

# DBMS

FEBRUARY 1990 • \$2.95

(Canada \$3.50)

VOLUME 3 • NUMBER 2

INCREASING DATABASE PRODUCTIVITY



SPIDERANALYZER P320



3835175230 9



## THE NEW ORACLE?

*Gupta Technologies' namesake reveals his strategy for taking the lead in the coming age of graphics-based environments.*

Umang Gupta goes back to the very beginning of SQL on PCs. His experience with a group of IBM upstarts explains why Gupta Technologies got its head start in SQL database servers on LANs with SQLBase — the first to ship by several years — and in graphical database front ends with SQL Windows. His experience at Oracle also seems to have had an impact; following in the once small, now giant footsteps of Oracle, Gupta Technologies has thus far doubled its revenues every year of its existence. The company's newly introduced SQLBase 4 (see this issue's featured new product) and a retail version of SQL Windows may allow Gupta to maintain that pace.

The story begins in 1983. The IBM Entry Systems Division in Boca Raton, Florida was riding a long hot streak; they could do no wrong. Boca had built a billion dollar business from nothing, faster than IBM ordinarily put together a request for bids.

The IBM PC AT was about to come out, which would put Big Blue's PCs back in front technologically. Someone in Boca realized the 80286 at the heart of the AT had enough horsepower to run a SQL database that could be a peer with databases on IBM's large machines, yet offer the responsive user interface people had come to expect on PCs. And so it was that the Entry Systems Division put out requests for bids to a handful of firms for a DB2-compatible database optimized for the AT platform. Umang Gupta, then working for Oracle, was one of the people who put together a specification proposal for IBM's PC version of DB2.

Insiders say that when IBM's Santa Theresa lab, birthplace of the relational theory and DB2, heard of Boca's move, they were up in arms. Relational databases were a strategic IBM technology; the PC hardware people, they said, had no business messing around in *Santa Theresa's* key technology. It was a turning point, because the upper echelon at IBM listened to Santa Theresa. Relational databases were too important to be left to an independent entity; indeed, perhaps PCs were becoming too important to be left to an independent entity. It was the beginning of the end of the independence of Entry Systems; the mavericks who created the PC were soon folded in with the rest of IBM.

Developing that relational database bid for Entry Systems was also the impetus that convinced Umang Gupta to leave Oracle, where he was vice president and general manager of Oracle's Micro Products group. Oracle had high hopes for the PC business; the company Gupta founded eventually built the first version of PC Professional. But Gupta leaned more toward the philosophy of the IBM bid, which was to build each database engine and user interface to fit the platform it ran on. That's where the conversation between DBMS Editor in Chief Kevin Strehlo and Umang Gupta began.



**DBMS:** Why did you leave Oracle?

**GUPTA:** I felt that to bring SQL to the PC you need an engine that's optimized for PC usage, not just for the hardware. Oracle's user interfaces, having emerged from the minicomputer world, would not be attractive to the PC marketplace. At Oracle there was — *is* — a clash of cultures. Portability is the key at Oracle. Make this product look, touch, feel the same on minis, mainframes, and micros. That's a valid concept if your primary goal is to sell products on minis and mainframes, and, peripherally, PCs. It's not how you sell PC software, though.

**DBMS:** Oracle professes to be very serious about PCs, particularly the LAN business. They've introduced database servers for PCs now for all the network platforms.

**GUPTA:** They *are* serious. But there's a difference between selling a database server on LANs and selling a LAN DBMS





PHOTOGRAPHY BY LARRY CLAY

system for PC users. A server on a LAN can have many of the same characteristics as on the mini and the mainframe. But a LAN DBMS for PCs has to differ. To optimize performance on a PC, you may want a client-server architecture. You can make *engines* portable to the PC — at the expense of performance — but not user interfaces. You cannot build a user interface that's optimized for the new graphics environments — Microsoft Windows, Presentation Manager, or the Macintosh and X-Windows platforms — and port it to dumb mini and mainframe terminals without sacrifice. I don't believe Oracle has been willing to make those hard choices.

**DBMS:** A lot of companies working on front end strategies extend application portability not only to graphic workstations — the Mac, to Windows and PM on PCs — but also to character mode on PCs. Gupta has chosen

**to ignore character mode. Why?**

**GUPTA:** I believe in some of the early landmark principles of user interfaces. To allow the human imagination to flower you have to have bit-mapped graphics. For example, you want to write a check-writing program? Put a picture of a check in front of the person. You want special help facilities? Draw pictures. You can do that in a graphics-based system, but not in a character-based one.

**DBMS:** But even in character mode, you can have an interface that uses the mouse, pull-down menus, scroll bars . . . . You can get the chief advantages of graphic interfaces, yet run on most existing PCs.

**GUPTA:** It is possible. But I don't think the graphics environment will stop at scroll bars and pull-down menus. We're going through a transition where people are experimenting and changes are coming in incremental steps. They've gone

away from function keys to hierarchical menus, for example. Scroll bars have replaced function keys for browsing. And graphic images are becoming more prevalent. Look at our SQL Windows products. Look at Excel. Its graphical capabilities are phenomenal! So if you limit yourself to character-mode systems to accommodate existing systems you'll always have an upper limit. And hardware prices are coming down. People are willing to buy two-, three-, four-megabyte 386 machines that can run Windows and PM, so why not be at the starting point of a graphics-based environment cycle?

**DBMS:** My own company is building a sales support application using the SQLBase engine. Our MIS director loved SQL Windows. But he looked at the installed base of machines with lots of 8088s and one-megabyte 286 machines, and thought, "If I want good performance, I'd have to upgrade all



these machines to 386s with four megabytes. There's no way."

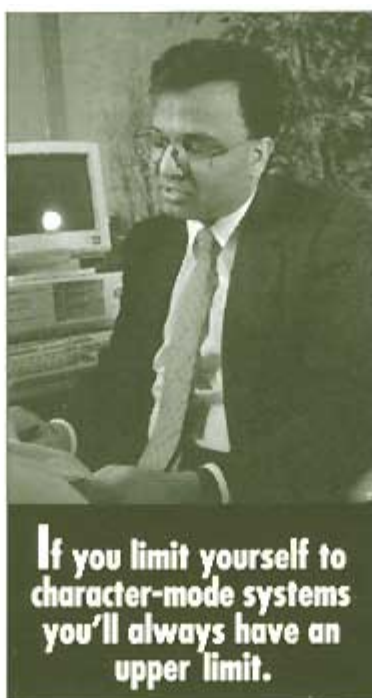
**GUPTA:** As a company, we are depending on the fact that over time most people will have 386 systems with four megabytes. However, we're not depending on that for revenues this year or next. Our strategy is to offer a range of products. You want to use our database engine in single-user or multiuser mode? You can. You want front end tools that have the Windows graphic interface, and you have the right hardware? You can. If you don't, no problem. Use Clipper with the Planet Library. Use COBOL. Use C. We have customers using our systems on 8088-based machines. We're at the start of a product's life cycle, so we need to offer choices well into the 1990s. This is the beginning.

**DBMS:** A large part of your business will come from Fortune 2000 companies that today run SQL databases on large platforms. Let's ignore price and performance and say they want better user interfaces, so they're moving to PC LANs. But won't they go with those large system database companies with whom they've established relationships, whose support organizations they know and trust?

**GUPTA:** Fair question. But you could also ask, "Why doesn't everybody just buy IBM?" But everybody doesn't just buy IBM. People buy things they need from a particular vendor that they don't get from another vendor. It comes down to what market the vendor wants to emphasize. Traditional minicomputer DBMS vendors emphasize the one-size-fits-all philosophy. Here's my DBMS. Here's a set of tools. Run them on your micro or your mini, and I'll connect it all together as long as you buy my product for every one of your machines.

**DBMS:** IBM hasn't done that. They have different source code for all of the platforms, the AS/400, PC, two mainframe products. They will have real connectivity with a distributed database in which the location of data will eventually become invisible. How do you compete with that?

**GUPTA:** IBM's connectivity strategy makes a lot of sense. They're not trying to sell a desktop machine with 370 architecture. They're really different architectures. The operating system is different. They're optimized. And they plan to provide universal connectivity through SAA. Our connectivity strategy builds upon the SAA concept. We agree that you should optimize your engine for the platform. But IBM says, "Buy an IBM product on your mid-range mini and mainframe, and we'll connect it all together." We say, "Buy any DBMS engine of your choice, and we'll connect it all together."



**DBMS:** Your SQL Network product provides connectivity to DB2, but how many people are actually using it today? Isn't it more of a check-off item to assure people they can implement future strategies?

**GUPTA:** It's a more important part of business measured in dollars than in units because of its price point. We have more than 2000 locations right now. Not all of them are connected to DB2. But because most of those customers are major corporate accounts, almost all want to be confident that they can one day connect to DB2 or Oracle. And those people who are connected to DB2 are doing serious work. Citicorp, for example, has a foreign currency trading application that uses a connection to DB2.

**DBMS:** Why is DB2 the database to connect to on mainframes, when SQL/DS actually has a greater number of installed sites.

**GUPTA:** DB2 and SQL/DS are not that different. They do GRANT and REVOKE differently, but most of the data manipulation statements — SELECT, UPDATE, INSERT, and DELETE — are the same. The main reason you hear more about DB2 is that it is used in corporate production shops; it is being sold as a potential replacement for IMS. That's where the corporate data really is. SQL/DS shops tend to use it for decision support.

**DBMS:** Why isn't SQL/DS a more logical match to networks and database servers. Aren't they going to be used primarily for decision support?

**GUPTA:** I disagree. Networks will continue to replace traditional mini- and main-

frame-type applications for mission critical work as well as for decision support. I think the lines between the two will blur, and you'll have to build systems that accommodate both from the same corporate data sources.

**DBMS:** One thing that got a lot of play in the press was the concept of groupware. We're seeing some groupware actually come out now; for example, Saros has a document tracking system built on SQL Server. Are there groupware applications like that under development on SQLBase?

**GUPTA:** First, let's define groupware: It is usually a set of tools that increase the productivity of a group of office professionals, rather than an individual. For example, Lotus 1-2-3 is a personal productivity tool, while electronic mail, or products like Lotus Notes or OfficeVision from IBM, will be groupware. You can expect to see a number of groupware products on our system. For example, the PSDI project development application will run off SQLBase. Then there's another group, the multiuser data query and access tools. To the extent that they work with a database server, products like Excel are also groupware. Over time, SQL Windows will evolve into that kind of groupware tool for office automation and end user ad hoc query and reporting. It will compete with Paradox and other PC DBMSs.

**DBMS:** How is the connectivity handled between Excel and SQLBase?

**GUPTA:** In two ways. Excel can have a direct connection to SQLBase. One of our distributors in the United Kingdom, Henly Software, built a product called SQL Vision that allows you to use Excel as a front end to SQLBase and automatically connect to DB2 or to any of our back end engines. It uses DDE as the direct router to talk to SQLBase. We expect to introduce SQL Vision in the United States, shortly. Excel can also connect as a cooperating application with SQL Windows. You write an application in SQL Windows and use SQL Windows' DDE functions in SAL, our 4GL. You could have a set of Excel macros written on the Excel side and a set of SAL DDE functions on the SQL Windows side, and the two cooperate.

**DBMS:** You've just hired Chuck Ellison, who headed Ashton-Tate's retail efforts during their explosive growth years. The retail channel is an interesting place for you to go, because it seems that your focus is on Fortune 2000. Is that a necessary channel for these decision support products that will tap database servers?

**GUPTA:** Our company started as a technology company building database serv-



ers. And we recognized that such technology had to be sold directly to companies who had the technical expertise to want to use it. That means the Fortune 2000. But there are explosive growth possibilities for database servers in the 1990s, not just for accounting or OLTP applications, but for office automation. To make that happen, we have to offer easy-to-use tools that the average business professional can use on a PC, either stand-alone or on a LAN with a database server or perhaps connected to a mainframe or a VAX. Offering those tools to a mass market is essential, and that's why we're going retail.

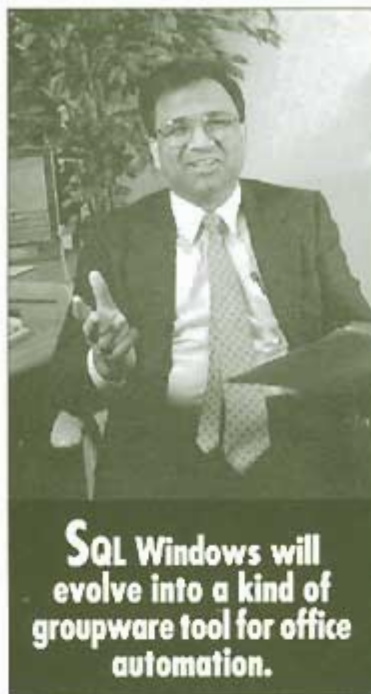
**DBMS: What about the retail DBMS competition? Beyond today's players, Lotus has announced and demonstrated DBMS tools, and although Microsoft hasn't made any official announcement, everyone has heard of Omega.**

**GUPTA:** There are clearly a set of front end tool companies in the PC DBMS business: Paradox, dBASE, Microrim. Existing products will be retrofitted. Some time next year, you can expect the Microsoft Omega product and the Lotus DBMS product, and who knows what else. We feel we have an incredible technology lead in terms of the front end tool business. Anybody who comes into this market — six months, a year from now, or whatever — will have to evolve their product over time to catch up.

**DBMS: But how do you win the necessary mind share. Take servers: We ran a contest where we asked our readers to vote for their favorite products. We asked them to vote only for ones they actually used, but that didn't happen. The winner in the database server category was Oracle Server, their older product that runs only on Xenix and requires Excelan LAN hardware. That's a small percentage of the market. If Oracle wins in a very database-literate group's mind with a product with a minimal installed base, how do you get up to the level of mind share you need to seriously compete?**

**GUPTA:** There's a big difference between mind share and market share. How many of those people actually bought Oracle? Mind share is an important element of establishing good market share, but over time mind share changes. We believe we can swing things our way by leveraging our technology position with good management, strong marketing, and a strong ability to get products through distribution channels.

Our products span back end servers, front end tools, and connectivity. It's important to sell these products through channels that make sense. For end user PC tools, a portion of our marketing must



be strong in mass market channels. On the other hand, we need to sell to the technically oriented people directly. Our alliances aim to round out our distribution capabilities, because we *have* all the products we need. The best way to do that is to establish strategic partnerships with companies that have access to a large corporate sales force and to large markets that we could not otherwise get into.

**DBMS: It seems that the key is to leverage the sales and support force that got Novell into nearly every large company in the U.S. And no database server vendor seems to have gotten a grip on that channel.**

**GUPTA:** Novell has done a marvelous job with the VAR channel.

**DBMS: How do you leverage that?**

**GUPTA:** We have a high regard for Novell's leadership position and the VAR base that sells Novell products, and we are actively working with VARs in the LAN business. Most of them are Novell resellers.

**DBMS: What do you have to do to make it attractive for Novell VARs to sell your products?**

**GUPTA:** We offer them a complete solution that they can sell to their customers. We've assembled the most qualified set of technical and sales people in the LAN DBMS business. We are *the* experts in the business.

**DBMS: You mean because everybody else is just getting started with client-server, because their products are just shipping.**

**GUPTA:** And we've been there three years now.

**DBMS: What about Dbase? On the PC the predominant database technology is the Dbase language, and some people seem to be waiting to see what the major Dbase language vendors deliver before making any decisions on database servers. In a plan-to-buy survey we did, Fox Server is very strong, and people are still waiting for Ashton-Tate's connection from dBASE to SQL Server. What will have to happen for you to make inroads?**

**GUPTA:** We're already making inroads. We have former dBASE customers, who would rather use a product that does not use dBASE but works with SQL Windows. We also have customers who've said, "We'd prefer to stay with dBASE," and they're buying the Clipper connection to our system. The whole world doesn't have to switch to graphics for us to be successful; it's only necessary for some percentage of the world to buy our graphics tools, some percentage to buy the character-based connections to our SQL database engine, and some percentage to buy connectivity products. We're not looking for a hundred percent market share.

**DBMS: You have serious competition in the Dbase world. Nantucket sees the same potential in the client-server architecture. Their next version of Clipper has a "choose your back end" strategy, where you pick an engine, but the front end tool will be Clipper.**

**GUPTA:** And we want to encourage every Clipper developer to develop for SQLBase as the back end, or our SQL Gateways as connectivity to DB2 . . .

**DBMS: So it's a two-prong strategy.**

**GUPTA:** It's a three-prong strategy. If you buy our front end tools, we love it. If you'd prefer existing tools, buy our database server. If you'd like to buy someone else's database server, use our connectivity tools. You don't need to buy SQLBase from us to have connectivity to DB2. You can develop an application in Clipper and have that application talk to DB2 without a SQLBase engine in the middle — just buy our SQL Gateway and a SQL network product.

**DBMS: The other Dbase question is whether Dbase is a good fit to SQL.**

**GUPTA:** A lot of people, especially dBASE people, say you don't have to change your existing dBASE applications to move into the SQL world. That's a myth. You can't use most real-life applications with SQL without some reengineering. You can easily map certain queries, such as



a single table query saying, "Give me the name" from a dBASE statement to SQL using the SQL GROUP BY operator. But nested queries are impossible to map from dBASE into SQL because it has no equivalent capability. dBASE, of course, writes a program to do the job. You could always do all the work in the front end and treat the back end SQL engine as a table server having a single table. That can be easily mapped, but you're unlikely to get good performance.

**DBMS:** Another issue that arises is the need to mask parts of Dbase. For example, Dbase has the concept of record number, time stamp, and of a record that's been marked for deletion but is not yet deleted by a "pack." If that information is in a SQL table and another front end accesses it, SQL Server hides the Dbase-specific records, or hides records that have been marked for deletion, or whatever is necessary. Doesn't that give SQL Server a competitive advantage in terms of linking to Dbase front ends?

**GUPTA:** You're saying that SQL Server has some features that make it perform invisible SQL better. I'm questioning the very basis of invisible SQL. Typical applications don't require single table JOINS, single table queries; they do JOINS. Dbase

**We say, buy any DBMS engine of your choice and we'll connect it all together.**

apps use aggregation and nesting queries, and those are not mappable to SQL.

**DBMS:** Do you have customers who use the Planet libraries and have found that their Clipper applications don't translate over?

**GUPTA:** Most people who use Planet recognize that Dbase applications do not translate. They do it because they want to take advantage of the additional features of our SQLBase engine; they access raw SQL capabilities.

**DBMS:** Despite the poor matchup between Dbase and SQL, wouldn't you like to tap the widespread expertise in Dbase as a programming tool?

**GUPTA:** Sure. Third-party programmers are developing dynamic link libraries for SQL Windows to handle graphics imaging, to attach Excel connections, and potentially even to make Dbase code invocable from SQL Windows, which is a

possibility we are currently exploring.

Dbase is also important as a model. I think the biggest reason for Dbase's success is that it is a very strong programming platform that allows anybody to develop the most sophisticated of applications. Over time it has developed an incredibly loyal base of third-party developers who have built applications that allowed Dbase to be used by an average end user. At the same time, Ashton-Tate kept extending its end user capabilities until it was also a stand-alone data query tool, an alternative to products like Symantec's Q&A. I believe that the key to success in the new graphical marketplace will be to follow those footsteps. That's why we came out first with a programmable DBMS. You can develop sophisticated applications with it. In the coming year, we will deliver a version that lets the average end user leverage SQL Windows, so when a corporate buyer looks at it he sees a strong product his end user clientele can use straight out of the box as well as an extremely strong product to write applications that connect to his database servers. The original dBASE II didn't do all those things. It was dBASE III, dBASE III Plus, and maybe one day dBASE IV. And I suggest to you that just as we brought out SQL Windows I, there will be a SQL Windows II next year, and a III beyond that. ■