compiled with the program, and remains as a permanent part of the program. A dynamic query is one that is made up on the fly at run-time; the user types in a "select" statement at run-time, and that statement is the query passed from the program to the DBMS. This is a much more complex environment than one handling static queries. You don't know until the query is entered what information will be requested, what buffer sizes are required, etc. Capabilities that are available when embedded queries are compiled must also be available at run-time to support dynamic queries.

To gain perspective on this issue, it should be noted that at least two of our developers stated that no two versions of SQL are the same, and that the vendors' vaunted claim of portability among Ingres, DB2, Oracle, et al. is not truly possible today.

MULTIPLE PLATFORMS. The fact that Ingres runs on a variety of different hardware/operating system configurations is a definite plus. Ingres runs the gamut from the the mainframe (VM, with MVS to come) through VAX/VMS, large Unix systems, Unix workstations, to the PC AT. As the Unix systems have become more powerful, the need for a subset of Ingres, as was originally implemented on systems like the low-end AT&T line, has disappeared. Relational Technology's success in putting a full implementation of Ingres on the PC AT is no mean feat.

NETWORKING. The developers speak positively of Relational Technology's networking efforts. "With its networking, Ingres/Star, and the future for gateways to other DBMSs, Ingres will be a super product."

The ability to have the PC act as a front end, accessing data across a network, will reduce the load on the database machine, reduce retrieval time, and free up CPU cycles for other tasks.

TRANSACTION PROCESSING. The Ingres transaction-management facility is another strong feature. Ingres's transaction-processing environment was described by one developer as "robust" and much more sophisticated than those found in products like Unify and Informix. Relational Technology has extensive experience with customer applications involving hundreds of megabytes of data. "The company can't afford to say to customers, 'Oops! You're going to have to rebuild the last three days of transactions.'" These large applications must have guaranteed transaction integrity with excellent recovery and rollback capabilities. Ingres has high-quality, fast, straightforward tools for maintaining transaction integrity.

QUERY OPTIMIZATION. All of our developers included the Ingres query optimizer as one of the product's strengths. The query optimizer, when it receives a query, normalizes it and then develops several ways to perform the query (called Query Execution Plans, or QEPs). It scores these QEPs on factors such as resource intensiveness and picks the best one in terms of overall performance. On a small database, the time spent optimizing queries may not be worthwhile. However, once a database grows to a "serious size," a query optimizer is essential. And the developers feel that Relational Technology has excellent technology here.

Relational Technology has invested a great deal of development effort in its query optimizer, and the consensus is that the company will continue this investment to further improve performance.

PC PRODUCT. The developers give Ingres high marks for its PC version. They find the product very fast on a PC AT.

Implementing a full-fledged version of Ingres that runs within 640K means that the product can run on the standard PC found everywhere today. In contrast, the new Oracle PC product requires 1.5MB of memory.

Weaknesses

REPORT WRITER. The Report Writer is one of Ingres's weaker links. The Report Writer uses an SQL or QUEL statement to retrieve the data for the report, but it is limited to a single query. The Report Writer cannot handle the more complex reports, and it also runs more slowly than programs generating reports with C programs. Developers would like to see the Report Writer permit multiple queries and multiple data views.

The user interface for the QUEL (and SQL) terminal monitor was described by one developer as "a product out of the dark ages."

DATA DICTIONARY. In some areas, the Ingres data dictionary facility is weak. Ingres manages a very complex environment with tables, forms, reports, and ABF applications. If you change the definition of a table, Ingres does not provide cross-referencing of associated forms or reports; therefore, the developer cannot see if any of them need modification. Some users have written their own programs to do this.

Other concerns include the following: You can have two tables with the same name in the database as long as the ownership is different. You can also have two forms with the same name and two different owners. If one is the database administrator (DBA) and one is a non-DBA, the non-DBA will always get his or her own form first and will never get to see the DBA's form.

Relational Technology has "taken certain pains to prevent you from accessing the Ingres system tables." These tables are not listed in the data dictionary; to update them, you need special permission.

One enhancement coming with Release 6.0 is the ability to add a description field for each table listed in the data dictionary. For applications using hundreds of tables, the table names can get relatively esoteric. A descriptive or "comments" field would be very valuable for tables, fields, forms, reports, and applications—all of the objects in the Ingres database.

PERMISSIONS. While Ingres generally provides good protection for the database in its permission scheme, transferring ownership of Ingres structures can be very difficult. In many cases, to change the ownership of an Ingres structure, you have to unload the data, destroy the structure, rebuild the structure with a new owner, and reload the data. Although it may not take long, the process is cumbersome and should be easier.

Maintaining the permission scheme is also difficult. For example, if a user is deleted from the user list, that user's permissions are still in the database and must be removed separately. Also, it is possible for someone other than the DBA to create a table within the database without the DBA being aware of it.

REQUIREMENTS. Ingres is big. Installing Ingres means dedicating a large amount of memory to the product. And, because the product is big, its performance on smaller systems (such as 68000/68010 machines or a VAX 11/750/780) may be sluggish.

One of the performance concerns mentioned several times is the overhead involved in starting up the Ingres forms system.

QUERY INTERFACE. The user interface for the QUEL (and SQL) terminal monitor was described by one developer as "a product out of the dark ages." Unlike the interactive editor accessed from Ingres/Menu, there is no menu across the bottom of the screen, and the user must enter commands like "\g" for go, or " \q" for quit. This cannot compare to the interactive editor within Ingres nor with the ring menus implemented in Informix-SQL.

DOCUMENTATION. One of our developers agreed that the Ingres documentation leaves a lot to be desired. On the one hand, it is voluminous; on the other, it is difficult to find information. One example the developer gave was trying to look up basic information, such as what data types Ingres supports in different operating environments. This information is buried in the text, not listed in a table format nor included in an appendix. If you want to see the structure of the system catalogs, you must send for an internal document from Relational Technology. The documentation also lacks focus—it does not successfully address either the novice or the experienced user.

TEXT. One developer stated that Ingres does not have the ability to manipulate blocks of free-form text in a database, such as a scientific research bibliography (although this individual does not see any DBMS handling this requirement well). Ingres lacks the facility for editing text and searching.

Summary

On the plus side, from the end user's perspective, it is easy to create and define Ingres databases and tables. The Ingres forms-based Vifred and RBF modules provide easy-to-use and consistent tools for designing forms and simple reports. The interactive query editor is well designed for entering, debugging, and running queries. (To be totally successful, however, you still have to master the syntax of SQL or QUEL!)

Areas of improvement include:

- Implementation of an easy way to modify the structure of tables (add, change, delete fields) in an interactive mode without having to resort to SQL or QUEL.
- A more sophisticated interactive Report Writer with the abilities to add more complex formatting and to easily run the report for debugging purposes. (The PC version also needs end-user report-writing capabilities.)
- Implementation of a visual query language between the limited capabilities of QBF and the complexity of writing SQL or QUEL commands, a point-and-pick approach, if you will, for the end user. We hope to see this part of the Ingres PC product implemented in other versions of Ingres.

From the developer's standpoint, Ingres is very powerful and can handle many types of complex applications. On the negative side, the developers mentioned the Report Writer (probably a unanimous choice for the weakest link in the product).

- Improved documentation and a tutorial for the end user. These are musts. Our impression is that the developer would also appreciate some major documentation improvements. Ingres seems ambivalent about the audience for its manuals. They are designed for both the novice end user and the sophisticated developer but manage to satisfy neither. It is not realistic to expect to cover both audiences with the same material. Relational Technology would better serve its users by developing an end-user manual and a separate developer manual. Better organization and packaging would be invaluable. The company indicated that it is addressing the documentation issues and that installation will also be much easier with the next release of Ingres.

From the developer's standpoint, Ingres is very powerful and can handle many types of complex applications. Strong points include the applications development tools, the query optimizer, the transaction-management function, the networking, and its availability on multiple platforms. On the negative side, the developers mentioned the Report Writer (probably a unanimous choice for the weakest link in the product), some missing features in the data dictionary, and difficulty in maintaining permissions. On the whole, the developers are very positive about Ingres and its ability to handle current and future applications with good performance and high productivity for the applications developer.

Futures

Relational Technology has a long list of plans in the works for both functional and performance improvements to its family of Ingres products. There will be a major release later this year, in conformance with the company's goal to introduce significant improvements to the product every year. Here are some of the things you can expect in future releases:

DATA TYPES. Relational Technology plans to implement a long, variable-length text field in an upcoming release. Ultimately, the product will support abstract data types, where the user can define the data type. An example would be the ability to store and manipulate three-dimensional coordinates with vector analysis.

USER INTERFACE. Some of the developments in user interfaces will include windowing for help and pop-down menus. We would also expect Relational Technology to eventually support the XWindows standard.

PLATFORMS. As we mentioned, a version of Ingres for the IBM MVS operating system on the mainframe is coming. Ingres/Star will become available for Unix. Also in the works is a networked PC version.

DISTRIBUTED PROCESSING. The company is working on a server architecture for the back end data manager that will be more efficient than running multiple back ends, as is done in the current product.

NETWORKING. Gateways to non-Ingres databases (e.g., IBM's IMS and DB2) will be implemented in Ingres/Star, along with additional distributed database capabilities.

SQL COMPATIBILITY. Relational Technology is addressing SQL compatibility issues, and the next release will introduce more features in this area.

PERFORMANCE. Performance is a major area of concentration for the company. It plans enhancements in several significant areas:

- Transaction processing. The company is completely rewriting its transaction-processing system to achieve much faster performance and to better handle online systems.
- Complex queries. Relational Technology will continue to improve its query optimizer.
- Distributed environments. A new version of Ingres/Star will improve performance and add functionality.
- Shared Memory. Use of shared memory on Unix will improve table-locking capability.

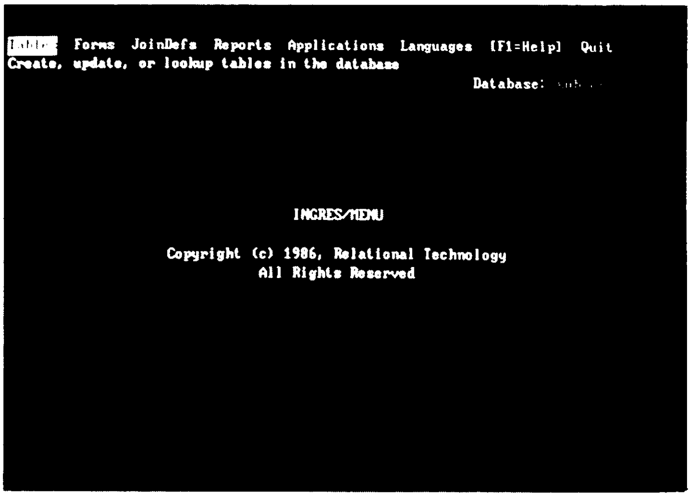
DATA DICTIONARY. Relational Technology is also addressing the issues of referential integrity and improvements in the Ingres data dictionary facilities.

The Bottom Line

Ingres is fighting it out with Oracle and DEC in the VAX/VMS market, and with Oracle on large Unix machines. Ingres and Oracle tend to play a game of leapfrog when bringing new features and functionality to the market. The key to the future will be the ability to take advantage of the network and allow users anywhere on the network to easily and transparently access databases in multiple locations. From the end-user perspective, the development of easy-to-use and powerful end-user applications development tools is very important. This is particularly true as these sophisticated DBMSs are ported into the IBM PC environment with its wide mix of user experience and expertise.

The issue of performance is critical as relational databases vie for the right to be a platform for large transaction-processing systems.

Relational Technology has a good handle on these issues and has development plans in place to enhance its products in all the right places. And it has a prominent position in terms of its installed base. If the company successfully manages its growth, continues to focus on the networking and performance issues, and adapts the product to the end user as well as the applications developer, we expect that Ingres will enhance its position as a key player in the networked relational DBMS market. ◐



```
Tables  Forms  JoinDefs  Reports  Applications  Languages  [F1=Help]  Quit
Create, update, or lookup tables in the database
                                                 Database: [illegible]



                            INGRES/MENU

             Copyright (c) 1986, Relational Technology
                        All Rights Reserved
```

Ingres on the PC uses a ring-menu interface.

Informix-SQL, Ingres, and Unify: A Comparative Look

Feature	**Informix-SQL**	**Ingres**	**Unify**
Underlying file structure	C-ISAM	Unix file system	Unix file system or proprietary (option)
Database parameters			
Tables/database	No limit	No limit	256
Records/database	Limited by storage	No limit	2 billion
Fields/record	No limit	127	256
Record size	32KB	2KB	25.6KB *
User interface	Ring menus	Single-line menus	Full-screen menus
Menu bypass	No	Yes	Yes
Contextual help	Yes	Yes	Yes
Tutorial	No (demo database)	Yes	Yes
Data types			
Character	Yes	Yes (255 char. max.)	Yes (256 char. max.)
Numeric	Yes (small integer, integer, small float, float, decimal, serial)	Yes (integer—1, 2, 4 bytes; float—4, 8 bytes)	Yes (numeric, float)
Currency	Yes	Yes	Yes
Date	Yes	Yes	Yes
Time	No	Yes	Yes
Combination (links between 2 fields)	No	No	Yes
Binary	No	No	Yes
Variable length text	No	No	Yes
Field attributes			
Case conversion	Yes	Yes	No
Default value	Yes	Yes	Yes
Required field	Yes	Yes	No
Acceptable values	Yes	Yes	Yes
Verification (enter data twice)	Yes	No	No
Verify joins	Yes	Yes	Yes (explicit relationships)
Lookup joins	Yes	Yes	Yes
Composite joins	Yes	Yes	Yes
Formatting of data	Yes	Yes	Yes
Calculated fields	Yes	No	No
Display only (no entry/update)	Yes	Yes	No
Prompt (for data entry)	Yes	Yes	Yes
Error message	Yes	Yes	Yes
Customized help	No	Yes	Yes
Screen forms			
Default form generator	Yes	Yes	Yes
Customized	Yes	Yes	Yes
Multiple tables/form	Yes (14 max.)	Yes (10 max.)	Yes (single screen limit)
Multiple screens/form	Yes	Yes	No
Embedded processing (if-then-else logic, display aggregates)	Yes	No (use 4GL)	No (but can be linked to SQL scripts)

** Unify can create special parametric files to allow increased database parameters.*

Informix-SQL, Ingres, and Unify: A Comparative Look

Feature	**Informix-SQL**	**Ingres**	**Unify**
Query-By-Forms			
Exact match	Yes	Yes	Yes
Relational operators	Yes	Yes	Yes
Ranges	Yes	Yes	Yes
List of values	Yes	Yes	No
Wildcards	Yes	Yes	Yes
Maximum/minimum values	Yes (only indexed)	Yes	No
Print query results	Yes	Yes	No
Pass results to Report Writer	No	Yes	Yes
SQL			
Standard SQL statements:			
Data definition language (DDL)	Yes	Yes	No
Data manipulation language (DML)	Yes	Yes	Yes
Query language	Yes	Yes	Yes
Extensions to SQL:			
Commit/rollback transactions	Yes	Yes	No
Execute operating system commands	No	Yes	Yes
Load/unload data to/from ASCII file	Yes	Yes	Yes
Additional data definition statements (alter, drop table, etc.)	Yes	No	No
Can be embedded in C/Cobol programs	Yes	Yes	Yes (C announced, Cobol coming)
Can create a new table with query results	Yes (temp. table)	Yes (permanent or temporary)	No
Stored queries	Yes	Yes	Yes
Case-sensitive (e.g., field names)	No	Yes	Yes
Optimizer	Yes	Yes	Yes
How create SQL queries	Interactive editor or Unix system editor	Interactive SQL/QUEL interface	Unix system editor
Report Writer			
Nonprocedural	Yes	Yes	Yes
Default report generator	Yes	Yes	No
Interactive report generator using screen forms	No	Yes (RBF)	No
Interactive debugging	Yes	No	Yes
Input source	SQL	SQL/QUEL	SQL, ASCII file, screen form, program
Multiple tables	Yes	Yes	Yes
Page formatting	Yes	Yes	Yes
Headers and footers	Yes	Yes	Yes
Data formatting	Yes	Yes	Yes
Sort data	Yes (8 levels)	Yes (unlimited)	Yes (unlimited)
Aggregate functions	Yes	Yes	Yes
Logical processing (if-then-else-logic)	Yes	Yes	Yes
User variables	Yes	No	Yes
Prompt for input variables at run-time	Yes	Yes	Yes

Informix-SQL, Ingres, and Unify: A Comparative Look

Feature	**Informix-SQL**	**Ingres**	**Unify**
B-tree indexing			
Max. no. indexes	No limit	No limit	255/database
Max. no. fields/index	8	10	8
Max. size of indexed fields	120 chars.	2,000 chars.	240 chars.
Order options	Ascend/descend	Ascend/descend	Ascend/descend
Unique index	Yes	Yes	Yes
Database security			
Login password	No	No	Yes
Database-level access	Yes	Yes	Yes
Table-level access	Yes	Yes	Yes
Record-level access	Yes	Yes	Yes
Field-level access	Yes	Yes	Yes
Access by time of day	No	Yes	No
Access by location (workstation)	No	Yes	No
Ability to customize standard menus	No	No	Yes
Ability to design application menus	Yes (1/database)	Yes	Yes
Default menu generator	No	Yes	Yes
Custom help	No	Yes	Yes
Ability to create views of database	Yes (in data dictionary)	Yes	Yes (only in forms)
File access methods			
Hash indexes	No	Yes	Yes (primary keys)
Links (explicit table relationships)	No	No	Yes
B-trees	Yes	Yes	Yes
Buffered sequential	Yes	Yes	Yes
Transactions			
Logging	Yes	Yes	Yes
Commit/rollback transaction	Yes	Yes	No
Roll forward	Yes	Yes	Yes
Referential integrity	Built in at application level	Built in at application level	Part of data dictionary
Concurrency control— locking levels:			
Database	Yes	Yes	Yes
Table	Yes	Yes	Yes
Record	Yes	Yes	Yes
Raw input/output	No	No	Yes
Database can span multiple physical devices (disks)	Yes	Yes	Yes
Networked version for IBM PC	Yes	No, but coming	Yes

•HOT BOXES•

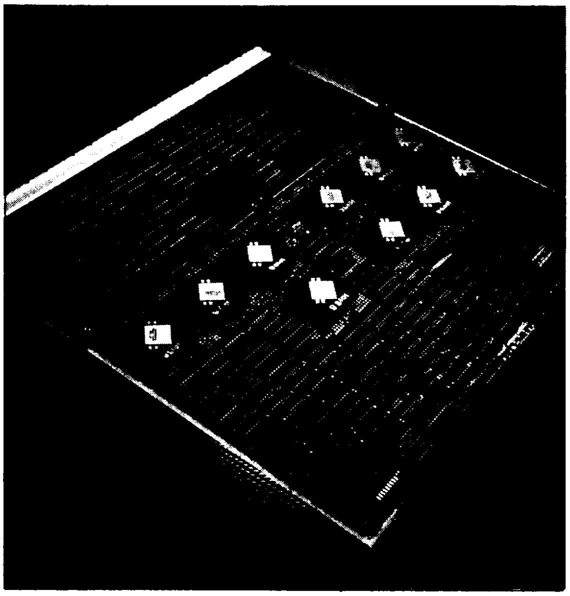
The MIPS Are Almost Free

Michael D. Millikin

There are executives in the computer industry who are trying to plan what their companies will do when MIPS are "free": when the cost of hardware components drops to such a degree that, in effect, the cost per MIPS is minimal.

MIPS might not yet be free, but from one company, at least, they certainly are a blue-light special.

Edge Computer has been selling its 6 MIPS (sustained), 68010-compatible uniprocessor compute engine for about a year. A dual-processor implementation cranks up to 12 sustained MIPS.



The basic Edge 1000 CPU board with 10 VLSI gate arrays.

Peak rates on these processors are 8 and 15 MIPS. (In DEC terms, this translates to a product line comparable to that of the 8650 to the 8800.) These processors have two extremely important features.

First, although the CPUs are proprietary gate arrays, they use a 68010-compatible instruction set. In other words, your 68010 code, utilities, and tools that have been working on the relatively pokey Motorola chips will run on Edge. Edge is providing a high-powered extension to any vendor's 680X0-based product line. For OEMs, this compatible extension is valuable. It saves porting time and costs, allowing vendors to be out on the market more quickly.

Second, the price/performance is amazing: $4,600/ MIPS.

Nor is the company through. The second-generation Edge 2000 processors will start at 16 peak MIPS for the uniprocessor and zoom up to 66.8 MIPS for a quad-processor version. Edge expects to push the performance on these processors higher within a year: to 28 MIPS on the uniprocessor and 112 on the quad. Concomitant with this comes a drop in the cost per MIPS to $1,200.

The performance statistics of even the current generation are phenomenal. On the Neal Nelson benchmarks, only a 30 CPU Sequent and Edge reached the 100 task level. Edge maintains a relatively flat performance curve over these tasks as well.

WHO ARE THESE GUYS? Edge started life as "just another workstation company." When the market reacted coolly, Edge switched to producing an OEM-oriented product.

The Edge processors are powerful, inexpensive, fast, and small. The Edge 1000 is a deskside processor perfect for use in an office. The cabinet for a single-chassis computer measures 17 inches by 29 inches by 30 inches.

Edge uses an implementation of the Harvard supercomputer architecture for its processors: dual 32-bit buses that fetch instructions and data concurrently (effectively, a 64-bit-wide aggregate bus).

The dual four-staged, pipelined processor achieves very close to Reduced Instruction Set Computer (RISC) performance, performing 90 percent execution of instructions in one clock cycle. Many vendors with 68K-based products are looking to RISC alternatives (or parallel architectures) to give them the needed power to remain competitive.

Edge points out, however, that the move to a RISC-based architecture is tricky, if only in terms of the time it takes to develop sets of utilities and tools. Look at the snags HP has hit in its move to RISC.

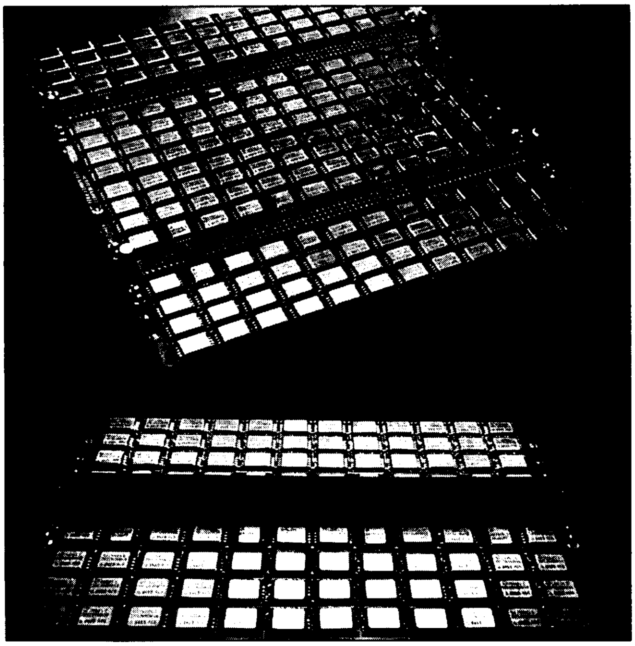
Edge offers a version of System V, Release 2.2, with Edge and Berkeley enhancements called GSX. GSX offers a shared-scheduling algorithm that provides CPU partitioning among multiple job classes and dynamic control over the allocation of resources. GSX also offers full Sun NFS connectivity with Ethernet TCP/IP.

THE PICK CONNECTION. Even more interesting is Edge's support (through recently-acquired Toltec) of Pick under Unix. With a product called Symetrix, Unix users have transparent access to Pick applications without running into some of the performance constraints of other implementations. The Edge systems see Pick and GSX as two separate environments.

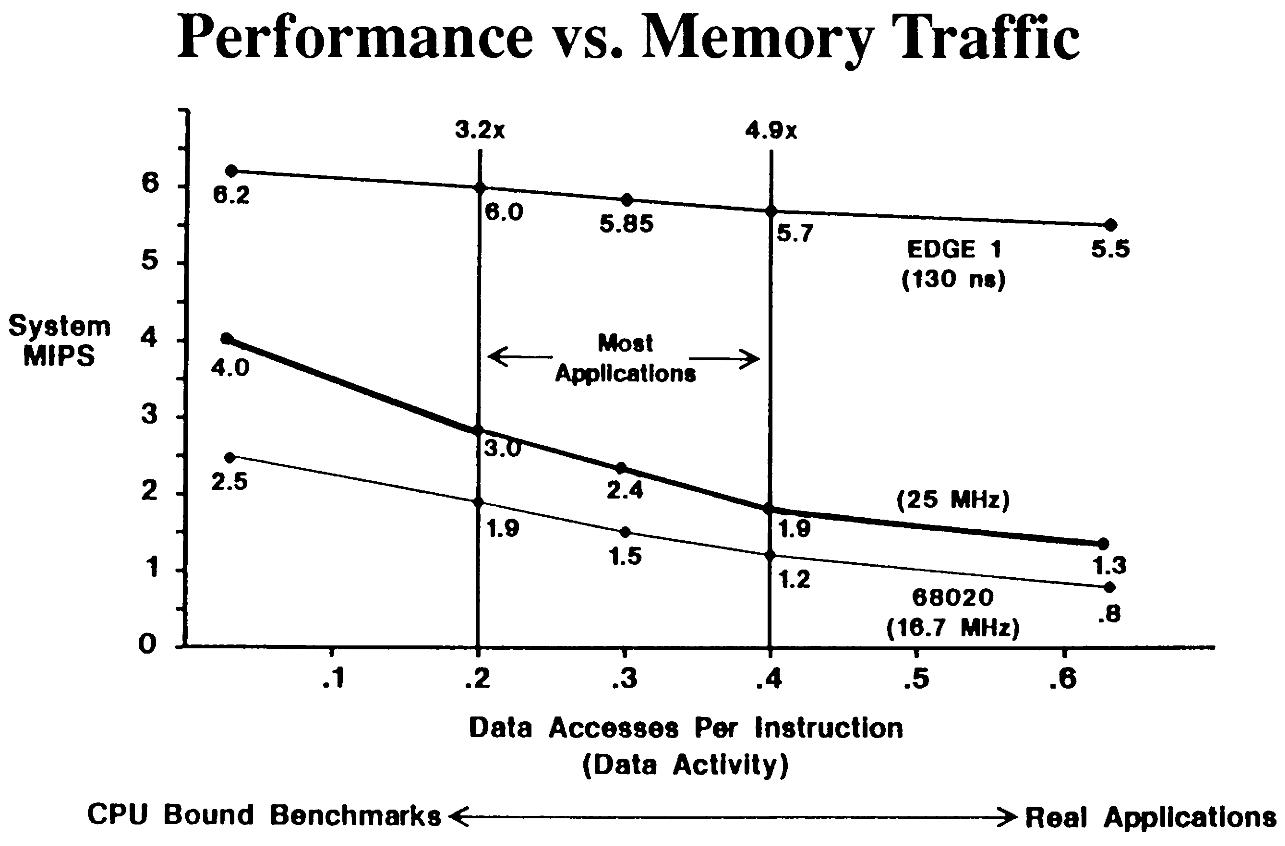
A piece in the kernel called the hypervisor determines which of the two environments gets the clock interrupt as they come into the CPU and allocates the task to either the Pick or the GSX segment.

Edge took the I/O driver from Unix and made it into a generic I/O service layer that addresses both Pick and Unix applications. Shared memory and named pipes integrate Pick and Unix, and any Pick file can map on top of a Unix file. One result is that Pick applications now have access to the networking and communication facilities of Unix. Furthermore, you can bury access to Pick within a Unix application. A Q-Office menu could, for example, take you into a Pick database.

Vendors with a 680X0-based product line should look seriously at Edge for a high-end component. ◐

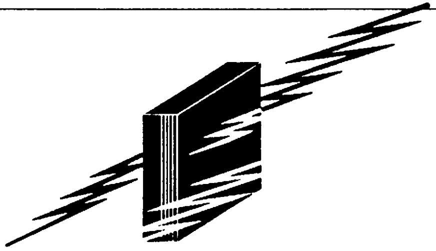


Edge's dual-surface mount memory boards pack 32MB per module.



Edge proprietary gate arrays provide higher and more consistent performance than the 68020 family and maintain compatibility as well.

•HOT BOXES•



Moving beyond MIPS

Judith S. Hurwitz

Pyramid Technology Corporation wants to change its image. Instead of being known as a hot box vendor, it plans to transform itself into an applications company. In most markets, this notion would be less than radical. However, in the MIPS-crazed Unix world, such an idea is tantamount to heresy. "We'd benefit from a move away from the MIPS race," acknowledges William Gimple, vice president of Systems Technology. The company plans to emphasize databases and software development applications.

To accomplish this goal, Pyramid is trying to become a "full-service computer company," insists Gimple. First, the company is playing the role of a systems integrator by forming strategic partnerships with companies like Sun Microsystems. Also, Pyramid hopes to attract more value-added resellers (VARs) and OEMs.

In order to switch from an iron pusher to a commercial systems vendor, Pyramid has restructured its sales force. "We are changing to a vertical market focus," says Gimple. He sees the market breaking down into four areas: database, online transaction processing, office automation, and connectivity. To sell these applications, Pyramid has established five sales groups aimed at five key industry segments: Telecommunications, Government, Manufacturing and Distribution (to sell via VARs), Computer-Aided Software Engineering (CASE), and Financial Services (aimed at Wall Street trading and business transaction-processing needs).

The company is also driven by demands from its international customers, at least 50 percent of its customer base, including Nixdorf, which sells its 98XE and 98E systems and its new Series 9000 under a private label OEM agreement. Nixdorf sells Pyramid systems as its Targon 35. These international customers are looking for the transaction processing focus as well as adherence to international standards.

Database Environment

The database and software development arena is where Pyramid is placing its bet. Seventy percent of the current customer base already uses Pyramid's products as database engines. Pyramid supports 14 proprietary database management systems (DBMSs), including Oracle Systems Corporation's Oracle, Relational Technology Incorporated's Ingres, and Informix Software's Informix database. Other database vendors, including Information Builders, VMark, Data Languages Corporation, and Britton Lee, use Pyramid's computers for product development. Pyramid also has a co-marketing and co-development relationship with Oracle, Relational Technology, and Information Builders.

Because of this orientation, Pyramid has made its technology more receptive to the needs of database and, specifically, transaction processing. For example, Pyramid recognizes the need to have seamless connectivity between its systems and the mainframe world.

Keeping an eye on developing markets, Pyramid is also offering common Lisp. Called PyrLisp, the product includes an interpreter, compiler, and debugger, and it operates with the full support of the dual-port Unix operating system. PyrLisp is being offered in the hopes of attracting development of artificial intelligence applications. Pyramid will offer both a development environment and a runtime environment.

Building Alliances

BUYING INTO PICK. Pyramid has wisely chosen to make Pick applications available on its systems. "There is a need to consolidate between Unix and Pick," says Gimple. Pyramid has therefore adopted the Universe software from VMark that allows Pick to run on top of Unix. "There are a lot of strong applications written for Pick," Gimple notes.

ENLISTING INDEPENDENT DEVELOPERS. Pyramid has implemented a program called PRISM (Pyramid Referred Independent Software Manufacturing) aimed at building its software base. The three-year-old co-marketing and development program has resulted in 3,000 applications being ported to Pyramid's platform. The company has encouraged software vendors in areas such as language development, database, and telecommunications to work in partnership with Pyramid. "This program allows us to enlist independent software vendors to port applications to Unix," says Gimple. Once these applications are ported to Pyramid, the company works with the software developer to take advantage of the closely coupled multiprocessor architecture. For example, Pyramid will show a database vendor how to put special I/O services into the database kernel. Also, Pyramid has shown vendors how to implement special scheduling and dispatching algorithms to help accelerate performance of the database. In some cases, the company has developed special lock drivers to handle semaphore operations that manage shared access to a database.

The Mini Mainframe

All this should not be misinterpreted to mean that Pyramid has decided to abandon the quest for ever greater heights in MIPdom. In fact, the company made a new statement of direction toward a commercial focus as it introduced its newest Unix hot box, the Series 9000. Pyramid calls its new series "minimainframes." Pyramid is aiming its new boxes at DEC's VAX 8500 and contends it offers them at about half the price.

The standard version of the low-end 9805 comes with 4MB of memory, 470MB of mass storage, and 16 RS-232 ports. It can be expanded to 128MB of memory, 32GB of mass storage, and up to 256 RS-232 ports. The 9805 is rated at 3.5 MIPS and sells for $129,000. Included in the series are two high-end multiprocessor systems with three or four processors each. They sell for $424,000 with three processors, and $514,000 with four. All systems are upgradable, from the low-end 9805 to the high-end 9840.

In addition to the new processors, Pyramid has announced a new I/O expansion chassis and a 1GB disk drive. The expansion chassis expands the I/O of the series 9000 to 512 asynchronous ports and more than 32GB of disk storage.

All this horsepower is intended to move the company more aggressively in its intended direction: online transaction-processing applications that include telecommunications, software development, manufacturing, administration, wholesale distribution, and financial services. Pyramid has also announced an ANSI 85 Cobol compiler it built using ACT's front end. The company added a proprietary code generator to optimize it for the Pyramid multiprocessor environment and SNA/3270 communications so that Pyramid systems can more easily coexist in a traditional commercial data processing environment. The communications software, called Pymet SNA/3270, is licensed from Systems Strategies in New York, which, according to Pyramid, was the first company to offer portable SNA and BSC packages under the Unix. The SNA software runs on Pyramid's intelligent synchronous communications controller and uses IBM's Synchronous Data Link Control (SDLC) communications protocols. The SNA/3270 enables Pyramid's computer to emulate IBM devices, including 3274 Model 51C controller, an IBM 3278 and 3279 information display station, and a 3287 printer. The company has also adopted Sun's NFS to communicate to the workstation environment. In fact, Pyramid was a prime mover in suggesting to Sun that it open its networking standard to the industry.

RISC Technology

Pyramid was one of the first companies to adapt the Reduced Instruction Set Computer (RISC) technology to approach a single cycle per instruction. The Series 9000, the company's second-generation RISC product, includes tightly-coupled processors that operate in a shared global-memory configuration, which enables quick memory access. The Series 9000 CPU is a CMOS VLSI implementation of Pyramid's proprietary RISC architecture. It includes discrete TTL and a 100 nanosecond clock. The company did its own development on the I/O processor to enhance functional parallelism. For example, each disk processor does its own processing. The terminal processor is itself a 1.5 MIPS processor with its own proprietary bit-sliced processor to handle I/O. To balance I/O with processing power, each processor has the performance of the VAX 8700, and each I/O channel has a throughput rating of 11 MBps.

Pyramid Technology runs a proprietary operating system, which is a dual port of both Berkeley's BSD and AT&T's System V versions of Unix. Users can choose either version as the operating environment and switch between the two with one command.

Conclusion

Pyramid has wisely recognized the importance of moving away from chasing MIPS and towards spending resources to develop more sophisticated software. It is critical for vendors in the Unix market to realize that the move into the commercial arena requires more than hotter iron.

At the same time, companies like Pyramid clearly have their work cut out for them. Vendors like Edge Technologies (see page 25) are demonstrating the speed that can be accomplished without using RISC. As vendors move out of the scientific and engineering marketplaces where Unix has reigned supreme, they will have to compete head to head with conventional mini and mainframe companies such as DEC and IBM. This challenge will be substantial. Pyramid and its counterparts in the Unix market will have to show that Unix has something special that will lure a commercial user away from the proprietary environment. ◐

NEWS

PRODUCTS • TRENDS • ISSUES • ANALYSIS

ANALYSIS

• INSIDE •

Altos offers a 386 system running on its WorkNet LAN. **Page 29**

The Arete 800 is an inexpensive alternative to low-end systems. **Page 30**

The Navy selects Xenix System V as it's operating system. **Page 30**

Three new products from Oracle include Proffesional Oracle, LANserver Oracle, and Networkstation Oracle. **Page 31**

• ALTOS •

Marketing the 386

The 386 deluge has begun in earnest. Most of the early announcements have been for PC or controller type products, but now we are beginning to see the first of the system-oriented 386s. Last month, Prime's EXL 386 hit the streets (see Vol. 2, No. 5) and was quickly followed by Texas Instruments' product family (see next month's *Unix in the Office*). This month, Altos Computer Systems has introduced its 386 system, the Altos 386 Series 2000.

The 80386 hot box runs at 16 MHz with a 32KB data and instruction cache, and an 80387 floating-point coprocessor. System RAM can be expanded to 16MB. The 3 MIPS systems will be available in four models. The low-end Model 2408S will accommodate up to 20 users and includes 4MB of RAM. The high-end Model 2817M includes 8MB of RAM and a multidrop cabling and transmission system. All models include an intelligent file processor subsystem, system memory, a communications processor, a 143MB ESDI disk drive, a 1.6MB 5 1/4-inch floppy drive, a 60MB streaming magnetic tape unit, and an Altos terminal. The Series 2000 will run Microsoft's Xenix System V operating system. Prices range from about $25,000 to $35,000.

NETWORKING. Altos sells its own terminals, but it will also allow PCs equipped with an Altos emulation board to serve as workstations on its network. Altos also plans to offer DOS running under Unix, allowing users to access DOS programs. The company is not ready to say if it will be dealing with Locus or Phoenix.

In order to network terminals and PCs into the 386 box, Altos is providing the same WorkNet local area network it sells with its other systems. It uses a four-wire, twisted-pair wiring scheme and includes an RS-422 interface. WorkNet resembles Sun's NFS in that it lets files be shared and transferred among users. Altos already has almost 3,000 of these networks installed. Altos also plans to implement Ethernet in the fourth quarter of this year. The company is committed to adapting AT&T's RSF when it becomes available. With the addition of a multidrop board, up to 128 terminals can be connected—64 of them concurrently. To connect to other systems, Altos is offering 3270 SNA, X.25, and DDN protocols.

THE COMPETITIVE ENVIRONMENT. By the end of 1985, Altos had almost 40 percent of the Intel-based Unix/Xenix market, according to research by InfoCorp. Obviously, the company has a lot at stake in keeping current with technology. Therefore, it is not surprising that Altos took the 386 route. To keep competitive with emerging pricing for the 386, per workstation cost is between $1,200 and $1,500, according to Jeff Bork, vice president of marketing at Altos.

Altos sees itself differentiated from other 386 vendors primarily because it is building on an existing product line. In essence, Altos has been able to swap out a 286 processor and plug in a 386. This means that all components have been battle tested in Altos's previous product line. All of Altos's existing products will be upwardly compatible with the new 386-based product family. Therefore, existing customers will be able to upgrade to the new systems. In addition, the same applications and language products will be available on the new box. Key software packages include Informix's relational database and Redwood International Ltd.'s Uniplex-II Plus office software (See Vol. 1, No. 7). Other packages include the 20/20 spreadsheet and Multiplan. Prochart graphics is also available. Languages such as C, Basic, Cobol, Fortran, Pascal, and RPG are being offered. This is a key advantage for Altos. Companies new to the Unix market will be scrambling to port applications software. "We will be shipping applications software from day one," says Bork. He notes that the company has already shipped 200 machines and has sold out its first two months of production through June.

Bork also notes that Altos has a design advantage. “We are ahead of Prime in terms of design.” He believes that Altos distinguishes itself from Prime's EXL 316 with regard to disk I/O. “Prime has not put in a separate DMA controller. They are using the SCSI disk with no microprocessor,” notes Bork. He points out that the Altos Series 2000 includes an ESDI controller that includes a separate microprocessor (an 8086 which will be upgraded next year to a 68020) and a DMA controller.

CHANNELS OF DISTRIBUTION. Altos will rely on its existing channels of distribution to sell its new hot box. The three primary channels include:

- Small- to medium-sized VARs that sell in the commercial arena, including such markets as health care, wholesale distribution, and accounting.
- OEMs, which have been a strong channel for Altos's 68020 product, will be used as a channel for the 386 product. Current OEM customers, including EDP, Tandem, and ITC Europe, have already expressed interest in the new product.
- Federal government has been a channel exclusively for Altos's 68020 product. There are now requirements (specifically the AFCAC 251 bid) that Altos is hoping to meet with its new product.

OPERATING SYSTEM. Altos is entering the market with Microsoft's Xenix System V. However, as soon as Unix 5.3 is available, Altos will migrate to it. Bork anticipates that this will happen in the first quarter of 1988.

CONCLUSION. There is no doubt that the 386 multiuser system market is growing hotter every day. We anticipate a slew of announcements before the dust settles. How successful all these vendors will be depends largely on their ability to port applications software and find innovative distribution channels and marketing alliances. Altos is well positioned for this market. It has established distribution channels and a track record in the Unix arena.

However, the company will have to do a lot more work in the applications arena to stand out as the competition heats up. Offering a strong database product like Informix and office software like Uniplex-II Plus is wise. But this may not be enough in a fiercely competitive market. Although components like spreadsheets and graphics are important, we'd like to see more functional integration in software offerings. In general, more applications software will need to come through the pipes. ◐

• ARETE •

Arete Moves Down

While companies like Pyramid are developing ever larger Unix boxes, Arete Systems Corporation is moving into the low end of the market. Its multiprocessor Arete 800 is a low-end system designed to support from 16 to as many as 128 users. An entry-level, single-processor system sells for less than $30,000. The multiprocessor configuration starts at $38,000; the 800 is targeted to compete with NCR's Tower 32/800. Arete claims that its system provides three times the performance at half the price of the Tower.

Arete is aiming the product at its traditional online transaction-processing market. According to Arete, the system processes transactions at between 5 and 10 transactions per second. The 800 system offers a distributed architecture that includes up to two tightly-coupled applications processors, up to two database processors, and up to five data communications processors. Also included are a 12.5 MHz CPU with a floating-point coprocessor, up to 16MB of shared memory, and up to 5.5GB of magnetic disk storage. Up to 14GB of optical disk storage is also available. The system runs a proprietary version of AT&T's Unix System V.2 with System V.3 enhancements. ◐

• XENIX •

Xenix Joins the Navy

There is little doubt that Unix is catching on fast with the government. In its latest move, the Navy has specified the Santa Cruz Operation's (SCO) version of Xenix System V for Zenith's Z-248. Zenith will provide the Xenix operating system as part of its contract with the Navy's NALTOACS program, and SCO will provide a text processing system. The contract will be used at the U.S. Navy's engineering laboratories, which will support advanced engineering CAD/CAM applications, word processing, accounting, database applications, and software development. According to SCO, Xenix was selected over MS-DOS because of its multiuser capabilities. ◐

• ORACLE •

Oracle Exploits Protected Mode

Oracle Corporation (Belmont, California), whose Oracle database has long reigned as one of the premiere Unix databases, has joined several of its counterparts by announcing three new products for the PC and PC LAN environments. Oracle, the company's SQL-based relational database management system (DBMS), was originally introduced on the DEC VAX under VMS in 1979 and now runs in a wide variety of mainframe and minicomputer environments as well as on the PC. The company indicated that these latest PC announcements are the first in a series intended to bring full mainframe capability and applications development to the PC world.

PROFESSIONAL ORACLE. Professional Oracle implements the full Oracle relational DBMS, Version 5.1, on the PC. Significantly, this is the first DBMS to bypass the 640K DOS memory limit by supporting the protected mode of operation available in the 286/386 PCs.

Some background: Version 4.0 of Oracle, requiring 640K of memory, has been available on the PC since 1984. However, Oracle Version 5.0 (developed for the mini and mainframe environments) added significant functionality to the product, and SQL*Star (which endowed Oracle with distributed database capabilities) was packaged as a standard part of the Oracle DBMS in Version 5.1. As a result, Oracle 5.1 would not fit in the DOS 640K memory box. In fact, the complete product requires 1.5MB of main memory. Rather than port only a subset of Oracle to the PC, degrade performance to adapt to limited memory, or wait for Microsoft to bring ADOS to market, the company decided to take advantage of the protected mode/extended memory features of the 286 and 386 chips.

Professional Oracle uses protected mode rather than the Expanded Memory Specification (EMS) or Enhanced Expanded Memory Specification (EEMS) supported by other PC vendors (Lotus, Ashton-Tate, etc.).

Professional Oracle has eliminated other DOS boundaries as well. An Oracle database can span disk partitions, so a database file can be larger than 32MB. And the user need not be concerned about the DOS limit on the number of files that can be open at one time. Since Oracle stores each database as a single file, the user can have an unlimited number of tables per database. Many other DBMS systems store each table as a separate file.

System Requirements. Professional Oracle runs under DOS 3.x on a 286, 386, and IBM Personal System/2, Model 50 and above. It requires 1.5MB memory and a minimum of 6MB of disk storage. The product uses the memory above 640K for the DBMS code and data, storing only an 80K Oracle Executive within the 640K managed by DOS. This leaves a sizable chunk of DOS memory (500K) available for user applications. The Oracle Executive is essentially an extension to DOS 3.x, managing the necessary transitions between protected mode (DBMS operation) and real mode (DOS operation).

Market. Professional Oracle is a standalone product aimed at value-added resellers (VARs) and applications developers in Fortune 500 companies. The benefit of having a complete implementation of Oracle 5.1 on the PC is the ability to develop Oracle applications on the PC and port them directly to a mini or mainframe. The source code for all Oracle products is the same, regardless of the hardware or operating environment. In addition, Oracle is compatible with DB2, a prevalent DBMS in the IBM MVS operating environment. So users of Professional Oracle can also develop DB2 applications that are directly portable.

A networking option is also available for Professional Oracle, giving it the capability to create distributed database applications.

Oracle will also run under the IBM/ Microsoft OS/2 operating system when it becomes available early next year.

NETWORKSTATION ORACLE. The second product announced is Networkstation Oracle, which permits any PC in an organization to access information in distributed databases regardless of location. It allows all IBM PCs and compatibles, including the Personal System/ 2 family, to act as application processors in conjunction with an Oracle DBMS running on another PC, mini, or mainframe. The communication between the application and the distributed DBMS can be asynchronous, 3270 coax, Ethernet, or token ring.

What Oracle has done is to split the operation of the DBMS into two parts: a front end and a back end. The Oracle application and user interface (front end) run on the PC, while the actual manipulation of the database (back end) takes place on a database server. Through Networkstation Oracle, the application makes high-level SQL requests of the server, and the server responds with the selected information. Benefits here include the ability to offload processing to the workstation and reduce traffic on the network.

Networkstation Oracle runs on any IBM PC with 512K of memory.

LANSERVER ORACLE. To complete the PC picture, LANserver Oracle introduces a dedicated, distributed, multiuser database server that runs on a 286/386 PC. This product provides the back end of the Oracle DBMS for a multiuser PC environment. LANserver Oracle works in conjunction with Networkstation Oracle (the front end) on a PC LAN. Each user workstation on the LAN must have Networkstation Oracle installed to access the database server. The LANs supported include Novell (Advanced NetWare), 3Com (3Plus), and IBM (Token-Ring).

LANserver Oracle differs from most PC-based DBMSs that have been adapted to LANs (such as dBase III Plus) in that a single database back end runs on the server where the data is stored, and only the front-end software runs on each workstation on the network. In the case of dBase III Plus, each user has the full DBMS running on the PC workstation, while only the data is stored centrally on the server. Thus, the DBMS cannot automatically provide data integrity and concurrency control; these security features must be built in at the application level. LANserver Oracle, with its single back-end processor (resident on the server with the data) for all DBMS users on the network, can provide concurrency control and database integrity across the network.

In addition, as we mentioned above, this architecture reduces traffic on the network. The front end sends SQL commands to the back end and receives the requested information in return. Processing the SQL request on the server eliminates the need to pass all of the data across the network to the

DBMS, as would be the case with dBase III.

LANserver Oracle requires a dedicated 286, 386, or Personal System/2, Model 50 or above, and DOS 3.x. Memory and storage requirements will be announced closer to actual shipment (fourth quarter), but LANserver Oracle will probably require at least 2MB of memory. There will be a stated maximum number of users for LANserver Oracle; the number of users it can support will depend on memory, storage, and transaction load.

COST AND AVAILABILITY. Professional Oracle ($1,295) and Networkstation Oracle ($695) will be available this quarter. Both products include SQL*Forms (forms-based applications generator), SQL*Calc (Lotus-like spreadsheet with built-in SQL access to Oracle databases), SQL*Plus (complex ad hoc query facility), and SQL*Report (multitable report generator). Optional software includes a Cobol precompiler and a networking option for Professional Oracle ($395 each). LANserver Oracle will be available in the fourth quarter at a cost of $2,495. ◐

Order Form

Subscription rates:		U.S.A.	Canada	Foreign
Patricia Seybold's UNIX in the Office	*12 issues per year*	$495	$507	$519
Patricia Seybold's Office Systems Report	*12 issues per year*	$385	$397	$409
Patricia Seybold's NETWORK MONITOR	*Newsletter only, 12 issues per year*	$595	$607	$619
Patricia Seybold's NETWORK MONITOR	*Newsletter (12 issues per year) and Reports*	$2,195	$2,207	$2,219

□ Please send me a sample issue of *Patricia Seybold's Office Systems.*

□ Please send me a sample issue of *Patricia Seybold's NETWORK MONITOR.*

TO ORDER BY PHONE, CALL (617) 742-5200

Multiple-copy discounts and group rates are also available. Additional copies of UNIX in the Office *and* Office Systems Report *are $35 each to subscribers and $50 to non-subscribers. Copies of* Network Monitor *are $49 each for subscribers and $65 for non-subscribers.*

□ **My check for $__________ is enclosed.** □ **Please bill me.** □ **Please charge my subscription to:**
Mastercard/Visa (circle one)

Name: ____________________ Title: ____________________

Company Name: ____________________ Tel. No.: ____________________ Card #: ____________________

Address: ____________________ Exp. Date: ____________________

City, State, Zip Code: ____________________ Signature:

Country: ____________________ ____________________

Checks from Canada and elsewhere outside the U.S. should be made payable in U.S. dollars. You may transfer funds directly to our bank: Shawmut Bank of Boston, State Street Branch, Boston, MA 02109, into the account of Patricia Seybold's Office Computing Group, account number 093-118-7. Please be sure to identify the name of the subscriber and nature of the order if funds are transferred bank-to-bank.

Send to: Patricia Seybold's Office Computing Group: 148 State Street, Boston MA 02109; (617) 742-5200