

programs, and so on. The developers of these other types of application programs can take advantage of the database management program without going into the database business.

But the most important advantage of database servers is they are the first step on the road to distributed databases, and further down the road, distributed processing. There are a couple of true distributed databases on the market today. But keep in mind that a database server is not necessarily a distributed database. A database server has one server and many clients. A distributed database has many servers and many clients, but the multiple servers are transparent to the clients.

SQL: The Good and The Bad

Most database servers support Structured Query Language (SQL). SQL (sometimes pronounced Sequel), a query language developed by IBM in the 1970s, is the *de facto* standard for getting information from relational databases. Within the last year or so, SQL has been adopted by the majority of PC database vendors. The minicomputer database people have been supporting SQL for a while longer.

SQL is an English-like query language. It allows you to do things like "SELECT employees FROM employee database WHERE salary \$50,000." It is important to note that while most database servers also support SQL, support for SQL does not make a product a database server.

SQL is a boon for distributed databases. It is the high-level mechanism by which distributed databases will be able to get information from each other. SQL is being supported on every computer platform, from Macintoshes to IBM PCs to minicomputers and mainframes. Ignoring any compatibility problems, SQL support means that any database program can get information from any other one, providing they all support the same SQL set.

SQL is a *query* language, not a procedural programming language for building programs and applications. It does not contain things like conditionals or looping. For that reason, SQL commands are often embedded into a programming language, such as C.

Several companies have written extensions to SQL and incorporated it into an Application Programming Interface (API). For the Macintosh, there's the Connectivity Language/One (CL/1) API

DEERE'S DISTRIBUTED DATABASES

John Deere isn't just tractors and farm machinery. It certainly isn't in the backwoods. Deere is on the forefront of computer integrated manufacturing (CIM) and distributed databases.

"A distributed database environment has grown up on LANs. We feel that as Deere tries to embrace distributed processing and data management, new techniques are needed. That approach has grown up from a LAN to a WAN approach," says Doug Foster, a network systems integrator with the Network Consulting Group of Deere Tech Services in Moline, IL.

Deere Tech Services is the official tech support arm for all John Deere factories. It also sells software for computer aided process planning and other computer integrated manufacturing (CIM) type projects. Deere Tech Services also does consulting. According to Foster, with 30,000 SNA nodes and an entire floor of 3090s, Deere's computer center is among the largest IBM installations in the country.

"Within Deere, people have used Oracle on the individual workstation for application development as well as for distributed application development. It works fine on a LAN level. What we need is to tie that LAN into more and more data resources," says Foster. One of the ways is distributing DB2 databases.

One of the ways of distributing data is installing DB2 and Oracle for MVS on an MVS host. For MVS machines, Deere uses Oracle's SQL*Connect which lets a local Oracle user access remote DB2 tables as if they were local. "We've essentially distributed DB2 down to a LAN," says Foster. To make the distribution of data feasible, the LANs at the offices in Dubuque and Davenport, IA, as well as the headquarters and Deere Tech Services, both in Moline, IL, have been tied into a WAN.

Information is distributed in a couple of ways. One is a TCP/IP-to-LU6.2 gateway between MVS and one of the LANs. "This allows us to take a Sun workstation on the LAN and hit the MVS machine running DB2 as if it held local tables. We are actually going away from a TCP/IP-to-LU6.2 gateway. Any time you go through a gateway, you have a bottleneck," says Foster. An IBM PC RT is the gateway. If many people on the WAN go through this gateway, performance will suffer. While one solution is to get a more powerful gateway machine, Deere chose another method. "One of the things we want to do is to make the host appear on a LAN. The products from ACC allow us to do it," Foster says.

"The need for distributed databases within a TCP/IP environment manifests itself in providing MVS connectivity to Unix workstations," says Gary Krall, Advanced Computer Communication's (Santa Barbara, CA) director of marketing. Acces/MVS is TCP/IP for MVS mainframes. In version 3.0 of Acces/MVS, there will be an API such that MVS-based applications can use the services of TCP/IP.

Deere installed ACC's Acces/MVS. "We are about to take it into production mode," says Foster. Deere has attached a 3090 to the LAN. The 3090 supports both SNA and TCP/IP. "We'd rather do peer-to-peer than LU6.2 gateways. We'd like to go straight to the host. We can do that with a product like Acces/MVS and Oracle's SQL*Net that supports TCP/IP," says Foster. SQL*Net for TCP/IP runs on MVS and gives Oracle users access to DB2.

"The ideas is we have intelligent workstations of different classes: PC RT, Sun workstation, Macintoshes, DEC workstations. We take an intelligent local workstation, put it on the LAN, interconnect into a WAN. By placing an IBM MVS mainframe as a peer on the network, users can share the resources, namely CPU cycles and data. Our CIM applications will take advantage of the distributed environment. It shouldn't matter what the workstation is on the front end or what machine is the back end for the compute station or data server. The idea is to make it transparent," says Foster.

"It's pretty neat," he continues. "Conceptually, it's going from the workstation to the host for data management. Take Oracle for Macintosh and SQL*Net for Mac. Think about the front-end to a DB2 database being a Mac with a HyperCard. For the executive who doesn't want to bother with selects and the syntax of extracts, he can click on a Mac and get a distributed database," says Foster.

(see box). DB-Library is the API for SQL Server that Microsoft says is better than using embedded SQL. Some database engines make no extensions to SQL.

A BILLION DOLLARS

A billion dollar bills weigh 1,020.4 tons.