RDBMS Workshop: Oracle

Moderator:
Burton Grad

Recorded: June 12, 2007
Mountain View, California

CHM Reference number: X4069.2007

Table of Contents

RDBMS Workshop: Oracle

Conducted by Software Industry SIG – Oral History Project

Abstract: The representatives from Oracle and from IBM covered the whole range of Oracle history from its founding in 1977 through the 1990s and even some discussion of Oracle as of the date of the meeting. They emphasized how much the company was driven by the vision and style of Larry Ellison from both a marketing and sales and from a technology viewpoint. Oracle adopted SQL to be IBM compatible, but intended to be all things to all people by porting to every viable platform. The various versions of Oracle are traced along with each of their technical advances. Product quality problems were discussed along with how they were solved. The incredible growth of the company (and the relational database industry) was analyzed. The financial crisis in 1991 was covered with comments on the causation and on what was done to save the company. And the session closed with a discussion on Oracle's competitors and why Oracle has been such a successful company.

Participants:

Name	Affiliation
Burton Grad	Moderator
Michael Blasgen	IBM
Peter Capek	Historian, IBM Research
Sharon Codd	IBM
Chris Date	IBM
Thomas Haigh	Historian, Wisconsin University
Jerry Held	Oracle [part of meeting]
Mike Humphries	Oracle
Ken Jacobs	Oracle
Bruce Lindsay	IBM
Jan Phillips	SI SIG, DEC

Burt Grad: The purpose of this workshop is to review the major events in the history of Oracle. What were some of the most significant technical achievements that were made? What were some of the major marketing decision? Let's start at the beginning. What did Larry

Ellison, Bob Miner and others, see as the initial opportunity and how did they get started? I guess we should start with Ken Jacobs. You were at Oracle pretty early.

Developing a Database Product

Ken Jacobs: I was there early. I joined in 1981 as employee #18 and numbers 2 through 17 have left so when Larry leaves, the place is mine. The board doesn't know that though. Actually I know a lot of the initial stories, although I wasn't there at the time. Certainly what had happened was that they were inspired by the IBM paper on SQL, made contact with the System R team at IBM and set out to build an SQL implementation. This was after the founding of what became Oracle as a consulting business. They wanted to become a product business, saw the opportunity to go into database, saw the potential of relational, and the importance of being IBM compatible. The company had some extra funding from a consulting project that they had completed early and under budget. And so they built their first database product. Their background had been in large scale memory devices. Larry Ellison had been working on a terabit memory project for Ampex. And the idea of a large memory was quite different then. But they had worked as well on a CIA project called "Oracle" and they asked if they could use the name for the product and subsequently changed the name of the company.

Mike Humphries: Before that it was called RSI [Relational Systems, Inc.].

Jan Phillips: And RTI [Relational Technology, Inc.] became Ingres.

Jacobs: And RDS became Informix, I think, Relational Data Systems. We were RSI until about 1983 or so. They built the SQL product and provided it to their first customer, the CIA, in 1979, and that was the first real product shipment.

Grad: What language did you use?

Jacobs: Originally it was written in PDP11 assembly language. And it ran on all of the PDP11 operating systems: RSX, IAS and RSTS, I guess.

Grad: Why that particular machine?

Jacobs: I wasn't there at the time. I think it may have been because it was an affordable machine to get. I don't know about the strategic decision making process. There probably were other alternatives at the time.

Humphries: The customer was the CIA and that drove it, right?

Jacobs: I think so but, I wasn't there so I have no first-hand knowledge.

Humphries: It was probably a machine that would fit on a submarine or something because it was the CIA that was behind the initial project. Let me tell you what Bob Miner said. We all heard different things at Oracle because it was that kind of company. Bob Miner told me that they had two projects in front of them that were big projects and were the kind of things they wanted to do. Remember, they were kind of a development company. One of them was writing a new compiler for somebody for the PDP4. The other one was the CIA customer for a relational database product. Bob said, basically, "We flipped a coin." Ken knows Larry better than I do. But I know Larry well enough to know that he wouldn't allow the coin flip to determine the result. Larry already knew what he wanted to do, I'm sure, and the coin may have just happened to end up on the database. But that's what they ended up doing. .

Grad: Bob Miner is another person that contributed much to the relational database industry who died a few years ago.

Jacobs: Ten years ago, I think.

Grad: There's a major endowment that has been given to the Computer History Museum in his memory.

Jacobs: Excellent. .

Grad: So you have the first product, for the CIA. Were there any other customers for that product that you know of?

Jacobs: I was in Washington, so most of the early customers I knew were government.

Grad: What was your role at that time, Ken?

Jacobs: When there are just two people in the office, you get to do everything. I ran the computer room; I did sales, consulting, and training. I had to write a user manual in punishment for inventing a requirement that they implemented in our forms processor. I ran support eventually for the East Coast.

Grad: Was this because your customer was located in Washington?

Jacobs: We had customers all up and down the East Coast.

Grad: This is by 1981

Jacobs: Yes, by 1981 we were a national company. I helped open the Washington office with a fellow from whom I was taking a graduate course in databases. He mentioned he was

going to join RSI all the way across the country. I went up to him one night and said, "If you ever need anybody, give me a call." I ended up meeting Larry and him in a hotel in Washington. And as far as I knew it was two guys and a suitcase. I decided to come out to California and interview the rest of the company and learn what they were doing and what they saw the potential to be. They all said the same thing, and I thought relational database was hot. IBM compatible was smart. They all said that the problem and the challenge was going to be managing the growth. This is when they were 12 people. That was, indeed, the challenge, doubling every year for ten out of the first eleven years.

Early Design and Development Issues

Grad: By 1981 what did they have besides the PDP implementation?

Jacobs: Around 1980, I guess, we ported or ran the PDP version on a VAX in emulation mode. And then it was ultimately, with version 3, rewritten in C which ran natively. So the first two years I was with the company it was essentially a PDP11 program running on a VAX. And it was still constrained by the memory space and it was a two-process architecture. On the PDP11, I think, memory was 64K. It was small. We had to separate the code between multiple processes or at least two processes -- and they communicated. So it was mostly the SQL layer talking to the back end across a wire or across the internal memory

Grad: Now the development work was going on in California.

Jacobs: Yes, it was all done there.

Grad: Bob Miner was in charge of the development work at that time?

Jacobs: Yes. And the three of them decided that Larry was the worst programmer and they sent him out to be the salesman. I'm not sure that's a fair statement but he was certainly the best salesman.

Grad: Who were the other people?

Jacobs: Bob Miner and Ed Oates. Ed retired from Oracle, I guess, almost ten years ago.

Grad: What was Ed's role at that time?

Jacobs: Ed, in the very beginning, was one of the coders, one of the developers.

Grad: He had a technical background?

Jacobs: Technical role, absolutely. But Larry actually designed the data in the Emp and Debt tables very carefully so that we could illustrate things like: "Find me the employee who made more than his manager;" that kind of demonstrated the power of SQL. And so Larry wrote the very first introduction to the SQL book using the power of SQL and demonstrating it via sample queries.

Grad: So he, Larry, understood the marketplace.

Jacobs: Absolutely. But he also understood the technology and still does. I mean, he is very deeply involved. When I first met him, I asked him what does a president of a 12-person company do because I'd never met such a person before. The whole idea of a start-up was very new to me. And he told me he was very engaged with customers but what I came to find out after joining the company was that in the weekly bug meetings that he was engaged in specific technical issues, driving a lot of the development decisions. He was very deeply involved and remained so until the very present.

Early Customers and Other Development Issues

Grad: We're now in 1981. You've joined the company. You have customers. How soon do you have customers up and down the East Coast?

Jacobs: When I joined we already had customers like the Wright-Patterson Air Force Base, which was an early customer, and Fort Meade.

Grad: Were there a lot of military and government customers?

Jacobs: Yes. One of the very first things I did was to sell Oracle to the Food and Drug Administration, the FDA. And as a consequence Oracle came to have a very large role in pharmaceuticals because they wanted to share data with the FDA. So we had both civilian and military customers. We had some commercial customers as well on the East Coast. But, our headquarters for the whole East Coast was in Washington.

Grad: By this point the product's in C, and you're running on VAX?

Jacobs: It was in 1983 that we released version 3 in C. So we had version 2 which, as I mentioned this morning, was essentially the first version. I think there was a version that was never released, version 1. Version 2 was released in 1979 and it was actually a fairly complete implementation of SQL. It had views, it had sub queries, it had outer join, it had connect-which is our hierarchical query mechanism. It had very rudimentary concurrency control and integrity capabilities. It lacked any kind of query optimizer. It was pretty simple. Its SQL statements were atomic at the role level, but it was pretty inconsistent when it came to protecting data. And it wasn't very reliable in terms of stability. But the transition to version 3 was relatively good. It

introduced portability and it began a period of time where we ported to many, many different platforms.

Bruce Lindsay: When was version 3 released?

Jacobs: Version 3 was released in 1983

Porting the Oracle Database System to Other Platforms

Grad: What were some of the platforms you ported to?

Jacobs: Oracle ran on virtually everything. There were many jokes about this. People would say, Oracle would run on anything, and the best platform it ran on was a slide projector. It ran on Pyramid, Sequent, Data General, Apollo -- you just run down the list of everything. And surprisingly, version 4 was ported to the PC under MS DOS and ran in 64K. Version 5 was the first product to use extended memory on the PC and it would switch the processor in and out of different addressing modes. This is a story I was told, I don't know if it was true, but apparently that mode of switching would cause the processor to overheat. So running Oracle on that box could cause it to melt the CPU. But version 4 was also ported to the IBM mainframe and it still runs on that mainframe.

Grad: I gather that the PC Version wasn't terribly successful at that time.

Jacobs: Not at that time, no.

Grad: Did you ever consider porting to any mainframes other than IBM?

Jacobs: We actually did a lot of arrangements with hardware vendors. So I believe in Europe there were ports to certain other platforms, like Siemens, but I'm not sure. I remember very much that we were really the only SQL product in town. And so companies like Burroughs or Siemens were covered.

Humphries: UNIVAC was a port that we did early on.

Grad: So you were porting it to mainframe hardware.

Jacobs: Yes, what we'd often do is get money to give the source to some hardware vendor and they would do the port with our support. I remember personally getting involved with a port to a Harris machine, which was a very strange architecture.

Humphries: It was a military oriented machine out of Florida. I headed the OEM sales business for two years, so I did a lot of those deals.

Grad: When did you join, Mike?

Humphries: I joined in 1984.

Grad: We are starting to get into that time period, so you are allowed to pitch in.

Humphries: I was the 130th employee. What a lot of people don't realize is Oracle started in 1977. By 1984, when I joined, it was at a $12 million point. We had a weird fiscal year. So our fiscal year didn't end until into the next calendar year. But to comment on something that Ken said earlier, when I joined, I got to know Bob Miner really well right off the bat because I was doing the OEM business which meant that we had responsibility for the porting that had to be done. We didn't do it. We taught other people how to do it and we supported them. Miner was responsible for making sure that the classes got taught and they supplied some people to support them. So I got to know Bob really well. Bob told me when I came to work there that Larry still had himself on the team listed as a developer. And he said, "He's the worst developer I've got. He never gets anything done and he never shows up." And so shortly after that he finally got him officially off. It was an ego thing with Larry. He liked being a developer.

Jacobs: Larry was not the first employee, actually. He was still working for Amdahl I think, when the company was founded.

Grad: Miner and Oates were really the first two or was Bob the first one?

Jacobs: I couldn't tell you which of the two was first. The two of them together, I think.

Grad: So the company has been in existence seven years, and it's up to $12 million in sales, and 120 employees. The product now runs on a lot of machines. Technologically, what had been going on during those seven years?

Technical View of the Oracle Database

Jacobs: Well, a couple of other things besides database. I read the manuals for version 2 which Larry had given me prior to my first interview with him, and they had a report writer and they had a product called UFI which was a takeoff on IBM's UFI. It was called UFI, User Friendly Interface. But far from being user friendly it was a command line product. When you did a query it didn't display the headings of the tables, column names. You'd just get the data back. It was pretty rudimentary. I remember my first experience with Oracle. This was either just after I joined the company or during that first interview trip. I sat down and did a query. I said, "Select *,* from emp." The thing blew up. That was one of the very first things I ever did

with it. But when I had read the manual there was a crude report writer with a kind of a markup language. When Larry asked me what I thought of the product I said that it was pretty interesting, pretty impressive. I wasn't too sure about the report writer. It was either the manual or the product that sucked.

Humphries: It was called REPORT.

Jacobs: RPT and PRF. PRF was the markup language, essentially, and RPT ran a loop. And people used the report writer as if it were a procedural language. It would embed updates and stuff in this reporting language.

Humphries: And it was odd, all of our early products were named with three letters.

Jacobs: Yes, that's right. One of the other products we had pretty early was a product called Fast Forms that later became the interactive application facility which was a forms generator. And in some ways it was modeled a little on QBE [Query By Example developed by Moshe Zloof] in that you could do a query by filling in a value in a field and hitting "query." And the feature that I invented that caused me to have to write the manual was a trigger, so that when you put a value in a field and then pressed "delete" there was a pre-delete trigger that had to fire. But the generator, the tool that we used to generate these forms was a question and answer dialog. It was an absolutely horrible user interface. But it became the basis for our philosophy, of producing non-procedural application development tools. That has really been pretty central to our success along the way. Not only in a non-procedural database language but a model based development approach. And that extended to Case tools and our current generation of products for developers as well.

Grad: You raise an interesting issue which we haven't really discussed here. The tools used in building the systems. Was that a specific focus? Did someone work on that?

Jacobs: When you say "building the systems" you mean building the database or building the applications?

Grad: Not the database or the applications themselves but building the programs that would manage the database. You were using C as your language to write the programs?

Jacobs: Yes, but I'm talking about application development tools that our customers would use to build their applications. They're very simple tools, used to build menus, forms, and reports.

Thomas Haigh: On this topic, would Oracle have started from the beginning with libraries so that C programmers could embed calls within the programs.

Jacobs: Absolutely. We had both a call library called the OCI, a very creative name: Oracle Call Interface. And subsequently we implemented a free compiler for modeling. This was like the IBM model of embedding SQL in programming languages. But in addition to that we had, from the very early days, tools designed for non-programmers to use to build applications, simple applications for transaction processing reporting.

Grad: We're into 1984/1985 time period basically.

Jacobs: Yes.

Competitive Technical Issues

Grad: Who were your competitors and what was the relationship?

Jacobs: The first significant competitor we had was Ingres. At that time they had not yet done SQL, they were still doing QUEL and a lot of the debates were about the merits of SQL versus QUEL. And as I said this morning, it didn't really matter which was the better language, IBM compatibility was pretty critical. We had other debates with the folks at Ingres around topics such as how data integrity ought to be enforced. I believe the model in QUEL was that if you did an update statement that violated a constraint it simply didn't change the rows that would have violated the constraint. And silently you would process this subset of rows that satisfied the Where clause. I never thought that was right. We had done it differently, so that if there would have been a row violation, we rejected the statement. So there were some very narrow technical issues. By the time we got to 1985 it was interesting. We had been talking a lot about query optimization. They were ahead, no doubt, in query optimization. We still had a rule-based optimizer in 1985. They introduced Ingres Star which was a distributed database capability where the architecture incorporated a central node that had a catalog of all the tables in the universe. And all queries would go to that central node and become optimized and then routed and data would be assembled.

Bruce Lindsay: Pretty much standard shared nothingness.

Jacobs: Well, it was a kind of shared nothing approach except that Oracle unconventionally took a different approach. In fact, I'd say that's been true about Oracle a lot. And maybe this is boasting, but I think a lot of the things that Oracle did were unconventional, not done the way the book says. And that was true about concurrency control. It was true about this shared nothing, shared disk thing. We should spend some time talking about that because that's a very big part of Oracle's history, parallelism as well. But let me go back to this distributed database question for a second. I remember Larry calling me one weekend to tell me that we had implemented distributed database by creating a forwarding capability. We could send an SQL statement to a remote node and we would parse it and decouple it so if we were joining a local table to a remote table, we would fetch the rows from the remote table and do the

join. While very crude in terms of query optimization, it has subsequently, of course, gotten better.

Lindsay: At that time did it handle the distributed committed deadlock?

Jacobs: The first release didn't support remote updates. So it was read only. But subsequently, maybe it was version 7, which came out in 1992; no, it was before that.

Humphries: Yes, it was before I left. I left in 1989. I was probably showing slides on it.

Jacobs: You probably were. We marketed version 7 for a very long time. I'll talk more about the importance of version 7 in a minute. But in version 7 we finally supported distributed transactions and two-phase commitments.

Humphries: Ken, didn't the kernel get totally rewritten somewhere?

Jacobs: Yes, let me take a minute and give you the whole history so you can put it all in context, because we'll probably pick up different technologies and place them on the Oracle timeline. Version 3 was rewritten in C and released in 1983. It introduced the multi-version concurrency control capability with the four-image file that I mentioned to allow queries to run even while updates were happening. But it still had table-level locking and it was not as reliable as it should have been. Version 4 became a lot more reliable and we completed part of the multi-version concurrency control by really adding read consistency. There were some holes in the implementation of concurrency in the version 3 timeframe where if you did a query and you said, "Insert into emp, select from emp," the query would see the rows that were being inserted and so you'd get an unpredictable number of rows in the result.

Haigh: And how about transaction support?

Lindsay: And the Halloween problem?

Jacobs: Those were there in version 3. So it was table-level locking with transactions, but the concurrency was a little incomplete until version 4 In fact, I remember going to one of these military government installation where they were experiencing difficulties as a result of this Halloween problem. And I said, "Well, that's easy. Just go to version 4, it was fixed in the next release." Where have we heard that before, right, in our industry? But indeed it was. And the guy, I remember saying, "What a hell of a way to release a product," because they were relying on it very much for a critical national security application.

Grad: I gather there were a significant number of problems but people didn't have any other solutions.

Jacobs: In the very early days I think people bought into the vision so thoroughly, they were willing to put up with horrible stories. I ran support so I encountered a lot of that. Very often the right thing for me to do when a customer called was to ask them, "How old is your most recent backup?" Because it was easier them to reenter all the new data than it was to fix the problem. But certainly there were a lot of people who bought into the vision very, very early. They wanted this stuff really badly. I remember someone left a pink slip-- this is before voice messaging -- and it said, "I want to buy Oracle. If you don't call me back I'm going to die." So the demand was absolutely there from the very early days.

Customer Reasons for Choosing Oracle

Grad: I want you to go ahead with the sequence, but was it relational they were buying or was it the application? What was it that was turning them on?

Jacobs: I believe it was the power of SQL, the ad hoc query capabilities. I remember doing the demos. I used to sit down with UFI, the command line interface. We would do queries. I would show people a sub query and their jaws would drop. The power of it was just incredible. So it was largely the ad hoc capability. They certainly weren't going to use it for mission critical transaction processing, and the notion of data warehousing didn't really exist. These were decision support type of systems, simple departmental transaction processing applications. But, it obviously moved pretty rapidly. The demands were more than we could satisfy for many years in terms of reliability, scale and so forth. But we always targeted the very high end. Our ambitions were always very large; for example we wanted to build a database for the airline industry, for airline reservations. Culturally, while we have always sought to go to the high end, as IBM did, we just delivered a product that was actually more successful at the low end.

Lindsay: So I'm seeing a big contrast between what Don Haderle [of IBM] was saying, "We can't ship DB2 until there are no bugs."

Jacobs: You bet. Culturally IBM was for Fortune 500 companies, mission critical applications. And in addition to the lack of an SQL product, that was the other opportunity that IBM left open to us, this other emerging market, departmental users, less mission critical, people with smaller budgets, a lot of other things that gave us a great opportunity.

Lindsay: I don't know if it's relevant from a historical point of view, as to whether the viewpoint was, "Let's get it out there with warts on it or let's get it polished perfectly round and smooth and shiny before we let anybody use it."

Humphries: I've got an interesting story about that. Boeing was a big customer of ours. And Boeing had come in to renegotiate their contract and it was roughly a $25 million contract. So they do the whole meeting and Larry was famous for being scheduled to show up at big

customer meetings and either not showing up at all or showing up really late. And this time he showed up late. And he's talking to these guys and they're chatting with him because all the other stuff that they had on their list was covered. So one of them said, "So, Larry, you've just gotten out version 8," (I think it was 8). How is it?" He said, "What do you mean?" He says, "It's running, and we have released it." And they said, "Well, are there any bugs?" And Larry says, "Well, let's see. I just was in a meeting yesterday. I think the bug database contains 11,000 and something bugs." So there's dead silence in the room. The Boeing guy is really shocked. And then apparently in an angry way he says, "That's unacceptable. If we were to build airplanes that way, I mean just think what would happen if we allowed there to be bugs in the airplanes." Well, Larry, who's a guy to never suffer insults, Larry said, "Oh, yes?" he said, "Well, I just flew back from the East Coast on a 777 in first class and my personalized LED screen for the movies didn't work," and so on. This is when the 777 first was in commercial service. And so he's bitching about Boeing airplanes in a meeting where his guys were trying to close a $25 million contract. So the Boeing guys got really in a huff and had an argument with him. They went away for a few weeks. But then the deal got signed. But that was the truth; it was 11,000 bugs.

Grad: Did they have options at that point in time?

Jacobs: Well, Boeing in particular was a big IBM partner and customer.

Grad: So, by then they did have options on the mainframe.

Jacobs: Yes.

Humphries: Yes, but can you think about what they would have to redo? They had been an Oracle customer at that point in time for probably, what, eight years, ten years.

Jacobs: Yes, I don't know the details of that story, Mike, but they'd been working with IBM very closely for a long time prior to that, I'm sure.

Michael Blasgen: I have two questions. First of all, you mentioned the Halloween problem. Just for the record, the Halloween problem has nothing to do with being frightened or anything being scary. It's just that the problem was discovered on the work day that was Halloween. So we called it that because the problem was discovered on Halloween.

Jacobs: Can you describe what that problem is in laymen's terms?

Blasgen: It sees the insertions that it's making.

Lindsay: Yes, I remember, where it manifested in ours was that there was an index on salary.

Jacobs: You updated salary and it would move the row to a later place in the index.

Lindsay: Yes, the query was: "give everybody a ten percent raise."

Grad: And then you'd give everybody a ten percent raise.

Jacobs: And it was in ascending index and so it kept moving up in index and so you would see that the same guy who had been given a raise, but you then gave him another raise.

Grad: Another ten percent raise, ad infinitum.

Setting the SQL Standard

Blasgen: Can you tell us about Oracle's involvement in SQL standards? How did SQL standards come up? How did the standardization process work in the early days? Who was participating?

Jacobs: I'm very glad you raised that. I think it is a subject we ought to return to in the technical plenary session because I want to ask you, Chris [Date], how was it that IBM donated the SQL spec and whether it was you, in fact, who brought it to the committee?

Chris Date: It was not me. I thought that Phil Shaw took it to the standards committee.

Jacobs: Was that over your dead body?

Date: No, that would not be fair to say. I had some reservations which grew stronger over time.

Jacobs: Well, we should have a whole discussion around that.

Grad: Yes, let's spend a minute here because it's part of the Oracle piece.

Jacobs: The short answer is that before 1985, Jerry Baker was Oracle's representative on that committee. There was a small committee at that time. I don't have much of the history before that time. That's when I joined the SQL committee and I represented Oracle from 1985 to 1993, essentially as the sole oracle participant in the standards process. And I did bring in some things just before the SQL 1986 standard was finished up. That's when we were busy doing some pretty basic things, and there were some really hard things. Again, we'll come back to that in the technology session. I think there were some real contributions made, with positive and negative results, within the SQL committee especially around RI [Relational Integrity]. It was really a hard problem.

Blasgen: When did SQL standards efforts begin?

Jacobs: Well, it goes way back. There was a Codasyl database standards committee. And the same committee pressed the reset button and did SQL.

Grad: Was this the old Codasyl database committee?

Jacobs: Yes, the same committee. Don Deutsch, the current chair, who has been chair since the inception. [added note from Ken Jacobs: Don Deutsche was at NIST (National Institutes of Standards & Technology) in Maryland and then at General Electric, before he moved to Sybase for some years. I then helped him get hired into Oracle, where he has been Vice President, Standards Strategy and Architecture, for about a dozen years.] Prior to it being called the SQL Committee, it was the ANSI X3H2 and it was the database committee way back then.

Grad: Was this the group that Charles Bachman was involved with?

Sharon Codd: No.

Grad: Well, there was a Codasyl committee on database.

Codd: It was the ANSI Committee.

Grad: Was ANSI a successor to Codasyl?

Jacobs: Codasyl was the committee on data systems and languages.

Codd: But only for COBOL. Codasyl was the standardizing body for COBOL.

Grad: There were a whole bunch of Codasyl committees. It wasn't just COBOL. That was just the first.

Haigh: In the big picture ANSI standardizes everything, right. They standardized light bulbs and what have you.

Jacobs: This was the technical committee on databases. And so its first scope was around what we now call network database. And they did develop a spec that I don't think corresponded to any product at the time or after in terms of the network model.

Blasgen: And it was standardized.

Jacobs: It was standardized. But we'll talk more about that later.

Haigh: Okay, so the first SQL ANSI standard is in 1986; is that right?

Jacobs: Yes.

Haigh: Now you said in passing that it did something that was very important.

Jacobs: I think a lot of good work was done in the committee to take Ted Codd's theory and turn it into a specification and that the specification had to do with what is the behavior of a system that implements this language. And in particular when it comes to referential integrity, there were some anomalies that would arise as you try to describe set oriented processing in a specification that inherently had to be an algorithm that the computer was going to process a row at a time. And so problems like cycles and revisiting the same row multiple times arise. And a gentleman named, Bruce Horowitz from AT&T Bell Labs, made very significant contributions to implementing a behavioral spec for referential integrity. Particularly around things like update cascade where it got really gnarly. So that was one of the things that we were working on during the time. And I remember, Chris, you had some real objections to some of the things that the standard was doing in implementing RI which was in the 1989 addendum, I believe, rather than in the 1986 standard.

Date: I'd have to go back and look at my notes from the time, but there are complexities, as you say. But those should have been implementation complexities and they were surfaced somewhat. This was true of the products, too. So certain situations were outlawed because they were regarded as being too hard to implement.

Jacobs: Yes, I don't think it was implemented. We had to find a way of specifying what the correct behavior was and so there was difference between theory, specification and algorithm. It was hard and it was ground breaking. There wasn't any work going on in academia or anywhere else that I was aware of to take Ted's and your work that described a theory, create a specification that could then be turned into an algorithm or, could lead to an algorithm. But during those times it was a very small understandable standard and it rapidly grew.

Grad: Well, but the point is that from Oracle's standpoint having SQL become the standard and basically the IBM SQL become the standard was a significant fact.

Standards and Security as Factors of Oracle's Success

Jacobs: It was very critical. When I joined the company I remember almost being branded on my forehead with compatibility, portability, connectability. It was compatibility with "the standard" which was IBM. And then came the work with the SQL standards committee.

Portability was part of our philosophy from the beginning with writing it in C and porting it to everything that you could plug in. And connectability being networking and client server ultimately and things like that. But let me go back to my history of the releases because I'd like to capture that. Version 3 was rewritten in C; version 4 addressed the concurrency problem and broadened the platforms that we ported to including the PC and the mainframe. Version 5 actually turned out to be a very reliable product and it was released in 1985. It introduced the distributed database capability, and it introduced some new security features. One of the things I did in my early days was work with some of the intelligence agencies to try to understand the military security requirements. I sat down with what was called "the orange book" which was the specification of criteria for evaluating operating systems. And in my own mind I tried to figure out how this would apply to a database. So, working with people in California we developed the audit capability and ultimately laid enough groundwork to get what was called a C2 certification back at that time. C2 was a security level.

Lindsay: Non-discretionary or discretionary?-

Jacobs: It was discretionary. And I got involved a little bit with security more broadly and ultimately I took the specification for SQL roles where you can create a group of privileges and grant them to a user and made that part of the SQL standard.

Lindsay: How close is that to the RBAC [Role Based Access Control]?

Jacobs: It's pretty close.

Lindsay: Yes, because RBAC is actually vague on whether it's network or hierarchical, right?

Jacobs: This was hierarchical in the sense that a role could be granted a role.

Lindsay: No, lattice or hierarchy, I guess, is your choice in RBAC, isn't it?

Jacobs: You know, I don't remember, Bruce. It's been too long. This project was funny. I remember the bell, the modular model and all that other stuff, too, because we worked with the military on a multi-level secure version of Oracle. It's amazing what stuff you remember. When I was thinking about coming here I was recalling my early days of graduate school. Some of the things I learned about-- like the Quine-McCluskey Method -- how many of us have heard that word in the last 20 years. But this multi-level secure version of Oracle was an interesting project. When I was in Washington I worked with the NSA on understanding their requirements for multi-level secure database stores where you have secret, top secret and unclassified data stored in the same database with the proper privileges to either see or modify data. And we ended up getting a sole source procurement from the NSA to build a multi-level database. And I was moving to California just at that time and the folks in Washington received this RFP that

basically I was behind. And they sat on it for six weeks and they brought it out here for me to write the proposal, which I did. And we won the contract, of course. And then they sent me a plaque to thank me for my work on the proposal…and then they sent me the invoice for the plaque. But we did actually build a multi-level secure version of Oracle that handled this problem.

Grad: Which version are we talking about?

Jacobs: That was in version 6. So I stopped my narrative at version 5.

Grad: Yes, that's what I was asking.

Problems with Version 5 and Development of Versions 6 and 7

Jacobs: Version 5 was pretty successful but it had some serious problems. It still had table-level locking. It had no real scalability. You didn't need it with table locking. You couldn't do much anyway. So we set out in about 1986 and made a fundamental decision to rewrite half of the product. We threw away, and literally deleted the directories for the lower half of the database. We kept the SQL layer but re-architected the process model, the storage format, the logging, the locking, the multi-threadedness. And, indeed, we laid a lot of ground work for our shared disk cluster implementation. And this is quite important. In fact, our work on this pre-dates version 6. I was involved in obtaining a piece of hardware from Digital to accelerate the development of our cluster technology. And so we actually had in the version 5 timeframe something we called "parallel server" that allowed two computers to access the same database on shared disk. But because of table-level locking and lack of, what we called "fast commits," we flushed all the buffers to disk on commit. It was very easy for one computer to read what was on disk when the other one had finished writing it and not have any conflicts on updates because it was table-level locking. But it provided some degree of availability and -- for the kinds of applications we were doing which were more read intensive, some scalability. So that was on version 5.

But version 6 truly laid the groundwork for our product for the next millennium, I think, by incorporating three important things. One was SMP scalability, which in 1988 was still an emerging requirement, working with companies like Sequent and Digital, to make sure that as you add in another CPU on a shared memory machine it would scale. Second thing that we did in that release was implement row-level locking. So in addition to the multi-version concurrency control, we had row-level locking without escalation to page or table level. This product eventually was released in 1988 but the work started in 1986, maybe even 1985. And the third thing that we introduced in that release that was pretty pivotal and also enabled this clustering that we now have and we call "real application clusters," was the storing of undo information in the database itself. Typically in a database you have two kinds of logging you have to do. You have to log the committed changes so that you can recover after a failure for changes that have

been made in memory and not written to disk, write-ahead logging. But you also need to record undo information to be able to reverse transactions that have rolled back.

Lindsay: Those are the snapshot pages?

Jacobs: Well, in version 5 we used the before-image file to roll those back. In version 6 we created what we called "roll-back segmenting." A roll-back segment had all this undo information in a chained fashion. And we used it not only to roll back a transaction but also to query the consistent view of data as of the time the query began. So this was pretty fundamental. But the fact that we stored that undo information in the database was very unconventional. Not by the book, not by Jim Gray's classical model, or Graham Rider's. This was a unique thing to do and it enabled us to build this shared disk architecture because now we had the ability to see the undo information from another instance and another computer and provide that consistent query.

Grad: So is it multiple levels of the undo information or a single level?

Jacobs: It's a single level but the undo itself is protected by redo because it's stored in the database. So in a sense the redo log has enough information to reconstruct not only the data but the undo information to roll back that data. And this particular model has allowed us to do a number of things, not only the clustering implementation that we have today, but something we call "flashback." We're now in the 1990s so I don't have to get into that. But the ability to look back in time, an arbitrary amount of time, is a very important technology. But this notion of parallelism and clustering has been something Oracle started back in the mid-1980s with this parallel server on version 5 and subsequently the implementation of version 6. At the same time we were interested in parallelism and in massively parallel computers like the Ncube. And Larry invested in the hardware platform; and it was in the mid-1980s where we started implementing parallel query and starting to take advantage of that parallel hardware. So let me get back to my timeline. In Version 6 we threw away and rewrote half of the product and in so doing addressed the scalability issues because of locking and SMP scalability. But we had serious reliability problems. And it took us a long time to stabilize it. And at the same time we started to have some financial challenges. And so in 1991 the company had a near death experience because of its aggressive accounting,

Version 7

Grad: Were you still in version 6 then?

Jacobs: We were still on version 6 but I have been talking a lot about version 7. Now during that version 6 time, just to tell a little story, we announced version 7 in 1998 by taking the press and analysts on a flight to nowhere on the Concorde. So I got to ride on the Concorde, to have dinner up in the sky with the analysts. It was very much a promotion of the speed and

scalability of Oracle. And this was interesting because at the time Sybase was making a lot of hay about its performance and its locking which was page level and Oracle was still on table level. But their biggest selling point was their stored procedures and triggers and so forth which we didn't have at all in version 6. But what Sybase did not have was SMP scalability and they did not have role-level locking. So we were able to address those things. So version 6 was a pretty important release despite these stability problems. Version 7 is really the release that everybody would acknowledge saved the company's cookies. As I said, in 1991 the company really had a near death experience. We needed to get some funding to meet payroll and sold an interest to a Japanese steel company to get some funding. But we developed version 7 and outdid Sybase in terms of stored procedures, distributed technology, referential integrity, declarative rather than programmatic which Sybase had at the time. And it was a killer product.

Grad: When was version 7 released?

Jacobs: It was released around 1992, that sounds about right. There were a series of version 7s, 7.1, 7.2, 7.3.

Grad: Yes, but the first one was when?

Jacobs: I think 7 was around early 1992.

Lindsay: Then you just invented your own stored procedure language.

Jacobs: And to my regret I never standardized it.

Lindsay: And Sybase already had one and the standards people were already thinking about PSM [Persistent Stored Modules] at that time?

Jacobs: No, no.

Lindsay: They weren't yet?

Jacobs: No, the standards committee was late to consider a stored procedure language and it was, I don't remember when the standards committee actually finally got around to it, but I regret that I didn't have the resources or time to propose our language which was modeled on ADA and ultimately it's not that different from PSM in terms of its robustness with exception handling and so forth. Sybase's language was very crude, more of a macro language than a development language. So it would have been a shame had Sybase's language been standardized.

Marketing and Sales Channel Issues

Grad: I'm going to stop you for a second. You're up to version 7. I'd like to back up. Mike, you joined the company in 1984, you were there until 1989. What was happening in the marketplace during this period of time?

Humphries: Well, I think the most important thing to remember is that the marketplace drove a lot of Oracle's behavior. Larry had set the goal that we were going to double in revenues every year. Once you decide that, it drives all kinds of other behavior. You have to look for new channels. You have to look for new markets. So from the business side, independent of our releases and other things which were making sure that they raced to stay up with this voracious drive to be able to double every year, we were looking at what other ways we could sell. So we put a big European organization in place and in other spots around the world grew pretty quickly. CERN was a big customer.

Jacobs: CERN was a big customer. Yes, and we actually had distributors around the world.

Humphries: That we later acquired.

Jacobs: Yes, when I joined the company we already had distributors in Europe. But over time we acquired them and they mostly became subsidiaries or sales offices.

Lindsay: So you're saying that the marketing and sales force was enhanced.

Jacobs: Sales force, not marketing. Marketing was almost always an afterthought at Oracle. And it really surprises me when people say Oracle's a great marketing machine because by definition, if you think a company has great marketing, they don't. If you think they have great technology then they do have a great marketing machine. Larry's belief has fundamentally always been that marketing isn't critical; he's famous for saying, "If you're not building the product and you're not selling it, tell me what it is you do." Because those are the things that he felt were important. So marketing has never been a focus,

Grad: You're saying marketing in terms of positioning, things of that strategic nature

Jacobs: Yes.

Humphries: Larry was really the official vice president of marketing, because the marketing strategy came from Larry. Well, we had to look for other channels and other markets. So, we built up our ISV channel. We misnamed it as the VAR channel but strictly speaking it was independent software vendors. That's where we had people who wrote applications that would embed Oracle; and that was a really good business. It was impacted in a big way when we

started our own applications business. We had more financial applications than anything else. The next group was manufacturing applications and they (the ISVs) rightly guessed that that would be the next ones that we did. So we had a crisis for a little while on that but after awhile people discovered that Oracle didn't really do a great job on Oracle financials or manufacturing or on other Oracle applications for like the first three renditions, right? How many years did it take before that stuff stabilized?

Jacobs: Quite awhile.

Humphries: So we weren't a threat to our own ISV's very much. We opened other channels as well. We opened the first distribution channel that we'd ever done in the U.S. We had what you'd consider resellers worldwide but we started with companies like Tech Data. We had never done that kind of thing before. We called them a master VAR, but they were really a master ISV. So we explored every possible thing. We started our professional services group in, what, 1986 or the end of 1985?

Jacobs: It might have been 1985. I don't remember the date.

Humphries: And because we were doubling every year-- this was part of the problem. After a few years, people at Oracle said, "You know, if we just had grown it like, 80 percent a year, it would have been manageable." But growing the professional services as fast as we had to, we actually ended up with a very undesirable result in that we had people from our professional services organization that had been hired two weeks before, in a class with customers who they were going to be doing the project for two weeks after that. And we had some really angry customers who realized that the guys that showed up at their door two weeks later had been in the same class.

Jacobs: The other problem with the professional services group was that we were competing with some of our key partners, the big systems integrators. And so a conscious decision was made much later to scale that back and focus it in a much more narrow way. And we've more recently come to realize the channels are really important. So both with respect to software partners and systems integrators they are doing a much better job now. But it's not been one of Oracle's historical strengths.

Grad: One of the interesting analogies is in the hardware area where DEC and the others were selling through VAR's and yet when the VAR was trying to make the sale, they would compete with him to get the hardware for the deal. So it ended up that the VAR said, "To hell with the hardware. I don't care whether I sell it or not and I'll pick whatever platform I want to support."

Jacobs: Well, that was one of Oracle's key strategies, I think, was to be platform agnostic. Whatever it was the customer wanted was fine with us, both in terms of hardware and operating

system. And so we were truly market driven in terms of porting to anything the customer wanted.

Haigh: Right, but not in terms of the applications. The applications have to use the Oracle database to this day, right.

Jacobs: That's correct, except for PeopleSoft and Siebel [both of which were later acquired by Oracle].

Grad: But then you're competing with them in a sense in trying to sell your application.

Jacobs: And there was a reason behind that. I mean, we believed that we could deliver a much better product by having the integration. We could take advantage of unique things that the database provided. If we were using the least common denominator approach being portable across everything, we couldn't leverage the features of the database. And there are a couple of other advantages. One is we had a built in customer. The applications business was a driver for what we did in the database. And we could deliver to customers a stack that worked, was tested and had a common tool set to administer it. And so we've continued that philosophy.

Grad: But this was happening in the 1980's?

Jacobs: 1989.

Humphries: No, it was before 1989. The first applications business was either 1987 or 1988. Jeff Walker had been hired to be CFO. But he was really secretly hired to start the application business and he came out with the financial application, because I remember sitting in a meeting with my ISV's when the announcement was made. And there was this firestorm of anger that I had to put up with when Oracle did that.

Grad: Jerry Held has just joined us in the meeting. What period of time were you there at Oracle?

Jerry Held: I was there in the mid-1990s. I was there in what I would call the period of the end of the independent database.

Grad: That's a later period of time. I want to make sure I haven't missed you in this earlier period of time, then.

Jacobs: No, we haven't gotten there yet.

Held: That was the period of elimination of other independent database companies.

Grad: We will come back to that. We're still in the 1980s, finishing up there. The market's still growing like crazy.

Jacobs: Doubling.

Grad: But yet there's a near death experience.

Jacobs: I'll tell a story. I remember to the day when I did this. October 20th, 1986 there was a cover of *Business Week* that was out and it had a picture of a CEO, I don't know who it was, sitting in front of a set of venetian blinds. And I had seen this picture of Larry in an office with venetian blinds behind him. So I became inspired to create a mock-up of *Business Week* with Larry on it. This was, of course, long before he was ever on a cover of a magazine. And what I did was I titled it "Larry Ellison, King of Relational Software: How He Took on IBM and Won." And across the top, "Relational Technology Files for Chapter 11." And up in the upper left-hand corner, "Cullinet: Can it Avoid a Takeover?" And the other headline was, "The Man Behind the $5 Billion Dynasty." So I think, even in 1986 we were anticipating-- I think it was five billion dollars in five years. It was going from one billion to five billion in five years.

Grad: Mike, when you left in 1989 it was how much?

Humphries: It was not quite $700 million, I think. Does that sound right?

Jacobs: Possibly yes.

Oracle Recruiting Practices

Haigh: I wanted to ask about Oracle's relationship with research. Were you hiring a lot of PhD database people, was there any kind of Oracle research lab set up at any point, were there collaborations with academic researchers?

Jacobs: Oracle's philosophy about this has pretty much been to commercialize academic research or research that was freely available elsewhere. We have hired some excellent PhD's and researchers. We've worked with them but not in as formal a way as other companies have. Franco Patsulo who was mentioned earlier and Dieter Goelich, both of whom came to Oracle, are some names from academia who are just outstanding. But we had a very proprietary view of our technology and didn't engage with universities by giving them the source code; that is where Ingres had an advantage.

Held: But the story about recruiting was, even before I was there, one day the MIT college placement guy said to me, "Who is this company Oracle? They're hiring more of our

best students than any of the big companies." Because Larry's philosophy was always, "Only go to a handful of universities," and it was this restricted list. It was MIT, Cal Tech, Berkeley, Stanford and you could only go to those universities and you could only hire from there.

Jacobs: And Wisconsin.

Held: So it was a very limited list and you had to go get the best people.

Humphries: Also Carnegie-Mellon, I think. And what we did, I remember one of the things Larry had great pride in, is to try to hire the winner of the math contest every year at MIT. Apparently the math contest was like the ultimate. So wasn't Derry Kabcenell the winner of the math contest one of the years?

Jacobs: He might have been. I don't know.

Humphries: Derry Kabcenell, who couldn't come today, was the head of the curling team. So Larry prided himself in harvesting people.

Jacobs: Earlier someone asked about a discrepancy in the use of triggers. In Oracle 6, we dealt with stored procedures. The kernel, the engine, would execute a PL SQL procedure, but it didn't have the ability to store it. So, you would have client side programs that would create these packages of multiple statements with procedure logic around them and submit it to the database as an executable unit. And, so stored procedures came in version 7, but PL SQL was in versions 6. In fact, we used the same capability in our forms product that I was mentioning earlier. So those triggers that I was talking about were written in PL SQL executing on the client side. So, that explains that discrepancy. It's not really a discrepancy.

Grad: Anything else on the 1980s that you think would be significant

Humphries: Well that was my entire span at Oracle, so I can't talk about anything after the 1980s. I can tell you, though, some interesting things. The recruiting went well beyond technical people. There was an expanded list of universities that Larry approved of for other recruiting. So we went almost like a wholesale buyer, like somebody at Cargill buying crops. We'd go into places like Stanford and we'd hire 50 people. And sometimes there weren't enough technical jobs. Some of the technical people went into the available development and programming jobs and the other people pushed coffee carts, were receptionists, until the right opening occurred.

I'll give you a quick example of Larry's philosophy. When I was running the OEM business, I was gone all of the time and I needed somebody to take my phone messages and do stuff. There was a lady who was at the front desk at that point in time, and she wanted to get off the front desk. So, I put in a requisition to move her to be my assistant. Larry disapproved it. So, I

went into Larry, and I said, “Larry, what’s the deal here? You know, this lady really wants to get off the front desk.” And he said, “Well Mike, she’s not very smart.” And I said, “But Larry, you know, I just need somebody to take phone messages and let people know where I am and stuff like that. I don’t need anybody really smart.” And he said, “Mike, listen. Our company is going to be really, really big someday and every year there are going to be jobs created, new positions, and we’re going to have to be able to move people into them. We’re going to have to remember that.” And he said, “Remember this Mike, it’s a lot easier to take somebody smart and teach them how to type, instead of taking somebody who knows how to type and teach them to be smart.” That was his whole philosophy - so we had people that were really great people from Stanford that were majoring in biology and other stuff who were pushing coffee carts, and I used to be embarrassed sometimes in the meetings. We’d be having a meeting with customers and here would be somebody that graduated at the top of their class in biology pushing the coffee cart into the meeting.

Grad: Why was that acceptable to people?

Humphries: Because those people were smart enough to see what Oracle was doing and they were willing to come in and pay the price of being kind of humble for maybe a year.

Grad: So they’d be warehoused for a year.

Humphries: And they were paid.

Jacobs: I mean the mail room clerk would be an MBA. But, I think this really goes to a lot of the culture of Oracle, very smart people, very aggressive type A people, worldwide and all marching to the same tune. I remember getting the whole company in one room. This could not have happened at IBM, of course, where you get the entire company telling the same story because we heard it all at the same time, in the 1980s. I mean I remember a meeting in Boulder Creek, that kind of thing where you get everybody in one room.

Humphries: We went public in 1986. So we had money then. We weren’t just self funding anymore. We had enormous amounts of money to do all kinds of things, and so the pace just got even faster. And when you listen to the rollout of the products and stuff, it sounds one way. When you think about all of the stuff we were doing to double every year, it was kind of a mad house all of the time and for many years, it was crazy.

Grad: And yet, somehow, the pace continued. The products got better.

Jacobs: And the company reinvented itself several times. I mean, not only, did we survive the debacle of the early 1990s, where we had some financial reporting problems, and an excessively aggressive sales force, and things like that. But, we moved from a minicomputer architecture with green screens, kind of interactive terminals, to client server in the mid 1980s,

at the same time as version 5. Sybase was making a lot of noise about client server. We were doing the same, where we separated the application from the back end engine. And, we subsequently moved to an Internet model of computing and now to what we call grid computing.

Dealing with the Competition

Grad: Let me go back, you mentioned Sybase. You first mentioned Ingres as being a primary competitor.

Jacobs: Right.

Grad: You've mentioning Sybase at a later time. You never mentioned Informix.

Jacobs: Well, it didn't come up. But, I think it was pretty much a case that we would focus on one competitor at a time. And we would aim all of our ammunition at that one competitor.

Grad: But the fact that Informix was in the UNIX market, did that make a difference?

Jacobs: It was a company whose technology, I think, we respected most, but it was not a market threat. But, once we started to get interested in UNIX, it became of interest.

Humphries: Yes, UNIX grew slowly. And we were growing 100% a year, so we couldn't throw a lot of chips on UNIX at the beginning.

Grad: And you didn't worry much about the IBM mainframe market?

Jacobs: Remember, SQL/DS didn't come out until 1983. And the DB2 on the mainframe, MVS, when was what?

Grad: Around 1985.

Phillips: 1985 was general availability.

Jacobs: And we were not really ready for that kind of business.

Grad: Were there any other competitors during that time period that were of significance to you?

Humphries: Unify was kind of a pain.

Jacobs: Yes, a pain, in a small way.

Humphries: Unify was an early UNIX provider, also.

Jacobs: I remember one of our first ads was against Dbase though. We had a fighter pilot plane against a biplane.

Humphries: An Ashton-Tate biplane. That was marketing for sure. And the important thing though is what Ken just said about on focusing on one competitor at a time. Larry started, I forget what year it was, it was probably 1985, he started a campaign called shut off the oxygen, aimed solely at Ingres. And it was from the movie Airplane; we talked about this in the meeting this morning, the movie Airplane, where the nun is playing the guitar to some little sick kid and she ends up standing on his oxygen tube and the kid can't breathe; it was based on that, Larry said, "Shut off the oxygen" and we started the campaign on Ingres. And we were focused, not 100% but pretty close, on putting Ingres out of business and that allowed Sybase, at least, from our thinking, to develop. It's like letting weeds in your yard get big enough and now you've got a big problem because they're starting to produce seeds. By the time we realized that Sybase had established itself in the financial community it was too late to squash them. …

Jacobs: We were also late with product that was directly competitive, since it took a while for version 7 to come out.

Oracle in the 1990s

Grad: Let's move into the 1990s.

Held: I think the 1980s were a time of pretty heady growth for everybody. So you picked a competitor, but everybody was growing up. Sybase was growing like crazy. Ingres was still doing pretty well. Oracle was doubling every year. And it was probably okay for everybody. You compete with everybody, but you're all doing well. It was good times for the database industry. But when I came around in the early 1990s, the wheels came off, and Oracle put them back on and brought in some really terrific managers, Jeff Henley and Ray Lane and others. So, the company was coming back and starting to go to another level. And, I think, that period of 1993 to 1997 to 1998 when I was there, it was the period of not just compete with, but eliminate the other competitors that were the independent relational database guys. And Larry's philosophy was very clear: you only compete with one company per year. When I took over the database group, it was just the end of eliminating of Ingres. Ingres was basically dead and bloodied on the floor never to be heard from again, in terms of being a serious competitor. And they've been resurrected as an open source, but not a real threat in any way. And each year, as I went into Larry's office for the planning meeting, it was the question of who is it this year? So, next year, it was Sybase. And you actually get up and give speeches about, "Well Informix is doing some pretty good stuff," to take the pressure off, and every ounce of effort was focused around how do you basically eliminate Sybase? And, it was not good enough for Oracle to win. It was important for somebody else to lose.

Jacobs: Well, fortunately, we had some help from Sybase on that. They made some fundamental mistakes.

Held: Sybase made some big mistakes with their release 10. But, when that mistake happened, it was: how do you put every ounce of product marketing, sales effort and focus on that.

Grad: On how to exploit that.

Held: And Sybase in that year, I can't remember, it was 1996 or something, where they had their fall, which they really never recovered from. And, Informix then, came to the top of the heap and you ask, "What about Informix?" Well, Informix had done some pretty good work in the high end data warehousing space. They found themselves at the top of the heap being the number two player. And, in one year, we kind of turned the focus from Sybase to Informix, and basically they never knew what hit them. It was such a concerted effort. I mean, it was amazing how you could get product management, marketing, and sales focused. We had a thing, "Where in the world is Phil White," because Phil White was their best sales person. If he went into an account, one of our best guys was in right after them to make sure that they didn't win that business. It was such amazingly focused effort. And within 12 months, Informix was basically on the floor.

Jacobs: But, again, Informix made its own mistakes, in terms of restating orders and booking things that weren't legit.

Held: Phil actually did spend a little time in jail.

Haigh: So the interesting thing, that five year period from 1992 to 1997, that was version 7 Oracle all the way through?

Held: There were still remnants of object database companies that were making noise that object databases were going to be good, and so we did the Version 8 release, which technically was not as good as the object databases for pure object database, but, it had enough features. And with the concentrated effort on convincing customers that if you wanted this set of features, version 8 had it; and that was the end of all of the object database companies. So, it was one year after another, one focused effort. And by the end of 1997, when version 8 was out, the only people left standing were IBM, Oracle, and Microsoft.

Held: Well, Sybase was gone.

Lindsay: When did they take Sybase public?

Jacobs: You know, I was talking to Mark [Hoffman] earlier, I don't remember. It was like mid 1980s, 1986 or something.

Lindsay: Mid 1980s. Now, Oracle was growing the top line, doubling every year. What was the rate of growth of your employees?

Jacobs: Doubling.

Held: I was there from 1993 to 1997; it was actually a 4 year period. When, I went there, it was $1.5 billion in revenue. When I left it was $6 billion in just four years. And it was 15,000 employees at the beginning and 45,000 people 4 years later.

Jacobs: Everything doubled.

Peter Capek: Well, that's not doubling each year, that's about 70% a year. But it's still incredible.

Lindsay: And the scale, did all parts of the company grow at the same rate? Because, a little bit of IBM laundry, at one point they have money and they say, "Where is it going to go?" And they have this fire hose and some weeks they point it at marketing and then they point it sales. They don't seem to have any dividers in their hose. And then, they pointed it one year at development and they hired a ton of people. Everybody who knew how to do anything had to become a manager, so they didn't do anything anymore, right. And it was actually, for quite some time, a really negative effect

Quality Problems and Putting Quality Control Processes in Place

Held: One of the things that happened during that period in the product side, was we went from a company that was, I think, scrambling quickly to get products out and keep up with all of the features during that period, to putting a lot of that dollar investment into process and quality. And version 8 was a high quality release that came out, sort of on time with high quality.

Jacobs: We changed the model of product development. I was the program manager for version 8. It was the first time we had an intramural cross product division process for putting a product together. So, not only the base development group, but the porting group, marketing group, support, and sales, were all in on the process of building a product. And so, it was the first, and maybe it's the last time we released a product on seven platforms and eight languages all on the same day on schedule. And it was a reliable product, as well.

Accounting Issues in 1991

Grad: I need to go back for a minute. The 1991 debacle, as I've understood it, it was primarily a financial issue, not a technical issue. Is that correct?

Jacobs: Well, no. It followed a period of bad releases of version 6; they were not stable. So we had a lot of customers who were not very happy with that release. As I said, version 6 was a complete rewrite of the lower half of the engine, the most critical part in terms of the logging and recovery and all of the index stuff and all of that. So, it was unreliable. It was really rather remarkable, when Derry Kabcenell came and ran the development group. You could see his impact, when he joined in introducing stability. And, I think, it was to his regret that version 6 came out with so many problems, but eventually, we did manage to get it under control, but that definitely contributed to the debacle.

Grad: How did it contribute?

Jacobs: Well, I think it gave an opportunity for Sybase and Informix. We have, today, the reputation of being the reliable safe choice for database, but we didn't at that time.

Humphries: And this was building for a long time. A number of us who left at the end of the 1980s, we thought we were all Superman, because we were part of Oracle and we thought we could go anywhere and do the same thing. But, we also felt something uncomfortable because when you grow 100% a year, you have to have tremendous incentives for everybody to be able to do it, stock incentives, as well as compensation. The sales force and most other people were compensated on reaching goals and had huge amounts of money at stake if you exceeded those goals. That's how Larry helped drive 100% growth. So, what happens is when you're doing that and you've got a lot of money at stake, you go right up to the edge, to the line, on the rules. And, then, you look at the money that's available and you start thinking, "Well, maybe it's okay to go over that line." This is why Phil White ended up in jail, I think, because they did the same thing there. You started having very aggressive bookings, and sometimes, in order to get a booking, a sales guy will do something with the paperwork or other things.

Jacobs: Side letters or something.

Humphries: Yes, side letters. This all came out in 1992, so it is not new news. But, it was all building for years. That was my whole point about it, every year I was there, we grew 100%. It just builds that much more. You're sitting on a bigger pile that you now have to double over the next year.

Grad: I wanted to raise three issues, that's what I was trying to get at it. One, you can say, "My products are lousy I can't make sales. Because my products are lousy, I can only make sales by pushing the envelope. Or, I'm lying, straight out."

Humphries: No, you do the middle one, but you don't really say it. At Oracle, the word was commitment; you made a commitment to your number, whatever you're doing. It didn't matter if you were in marketing, or sales or anything, you committed to do something and you just did it. If you made a lot of excuses, it wasn't quite as brutal in some areas, as people would think because Larry's got more of a heart than people make him out to have. But, if you missed your number in sales for a while, or you had some really pathetic excuses you could be gone in a flash. And then the other guys noticed. That's a good example, so they worked that much harder to make sure that they didn't come up short.

Grad: How were your receivables during that period of time?

Jacobs: Well, during the very early times that was one of the issues. At the time of the debacle, we were delivering product to people that they couldn't use.

Grad: They wouldn't pay?

Jacobs: They wouldn't pay. But, when Jeff Henley came, and became CFO, he really cleaned up that stuff. He just completely redid all of that.

Grad: Prior to this period, Jeff Walker was the CFO and he was also in charge of applications.

Jacobs: And he had been in change in marketing, as well. And so, either he was spread too thin or had different priorities.

Why was Oracle so Successful?

Haigh: After Version 7 comes out, you go five years until the next major release. Is that the philosophy that's coming down from Ellison at the top? That it was time to clean the product up, stabilize things, to become the safe, reliable, company?

Jacobs: We have never felt that we wanted to be the safe company, only. I mean we were always innovating, doing things better than anybody else, and innovating faster than anybody else: doing things first, doing things in an unconventional way. As, I said, there are number of things about our fundamental architecture that were unconventional from the locking to the shared nothing architecture to just the way we designed a lot of things. So, I think, it was just a maturation process. We always wanted to build good solid product that supported the high end, and we knew what needed to be done. But we just couldn't do it all as fast as we would have liked.

Grad: You raise an interesting point. What I've been hearing at the other company workshops is that Oracle succeeded by having this incredibly aggressive sales force, reflecting

the personality of the leader, and how his directions were implemented, the kind of compensation, the levels of people he brought in. That Larry Ellison pushed the envelope all of the time and Oracle outsold everyone. They felt that it wasn't technical advantage. What would you say to that?

Jacobs: I'd say that's exaggerated. There are even books that tell one side of the story. And this is not the first time we've heard that kind of comment or that anyone at Oracle heard that kind of comment. I think, people sometimes confused aggressive "go to market strategies" with arrogance, and they're not necessarily the same thing. Yes, we're aggressive, absolutely. But, I think, it was more than just having an aggressive sales force. It was having a product that was first, that was capable of doing things that the other products weren't able to do.

Grad: Yes, but having products first in many cases, has not secured or retained the market. We've seen that in the PC world, over and over again, the first product gets replaced over and over again.

Jacobs: But during the dot.com days, it was very important to be first. So it's not an unimportant consideration.

Grad: The other analogy was in the accounting applications area back in the mainframe period, where we had John Imlay of MSA and we had the McCormick and Dodge people who felt they had a better product than MSA. But John was just a hell of a lot better salesman and sold a hell of a lot more. Until Dun & Bradstreet eventually bought both of them.

Humphries: But see, we had two eras. The era that I was a part of that we talked about, the period of rapid growth, it was like being children that turned out to be very successful. After 1992, Oracle grew up, I would say. Bringing in Ray Lane and Jeff Henley provided different kinds of leadership. The company became a very different place, because I dealt with Oracle while working with other companies and I could see how different it was. So, you've got almost two completely companies.

Grad: What was different?

Jacobs: What was different? It was still as aggressive on the sales side.

Held: Yes, but there's a thing that happened after the wheels came off, and, I think, to Larry's credit, he realized that if he was going to save his company, which he came this close to losing, he'd have to bring in adults to run the business. And if you looked at the team during the years that I was there, when I looked around the executive staff table, it was the most unbelievable management team. If you look at Jeff Henley and Ray Lane and the dozen other people each of whom could have been CEO of a Silicon Valley company, it was an amazingly good management team. So, it went from a hotshot start up for the first 10 years, doubling

every year, kind of living on the edge, to a wakeup call. And then, in the 1990s, it became a very professionally managed, but still very aggressive company. I saw the same thing from the outside. I knew in the early days Ingres was known to have better technology and Oracle had better marketing and sales. But, when I got to Oracle, when I stepped inside and I looked at the people there, it was a phenomenal team of developers. And by that time, it was the largest database team in the industry and growing with great people and people from IBM and Tandem and Sybase.

Database versus Application Products Revenue and Profit

Grad: By that point in time, how much were the relational database revenues, how did that compare to the applications revenues in accounting and manufacturing and financial?

Held: What I tell people, when I was running the database group, I was responsible for 75 percent of the license revenue of the company, and 150% of the profit.

Grad: That's an interesting point, that the application side was not profitable, right?

Held: It was never profitable when I was there.

Grad: But, could one make the argument that the applications sold Oracles' relational systems?

Held: The database business was responsible for every penny of profit through at least the year 2000…At least past the year 2000, there was never a penny made in applications…

Grad: Stay with me, because one of the arguments has been for doing certain kinds of things that it sells the base product. IBM said, "Hey, I'll put in these things because it sells hardware." You can make the same argument on applications, even if it takes a loss, because look, I'll sell the relational system.

Held: I don't think so.

Jacobs: I don't think it ever pulled database sales.

Grad: How about the other way around? Did the relational systems push the applications?

Jacobs: Yes, although selling applications and selling databases are two very different things. We sell to different buyers.

Grad: Did you have different sellers?

Jacobs: Not, in the first three years.

Held: We went back and forth.

Jacobs: We have separate sales forces now, but it's a whole different cycle, a different set of decision makers.

Haigh: So, did anyone ever suggest getting out of the applications business then?

Jacobs: Many people.

Humphries: Even when I was there. We went through two cycles of it when I was there. I think Larry paid for education in the applications business. And, even if he wasn't making money, today he's positioned so he should know how to do it. And he can compete with SAP. And he's bought the others that he didn't want to have to compete with: PeopleSoft and Siebel and the others.

Grad: You raise an interesting theory, and we're going into the 2000s, but the decision to buy PeopleSoft, which is basically an applications only company, if that wasn't a big money maker, and it wasn't driving the sale of the relational systems, what was the logic of that?

Jacobs: It's very simple. It created a strategic footprint in our customers. It gave us a whole stack, a credible stack. And we could now sell at a higher point into the companies, into the board room. And, our large customers wanted to consider us a strategic partner, rather than just a vendor of technology. So, it has, actually, had a big impact on the way our sales force could sell.

Grad: That was the second significant hostile takeover in the entire industry. The first one was Sterling Software of Informatics, in the 1980s.

Jacobs: Of course this was a lot bigger.

Grad: Relative to that time no; that was a $25 million company taking over a $250 million company in 1984. How big was PeopleSoft when it was acquired?

Jacobs: It was $13 billion or something, when we bought them right, so it's a little different.

Held: I think, if you look back at it, the strategy was a very good strategy that the database industry, as big as it was, if you looked into the long term, applications were going to be very important, The investment in the applications was a big investment, and it took a long time to pay off, and it required a number of acquisitions. But, the result is that Oracle is the biggest player in enterprise software.

Haigh: So, did Sybase or any of the other companies ever try to embark on that course?

Held: I don't think so.

Jacobs: I never heard of Sybase doing it. Sybase bought a tools company, PowerSoft. Informix never really did anything like that. But they had some tools from a company that they merged with.

Humphries: Informix made the mistake of buying that company and having dual headquarters for two years.

Jacobs: The company was In Kansas, yes. That was a big mistake. But, they never got into the applications business. And, I guess, the end game here is the stack story. IBM has a stack, except applications. Microsoft has a stack with new applications emerging. SAP has a stack but doesn't have a database. And so Oracle finds itself in a unique position of having an entire stack and now with the addition of Linux, really down to the hardware.

Integration of Applications and Database

Grad: Do Oracle applications run under anybody else's relational system?

Jacobs: Some of our applications, broadly, yes. PeopleSoft and Siebel do, and some of the applications we've acquired. Now, let's just say applications, because there's another part of our business, and again, this is later than the historical period but, the middleware business has become a $1 billion business on its own. We claim that it was the fastest to a billion dollars of any software company in the valley. In four or five years it became a $1 billion business from a standing start. And middleware people have a very different philosophy, actually, about this heterogeneous world and it's beginning to change the culture inside Oracle to interoperate and support non-Oracle databases more and more. And it's what our customers have demanded. So, it actually brings me to talk a little bit about some of the philosophy of our products, as I see it. I think portability was one of the key philosophies. Integration of all kinds of technologies into the core product was very important, whether it was text search or objects or spatial or XML or whatever, all in one engine, rather than have an OLAP engine that was external and a text engine that was external, and, this was very important to our customers. And the same applied to integration and then completeness of breadth of function, while complying with standards. And, I think, that's how our middleware business is driven as well, that we are open to

interoperate with other products that comply with standards. So it's integrated, yet, open is probably the bottom line.

Lindsay: I think that, Jerry made the point that when you target an application to a single database platform, you really have a huge advantage over your competition, customer loyalty to a particular database. You know and I know that some things work better in DB2 and some things work better in Oracle, that you have language features we don't. And we have language features, you don't. And to code an application to the lowest common denominator of the big four database systems requires losing some things.

Jacobs: Right, you sacrifice. You give up things.

Lindsay: You're sacrificing yourself and it gets even worse, if you're trying to really make your application be, what should I say, autonomic, or hands off, lower cost of ownership, whatever it is. Because all of that stuff is really product specific, "How do I start and stop the database?" Well, that's different in every product. And "How do I inquire what the deadlock frequency is?", because a good application is going to watch that, right. So once you get down below the SQL level, which really is where an application satisfies or it doesn't as to whether it will take care of some of the edge conditions, that's no standard there.

Jacobs: It's always been a tension that customers face too. They claim they want portability or they want database independence to be able to switch products, but it rarely happens, because inevitably, they do take advantage of product specific features. They develop skill sets that are unique to a product.

Lindsay: Skill sets are very important.

Jacobs: Very important stuff. So a database, even if it completely complies with a standard which, really, nobody does, is going to be a pretty difficult thing to get rid of.

Grad: Who does Oracle, to date, now consider their primary competitor?

Jacobs: Cleary IBM and Microsoft.

Grad: Those are the two players?

Jacobs: As a company, SAP, of course.

Grad: But in the relational area?

Jacobs: But at the deepest level, I have to say, in front of my friends from IBM, we don't particularly see DB2 on the UNIX and Windows platforms as a competitor. It's primarily us. We've got the dominant share on Linux and Microsoft owns Windows. So, at the database level, it's different. But, clearly SAP at the strategic level is the other boogey man.

Held: If you had to pick only one competitor for the company right now, it's SAP.

Jacobs: We're still a database company. And so, you know, despite all of these other activities, middleware and applications and consulting and training and all of that, fundamentally, Larry is a database guy. I mean database is where we live. So we continue to do things that press the envelope, technically.

Grad: We're going to draw it to a close. It's five o'clock. Thank you very much. That was very informative. I appreciate all of your contributions.