History of Sybase

Bob Epstein

One of the cofounders of Sybase tells about the birth of the company in 1984 and its strategy for entering what appeared to be a crowded marketplace. The article covers the company's initial capital funding, technical development of the early products, the technical advantages of Sybase (along with its marketing thrust to compete with Oracle), business relationships, and eventual acquisition by SAP.

''The world doesn't need another database company!'' With that message, yet another venture capitalist turned us down. The year was 1984, and we were determined to raise the capital needed to start Sybase and prove our detractors wrong. This is the story of how we created a multibillion dollar company and, seven years later, provided a 17 times return for the venture capitalists who did invest in us.

The World Before Sybase

Sybase was founded at a time when relational databases were being used for decision-support applications on terminals running on minicomputers. Serious, online databases were still written for and run on mainframe computers.

I thought the computing world was about to change. Microprocessors were beginning to outperform minicomputers and the IBM PC was three years old. At the time, a 1 million instructions per second (MIPS) computer was a big deal—and a bargain at $150,000! Ethernet had finally reached the point where offices were beginning to install local area networks. Unix workstations were just being popularized by Sun Microsystems, and the need for rapidly developed, specialized applications could not be met by centralized, mainframe development organizations. Our bet was that computing would become more distributed and applications development would become decentralized. The phrase we coined to describe this was client-server computing. This systems approach would break up monolithic applications into separate components that would interrelate over a network for faster processing. Clients (personal computers) supported the applications, and servers (minicomputer and mainframe computers stored the databases) supported the shared data, security, and data integrity rules.

Sybase was my third database effort. During graduate school in 1977–1980, I was the third manager of the INGRES Project at the University of California, Berkeley, under the direction of Michael Stonebraker. INGRES was one of a handful of relational database research projects started after E.F. Codd published his seminal paper on relational databases.[1] During that period, several hundred licenses for INGRES were issued, and it became clear that a market for relational databases was about to emerge. Relational Software Inc. started marketing ''Oracle'' during the same time period.

Although Stonebraker put the INGRES code in the public domain and launched Relational Technology Inc., I chose a different path. I was concerned about the incredibly slow performance of relational database software running on minicomputers, for it could only support a handful of users simultaneously. When Dave Britton and Geoff Lee approached me to be their technical person to help start a database hardware company called Britton-Lee, I jumped at the chance. Their goal was to build specialized hardware to optimize relational database performance. Along with Teradata, we established a new category called ''database machines'' and sold our products to performance-oriented customers.

After a few years, it became obvious to me that building dedicated hardware was no longer necessary and that we could in fact optimize database performance across a network while running on general purpose hardware. I quit Britton-Lee in 1984 and convinced Mark Hoffman, Tom Haggin, and Jane Doughty to join me in starting Sybase. I had met Mark at Britton-Lee, where he

More than one venture capitalist told us they were unconvinced that there could be any billion-dollar software companies.

impressed me with his strong operational and managerial capabilities. Tom and I were friends from UC Berkeley. Tom was the first person to describe a practical use for the one-byte INGRES data type that stored numbers between 0 and 127; he told me jockeys never weigh more than 127 pounds, which made INGRES the perfect system to track jockey weights! Tom and I knew Jane from UC Berkeley, where in addition to great programming skills, she was a natural at program and project management. Mark became Sybase's president. Jane was responsible for the engineering of the database server. Tom took charge of the application interfaces and development tools. I was executive vice president and did whatever was needed, including designing the software and managing the sales, marketing, and finance. I even wrote and gave the first customer training classes.

Raising Money

In 1984, Cullinet Software was one of a relatively few publicly traded software companies and they sold mainframe databases. Venture capitalists were not yet focused on software companies, and more than one told us they were unconvinced that there could be any billion-dollar software companies. It wasn't until 1986, when Oracle and Microsoft went public, that the public market for software companies was firmly established.

Mark and I started working full time in August 1984. We worked out of my house until Tom and Jane joined us in October. Working from home wasn't always straightforward. One afternoon, my wife Amy walked in, handed our three-month-old son Harris to Mark and said, ''Here Honey, take care of the baby for a while.'' None of the founders of Sybase had ever raised venture capital. We needed a hook to get venture capitalists interested. From our Britton-Lee experience, we had established a relationship with the TRW Systems Group and their customers in the intelligence community. There was a significant interest in developing systems that could meet the emerging security standards. At the time, the US military had established its own computer language (Ada), and TRW was interested in developing a database system in Ada. We entered into a contract to share software designs with TRW for their use in their market. The cash from those contracts helped us attract venture capital—but not fast enough.

I would alternate between software design, supporting our TRW contract, and looking for money with Mark. We were rapidly running out of cash and reaching the limits on our own credit cards. By January 1985, we had missed our self-imposed deadline for raising venture money. Fortunately, Britton-Lee went public in January 1985. A portion of the proceeds from the stock I sold were loaned to Sybase, and that gave us enough money to maintain our hiring plan until we closed a $1.5 million round of venture funding from Hambrecht & Quist, Kleiner Perkins Caufield & Byers, and TRW three months later. The ''Series A'' stock was sold for 38 cents per share (seven years later, our public offering would be at a split-adjusted price of $6.75 per share) with the employees keeping 40 percent of the stock. Combined with a $1 million contract from TRW, we finally had enough cash to carry us through the initial product development.

Product Strategy

We delivered on an 18-month development cycle, so by the time our product was ready for beta testing, it was the fall of 1986. The market leader was Oracle, which was a $100 million revenue company by the time we made our first sale in 1987. Other established companies included Ingres, Informix, Gupta, Esvel, and others. We obviously needed a strong and sustainable differentiation in order to compete. Using the client-server model, we assumed we would be running on a dedicated server servicing a large number of client workstations or PCs. Our goal was to make a database machine out of off-the-shelf hardware. Our development environment of choice was Sun workstations and servers because we all wanted a Unix development environment and Sun's hardware offered the best price and performance.

The features that made our initial product a success went by the acronym PIADT (performance, integrity, availability, distributed, and tools).

Performance

Our competition was slow running large numbers of users because they had a separate operating system process for every user. We developed our own threading so we could switch between users in tens of microseconds while our competitors required 10 times as long. In addition, the amount of memory needed per user was so small that we could manage a hundred of users in 1 Mbyte of memory. We also invented ways of managing data more efficiently. After having built two implementations of relational databases at Ingres and Britton-Lee, this was our third implementation of relational databases. We knew which portions of the system needed special optimization.

Integrity

We were also the first relational database that allowed data integrity and consistency checks to be programmed directly in the database design. In this way, the logic was shared independently of the many applications that could access it. The integrity was initially implemented as stored procedures. For example, to update a checking account balance, the application would call a procedure in the database that knew all the business rules associated with account updates. One of our early customers asked if we were also planning on implementing ''triggers''—code that would be called before an update to the databases was allowed. I explained that it was too hard to do before realizing that evening that our stored procedures would actually make the implementation of triggers a breeze.

Availability

The system supported features that were already standard on mainframes such as online backup and maintenance. We also supported mirroring of disks to protect against data loss caused by disk failures.

Distributed

Our system was built around the notion of a network and was naturally distributed across multiple computers. We included the ability to coordinate commits across multiple databases.

Tools

We provided development tools for both character-oriented terminals that were the predominant data-entry device at the time, as well as graphical tools that are taken for granted today but were still rare in 1984. Our tools could run in a timesharing mode on character terminals on a PC or Sun workstation.

Early Market Positioning

While we felt good about our design and technology, none of us had any marketing experience. In 1986, as we were approaching our first beta test sites, we went looking for a vice president of marketing and tried to recruit Stu Schuster from Ingres. Stu's initial reaction to our pitch was that the market was already crowded and didn't need another relational database company. He agreed to meet with us nevertheless and was excited by the transaction-processing capabilities of our system, which he hadn't seen anywhere else. When he joined our team, Sybase was still in ''stealth mode'' and keeping a low profile. We did not want to launch the product until we were in production with real customers (in the middle of 1987). Our tag line before Stu joined was ''fulfilling the promise of relational databases''—not exactly effective or creative marketing.

Stu designed our market positioning based on two key components. The first was that we were only going to compete against one company: Oracle. With many companies in the market, we wanted to create the perception of a two-horse race. We hoped that Oracle would become so irritated that they would choose to compete with us and thus raise our visibility. The second component was to focus on a new class of applications: ''online'' as opposed to ''decision support.'' Online is characterized by frequent updates, many users, and immediate response-time requirements. By contrast, decision support tends to be mostly reporting, infrequent updating, and fewer simultaneous users.

Combining the two components of the strategy, we claimed that Oracle was fine for decision support but that you needed Sybase for online applications. At the time, Oracle was particularly weak at update-intensive applications and multiuser applications. Some of our sales people whose prospects were planning on decision-support applications complained. Nevertheless, the strategy of clear focus worked well for us.

The business decision to work with Microsoft was anything but straightforward.

In August 1987, we completed another investment round totaling $3.3 million that included Apple Computer with a minority investment. Apple said the investment was part of its strategy to encourage development of products that would promote sales of its personal computers. Our original investors Hambrecht & Quist, Kleiner Perkins Caufield & Byers, and TRW also invested, as well as our second-round lead investor Charles River and the newcomer Oak Investments. We were clearly getting attention from the industry.

Our initial hardware strategy was to support Sun and Digital Equipment VAX/VMS. We did the development work on Sun and then ''ported'' the software to VMS. At the time, Sun was trying to break into Wall Street and was competing with Digital Equipment. The combination of Sun and Sybase proved powerful for both companies. If the Wall Street customer selected Sybase, their best hardware value was Sun. Correspondingly, if the customer first selected Sun hardware, they were likely to pick Sybase for the database. This positive reinforcement helped both Sun and Sybase become the strongest combination on Wall Street. Our market positioning, which was designed to contrast the strength of Sybase for online applications to Oracle's for decision support, was working. In fact, it was working so well that Oracle announced that their version 6, to be released in 1988, would be able to do online applications too.

That's when a new endorsement of Sybase threatened to turn us into a more formidable rival than Oracle had probably ever imagined.

Microsoft/Ashton-Tate SQL Server

In January 1988, Chairman and President of Microsoft Bill Gates and Chairman and President of Ashton-Tate Ed Esber announced the Ashton-Tate/Microsoft SQL Server for OS/2. The product was the Sybase database modified by us to run on OS/2. Two of the three largest PC software companies were endorsing Sybase!

The Microsoft relationship had been initiated 18 months earlier with a call from Rob Glasier, then Microsoft's director of marketing for the network group, on behalf of Bill Gates. They had heard good things about our technology and wanted to visit. I took responsibility for figuring out if a relationship with Microsoft could be secured and, if so, how to make it work.

At the time, Microsoft was entering the server market in combination with IBM using OS/2 and needed a server application to sell on OS/2. They were also developing their own database development tool (which eventually evolved into the current Access product) and needed a database server to work with it. From our perspective, we wanted all PC applications to work over a network with our database. We also wanted a low-cost version of our database distributed through PC channels, and we were looking for Microsoft to promote our product. Our own sales force could then promote the upgrade path to larger systems running on Unix and VAX/VMS.

The business decision to work with Microsoft was anything but straightforward. A friend had described a relationship with Microsoft like having a ''techectomy''. He said, ''They will extract the technology and leave the rest of your company behind.'' I thought about two alternative approaches. The first one would be to not give Microsoft access to our source code; this way they would get a marketing benefit but our technology would be completely protected. The second one would include providing them with full source code. In this approach, Microsoft would have a harder time developing their own product without violating our proprietary technology. I decided not to give Microsoft access to our source code. This left Sybase with a large technical support burden, and we would eventually be driven to change our minds and provide the source code under the agreement.

I outlined a plan to our Board of Directors assuming that the relationship with Microsoft would last for five years, thus allowing us to gain a significant marketing advantage and royalty revenue and giving us time to develop additional revenue-generating products.

As I was negotiating our contract with Microsoft, Rob Glasier called me to mention that they had approached Ashton-Tate about co-branding the database and positioning it as the high-end server for the dBase market. Microsoft wanted to be a major database player, and dBase was the leader in the PC-DOS market. Because Ashton-Tate was adding SQL to dBase, they also needed a database server. At the time, Microsoft derived its primary revenue from the DOS operating system, and they viewed the transition from DOS to Windows and OS/2 as their way to move farther into the applications market.

Supporting dBase required making a number of straightforward changes to our software. But supporting OS/2 was much more complicated. When we designed our database, we had assumed that it would run exclusively on 32-bit operating systems and hardware. However, IBM insisted that OS/2 also work on the Intel 286—a 16-bit processor—so we had to follow suit. I could not imagine a customer being happy with the performance of a 286, but we had to support it, and we spent a large amount of engineering time modifying our code. In the end, I suspect there wasn't a single copy sold for the 286.

Revenue Growth

We ended 1987 with $6 million in annual revenue. More importantly, we had just completed a challenging year working with all of our beta sites and getting our database to operate correctly. Our revenue forecast for 1988 was $24 million. At the January sales meeting, I was able to announce to our sales force that Ashton-Tate and Microsoft would be unveiling their joint product, SQL Server, two days later. We were about to define the standard for the PC industry. The year 1988 was off to an amazing start. Among an encouraging series of ''firsts'' in 1988, our first users' group meeting took place in the fall with Bill Gates as our keynote speaker. And in December 1988, we started our European operations in the UK and in France.

In October 1988, Sybase was part of the NeXT computer announcement by Steve Jobs. The previous winter Steve had called me to say that he was looking to bundle a database with his to-be-announced computer system. He wanted to meet with me that Friday, but it was my turn to prepare snacks for the regular Friday employee gathering, so I had to turn him down. Steve said he would come to the gathering and was in the kitchen with me preparing food while we discussed his idea. I paid a visit to NeXT shortly thereafter, where I was shown a demo of the ''black cube'' computer running the Unix operating system with the NeXT-Step user interface. The system had a one-megapixel black and white display and sported a writable CD instead of the standard floppy drive. The target market was universities, but at $6,500, the machine was expensive.

The bundled database turned out to be a great opportunity for Sybase as we received royalties. It became especially popular with Wall Street developers, who would buy the NeXT system and develop applications based on our database.

Throughout 1987, we had focused on establishing a customer base on Wall Street. Wall Street trading applications were all online, needing high performance in a distributed computing environment. In the first quarter of 1988, our first order over $1 million came in from Solomon Brothers as Wall Street firms were starting to pick Sybase as their standard database for online applications.

But Oracle's large sales force was not about to yield a customer to Sybase. They used steep price discounting at the end of the quarter to entice customers to stay with them.

We responded with a series of inexpensive ads. The first one said, ''For a better price on Oracle, call Sybase.'' The text detailed our assumption that Oracle would lower their price if they knew you were talking to Sybase—then again, if you called us you might just discover we had the best product.

Our second ad was more edgy—''The difference between Sybase and Oracle is integrity.'' We went on to explain that our database allowed you to define data integrity directly in the database while Oracle had no such feature. Of course, the reader of the headline might speculate that we were discussing something else altogether. At the time, Oracle's sales force was well known for aggressive end-of-quarter selling. Meanwhile, the image of Sybase as the small, friendly, coming-from-behind company was true to our customers' experience.

Benchmarking

Because we were selling high-performance systems, we needed a way to measure our speed and compare our products to those of our competitors. Originally, we used a standard banking transaction consisting of looking up a customer's balance, debiting one account, crediting another account, and recording the transaction in a general ledger log. Our initial results were nine transactions per second on a relatively inexpensive Sun/Unix system. We would tell people that our performance was so much better than anyone else's for online transactions that you could more than pay for the Sybase software through the savings it would allow in computer hardware.

The growing interest in benchmarking led to the creation of the Transaction Processing Council (TPC) in 1988, with Sybase as one of the founders. The TPC developed a standard way to measure database performance, starting with a banking transaction. Through third-party validation, the benchmarking wars were becoming well established.

OS/2 to NT

While sales continued to boom at Sybase on Sun and VAX/VMS computers, OS/2 was experiencing difficulties in the market. This had little effect on us, however, the divide between IBM and Microsoft was widening. When Microsoft unilaterally decided to make a clean break from IBM, they asked us to adapt SQL Server to run on Windows NT, their stealth operating system.

Microsoft was seeking to dislodge Novell Netware as the standard for PC file servers while also establishing NT as the new standard for PC servers. As it turned out, customers were looking for database servers, and SQL Server was causing some customers to select NT. Although this was not the only reason, it did make SQL Server more strategic to Microsoft.

The combination of low-cost PC hardware running NT and SQL Server was starting to sell well. So, given the growth in the number of Microsoft customers, Sybase could no longer manage the support requirements. We changed the licensing structure to provide the source code to Microsoft so that they could provide full support themselves.

Lotus Development

During 1989, we heard rumors of Lotus Development's teaming up with Oracle in order to combat Microsoft/Sybase. Lotus was betting the ranch on OS/2 and Unix on PCs and supported IBM against Microsoft in the server market. We approached Lotus to demonstrate that Sybase would be a better business partner than Oracle. We were more closely aligned with them in business strategy and technology, and we were not that tied to Microsoft.

Lotus was developing its own database-application-development tools in addition to database connectivity with its spreadsheet Lotus 1-2-3, which was the market leader at the time. We exchanged marketing agreements with Lotus for them to have some marketing rights to the Sybase database on Unix running on PCs. And we agreed to market Lotus's application-development environment. We also agreed to have them purchase 15 percent of our stock.

I called Bill Gates to inform him of the pending announcement—not the easiest of phone calls to make. In September 1989, Sybase and Lotus made the joint announcement of our relationship and investment. Our connections to Microsoft, Apple, Lotus, and Ashton-Tate provided us with a clean sweep of the PC industry. An industry newspaper ran a cartoon of ''Scarlett O'Sybase'' descending a staircase with various PC companies as suitors.

Over the next several years, Microsoft outmarketed their competition and took over the spreadsheet market, PC database market, fileserver market, and PC server market. None of this had any negative effects on Sybase. We achieved another record-sales year in 1989, with revenue more than doubling to over $54 million.

Back in 1984, we had projected $48 million in sales by 1989 with no exact idea of how we would get there. As it turned out, we were one of the fastest-growing software companies in history due to a combination of luck, timing, marketing relationships, and the quality of our products. The market was transitioning from mainframe computing to online client-server computing, and we were smack in the middle of it.

But our problems were just starting …

Layoffs

As we entered 1990, we looked to more than double revenue for the year. We had a small, but growing presence in Europe and a strong foothold in several regions in the US.

In 1990, we made our first and most successful acquisition: D&N Systems. We had tried to build a professional services organization from scratch, but we were not competitive. Customers were struggling to find trained services people as the market was very young. Hiring talent with experience in Sybase products was especially difficult.

I remember seeing an ad in the *New York Times* looking for someone with five years of experience of programming on the Sybase platform, and I realized that even I didn't qualify!

D&N was the largest and best professional services company servicing our customers. It provided a services business and a set of software products that worked with multiple databases. They had about 40 employees at the time and were based in Massachusetts. We changed the name of our new subsidiary to SQL Solutions. It was obvious to us that most of our large customers had multiple databases and would benefit from a company that could integrate data and applications across multiple database environments.

As we progressed further into the year, we had strong revenue growth but no profits. Our chief financial officer, Jeanne Wohlers, had previously told us that our pattern of focusing on market share and revenue growth was deferring the profits that we would need to become a sustainable public company. She made the observation that if we just froze expenses for one quarter while maintaining revenue growth, we could start to build profits. Unfortunately, we did not take Jeanne's advice when it would have been easy to do so, and by mid-year in 1990, it became painfully obvious that we had to cut expenses. In August, we laid off 5 percent of our 800-member staff and started looking to improve our operating efficiency.

We were not alone in dealing with such challenges at that time, as our competitors were going through their own growing pains. Meanwhile, the market demand for database software was strong and kept growing steadily.

Simultaneously, we were witnessing the early shift of customers' preference away from homegrown applications in favor of off-the-shelf products. Oracle was starting to promote their own application software. Meanwhile, application developers tended to pick the Oracle platform first and then would typically have a hard time moving to Sybase because there were no database interface or SQL language standards, and the differences were not trivial.

The rising star in the enterprise-application sector at the time was the German company SAP. They were strong in Europe and were just entering the American market. SAP had standardized on Oracle but wanted to support Microsoft, which meant talking to Sybase. SAP was written for Oracle and rather than changing their product, they felt we needed to change ours. They were right, so we embarked on a long development process. Lastly, multiprocessor computers were gaining some market attention on Unix, but we had deferred our multiprocessor implementation. We started working on multiprocessor support, but it would take us several product releases to do it well.

Sybase extended the distributed data model, which allowed databases to replicate changes made in one database to other databases.

New Products

Customers liked the client-server architecture with a standardized way to access relational data, but not all data was in our database. Our first major new product category, Open Server, was introduced in 1989. It allowed a standard way to make data available on a network without having to first load it into our relational database. The standardization of both the client and server application programming interfaces (APIs) was attractive to enterprises that needed to integrate data from a variety of sources. We developed versions of Open Server that ran on mainframes in order to access legacy data. Our standard interface on the client or PC went by the obscure name of ''dblib.'' (Microsoft created their own network-compatible API called ODBC that eventually became the standard.)

Sybase also extended the distributed data model, which allowed databases to replicate changes made in one database to other databases. This was a joint effort with some of our Wall Street customers who needed this data-replication capability for their highly distributed businesses. Replication Server became a significant new business opportunity for us; it significantly extended our ability to integrate an entire enterprise's data.

Replication Server would read the log of database changes from our database and allow specific changes to be ''replicated'' into other databases. For example, a customer's server in Tokyo might accept online transactions and then replicate the changes in their New York system. The model was based on data being changed in a primary system and then replicated to any number of secondary systems. It was much more reliable than the alternative of requiring the change to occur simultaneously in all systems. Replication Server became our largest source of revenue after our database products.

We closed 1990 with over $100 million in sales and a small profit.

Initial Public Offering

The path to the public stock offering began in late 1990 as we started to generate a profit and began looking for underwriters to list the stock.

At the time, it was typical for a company to pick a lead underwriter and a second underwriter. We had decided to lead with Lehman Brothers when we got a last-minute call from Goldman Sachs. They wanted a shot at being selected and flew their team out to convince us. This was one of the early clues that there would be strong demand for our stock. Because most Wall Street firms were customers of Sybase, we were a well-known brand.

Goldman Sachs won us over. We informed Lehman Brothers of our decision and offered them the second position. Subsequently, we got a call from Hambrecht & Quist who rightfully pointed out that they had supported us from the start. They became the third underwriter.

We had two primary purposes for going public. First, we wanted to provide the venture-capital investors as well as the employees the opportunity to sell their stock over time. Second, we intended to strengthen Sybase financially in order to attract and retain more customers because we knew that customers typically expected to work with a public company.

The ''road show'' has remained the most memorable part of our public offering process. With Mark Hoffman as the lead, I was the one to cover the technology and provide some comic relief, while Jeanne Wohlers covered the financials. We performed in San Francisco, Denver, Dallas, Minneapolis, Boston, and New York. At the end of the trip, the underwriters described the stock as ''frothy,'' meaning that there was more demand for it than we were selling, with investors placing large orders with the knowledge that they would be cut back on allocation. The allocation process was kept a secret from us. It clearly was a set of favors offered by the underwriters to better customers, many of whom, on the first day of trading, would end up ''flipping'' (or selling) the stock and making a 15 percent profit.

Sybase became a public company on 20 August 1991, with an initial sale of over 4 million shares at $13.50 per share, providing a 17 times return for our early shareholders' investment of $0.38 per share. (Prior to the offering, we did a reverse split of one share for every two so the initial price would be $13.50 instead of $6.75. Underwriters like to offer stocks above $10.)

It was obvious to us that the stock was underpriced. The underwriters told us that this was the way it was done in order to generate sufficient initial volume to make the early buyers happy. Never mind the fact that this process amounted in fact to a wealth transfer from Sybase to the first public buyers. Many years later, Bill Hambrecht, then at W.R. Hambrecht, would launch a public auction system for selling initial public offerings that would eliminate the somewhat unsavory practice of how initial public offerings were managed. Based on a Dutch auction system designed by Nobel Prize winning economist William Vickrey, OpenIPO auctions would introduce an unprecedented level of transparency and fairness. Google would famously make use of it in 2004.

Quarterly Results

We were now running a public company. Every 90 days, we needed to beat the financial analysts' expectations on our stock by at least a penny. Any shortfall and the stock price would fall (as we would find out in 1995) and with it the value of our employees' stock options. We developed a disciplined sales process so we could accurately forecast sales and have good visibility into future sales activities. That process worked like clockwork for more than three years. Large enterprise prospects would wait until the end of the quarter to sign deals, knowing that we would need the business to make that quarter's numbers and consequently might offer better pricing. As a result, more and more of our business was occurring in the last month of the quarter, with most of it happening in the last week, if not in the very last, tense day of the quarter. This cycle caused a lot of added tension for management. Our forecasting system designed by the Kappa Group and implemented by our Chief Operations Officer Dave Peterschmidt, however, performed well for us for the first three and a half years.

The database market continued its strong growth through the early 1990s. Sybase was doing exceptionally well in the US market, so we decided to increase our focus on Europe and Asia. Our bankers advised us that the best way to grow overseas was to relocate a senior executive so the corporate culture could be more easily transferred. In 1992, I moved to Paris with my wife and two young sons for six months. It was one of my best experiences at Sybase because I learned firsthand about doing business in Europe and shared the experience of Sybase employees who were operating remotely from the mother ship. I also had the chance to experience daily life in a different culture.

While I was living in Paris, Sybase launched a development project on a new type of database. Object-oriented databases were becoming the hot topic and claims were being made that this new style of database would supplant the relational database products that we and Oracle and everyone else were selling. We invested a lot of time and money into the effort, as did our competitors. New entrants came into the market. Ingres and Informix saw this as an opportunity to try to reclaim some lost ground. But in the end, object-oriented databases became a niche market and did not cause the sort of market transformation that had come with relational and client-server databases.

Meanwhile, two developments in the market compelled Sybase to position itself powerfully on two new fronts: mainframe integration and a development environment for PCs.

Mainframe Integration

Although client-server computing started out in individual departments writing department-specific applications, it also expanded the demand for data integration across departments and back to the legacy mainframe data sources. Sybase addressed this market with our Open Server, Replication Server, and Omni-SQL products that allowed integration across multiple SQL databases. We were not alone. MDI, a Boulder, Colorado, start-up made Sybase-compatible (and in some cases better) products for data integration with the IBM mainframe. After a couple years of cooperation and competition, we purchased the company in January 1994 for approximately $40 million (including an earn-out). In an unusual structure, we assigned our mainframe data integration team to report to the MDI management. The move set the stage for our entry into what we called ''enterprise client-server'' computing. The product line was called System 10.

Powerbuilder

Sybase never developed a state-of-the-art development environment for PCs. We went through different strategies of trying to buy or remarket development tools, but nothing worked out for us. Meanwhile, Powersoft Corporation had started out building a modern manufacturing reporting system (MRP). To do so, they hired Dave Litwack, who developed a Windows development environment for writing the MRP system. The development environment ended up being more valuable than the MRP application, so they marketed it as Powerbuilder and became the largest independent software company for database development tools for the Sybase and Oracle market.

We became increasingly concerned that if Powersoft ever stopped supporting our database, we would be in serious trouble. After a long courtship, we purchased the company for $900 million. At the time it was the largest software transaction in the history of the industry. The deal was announced in November 1994. In addition to Powerbuilder, Powersoft had a highly efficient SQL database system that was appropriate for both single-user and workgroup applications that was named SQL Anywhere. The name reflected the fact that it was small enough to fit into computers of any size, including palm-sized computers and eventually smartphones.

Hitting the Wall: Q1 1995

Our forecast model for the first quarter of 1995 showed us on the low range of making the quarter. After more than three years of beating expectations, we were confident however that we would uphold that trend. After all, Q1 was historically our weakest quarter of the year. It was also the quarter

Sybase would have been a stronger business had we better caught the wave of off-the-shelf applications that emerged in 1990.

when we would close the Powersoft transaction. So when we announced, upon finishing the quarter, that earnings would be positive but significantly below expectations, all hell broke loose. It was the beginning of a brutal period that would last almost four years.

Within 24 hours, the first shareholder lawsuit was filed. For companies in the technology industry to be sued by a shareholder after an earnings shortfall is business as usual. As such, this filing was run of the mill. It was the start of an intense period nevertheless, and the plaintiff's lawyers endeavored for more than a year to prove that we had misled investors. They knew how much liability insurance we carried, and they used that to assess their target damage amount.

Law firms that specialize in shareholder lawsuits usually find a sticky point upon which to build a solid case, or they simply keep up the fight to the point where the company will opt for a settlement as its cheapest option. This is precisely what we ended up doing on advice of counsel. Although we thought we could prevail in court, the cost of winning the suit was likely to be larger than the payment to settle. In addition, there was obviously no guarantee of a favorable outcome in a court case.

The year 1995 marked the end of the high-revenue-growth period for Sybase. This being said, we gradually restored customer confidence and returned to profitability. Over the next couple years, we crossed the $1 billion mark in annual revenue.

Executing a financial turnaround requires different skills and disciplines than launching a brand new enterprise. Consequently, it is common for members of the founding team to be augmented or changed along the way. As could be expected, the board made two management transitions as part of the financial recovery plan. Our founding President Mark Hoffman stepped down and Powersoft CEO Mitchell Kirtzman became the new president and CEO. And by 1997, the board made an additional change and brought in a new management team led by John Chen. John had made a name for himself as CEO of Siemens Nixdorf's Open Enterprise Computing Division, a post he left to become Sybase president and COO. He later became the chairman, CEO, and president of the company until its acquisition by SAP.

John placed a lot of emphasis on mobile computing based on the SQL Anywhere product that came with the Powersoft acquisition. He focused on cash, profitability, and financial discipline, which is what we needed. By 1998, Sybase had grown into a stronger, more profitable company and was building cash. But it had been a difficult four-year process.

Time to Leave

I regularly give guest lectures on entrepreneurship at UC Berkeley and have organized my thoughts into 10 rules. Rule number 10 is: ''Starting a company should not be a life sentence.'' I had originally planned on Sybase being a 10-year project for myself. When the company fell on hard times, however, it became obvious that I could not leave before it was back on track. By 1999, we had recovered from our missteps, earnings were stable, and the company had a new direction to follow. It was time to leave and I was ready. Over my 15 years at Sybase, I had done a bit of everything, from product architecture, financing, and marketing to sales, geographic expansion, strategic planning, and customer relationships. I transferred my remaining responsibilities of customer relationships and strategic planning and was confident I was leaving the company in good hands.

I was the last of the four founders to leave. Jane Doughty left prior to the IPO when she moved out of state. Tom Haggin left after 10 years. After Sybase, Mark Hoffman stuck with his entrepreneurial spirit and joined several subsequent start-up companies. Jane, Tom, and I pursued activities in other areas.

Sybase did well without me, growing its profits, cash, and stock price for 11 more years until it agreed to sell to SAP for $5.8 billion in 2010. It continues to develop innovative products, and 26 years after it started in the back of my house, it continues to serve its extensive customer base well.

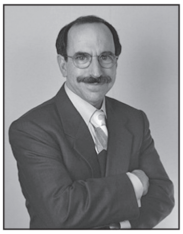
Many Sybase employees honed their entrepreneurial skills and went on to start their own companies. Due to our rapid growth, we offered opportunities for quick advancement and increased responsibilities. We also made our fair share of business and technical mistakes that were learning experiences in their own right.

Looking back, I believe that Sybase would have been a stronger business had we better caught the wave of off-the-shelf applications that emerged in 1990. As more and more customers were abandoning homegrown solutions in exchange for packaged, standard applications, most application providers were writing their applications for Oracle first. By the time we adjusted our product to be more compatible with Oracle, we were experiencing financial difficulties and missed some windows of opportunity. Few applications companies invested in the Sybase platform outside of a couple of special markets such as financial services, one of our earliest and most promising markets from the start.

Very few start-ups get to the age of 26. Even fewer get to redefine an industry. Relational databases already existed, but we brought them to online applications and the client-server architecture. Although today's applications are Web based, most of them continue to get their data from a relational database. The creation of Sybase was set at a special time in the history of computing. It was an exciting, competitive, and rewarding experience.

Reference

1. E.F. Codd, ''A Relational Model of Data for Large Shared Data Banks,'' *Comm. ACM,* vol. 13, no. 6, 1970, pp. 64–69.



Bob Epstein is an entrepreneur and engineer and the cofounder of five companies: Sybase, New Resource Bank, GetActive Software, Colorado Microdisplay, and Britton-Lee. As the manager of the INGRES Project at the University of California, Berkeley, and vice president of development and principal architect for Britton-Lee, his early career was focused on the commercialization of databases and networking. He is currently a cofounder of Environmental Entrepreneurs (E2), chairman of the NRDC Action Fund, and trustee of the Goldman School of Public Policy. Epstein has a PhD in electrical engineering and computer science from the University of California, Berkeley. Contact him at bob@bobepstein.to.