RDBMS Plenary 1: Early Years

Moderator:
Burton Grad

Recorded: June 12, 2007
Mountain View, California

CHM Reference number: X4069.2007

Table of Contents

RDBMS Plenary 1: Early Years

Conducted by Software Industry SIG – Oral History Project

Abstract: This session briefly examined the early database management systems (DBMS) software which preceded the development of relational database management systems (RDBMS) which are the focus of this meeting. The early effort by Charles Bachman in the development of the Integrated Data Store (IDS) is mentioned as well as other early dbms products, including IBM's IMS, Cullinane's IDMS, Cincom's TOTAL, MRI's System 2000, CA's Datacom/DB and Software AG's ADABAS. After a discussion of the pioneering efforts of E.F. Codd at IBM Research, significant relational packages are examined by individuals involved in their development and marketing, including: Ingres, Oracle, Sybase, Informix and IBM products such as System R, SQL/DS and DB2. Further discussion will be found in subsequent sessions.

Participants:

Name	Affiliation
Burt Grad	Moderator
Greg Batti	Ingres
Michael Blasgen	IBM
Marilyn Bohl	IBM, Ingres
Paul Butterworth	Ingres
Don Chamberlin	IBM
Sharon Codd	IBM
Chris Date	IBM
Don Haderle	IBM
Roy Harrington	Informix
Jerry Held	Ingres, Tandem, Oracle
Mark Hoffman	Sybase
Mike Humphries	Oracle
Ken Jacobs	Oracle
Bruce Lindsay	IBM
Robert MacDonald	Informix
Stu Schuster	Sybase
Roger Sippl	Informix
Jim Strickland	IBM
Moshe Zloof	IBM
Peter Capek	Historian, IBM Research
Thomas Haigh	Historian, Univ. of Wisconsin

Doug Jerger	SI SIG, ADAPSO/ITAA
Luanne Johnson	SI SIG, Argonaut Systems
Michael Mahoney	Historian, Princeton University
Jan Phillips	SI SIG, DEC

Burt Grad: It is June 12, 2007 and we are at the Computer History Museum in Mountain View, California. We do appreciate all of you taking time from very busy schedules to come here and join us. As you know we invited others, but some couldn't break away. This particular meeting is sponsored by the Software Industry Special Interest Group (SI-SIG) which is a volunteer group within the Computer History Museum. Luanne Johnson and I are the co-chairs of the SI SIG. The others who have helped to organize this meeting are Michael Blasgen, Sharon Codd, and, and Mike Humphries. I thank them very much for their help. Doug Jerger who is helping out here has been a long-time member of the SI SIG. In addition, we have a few historians who are here to join us and to help solicit the information we're looking for and to maybe use some of it at later points in time.

The purpose of the meeting is to get your recollections of what happened during the phenomenal development of this particular business segment, relational database management systems. We're going to follow through from the beginnings with the mathematical paper by Ted Codd through the research work at UC Berkeley and at IBM. We will focus on the commercial exploitation of Ingres and Oracle and Informix, IBM and Sybase. While there were others, these were the five that we decided were the most significant -- and since we can't cover everything, we cover a segment, put fences around it, and hope we get useful information out of it.

This first session, which will last about an hour and a half or so, is a broad picture introduction. We're going to start with an overview of what happened from the late 1960s, early 1970s, through to the early 1980s. We'll go as far as we can go. We want to set a framework to help us in the individual workshops that we'll be doing until the closing meeting where we'll sort of bring everything together, cover what we left out in the previous meetings and say, "Oh God, I wish I had mentioned that." We'll do that in our final plenary session at 10:30 or 10:45 on Wednesday, June 13th.

The result of the early research work and development work, of course, were very highly marketable products that eventually built one hell of a set of businesses, brought in a lot of money, created more than a few millionaires; it was a very exciting era to be in.

I'll ask you to open your folders please. On the left hand side is a copy of the program. It's a very simple sequence. We have this plenary session. We'll have two workshops later this

morning, one on technology and one on business issues. We'll have lunch. In the afternoon we decided to have separate sessions for each of the companies. We would like to have had a separate session for Ingres and Sybase, but I couldn't get the equipment and the operators to run separate video sessions. I apologize and it's just one of the things we have to live with, so you'll get less attention than you should have. While the people are in the IBM and Informix meetings, the others of you who haven't had tours of the Museum will want do that and they'll meet in the lobby there for the tour. And then, those who won't be in the Ingres, Sybase or Oracle sessions can get the tours at that time. If you choose not to take the tour, that's fine. You can sit in on any other sessions you want to. There's no restriction. You don't have to have an IBM tag to get into the IBM session, for example. There will be a moderator for every session: Luanne, Doug or I will moderate all the sessions. We're asking that the historians talk to each other a little and make sure we have at least one of the historians at each session. We have dinner planned tonight at Shiva's Restaurant in downtown Mountain View. Tomorrow, we will continue with our technology discussions and our business discussions and then we'll meet back together on a plenary basis to summarize and bring up things we may have missed. Also, if we have time tomorrow, we can discuss what happened after 1995.

Also in your folder is information about the Software Industry SIG. Luanne will be talking about that at lunchtime. And on the other side of your folders is information about the Computer History Museum. I want to remind you that all sessions are videoed and recorded. You all signed releases so we can do anything with the material. Actually we will edit them and if anything is inappropriate, we will take care of it. Please speak clearly and remember no side talk. We know that there have been significant differences of opinion often expressed in print or in other forms by various people. But this is a non-confrontational meeting and we expect all of you to be that way. Express your opinions. Express what you believe, but it's not to be to the person. It's always to the subject and to the topic that we're dealing with. Now, I don't know if this is appropriate or not but I know that two of the very significant pioneers in this industry are not with us at this point because they've died: Ted Codd and Jim Gray. I'd like to take just a moment to remember them and to remember their contributions to all that's happened in this industry.

Okay, let's get going. I want to do a very quick tour of the room. Many of you know each other but some do not, so please state your name, your affiliation during this time period and your experience with relational systems.

Introductions

Jim Strickland: I came to the Santa Teresa lab (IBM) in 1978 from the marketing division and worked in various planning aspects of DB2.

Don Chamberlin: I've been with IBM for many years. During the time that we're discussing today I was at the San Jose Research Laboratory. I worked with Ted Codd on the early relational prototypes including System R and the development of the SQL language.

Mike Humphries: I was at Oracle from about $12 million in sales to about $650 million. That's how we measure things at Oracle for progress. I was on the business side, OEM sales, ISV and some of the other parts of it so I'm lost here among the technology people.

Bruce Lindsay: IBM Research. I worked on the tail end of System/R and DB2 for quite a while.

Michael Blasgen: I worked for IBM for a long time, joined the San Jose Research Laboratory in 1974 and worked basically on System R for the next five years.

Moshe Zloof: I worked on a small project called Query by Example at Yorktown Heights. We collaborated with Ted Codd. He gave us lots of input and we built the first relational database management system.

Sharon Codd: I was with IBM in 1970 when I first heard Ted Codd speak at a database symposium at New York University. This was the first public lecture of what is known as his seminal paper, the 1970 paper that was published by the ACM in June. [Codd, E. F., "A relational model of data for large shared data banks*," Communications of the ACM*, 13/6 (June 1970), pp. 377-387] I became his first supporter. As my ex-husband who was sitting next to me often says: "Sharon turned around and at a stage whisper, a New York stage whisper, said: 'This man has solved the problem.'" I became friendly with Ted and later married him.

Stu Schuster: I was first involved in the relational industry as a Ph.D. student when I read Ted's paper and it inspired my thesis topic in geographic databases. This was kind of early in the game of geographic systems. I entered the industry -- the relational database industry -- with Ingres when it was about $1 million company and had my first marketing job as a product marketing manager, worked my way up to vice president of business development there for three years and then was recruited by Mark Hoffman to become vice president of marketing at Sybase. I was there from zero revenues to a $1 billion run rate working with Mark over a nine year period.

Mark Hoffman: I had started in this area with Britton-Lee and came out of Britton-Lee and co-founded Sybase in 1984.

Roger Sippl: I started Informix in 1980 and was the CEO for about ten years, taking it public in 1986.

Roy Harrington: I worked at Informix basically from the beginning on the technical side until the early 1990s.

Chris Date: I was in IBM from 1967 to 1983. After 1983 I left to work for myself. When I was at IBM, I was mostly concerned with the interface between high level languages and database stuff and I read Ted's original paper and said, "This is it." And then I got into some battles. Since I left IBM, I've been teaching and writing in the database area and doing research of my own.

Ken Jacobs: I joined Oracle in 1981 to help establish the Washington, D.C. office and in 26 years have had a lot of different positions in technical, marketing, and product management capacities. I spent seven years representing Oracle on the SQL Standards Committee, which I think makes me unique among the group here, but certainly I have had a number of different roles trying to explain what Larry Ellison really meant sometimes.

Paul Butterworth: I was in the engineering group at Ingres from 1980 to 1990.

Peter Capek: I retired from IBM Research about two years ago. I'm here as an observer and historian, not as a contributor.

Jerry Held: I helped start the Ingres project at Berkeley. I was the first leader of the initial implementation of the Ingres project and I did my Ph.D. with Mike Stonebraker and then went off to Tandem and started the Tandem database group and built the Nonstop SQL product line for many years and then went over to Oracle and ran the Oracle database group for a number of years and have been working on a variety of other things. And most recently, I signed on as chairman of a company called Vertica which is Mike Stonebraker's latest company, and so after 33 years we've reunited.

Don Haderle: I joined IBM in 1967 and did a bunch of development stuff. I'm a technical developer. I was the technical lead for DB2 back in 1978 through its inauguration in 1983, 1984, or 1985 -- depending upon who's counting when it came out -- and then continued leading the development activities on the technical side through the 1980s. And then in the 1990sI was the CTO for IBM on the data management business side.

Marilyn Bohl: I joined IBM in 1969 and worked with Don Haderle and DB2 from late 1978 through 1987. I then worked with Paul Butterworth and Greg Batti in Ingres until '94.

Luanne Johnson: I'm co-chair with Burt of the Software Industry Special Interest Group here at the Museum.

Michael Mahoney: I teach the history of science and technology at Princeton University and I've been working on the history of computing for the last 20-some-odd years, with a special interest in the development of theoretical computer science and mathematical discipline and then the efforts to make software engineering an engineering discipline -- one success story, one story still open.

Thomas Haigh: I'm assistant professor at the University of Wisconsin Milwaukee. I provide historical consulting services on the side. I published a number of articles on the technical and business history of computing. There's one in particular on the history of the database management system concept that some of you might be interested in called a "Veritable Bucket of Facts" and if anyone's interested in that I can give them my card with my URL or email them a copy. [See: Haigh, Thomas, "A Veritable Bucker of Facts: Origins of the Data Base Management System," *SIGMOD Record*, 35/2 (June 2006), pp. 33-49. Available at: *http://tomandmaria.com/tom/Writing/VeritableBucketOfFactsSIGMOD.pdf*]

Grad: Tom [Haigh] and Michael [Mahoney] have been among the historians who have worked with us over the past seven years; Tom has written a number of papers based upon work that's been done in some of these meetings, We have about 15 historians who have worked with us to some extent, and the people at the *Annals of the History of Computing* have been very helpful. The back table has copies of some of the publications that we've put out over the years. There's a book that Luanne wrote based upon the meetings in 2002 on mainframe software and ADAPSO, the trade association. Then there's an *Annals* issue that Luanne and I co-edited that was put out in 2002 that has a number of interesting articles. Paul Ceruzzi and I were co-editors of an issue of *Annals* on PC software: word processing, which came out at the end of last year. We're now doing one now on spreadsheets. It will be out at the end of the summer. You're welcome to take copies of them. [See: *IEEE Annals of the History of Computing,* special issues: 24/1: The Start of the Software Products Industry (Jan-Mar 2002), 28/4: Word Processing for Everyone ((Oct-Dec 2006), 29/3: PC Software: Spreadsheets for Everyone ((Jul-Sep 2007).]

Bob McDonald: Back in 1983 I was the 13th employee of Informix. The day I joined, I was 50 percent of the sales force. That's how small we were back then. I had a number of executive roles at the company, including VP of corporate marketing when we were 4,000 people 13 years later.

Greg Batti: I worked at Ingres from early 1982 and worked in the engineering department through the acquisition by ASK Corporation (in 1990) and left in early 1994 before CA [Computer Associates] took over. It was a good time to get out.

Jan Phillips: I was with Digital Equipment Corporation from June of 1980 to June of 1990 and then retired. I am a member of the Software Industry SIG.

Early Non-Relational Database Systems

Grad: The first question I was going to start with was what was the status on non-relational database systems in the late 1960s and early 1970s? Who would like to tackle that one for me?

Codd: I can tackle that. The status in the late 1960s was as follows: The first I think what you would call a database system was offered by General Electric and it was called IDS. It was designed by Charles Bachman who was here in 2003 when there was a large database meeting. In about 1968, IBM came out with IMS. It was in 1968 that my late husband [Ted Codd] took a look at what was going on in the database world and decided it was really all very *ad hoc* and very unusable.

Grad: Let's hold it a minute there. Who else had anything to do with non-relational databases during the 1960s? Any of you were working in that area?

Held: Actually I was working on the CODASYL databases. There was a period of time where the first standards were trying to emerge. B. F. Goodrich originally developed a product called IDMS and it was a network data model. I was at RCA at the time and we were one of the first customers and it was an attempt to overcome some of the limitations the hierarchical model of IMS. It was a typical product of a standards groups which was a little too much for what was needed. But I think it actually helped the relational world because with the progression from hierarchical databases like IMS to CODASYL, network databases never really caught on big time and it left the door open for relational products.

Grad: I thought that the hierarchical came out of (IBM's) bill of material processor work that was being done. Do you all agree or is that just something of my background?

Strickland: I was in the field at that time and in 1968 IBM was still giving away all the software and so on. In 1969, they had the big reorganization and we started selling software. IMS and some other products were brought from what we called a Type 3 program where they

were developed in cooperation with some customer and the customer for IMS escapes me. I think it was a large aerospace company but it escapes me.

Codd: North American Rockwell.

Strickland: Thank you. So in 1969, we started selling it. It became a very big hardware seller. IBM was still a hardware company at that time and IMS took a lot of resources. It did a lot, didn't do it very stably, but it did a lot and sold an awful lot of hardware. So from 1969 until 1978 when I came out here, I had some sort of ancillary involvement with IMS and watching it grow to be a really powerful system but still very, very buggy.

Grad: The thing I was going to ask you though is there was no mathematical foundation or basis for the hierarchical work to my knowledge. Is that a correct statement?

Date: Well I'd like to comment on that. Jerry used the words network model and it's fashionable to say that the CODASYL stuff was based on a network model. IMS was based on a hierarchical model and so on. The truth is the first products were very close to the middle. There was no real notion of an abstract model at all originally. Ted Codd was the guy who came up with the idea of a model and a very clear separation between logical and physical levels. Retrospectively, various people, including me, decided that we would label things like IMS as being based on a hierarchical model and CODASYL as a network model. But it was after the fact, so the relational thing was the first case where you had an abstract model defined ahead of time before you started writing code.

Grad: Were there any other manufacturers that at that point in time produced either for free or at a price any kind of the database management software to your knowledge?

Strickland: There was a product called TOTAL that was very popular in the field. That was a big competitor for IMS.

Grad: That was Tom Nies.

Early Query Languages

Schuster: And then there was MRI which had this query intensive system which had a very powerful one of the early kind of user-oriented query facilities that gave a little higher notion to a language but it was mainly retrieval with bulk updates.

Date: You mean System 2000?

Codd: The Company was MRI.

Grad: One of the differences that seem to come out here is that when we were doing bill of material processing, we didn't think of it as a base for querying against, yet when they created IMS, IDS and so forth there was the concept you want to ask questions of the database. Is that a difference or not? I'm not hearing anything.

Bohl: You built segment search arguments in IMS. You could ask questions but they were pre-planned questions.

Grad: You had language, you had DL/1.

Bohl: Right.

Grad: Wasn't that a query language in a sense?

Bohl: No.

Date: A program is a program.

Grad: You wouldn't treat that the same. How about with IDS with Bachman; was there any kind of query language capability there?

Codd: There was a programming language and, in fact, Bachman wrote a stirring paper on the programmer as a navigator. [Bachman, Charles W, "The Programmer as Navigator," *Communications of the ACM*, 16/11 (November 1973), pp. 653-658]

Grad: But then you're saying with MRI there was clearly a query language involved.

Schuster: Right.

Grad: And what about Cincom?

Sippl: Well by the mid-1970s I'd never worked with TOTAL but Hewlett-Packard had a knock off of TOTAL called Image that did have a query language with it in the mid-1970s. But the big query language products were focused on Nomad.

Schuster: Nomad and Focus, yeah.

Sippl: And Information Builders played a major role even though I don't know that they had an API (application program interface) but they did have a lot of users storing data, using Focus and doing queries and report writing.

Grad: Focus comes out of RAMIS basically. Gerry Cohen worked at Mathematica where he helped to develop RAMIS.

Codd: There was another database product and that was Datacom/DB and that was a tabular type of a system.

Grad: When was that?

Codd: That also was in the late 1960s or early 1970s.

Johnson: They were down in Texas.

Codd: No, they were in Princeton. And, in fact, Marty somebody was--

Grad: Marty Goetz?

Codd: Marty Goetz.

Grad: He's later. He doesn't get Datacom/DB until the mid-1970s.

Codd: Okay.

Grad: ADR buys that later on.

Johnson: The Company that started Datacom/DB was a database company [It was called CIM which was bought by Insyte which later sold the product to ADR].

Grad: Luanne is leading a corporate business project in which we've been studying the history of some of the database management companies -- not the relationals particularly. We had a mixture of those and so we have quite a bit of information in the database on that. I want to move on if I can. If you think of others that we should know about that's a pretty good list. There was a lot of stuff going on. I was on the IBM Unbundling Task Force in early 1969 and we announced the first 17 IBM program products and priced them, including IMS and CICS which was one of mine and that was quite an exciting time but that sort of legitimatized the selling of

software products. There was some software being sold before, as you know, but it was very small potatoes.

Ken Jacobs: Burt, I don't know much about it but there was Model 204 which was a very interesting product because it used indexing technology [this was a product of the Computer Corporation of America].

Grad: There were a lot of different things being tried. Chris points out these were solving problems without a structure or a concept or an abstraction behind them. I believe that is an accurate statement.

Sippl: Well I think Chris is right. The models were based on disc sector size and what the word size of the processor was.

Grad: But they were solving problems, weren't they Roger?

Sippl: They were solving problems but, for example, software AG's ADABAS's big claim to fame was that it had variable length records and variable length fields and how it stored variable length records into sectors of the disc drive. So if you went to a sales presentation for ADABAS that would be half of their sales presentation -- to differentiate themselves from the others. So it wasn't a logical abstract mathematical model. When they talked about models they talked about the efficiency of how they stored and retrieved information.

Grad: John Maguire would sell very well whatever he had to sell for Software AG. He was quite a salesman, still is. Okay, let's move ahead to question number two. What led to Ted Codd's work? What led him to do the paper? I guess that's the question. Chris, were you there?

Ted Codd's Initial Work Error! Bookmark not defined.

Date: I came on the scene a little late. Ted started thinking about this problem in 1968 as Sharon mentioned a moment ago. His first paper was not the 1970 paper. [Codd, E. F., "A relational model of data for large shared data banks," *Communications of the ACM*, 13/6 (June 1970), pp. 377-387]. It was 1969. He wrote an IBM research report on August 19, 1969 RJ599. [Codd, E. F., "Derivability, Redundancy, and consistency of Relations Stored in Large Data Banks," *IBM Research Report RJ599* (August 19th, 1969]. So the paper in Communications of the ACM that most people think of as the seminal paper really was not the seminal paper. It was the one the year before. There are some interesting technical differences between the two which maybe we'll get to later.

Grad: In the technical session, yes.

Date: But I think the point was, is as Sharon mentioned, Ted looked at what was going on out there and saw what I like the call “higgledy-piggledy” systems. There was no abstraction; there was no mathematics. It was just write code and hope it works. And so having a mathematical background himself, and realizing that the fundamental purpose of a database is to store facts -- and facts are basically what logicians call propositions, he came up with a very elegant way of encoding propositions and the rest is history.

Grad: Was there any particular project within IBM or elsewhere that had triggered this interest or anything that got him to work on this to your knowledge?

Date: Not to my knowledge. Sharon, do you know?

Codd: You mean other projects that he was working on?-

Grad: Anything that he had previously done that would have led to this particular piece of work?

Codd: I heard him speak the first couple of times and this was early in the 1970s. I remember he got up at one meeting and said, “Look, you have to take a look at IBM business and you have to plan for the future. The only way that IBM is going to be able to sell more hardware” -- because certainly at that time it was a hardware company and the software was an afterthought -- “The only way IBM’s going to sell more is if we can get the average person onto a system to ask questions of the information that’s already stored there.” And he said, “In order for that to happen, you need something that is fairly simple but is bolstered up by something that is mathematical that you can essentially bet the farm on.”

Grad: Let me ask a question. Did any of you here either work with Ted prior to that time? [silence] None of you worked with him prior to this 1970 or were aware of his work prior to then either?

Codd: Well he was working by himself up to that point.

Grad: Well I’m asking the question for the record. So in other words, it (his research) was not specific; however he did have a business interest -- saying we need software to sell hardware and people to be able to use it.

Codd: Yes.

Blasgen: I don't know the whole history of San Jose research (IBM) before Building 28; Don Chamberlain probably knows. There was a significant change in the beginning of 1974 or the end of 1973 when basically all the management was replaced in that organization

Codd: Yes.

Blasgen: San Jose makes disc drives. At some point, disc drives represented something like 35 percent of IBM's revenues and so it was fairly natural for the small research organization. The research division at that point consisted of three laboratories: the large one, the Thomas J. Watson Research Laboratory in Westchester County, NY; a much smaller one in San Jose associated with the San Jose Disc Drive Design Manufacturing Facility; and an even smaller one outside of Zurich. The San Jose one, I guess through the initiative of Mike Senko but I don't know, had an explicit focus on information and data management and stuff. They thought, well they're going to store the information on these magnetic media. What are the models? What is that all about? And so I think that while Ted certainly, as far as I've always heard sort of worked alone, I think it was within the official blessing of the system.

Codd: No, it wasn't and, in fact, Mike Saranga's project was looking to essentially build on IMS. Ted was very much opposed to that and went to the director of the research lab and demanded that his ideas be heard and he asked to be switched to another group.

Grad: There's timing though.

Codd: I understand.

Grad: Wouldn't that have been after his paper?

Codd: No, absolutely not. This is why he worked by himself. He saw things differently than Mike Saranga's group and as a result he demanded that he be allowed to work in another group and develop his ideas and, in fact, he did.

Grad: But the context that is being pointed out from Mike Blasgen though, is that it was within the general context of the San Jose Research Facility's broad view.

Blasgen: I wasn't there. I was hoping that somebody might be able to--

Grad: Moshe, when did you join the Yorktown Research Center? When did you go there?

Zloof: At the end of 1972.

Codd: I'd like to get this into the record, okay? When Ted developed relational database (theory) he was working for Glenn Bacon in IBM Research. Glenn Bacon had a hardware group. Ted worked by himself but reported to Glenn.

Influence of Codd's Paper

Grad: Okay, thank you. Let's move ahead now. The next question is: "What was the reaction to the paper?" I've heard a number of you already mention that you read the paper and it motivated you quite a bit. What was the reaction to that paper and by whom and what did it lead to directly? That was my next question. Who wants to start on that? Did you read it?

Date: I read it when it came out.

Grad: Oh, before I forget, how long had Ted been at the San Jose research lab at that point?

Codd: Just about a year. He had just gotten there.

Grad: Where had he come from?

Codd: From Poughkeepsie.

Grad: So he had been at IBM before that?

Codd: Oh, yes, and, in fact, it was before Building 28 opened and the research lab was in trailers on the IBM plant site.

Grad: Now the paper is published. People start to read it. Chris.

Date: I was working for IBM in England at the time at the home of what was then the PL/I language and I was given the job of figuring out what PL/I should do about this new thing called databases. IMS, of course, was the IBM product at the time. I installed IMS and I played with it for a while. CODASYL was coming along. I read their specifications and tried writing programs on paper and so on. Ted's paper was published in June of 1970 and I read that. Now I'm also a mathematician, at least I was a few centuries ago. This was the first time in my computer career where I saw that my mathematics might have some relevance to computing

and to me it was obvious that what Ted was doing was the right way to go. And I wrote to him and said, "This is great. I want to incorporate your ideas into PL/I." With a colleague we did designs and language extensions to PL/I. Well there's a lot more I could say about that.

Grad: That was the start.

Date: That was the start for me.

Grad: Which others saw it? Bruce did you see the paper?

Lindsay: Is there anybody here that was at the famous meeting in Ann Arbor where Ted Codd and Charlie Bachman--

Date: Debated?

Lindsay: I think it was where they threw lightning bolts at each other!

Schuster: I was there; I was there for that.

Codd: What year was that, 1973?

Schuster: I don't remember exactly.

Date: May of 1974. I was supposed to have been there but I had visa problems getting out of England at the time.

I was supposed to support him and a guy called Kevin Whitney [deceased] versus Charlie Bachman, Ed Sibley and a guy named Jim Lucking in England. And Ted and I worked very hard ahead of time to prepare our position. We wrote, in fact, two research papers that still exist. I was supposed to present one but couldn't. Dionysios Tsichritzis [of the University of Toronto] stood in for me. Ted, of course, presented the other paper. It looked like Charlie and his team had not really done as much preparation work as we had. The proceedings still exist; they're something of a collector's item. The proceedings make it pretty clear who won the debate. There was a report on it shortly afterwards by Bob Ashenhurst about the great debate. Since I was not there, sadly I can't say how it went down but my impression was that some people believed that Ted was right and very few people believed that Charlie was right. That was my assessment. I could be wrong in that.

Schuster: It was a strong academic discussion.

Grad: Were you there?

Schuster: Yes, I was at the meeting. I was assistant professor of computer science at the University of Toronto at the time and almost all my work during that research period was on relational database. And the way I'd characterize it is that in the Ted Codd camp it was logic and reason; in the Bachman camp it was emotion and "by the way we have products." And clearly because there were so many strong researchers and academics there, there was a natural tendency toward the logic and reason side of the argument. So from everybody's feeling, both emotionally and intellectually, Ted had made the point and the team had made their point strongly.

Grad: What happened between 1970 and 1974? Who else saw the paper in that period of time and what effect did it have?

Schuster: Well, as I mentioned, I was a grad student.

Grad: When did you read the paper the first time?

Schuster: I read it when it first came out in the ACM and I was looking around for something to do that had some practice in computer science that would be tangible and that wasn't just an abstract or yet another language -- so it dramatically influenced me that you could put this mathematics and science to a computing problem and then you could expand from there. The question always was how would you build a system to do this? That was often the big question but the impact on me was that it shaped my entire career. It actually influenced my thesis from that point on.

Grad: Right away?

Schuster: Right away. It was like this is what I wanted to do for the rest of my life.

Grad: Don.

IBM's Reaction to Codd's Ideas

Chamberlin: I was influenced by Ted's paper also and I think a lot of people at IBM were. I was working at the Watson Research Center in Yorktown Heights at the time and I think a

number of people in IBM research felt that the best response to Ted's paper would be to try and build an industrial strength prototype to prove whether these ideas were practical or not. And for that purpose, a number of small projects were collected together from various places within IBM and consolidated at the San Jose Research Laboratory in a project that came to be known as System R.

Grad: What time period are you talking about, Don?

Chamberlin: System R began in 1973 and at that time it was Ted Codd, Jim Gray and Irv Traiger who were working mostly together. There were some people from Mike Senko's group which was called the DIAM project. Some of those people joined System R. There were some people from Yorktown. That included me and Ray Boyce, Frank King and Mike Blasgen who came from Yorktown to California to join the project and Raymond Lorie joined the project from Cambridge Scientific Center at IBM. So all these people came from disparate backgrounds and different points of view and came together in San Jose to try and basically create a working proof of Ted's concept.

Grad: Do you know what happened, where in IBM the decision was made to put research resources in that particular direction?

Chamberlin: I think the decision was made at research headquarters in Yorktown because resources were gathered from several labs and I think that required a headquarters level decision.

Grad: Was Ralph Gomory there at that time?

Chamberlin: Yes, Ralph was the director of the research center.

Grad: Would this have been a decision he would have been involved in? It sounds like it.

Chamberlin: It sounds like it but I have no personal knowledge of it.

Grad: Was Joel Birnbaum there at that time also?

Chamberlin: Joel was at Yorktown running the computer science department.

Grad: So it was not Joel's project though at any point?

Chamberlin: No, it was not Joel's project.

Grad: Interesting. Anybody else want to comment on this topic? That's very interesting. So there's a direct lead in within IBM, from Codd's paper, to discussions over some period of your work, and then a major piece of work being done at IBM research to pursue separately a relational database system implementation.

Chamberlin: The manager who was placed in charge of this amalgamation was Frank King. And Frank King was one of the Yorktown people who moved from Yorktown to San Jose to start this project.

Blasgen: Frank's boss also was from Yorktown, Leonard Liu was sent out so these people were all more or less hired at the same time (within I think roughly within a year of each other). The Yorktown laboratory was reformed because Herb Shorr had been brought back and Phil Dauber and John Cocke, and eventually Fran Allen had been brought back from the West Coast to take over computer science at Yorktown. That would have been 1968 or 1969 and they must have gotten authority to do fairly massive hiring because they hired a lot of people. I mean I can give you a list of the superstars that they were able to hire, including Moshe, Peter Capek, and Don Chamberlin.

Grad: Help me with one more thing before going down the IBM path and the UC Berkeley path. I was going to ask if you could talk to me about that what was being done with the Codd relational concept.

Non-IBM Projects Error! Bookmark not defined.

Held: This is exactly the same time. In fact, in 1973, when this massive talent was being assembled at IBM, there were also little projects being done elsewhere I think. I don't know how many universities but just like Stu said, it was a very interesting paper at our university because it gave you a mathematical model to want to do research around. I don't know how many universities -- but certainly Harvard, MIT, Toronto, Illinois -- all had a little project going to see if they could do something with this concept. At Berkeley we were just absolutely lucky because we ended up doing the same thing. It was Stonebreaker, Gene Wong and myself who did the original work and started the original group, but a number of things happened that made it a little different. One was we picked up this new operating system that came out of AT&T. We were the first ones to get the first copies of UNIX and we decided to build on top of UNIX. I had come from an engineering background and decided I wanted to build something real instead of a toy. Mike was a good promoter and all the stars came together, so we had a little project that was building something fairly real. I recall during that

next year Ted Codd would shuttle between San Jose -- watching how the System R team was doing -- and our group and it was kind of a race. Our three Ph.D. students, two Master's students and two undergrads against the mighty IBM -- but we sort of kept pace. Of course, the luck of being on UNIX -- within a year -- had spread to 200 universities and Ingres became a pretty widely used early relational database. I would say it was a lot of good work but a lot of good fortune too.

Grad: That's very interesting how that occurred. Chris, do you have a comment on that before we move ahead?

Date: I have a comment but not on the Ingres stuff. I wanted to go back and say something about the IBM track. I don't think people know how hard Ted himself fought to get these projects moving. I have some notes I made here. In September Ted organized a thing he called the prototypes symposium at Yorktown. I don't know if Don was there. And there were various prototypes within IBM that were demonstrated and discussed and so on. And then in January 1973, Ted proposed a multi-division project within IBM to build this whole architectural area. Ted was a mathematician and he labeled the components in Greek letters, gamma, gamma-one and so on. Gamma zero which he'd already written about in 1972 essentially I think became the RSS. Mike [Blasgen], you can probably set me right here. But Irving [Traiger] and Ted worked on gamma zero for some time and really just reading the paper and reading about the RSS they looked very, very close. So Ted was actively lobbying to get the project going in IBM.

Grad: It's interesting though that this was all public information. IBM had published everything, right? And apparently Ted was comfortable going back and forth to UC Berkeley and that was, in effect, competition.

Held: It wasn't competition for him. He wanted his idea out there and he didn't care who did it.

Grad: The competition with IBM in a sense.

Schuster: Yes, but it was research. It was all research at the time.

Codd: Well there was a lot of opposition to Ted's ideas in IBM development at that time. In fact, there was total opposition from IBM development and IBM marketing.

Grad: When you say IBM development, who do you mean?

Codd: IBM development is the product divisions where the technologies are built into products.

Grad: Remember though that IMS did not belong to a product division at IBM. It belonged to the data processing division.

Codd: But at that time after program products.

Grad: No, I'm telling you the facts. I have them.

Codd: I work with them.

Sippl: It is remarkable that IBM was publishing all these papers.

Grad: But that was typical research, IBM research was published. That was one of the ways you rewarded the people that were there.

Codd: Ted did go to the attorneys to see if they could file some patents but, at that point IBM was not interested in software patents.

Sippl: They weren't granting very many back then.

Grad: They weren't granting any to speak of. You had to do it on a hardware-fake if you wanted to get a patent at that point in time.

Sippl: Yes.

Grad: Because Marty Goetz, for example, got a patent in the 1960s but that was on a fake. He built a machine.

Codd: A sort.

Grad: Okay. Again, interesting pattern of how that developed and the two major parallel paths. Is that a correct statement incidentally?

Sippl: Well I was always fascinated by that. I never knew there was collaboration in visiting at least between Santa Theresa and UC Berkley.

Held: I wouldn't say there was any collaboration between the groups. Ted was not part. I would say he was more above the whole thing going back and forth looking at the work. There was real competition. We would go to all the conferences. And they would write papers and we would write papers and they got single variable *queries* going and we got them going. And then we had *joins* going and they got them going.

Chamberlin: Once in a while there would be a summer student from Berkeley who would come to IBM. Didn't Bruce do that in the early days?

Lindsay: Yes but I wasn't involved with the project.

Haigh: Actually I have a question on this.

Grad: Go ahead, Tom.

Haigh: The development guys at Honeywell claim to have been the first people to ship a relational database management system as an actual product. Does anyone know anything about that?

Date: They did have something called Mac AIMS. I never heard there was a product. There was a prototype. I've always regarded this as one of those strange situations like in astronomy that two people discover a planet at the same time or when Leibniz and Newton invented the calculus at the same time. These guys I think were working quite independently of Ted. They built a prototype and they wrote about it in the 6th ACM SICFIDET Proceedings [Special Interest Committee on File Definition and Translation] and then that was the end of it; they never followed through. They never developed their ideas. They never really wrote up the ideas about abstraction. They just thought relations were neat. So it is true that independently another system was built but it didn't go anywhere. I'm certain it never became a product.

Codd: Since we're collecting history, I would like to add one more historical fact and I know you want to shut me up but I have to say this. There were three prototypes and the third was Query By Example (QBE) that Moshe did at Yorktown Heights. And, in fact, of the three Ted thought that QBE was the best, since it supported much more of the relational model than any of the others. Also, QBE became a product for IBM long before SQL/DS or DB2.

Grad: Moshe, when did you begin developing QBE; when did you start working on it?

Zloof: We started in 1973. Again we were very impressed by Codd's paper. My background was also in math actually from Berkeley and I was very impressed by the logical

aspect of it. I came up with the idea of QBE and we started implementing it at Yorktown Heights. We had a small group and finally it became a product, but it was what they call at IBM called IUP [User Installed Product], which is really something developed in-house and we had about 150 customers at that time in 1975. However, it was killed in time because we couldn't support it. It had to be supported by our group and the development divisions, of course, took the System R concept and implemented it as DB2. But we had at one point 150 customers and again it is very important to stress the fact that IBM started with hardware and then they had systems software and then applications, right? We were really in the application business in a way and that was kind of the last thing they were worried about at that time.

SQL and Other Query Languages

Grad: Again I'm trying to get some background on a range of subjects. We have more time in the individual sessions to explore these in more depth: the languages, the query languages, SQL and QUEL. Did you call it QUEL? And SQL was originally called Sequel and then became SQL when they found the name was blocked for some reason. Where did SQL get developed and why?

Chamberlin: SQL was the user interface of the System R project and we were trying to do several things. The first thing was that we were trying to couch Ted's access primitives in a way that was more accessible to people that didn't have a mathematical background. We wanted to try and take these concepts and represent them in some way using familiar English keywords so that people who weren't mathematicians could use it. The intention was that they should be able to understand fairly easily at an intuitive level what a query meant just by looking at it. And also we wanted to make the queries easier to type on keyboards so we didn't want to have universal quantifiers and unknown things like that that were hard to type. So that was one motivation. The second motivation was that we wanted to study in more depth the data manipulation issues of *insert, delete, and update*. These had been treated only in a fairly rudimentary way by some of Ted's original papers and in relational algebra, for example, there was the use of operators like *set union* and *set difference* to do what we would now call *insert* and *delete* and we felt that once again that there were better ways to address the data manipulation problem. We wanted to address that in a uniform way with the query, so you could put a sub-query inside of an insert, for example, and the syntax would all fit together. We also wanted to extend that syntax not only to data manipulation but to view definition, which is really very similar to the query problem and to the definition of assertions and triggers and other kinds of administrative processes that had traditionally been treated quite separately from query. We wanted to sort of unify these things.

Grad: Did you have any models that you looked at or used when you started working on SQL? Was there some previous work that you had seen or done that said: "Can I use that as a base for doing this?"

Chamberlin: Well there was an earlier language called SQUARE. SQUARE was an acronym for Specifying Queries As Relational Expressions. It was based directly on Ted's work and, in fact, it was an outcome of the prototype symposium in Yorktown that Ted organized. I went to that symposium and it was a conversion experience for me. I had been working on CODASYL before that and I realized from Ted's presentations at that symposium that queries that took pages to represent in CODASYL could be one-liners in the relational approach and I thought that was beautiful. So Ray Boyce and I were members of the System R project. We were sort of in the language end of the business and we were trying to work on building a unified syntax for query, for data manipulation, for view definition, assertions and triggers, and we also wanted to think about some kinds of queries that were possible but in our personal judgment were kind of awkward to do in the existing languages that we'd heard of. And a lot of these had to do with grouping. We thought that queries like "*find the average salaries of employees by their department*" was logical. You don't need to be a rocket scientist to understand a query like that. We thought that it should be really easy to write that kind of grouping query. Some of them were more complex, where you filtered the rows before you formed the groups and then you filtered the groups after you formed them; the groups were formed by multiple grouping expressions and multiple aggregations were computed for each group, like the average salary and the maximum salary. We felt that all those things should be combined together into relatively simple query blocks and so that was one of the original query blocks.

Grad: Go further into this, Don.

Chamberlin: Okay. But could I just wrap it up quickly with one more point? The last thing that Ray and I were trying to accomplish was to build a language that could be used not only for queries but for transaction processing as well. And that required it to be interfaced to a programming language like C or PL/1 and required the addition of some concepts like cursors and updating for a cursor. And it required the development of some technology like pre-compilation and binding of parameters to pre-compiled plans and so on. And so all these things together formed the impetus for the development of Sequel, or later, SQL. The initial publication of the language was at this same meeting at Ann Arbor, the SIGFIDET meeting, where the famous debate was held between Ed Codd and Charlie Bachman.

Grad: Okay, QUEL. My question is similar. How was QUEL developed? Who is the source on that? Are you familiar with QUEL, Jerry?

Held: Actually, the driving force behind it was Gene Wong, who was our sort of mathematician, and his initial work was structured heavily after Codd's work. And it was fairly cryptic actually when it was first developed. There's an interesting story about that, sort of the opposite of lots of research that went into really thinking exactly every piece of the syntax. We were publishing the definitive paper on *Ingres*-- I don't remember what publication, but the last proof was on Friday and we had to have it submitted on Monday. We looked at it and I said, "You know, this is just too mathematical coming from an engineering background." I said, "Can I work on this over the weekend?" And we completely rewrote the language to more of an English-like language over a weekend, published the paper, and that was the final design of QUEL.

Grad: When was this, Jerry?

Held: This was in early 1974 probably. But much of it was based on Codd's work and Gene Wong should get most of the credit for the original developmental language. The syntax of it changed dramatically in a two-day period which came out to be QUEL, which was in a lot of ways similar to SQL. I think the biggest difference we had was how you approached *joins*, especially complicated *joins*. We thought we had a simpler way of looking at complex *joins*, but in the end it was a classic case of a large player who really had a dominant position in a market and was able to establish this as a defacto standard.

Grad: Who was that dominant player?

Chamberlin: It was Oracle.

Grad: I was going to say it was interesting how the languages got started; where was the beginning. Chris?

Date: I have a comment. We talked about Codd's 1970 paper, but in 1971 and 1972 he wrote three more papers that were incredibly influential. There was one on further normalization, there was one on relational completeness, and there's one on a language he called Data Sub-language ALPHA, which was a realization in concrete syntactic form of his relational calculus. DSL/ALPHA was never implemented as far as I know, but I know from Gene [Wong] it was very influential on the design of QUEL.

Held: Yes.

Date: The path to Ted's Alpha work.

Held: QUEL was basically a derivative of Alpha.

Grad: Before I forget, were there any other significant query languages developed based upon Ted's models?

Sippl: Well, Informix's origins were more based in query language and the report-writing aspects of database management, and there were a lot of query languages at the time. Now if you didn't have a relational database management system to put a query language on top of, it wasn't going to be a relational query language. But I had not found any of these papers when I started designing the original query language at Informix. I was just frustrated by report writers that sat on top of non-relational databases. The term didn't exist as far as I knew at the time I was doing this work in the late 1970s. All I knew was that if you laid out the data in these tables -- if you laid them out on your dining room table as index cards -- you should be able to go pick up one card and say: "Oh, that's Customer Number 12" and go find his orders in the order stack because they all have “Customer Number 12” in them. So I just felt that if there was data in common, you should be able to -- what I later learned was called a *join* -- join them together. And there were a number of other query languages that were starting to do that, so the original query language products that Don said were popular at the time, *Focus* and *Nomad*, were starting to realize that they could use their index forms to relate one table to another.

Grad: This is something I was hoping we'd talk a little bit about. My memory goes back to *Mark IV*, for example, as a report writer -- and there were a number of other report writers that were produced during the 1960s – so you had ways of getting information from them. I don't remember if we called any of them formal query languages. Then we had the 4GLs, with *Ramis* and *Focus* and *Nomad* and there were a lot of others -- their database was hidden inside. You couldn't get at the database except through their tools that they provided. So that's what I was trying to understand, whether the concepts, the kind of query languages done here, evolved explicitly from the models that Ted had done or whether they were a separate piece of work to create generalized query models. I've asked the question poorly, I'm afraid.

Sippl: Well, my observation of it as a student trying to learn the whole thing and trying to figure out what to do to build a commercial product and sell it was that it was a little of both. I mean, there was this abstract model that Ted came up with and, from my looking back on it, you've got understand that when he started this work in 1969, I was in junior high school. I had no idea.

<Laughter>

But in the late 1970s, I decided I was going to try to turn a buck in the computer software business and so I was very eager to figure out what the chain of development of all the database systems were. And by the mid-1970s, these query languages were very popular for people to look at data and databases. It was considered a liberating technology -- which end users could actually run their own queries without having to go get a custom COBOL program written. So to have a database management system without a query language would have been very awkward and it would not have become a popular database management system.

Grad: Did IMS have a query language by then? Anybody know?

Schuster: I think there were report writers that had a query capability that ran on top.

Lindsay: In a sense I would say that DL/1 was a query specification.

Codd: No. It's a programming language.

Lindsay: Perhaps the most fundamental and major features of both QUEL and SQL was that they were *ad hoc* query languages. You didn't need to shut the system down and tell it about your query and then start it up like you had to do with DL/1 and many CODASYL systems. Perhaps, the more important thing is that these languages included -- from the beginning – a data definition, and they took the attitude that you don't have to shut the system down to create a new table or to find an index. Of course, Ted didn't like the idea of defining indexes at all. But the fact that the data definition language was sort of at the same level as the query language and consciously meant to be *ad hoc* -- to provide some kind of program that could do it on demand, interactively -- in my mind is perhaps the thing that led most directly to the success of these systems as commercial products. I almost have a feeling that yes, query by relationship, great idea, but what really pushed it into commercial use was the ability to manipulate the system without shutting down and recompiling.

Grad: So you could do it on an *ad hoc* basis, Bruce?

Lindsay: Right.

Codd: Yeah.

Jacobs: I was an early user like Roger; I remember Datatrieve from Digital. And that was, I think, my first introduction to any kind of data management system. But one that I found particularly inspiring personally was System 1022 from a company called Software House, I think. It was very much a flat file, table-type structure and instead of a *join* it had what they

called the map operation and you could find those rows in one table that corresponded to another. Very efficient, very easy to use.

Grad: When was that, Ken?

Jacobs: That would have been -- I was at a timesharing company -- in the late 1970s.

Grad: Which one?

Jacobs: I was at a company called First Data that was later bought by ADP. And there I was, a user of System 1022, which later became 1032 on the DEC-10, I think. And somehow I think it became part of CompuServe or something like that. But it was a very interesting product. We used it to build a lot of government applications. And it did have the properties that I think Bruce was talking about, creating tables without shutting down and being able to empower end users to do queries.

Humphries: So you asked a question about when it dates back to. I was in the timesharing industry also. I was with Tymshare during the 1970s and I think 1022 dates back at least to 1975. But one historical footnote I wanted to mention is, at Tymshare we wrote most of our own stuff for different platforms, but we started acquiring other people's products or licensing other people's products, including 1022, which I think was from Boston. I think that was the firm you were trying to think about.

Johnson: It was Systemhouse, wasn't it?

Jacobs: Yeah, that's it. Systemhouse.

Humphries: But starting in 1974, somebody at Tymshare had been influenced by relational papers because a product called Magnum was starting to be developed in 1974. It was offered in 1976 and it was a relational product with both a query language as well as an interface at least with FORTRAN, because I had customers who wrote applications and did transactions. There was a customer of mine that had an order system and a billing system written with it. And it was written for the DEC PDP-10s but for our operating system. We ran our own operating system for the DECs. And it was never ported, to my knowledge, to any other platform except for DEC systems. But it was a little-known relational product that I wanted to mention. [According to Ann Hardy, the tech manager of Magnum was Dale Jordan with assistance from Dick Ouellette].

Grad: That's interesting. That's a good add in. We've got 10 more minutes, and I have about seven more questions.

<Laughter>

History of Ingres

Grad: I'm going to pick just two more to try and cover. But again, these will get picked up in the workshops that we're doing later on, so this is not your only shot at this kind of stuff. Let me bring in some of the other companies a little bit just to see what their connections were. *Ingres* was started at UC Berkeley and then became a separate company at some point in time, a separate product or a company name at some point?

Held: Several years later; I was not part of that.

Butterworth: That was in 1980.

Grad: Oh, it's that late. So it's not during this earlier time period?

Butterworth: No.

Grad: Okay. So *Ingres* continued to be done under the UC Berkeley aegis over that period of time, up to 1980?

Held: Right. Bob Epstein came after me to run the project. And then there were several releases of it because of UNIX; I think UNIX was a big driver.

Grad: Was it a commercial product? Were you selling it for UC Berkeley or giving it away?

Held: No, it was freely available and so as UNIX spread like wildfire, *Ingres* was sort of piggybacking.

Lindsay: And universities could get source code and use it as a research tool.

Held: Yeah, so *lots* of work was done around it.

Schuster: I think I came to Ingres after you, Paul; weren't you one of their first programmers?

Butterworth: Right.

Schuster: It was about a year into the product, and the story I remember was that a version of it was specifically put into the public domain. That was the version that was then taken by Mike Stonebraker and Gary Morgenthaler and others, you know -- the founding group -- and brought into Ingres to create the basis for the first product.

Johnson: My understanding was that they used some of the venture capital money they had to convert the Ingres product over to DEC, because they thought that would be the commercial marketplace.

Blasgen: You mean a different operating system?

Johnson: Yes.

Butterworth: Yes, that's true.

Held: Not UNIX.

Johnson: Not UNIX, yes. They felt that they needed to be on a different operating system.

Blasgen: Just to make a comment in support of Jerry's comment -- which I think is exactly right -- UNIX was a major player here because AT&T, which owned UNIX, chose to just make UNIX available for free to anybody, very liberally, especially universities. And remember, that was when AT&T was a monopoly and they were under various controls. And so it did. UNIX spread like wildfire through universities and research laboratories. In some ways, IBM Research was at a disadvantage because we weren't, if you will, in the mainstream of research because all the work that everybody else did was based on UNIX.

Grad: Interesting comment. So *Ingres* is still in UC Berkeley during this period of time. How'd you get into it, Roger? What triggered your interest?

Sippl: I was a student at Berkeley; I was an undergraduate and I was totally unaware there was any database research going on there whatsoever.

<Laughter>

Starting Informix

Grad: So what triggered you to start Informix?

Sippl: Well, I had to get a job because I had a fifth year of college and all my scholarships ran out, so I switched majors to computer science late because I had cancer. Because of that I wasn't going to go to medical school and I switched to computer science. I finally got a job. I built a database application at Bechtel Engineering to keep track of parts at nuclear power plants, which later was in the movie "China Syndrome" apparently. But that database was home brewed, and I was just terrified because the Nuclear Regulatory Agency could shut down every construction project that Bechtel had going worldwide unless we fixed it. So we had to rip it out and put in a database management system and it was under the Hewlett Packard operating system. We put in Image, and it worked great. And I thought, "Well, this packaged software concept just sounds swell to me. And the database thing is great, but this one is horrible and the query language is horrible and it's very restrictive and you can't *join* things arbitrarily together. But there's this KSAM product that has B-trees and that seems to be a much more flexible indexing system. So, if someone just built a database management system on top of B-trees, you'd be able to do this cross-referencing thing a lot more easily, and you could build much better query languages and report writers." So I kind of came to the same conclusion Ted did, ironically, but through a very pragmatic approach. And I was not a mathematician. I'm amazed that Ted, without a customer driving it, developed that model because that was the most flexible model. And I was forced by pragmatism – as a guy with 58 report programs to produce and no report writing language to produce them.

Grad: So you had started your company at that point?

Sippl: No. Three years later I started it. I went to work for Cromemco for a couple of years, and there I was in charge of their primordial single-file database product. But I designed a multi-table system; what I discovered was I was designing a relational database program. I just want to say that it wasn't Ted Codd's seminal paper that inspired me and motivated me. It was Chris Date's interpretation of those papers because I couldn't read Codd's papers. I tried, but I only found them through the bibliography in Chris' books. So it was Chris' books that made it so that I could understand the foundations of the relational model. And I've got to tell you, since this is a history process here, Roy Harrington and I were at Cromemco building some of the first microcomputer software on the planet and when I researched this -- the history of the relational data model -- I discovered: "Someone's been here before my ideas of relating this stuff together. Jeez, these people have been writing papers about this for a long time and there's even some products and stuff." And I looked back and I saw this 10-year history and I

thought, "What are these doing? Ten whole years and still we don't have any report writers worth a darn? This is like, you know, molasses."

Oracle

Grad: Let me bring in another player, Oracle.

Jacobs: There was a direct correlation to the work at IBM and Oracle. I thought it was 1976 when the SQL II paper was published. I wasn't aware of the 1974 paper. But the folklore, anyway, is that Larry was inspired by that paper. He had had a small consulting business called Software Development Labs, SDL, and they had a contract with the government and had, unlike many contractors, managed to complete the work under budget and ahead of time and so they decided to build a product based in part on that 1976 SQL paper. Larry and the founders, Ed Oates and Bob Miner-- the late Bob Miner, a wonderful guy-- created a relational database for the government, for the CIA, in fact, and they were the first customer. And the first product, of course, was Version 2 because Larry didn't think anybody would buy a Version 1 product!

<Laughter>

Jacobs: But indeed, we talked earlier about the openness of IBM to publish all its work and whether or not there was competition between UC Berkeley and Santa Teresa. There's another apocryphal story-- and I'm sure, Don, you know whether this is true-- that Larry wanted to be so IBM compatible that he wanted to know the error codes from SQL/DS so we could actually replicate that. And I think that was really the key to Oracle's success in the early days; they chose to be IBM compatible as well as to do SQL. And that was really a brainstorm.

Lindsay: And be on UNIX.

Jacobs: Before VAX it was RSTS and IAS and RSX on the PDP-11. The VAX came along really in big force in 1980 or so. And we did a port of the PDP-11 version to the VAX.

Schuster: I remember very distinctly the importance that the VAX played in this market.

Jacobs: Absolutely.

Schuster: Because you couldn't compete with IBM on IBM platforms. That just wasn't going to work at that point in time. And then UNIX was too early. It wasn't really commercially available and it was really only being used by the engineering and scientific community and

research community, so it wasn't really commercially viable at that point. So it was really VAX and VMS that really spurred the early database work.

Jacobs: And of course, there were a lot of other propriety operating systems, you know, including those from Data General.

Grad: Once you were down on the minis away from the mainframe--

Schuster: Then you had an open field.

Grad: -- A whole open field.

Jacobs: And it was quite appropriate for the departmental deployments of these relational databases as well.

Grad: Right. Last comment and then we close for the break.

Sippl: The fact that Oracle didn't succeed on UNIX early on because it was in the VMS market and the war was between Ingres and Oracle -- and that was isolated to the VMS marketplace -- made it possible for Informix to exist, grow and thrive because we were a smaller footprint database management system and we ran on .super microcomputers.

Grad: One of the fun things that we're going to pick up in the business sessions is the choice of markets, how much they were intentional, how much they were accidental, how much you observed them or you just fell into them.

Blasgen: The paper you referred to in 1976 usually is Codd's paper in ACM Transaction on Database Systems (TODS) [v.1 n.2, pp. 97-137, June 1976], and an unusual aspect of that -- which I don't know the whole story of -- is that a reviewer came back and said, "Would you please include a more formal definition of SQL." My question is, who was that reviewer and what was his or her motivation?

Chamberlain: I don't know the answer.

Blasgen: Because it's unusual for a paper of that sort to have a BNF [Backus-Naur Form] included as an appendix. And I wonder, did Larry want that?

<Laughter>

Sippl: But I do recall that that was helpful.

Mahoney: May I just make just an historical comment? I'd like to remark here tha I'm a member of the ACM History Committee and one of the things we're discussing is archiving ACM's records. And one of the issues is can we get at the reviewer's comments, and are people willing to put reviewer's comments on record. Our feeling is that reviewing is done under confidentiality. 100 years from now, it won't make any difference and it would be nice to be able to answer a question like that one -- the role of reviewers in the shaping of products, which is an unwritten history because we don't have the names of the reviewers. We don't have that. We also don't have the letter, which I know exists, where Larry wrote to Don, I think, isn't that right, and asked for the error code.

Chamberlain: It was a phone call. I took the phone call.

Jacobs: And your answer was, "No," wasn't it?

Chamberlin: Our lawyers would not allow us to release our error codes.

<Laughter>

Grad: I think we may have found out some of the errors in, like 1023, I think, Ken.

<Laughter>

We'll come back to it. I have one more thing to do. There's a gentleman here who's joined us a little later, Doron Swade. Would you stand up and say, "Hi?"

Doron Swade: Hi.

<Laughter>

Grad: Sit down. That's it. Doron is going to work for the Museum for the next three years in putting together the timeline exhibit, which will be a featured exhibit here at the museum. Doron wrote a wonderful book about the history of Mr. Babbage, and he was the one responsible for building the Babbage Engine at the British Science Museum. He's also a reasonable historian and he writes exciting detective novels.

Swade: Did you say "reasonable?"

Grad: Reasonable, not great.

<Laughter>

So we welcome Doron. And if he can, he'll join us whenever he can during the next couple of days.

We'll take a 15-minute break. Fifteen minutes. At the end of it -- please, before you get up yet -- we would like those of you interested in the business area to discuss that aspect upstairs to the Blue Room with Luanne. Those interested in the technical aspects, many of the companies here, you're stuck with me for another hour and a half or so.