SRC Technical Note

1997 – 018

August 20, 1997

The 1995 SQL Reunion: People, Projects, and Politics

Edited by Paul McJones

Systems Research Center
130 Lytton Avenue
Palo Alto, CA 94301

http://www.research.digital.com/SRC/

Copyright © 1995, 1997 by Paul McJones, Roger Bamford, Mike Blasgen, Don Chamberlin, Josephine Cheng, Jean-Jacques Daudenarde, Shel Finkelstein, Jim Gray, Bob Jolls, Bruce Lindsay, Raymond Lorie, Jim Mehl, Roger Miller, C. Mohan, John Nauman, Mike Pong, Tom Price, Franco Putzolu, Mario Schkolnick, Bob Selinger, Pat Selinger, Don Slutz, Irv Traiger, Brad Wade, and Bob Yost. You may copy this document in whole or in part without payment of fee provided that you acknowledge the authors and include this notice.

The 1995 SQL Reunion: People, Projects, and Politics

Edited by Paul McJones

Abstract

A reunion of people who worked on System R and its derivatives, including SQL/DS, DB2, and R*, was held at Asilomar on May 29, 1995. This is an edited transcript of the day's discussions, incorporating changes provided by the speakers. It provides an informal but first-hand account of the birth of SQL, the history of System R, and the origins of a number of other relational systems inside and outside IBM.

Contents

Preface 4
Remembrances 5
Ray Boyce 5
Vera Watson 5
Morton Astrahan 6
The Birth of SQL 8
Prehistory 8
System R 12
Spreading the word 25
IBM products 31
SQL/DS 31
Shoot-out at the OK Corral 34
System R winds down 37
VS/QUERY (QMF) 38
DB2 39
System R folks leave the fold 45
Esvel 45
Tandem 46
Oracle 48
And the rest 52
Intergalactic dataspeak: SQL standard, Open SQL, ODBC, DRDA 52
Teradata; Ingres family: Relational Technology, Britton-Lee, Sybase, Microsoft 55
Informix 57
Index 58

Preface

In the spring of 1994, Mike Blasgen decided there should be a twentieth anniversary commemoration of the System R project. By the fall of 1994, Mike had recruited Jim Gray to handle local arrangements and proposed to:

> "Invite those people who worked for IBM on the early relational systems. This would roughly be from the early 70s to the early 80s: a decade of progress. Include not only the original System R team but also people who worked in IBM on "derivatives" like R*, SQL/DS, and DB2."

The event was held at Asilomar in Pacific Grove, California, on May 28-30, 1995, following SIGMOD '95 in nearby San Jose. In addition to catching up with long-lost friends, walking on the beach, and enjoying a magical private reception at the Monterey Aquarium, we spent Monday, May 29, in a meeting room recounting the events of two decades ago.

I recorded and transcribed the day's talks, asked the speakers to make any appropriate revisions, and performed the final editing. The result is an informal but first-hand oral account of the birth of SQL, of the project – System R – from which it sprang, and of some of the other relational database systems.

I would like to thank the speakers for reviewing this document and providing revisions. I would like to thank Ken Beckman, Bob Taylor, and Digital Equipment Corporation for audio recording advice and equipment loan. Finally, I'd like to thank Mike Blasgen and Jim Gray for making the reunion happen.

Paul McJones
December 10, 1995

Preface to Second Edition

The first edition of this document was self-published on the World-Wide Web. In the hope of making it easier to find and to cite, I am reissuing it as an SRC Technical Note.

In the first edition I included a number of bibliographic references as starting points for readers interested in learning more about the topics discussed during the reunion. In this edition, I've made a few corrections and a number of additions to these references. There are several references that may be of general interest to readers of this document: an overview[1] of the database field at the time of the System R project, and a technical retrospective on System R[2].

I'd like to thank Cynthia Hibbard for her editorial help with this edition.

Paul McJones
August 20, 1995

---

[1] Special Issue: Data-Base Management Systems. *ACM Computing Surveys 8*, 1 (March 1976).

[2] D.D. Chamberlin, M.A. Astrahan, M.W. Blasgen, J.N. Gray, W.F. King, B.G. Lindsay, R. Lorie, J.W. Mehl, T.G. Price, F. Putzolu, P.G. Selinger, M. Schkolnick, D.R. Slutz, I.L. Traiger, B.W. Wade and R.A. Yost. "A History and Evaluation of System R" *CACM 24*, 10 (October 1981) pages 632-646.

Remembrances

Mike Blasgen: I would like to take a moment here to remember to commemorate three people who were involved in this project but couldn't be here today because they are not alive; three people who made important contributions at various points along this project. The three people that can't be here are Ray Boyce, Vera Watson, and Morton Astrahan. So can I ask Don Chamberlin to say a few words about Ray Boyce?

Ray Boyce

Don Chamberlin: Working with Ray was one of the great privileges I've had in my career. I didn't get to do it for very long, but it's something I'll always remember. Ray grew up in New York State on the west side of the Hudson River. He went to college in Providence, Rhode Island. He met his wife Sandy there. She was a nursing student. He got his PhD in Purdue and he was one of the few people I ever met who actually liked it in West Lafayette, Indiana. After he left Purdue he joined the group that I was working in, in Yorktown Heights, New York, where we were just beginning to work on database projects under Frank King.

Ray was a person who made things happen. He was a very smart and very ambitious guy and had a lot of energy. I really think Ray, if he'd lived, would have been in the class with Steve Jobs and Larry Ellison and Bill Gates – everybody would know Ray's name, I think, if he was alive today. Ray and I worked together in a very close collaboration in the early days on the SQL language – it was called SEQUEL in those days. This collaboration was so close that at the end of the day neither one of us could remember what ideas each one of us had contributed to the work. So it was a very close partnership. The main difference between Ray and me was Ray was a lot more interested in management than I was, so when it came time to choose a manager for the group, Ray was the one who got the job, and I thought that was a real good division of labor. So Ray was my boss for a while.

He and Sandy had a daughter Kristen just a few months after they arrived in California. Ray and Irv Traiger were the two managers under Frank King in the early days of System R. I had a car pool with Ray and one day in the spring of 1974 I drove Ray to work and after lunch I heard a kind of rumor in the building that Ray had collapsed at lunch time. He was the picture of health – he was strong and vigorous and I didn't have any clue that he had any sort of health problems. One day at lunch he just kind of fell over, and he was taken to the hospital. He had an aneurysm of the brain, which is a blood vessel that swells up and bursts inside the brain. He was taken to Valley Medical Center and was operated on and lived for a short time after his operation, but he died on Father's Day in 1974. His daughter was only about nine months old when he died.

His wife and daughter still live in San Jose and we've kept in close touch with them over the years. Sandy went back to school and got her master's degree in clinical psychology. She's working as a counselor now for children and foster parents. Kristen grew up and went to the University of California at Santa Barbara where she still is – my daughter is there, too. She got married last year and will graduate from UCSB this year with a bachelor's degree and she's going to stay there and work on her teaching credential.

So I think the most important things to Ray were his work and his family. I think he would have been real proud of what became of his work. In the short period that he had, which was not quite two years long, he invented Boyce-Codd Normal Form, which is still taught in textbooks; he developed the SQL language, which some people still remember. So I think he would have been real proud of that, and I think he would have been real proud of the way his family turned out. I wish Ray could have known the impact his work would have had.

Vera Watson

Mike Blasgen: Thank you, Don.

Vera Watson. I met Vera Watson when I moved to Yorktown to go to work in the Research Division in New York. One of the other people in the group I was in was Vera Watson. Vera has a very unusual background. She was born in China of Russian parents. That was part of a Russian community that occupied a portion of China. Yul Brynner also has the same background, in case you care. So she spoke Russian and came through England to the United States, I would guess in the late fifties, was hired into IBM Research because of her Russian language skills. That is when the optimism was running high about automatic translation of languages – this was text-to-text translation between languages. There was a big research project to do that. It was expected that it was just a matter of a few more months and this would be routine. It didn't turn out that way. But they needed the special skills of somebody who was fluent in Russian, so Vera was brought in. She eventually became part of several different groups in Yorktown; became a programmer, contributing in programming to several different projects that I also was involved in (graphics projects and other things). She moved to California in probably the beginning of 1974 or the end of 1973, about the time that several other people moved from New York to San Jose, and I moved out soon after and joined her and worked with her in the same department under Traiger.

Vera had an interest outside of work, which was mountain climbing. She was a very serious mountain climber, a member of the Alpine Club in New York City. She was a very serious climber – rock climber, mountain climber. In 1975 or 1976 she took off three or four months

from work and went to South America and did a solo ascent of Aconcagua, which is the highest mountain in the western hemisphere. I remember Frances King wrote a poem about Vera's ascent of Aconcagua. Then the following year, which would have been about 1977, she had a special opportunity to join an all-women's assault on Annapurna. Annapurna's one of the major Himalayan mountains. Many of us were involved in that and had fond memories of Vera going off to do that. One of the unusual things about that I remember is that, at the time to get a leave without pay, in this case for the three months or four months that was required to do an assault on a major Himalayan peak, you had to claim to IBM this was a once-in-a-lifetime opportunity, which it surely was, except the trip to Aconcagua was also a once-in-a-lifetime opportunity. So she had two once-in-a-lifetime opportunities within two years of each other.

So she joined the group to climb Annapurna, and was part of the second team to attempt the summit. You go up in pairs, so you do pairwise summit attempts – these Himalayan style things where you do base camps. So she was working her way to the upper camp as the first summit team was coming down between the topmost and the second topmost. They passed, and then she was lost – she and her partner were lost. We're quite sure they fell. They were roped together; we think one fell and took the other with them.

I learned of this from a phone call from John McCarthy. Vera had married John McCarthy, the father of Lisp and of artificial intelligence. John called my office to tell me that he had just learned of this mess. I remember going in to a meeting of the department. I even remember the conference room in Building 28 where we met. I told this story roughly like this that Vera was lost. They did send up others to try to find her. They were able to see the bodies in the snow way below but it was not considered safe to descend, and even if you could descend to the bodies, there was no way to bring the bodies back out without bringing in helicopters and things like that, which were not considered justified. So there's a memorial at the base camp at Annapurna today to Vera, among others who've died on that assault; it's a serious mountain.

And so I think it's nice that we can all remember Vera. Vera contributed a lot to this project. If you look at this piece of paper here, it says VM+[3]. That plus sign is Vera. She did the work to modify an IBM operating system to make it suited for running the multi-user version of System R. So, we all remember Vera.

Now I'll turn it back to Don for Morton Astrahan.

Morton Astrahan

Don Chamberlin: Morton was a real unusual guy. I first met Morton when I transferred to California along with Ray Boyce and Frank King, Vera Watson, and some other people, in 1973. This was a large infusion of new people into the environment at San Jose, which had a project underway, and that project of course was impacted by the arrival of the newcomers, and different people had different attitudes about that. The term "Yorktown Mafia" is indicative of one of the attitudes; Morton never used that term. Morton's attitude toward newcomers was, "Welcome to California. How can I make you guys feel at home?" Since not everybody felt that way, it was real nice to have Morton around, because Morton knew the ropes and he was the guy who helped us find places to live and places to shop and things to do at night. He was real nice that way, to make us feel like we were welcome by the natives. I say that although I'm a native of San Jose.

Morton had a cabin up in the mountains that he called Serendipity. Serendipity, as you know, means a kind of a surprising good outcome. I never figured out exactly why Morton's cabin was called Serendipity, but that cabin was an important thing to Morton, and one of the things that he did was to invite all of the newcomers from New York, one at a time, up to his mountain on the weekends. So we went up there and took our young daughter and it was a beautiful place and Morton really enjoyed sharing it with people. Morton claimed that he had a muse that lived in Serendipity and whenever there was some kind of technical problem that came up in our project that had everybody scratching their head, Morton would tend to disappear for several days at a time and would go up and consult his muse. A lot of times he'd come back and he'd have the problem solved. I thought that was pretty nice. When Morton disappeared, I always looked forward to what he'd have to say when he came back.

One of the things about Morton was he didn't really like to argue, and just about everybody else in the project liked to argue a lot *[laughter]*, so this made Morton kind of unique. Something that would happen a lot of the time was everybody would have meetings all week and do a lot of shouting over some technical issue, and by the time the dust sort of cleared, here would be Morton, who hadn't come to the meeting and sat in his office and wrote code all week, and he'd have the problem solved. He was real productive and real fast that way. He got a lot done with kind of a minimum amount of heat and political energy.

Another thing that I remember about Morton was his courage. Morton had a lot of health problems: he had Parkinson's Disease and he had crippling arthritis so he couldn't stand up straight and I think he was real uncomfortable a lot of the time. But I never heard Morton mention that a single time to anybody and it never limited any of his activities. You know, Morton was always first in line and had more energy than anyone else. It must have

[3] J.N. Gray and V. Watson. A Shared Segment and Inter-Process Communication Facility for VM/370. IBM Research Report RJ1579. San Jose, California (May 1975).

been very uncomfortable for him and taken a lot of courage to do that. But, you know, Morton was always kind of right out there carrying more than his share.

Morton retired from IBM sometime in the mid 1980's – I don't recall the exact date – and he died shortly after that – probably 1986 or thereabouts.

Morton is somebody who I remember for his courage and friendship and constructive attitude. If you had something you needed done without a lot of fooling around, then Morton was the guy you wanted to get in touch with.

The Birth of SQL

Prehistory

Mike Blasgen: So now we have a discussion about how it all began and how it proceeded. I have a timeline – some of you have seen it because I sent out one version of it – which acts to make me remember how to prompt people and also help me remember stuff that I remember myself. So I will do this. The earliest I remember is I was at [The University of California at] Berkeley and I remember a sign on the wall somewhere in the 2nd or 4th floor [of Cory Hall] saying that there were some interesting things going on in San Jose. I was still a student, so this would have been in 1968, roughly. So already San Jose was doing work in database. I don't think it was called that, then. It was called data management or file systems, or – I don't remember what it was called. But it had to do with work that Mike Senko was leading. And of course the research laboratory itself was always associated with data because the original development of the disk drive occurred there in the early fifties. So already by the late sixties there was a focus on software for the management of data. And I'm not familiar with that at all, nor was I involved in any of the work prior to the Phase Zero prototype of SEQUEL. But there was much work that went on in the company.

Irv, what led to Codd's paper[4], which was published in 1970?

Irv Traiger: I honestly don't know. There were two departments back then, the Systems Department under Jim Eaton and later Glenn Bacon, and another one – I think it was called Information Systems or something like that – under Senko, and they were very different worlds. People might play Ping-Pong together at lunch – there was a lot of Ping-Pong then – but essentially no technical interaction. You'd hear about things over there. In fact at one point there was a big project called DIAM[5, 6] with a very complex structure, a complex query language. And we knew that this man was over there named Ted Codd and that there were some disagreements, but I really don't know what led to what. At one point, Ted Codd suddenly showed up in the Systems Department and after some delay he built up a small group of people – it was actually three people originally: Dines Bjørner, Ken Deckert, and me. We began to work on a project called GAMMA-0, and I brought the GAMMA-0 paper[7] with me.

Mike Blasgen: Oh, really? Is it on the artifact table?

Irv Traiger: Not yet; it will be there. GAMMA-0 was meant to be the lowest-level thing that anybody would get value from, and even then there was the notion of supporting multiple things on top, which would happen again in System R and in Eagle, the big project at Santa Teresa. Nevertheless, what kicked off this work was a key paper by Ted Codd – was it published in 1970 in CACM?

Mike Blasgen: Yes.

Irv Traiger: A couple of us from the Systems Department had tried to read it – couldn't make heads nor tails out of it. *[laughter]* At least back then, it seemed like a very badly written paper: some industrial motivation, and then right into the math. *[laughter]*

Bob Yost: I went over there with several other people – I was in the Advanced Systems Development Division – I remember going over there in about 1970 to see this because we were working with the IMS[8] guys at the time. We couldn't believe it; we thought it's going to take at least ten years before there's going to be anything. And it was ten years. *[laughter]*

Irv Traiger: So we had this 1970 paper; there were a couple of other papers that Ted had written after that; one on a language called DSL/Alpha[9], which was based on the predicate calculus. Glenn Bacon, who had the Systems Department, used to wonder how Ted could justify that everybody would be able to write this language that was based on mathematical predicate calculus, with universal quantifiers and existential quantifiers and variables and really, really hairy stuff.

---

[4] E.F. Codd. "A Relational Model of Data for Large Shared Data Banks" *CACM 13*, 6 (June 1970) pages 377-387.

[5] M.M. Astrahan, E.B. Altman, P.L. Fehder, and M.E. Senko. "Concepts of a Data Independent Access Model" *1972 ACM SIGFIDET Workshop Report*, pages 349-362.

[6] E.B. Altman, M.M. Astrahan, P.L. Fehder. and M.E. Senko. "Specifications in a Data Independent Access Model" *1972 ACM SIGFIDET Workshop Report*, pages 363-376.

[7] D. Bjørner, E.F. Codd, K.L. Deckert, and I.L. Traiger. The GAMMA-0 *n*-ary Relational Data Base Interface: Specification of Objects and Operations. IBM Research Report RJ1200. San Jose, California (April 1973).

[8] IMS stands for Information Management System, IBM's first database management system.

[9] E.F. Codd. A database sublanguage founded on the relational calculus. *Proc. ACM SIGFIDET Workshop on Data Description, Access, and Control*, San Diego, California (November 1971) pages 35-68.

Somehow, again, I don't know how, there grew up around IBM a bunch of pockets of activity. There was a project in the Peterlee Science Center in England of all places. Peterlee was a manufactured town. The English government was trying to seed industry and business in different parts of the UK and they invented Peterlee and IBM said, "Sure, we'll put a lab there." There was a person – was it Terry Borden? – Terry Rogers who was heading up this project based on the relational algebra – a very weird language that occasionally gets used nowadays as an intermediate layer in a system. There was a project in Hursley (kind of interesting how much activity in England) called the Hursley Prototype – was that Peter King?

Raymond Lorie: Peter Tilman.

Irv Traiger: OK, Tilman. There was a project at the Cambridge, Massachusetts, Scientific Center. Raymond Lorie, Andrew Symonds, and others, were doing that[10]. And there was a predecessor project[11] that had been done at MIT Lincoln Laboratory by Paul Rovner (who went to school with Mike and Jim Gray and Mario [Schkolnick] and me at Berkeley) and Jerry Feldman, who later became a Stanford professor and is now the head of ICSI[12] at Berkeley. So there were these pockets, and so Ted Codd wanted to establish his own pocket, and that turned into this GAMMA-0 project.

At one point Codd decided to set up a symposium at Yorktown – you know, the seat of power in the Research Division – and it was to basically have a scan of all the activity across IBM related to his relational ideas. We went through that, with the various labs being represented, and a bunch of others, and somehow or other a few months later this project happened. It was to be in San Jose; it was to have an infusion of people from Yorktown; and we didn't know what that would be like, but it wasn't a problem. People like Frank King and Don Chamberlin and Ray Boyce were certainly aware of the fact that they were the incoming horde, but they were very sensitive about it and they tried very, very hard to involve the San Jose people. Mike Senko and his department were merged into the Systems Department, which was renamed Computer Science, under Leonard Liu. Glenn Bacon went off to SSD, or what's now called SSD[13]. Mike Senko went back east, stayed in IBM, and died not too long after that, I think in Europe on a business trip. Frank King kept us kind of in task force mode for quite a few months, trying all kinds of crazy management schemes, like mentors, and inner circles, and teams. Out of that grew System R. That's kind of the long story. I don't want to steal the whole stage here. That's kind of the vague memory of how it all began.

Mike Blasgen: That's great. So actually you mentioned a lot of the points in my list here: I have Mike Senko, the Ted Codd paper, PRTV[14], Cambridge, … So now, how did the Codd-Bachman thing come about? How did that fight come about? Is that related to DBTG?

Irv Traiger: Yeah, there was this standard going on. It was organized by the Database Task Group and it was called CODASYL[15]: Common Data something – Systems Language – how does that sound? It's kind of *deja vu* because you hear today about how important it is to follow standards, and if we had done it back then none of this stuff would have happened because DBTG was richer than

---

[10] The RM (Relational Memory) system supported binary relations; see:

A.J. Symonds and R.A. Lorie. "A schema for describing a relational data base" *Proc. ACM SIGFIDET Workshop on Data Description, Access, and Control*, (November 1970) pages 201-229.

R.A. Lorie and A.J. Symonds. "A Relational Access Method for Interactive Applications." *Courant Computer Science Symposia, Vol. 6: Data Base Systems*. Prentice-Hall, Englewood Cliffs, New Jersey (1971).

The successor XRM (Extended Relational Memory) system supported n-ary relations; see:

R.A. Lorie. XRM—An Extended (N-ary) Relational Memory. IBM Technical Report G320-2096. Cambridge Scientific Center, Cambridge, Mass. (January 1974).

[11] J.A. Feldman and P.D. Rovner. "An Algol-Based Associative Language" *CACM 12*, 8 (August 1969) pages 439-449.

[12] International Computer Science Institute.

[13] SSD stands for Storage Systems Division.

[14] PRTV stands for Peterlee Relational Test Vehicle. See:

Stephen Todd. "PRTV, an efficient implementation for large relational data bases" *Proc. VLDB*, Florence, Italy (1975), pages 554-556.

[15] Actually, CODASYL stands for Conference on Data Systems Languages, which was formed in 1959 to design the business data processing language COBOL. CODASYL's Data Base Task Group defined what has become known as the DBTG database model:

CODASYL Data Base Task Group. Report of the CODASYL Data Base Task Group. ACM (April 1971).

R.W. Taylor and R.L Frank. "CODASYL Data-Base Management Systems" *ACM Computing Surveys 8*, 1 (March 1976) pages 67-103.

IMS[16]; it was a network, which certainly includes a hierarchy; and for that matter, if you wanted flat files, you basically had that in DBTG. You could just omit the named relationships. What's the big deal, right? You want a good language, we'll give you a language. The technical community, which was kind of small then for database, had its own SIG and I don't remember what it was called. SIGMOD was new.

Raymond Lorie: SIGFIDET.

Irv Traiger: SIGFIDET. SIGMOD was the kind of grass roots, revolutionary, not taken seriously bunch and SIGFIDET and CODASYL just sort of ran the whole game, and Bachman was Mr. CODASYL[17]. On several occasions, and I don't remember them all, maybe one at an early SIGMOD conference, these people would go at each other, I mean just hurling thunderbolts, about better and worse, complicated and simple, and mathematical foundations, and who cares.

Mike Blasgen: One of those debates was published and widely circulated[18].

C. Mohan: NCC panel, I think. National Computer Conference.

Don Chamberlin: There was one at the SIGFIDET conference in Ann Arbor, Michigan in 1974.

Franco Putzolu: I think for a while people who eventually worked on System R worked on design techniques for DBTG databases. Also there was a project I remember in Yorktown in 1972-73 on how to design DBTG databases.

Don Chamberlin: I was working on that. I was recruited by Leonard Liu in Yorktown in 1971 to work on an operating system project called System A. Leonard Liu was a first-level manager in those days and I worked for Leonard for a year or so, until the System A project broke up in 1972. It seemed like every time there was an upheaval, Leonard got promoted and that was what happened in 1972. *[laughter]* Leonard got promoted to be a second-level manager and I started working for Frank King. We were in kind of a state of chaos in Yorktown in 1972 because our operating system project had broken up and we didn't have anything to do. Leonard was pretty astute politically and he thought that database was an important field to get into, so he kind of organized us into study group mode to try and figure out what needed to be done in databases. I got a particular job in this. I thought it was a plum of a job. My job was to study this CODASYL DBTG proposal and learn about it and give presentations on it and figure out what needed to be done to it and things like that. So I became an expert on DBTG and I just loved it and thought it was neat. It had all sorts of real complicated pointers and set-oriented selection rules and you could just study it all day. It was a real puzzle. I was kind of a programmer type; I really grooved on that and gave a lot of talks on it and things like that. I was the CODASYL expert in our group; other people studied other things: CICS[19] and IMS and different things like that.

We knew sort of peripherally that there was some work going on in the provinces, in San Jose. There was this guy Ted Codd who had some kind of strange mathematical notation, but nobody took it very seriously. Ray Boyce was hired at about this time, and we kind of got into this game called the Query Game where we were thinking of ways to express complicated queries. But actually before the Query Game started, I had a conversion experience, and I still remember this. Ted Codd came to visit Yorktown, I think it might have been at this symposium that Irv alluded to. He gave a seminar and a lot of us went to listen to him. This was as I say a revelation for me because Codd had a bunch of queries that were fairly complicated queries and since I'd been studying CODASYL, I could imagine how those queries would have been represented in CODASYL by programs that were five pages long that would navigate through this labyrinth of pointers and stuff. Codd would sort of write them down as one-liners. These would be queries like, "Find the employees who earn more than their managers." *[laughter]* He just whacked them out and you could sort of read them, and they weren't complicated at all, and I said, "Wow." This was kind of a conversion experience for me, that I understood what the relational thing was about after that.

Ray Boyce had just been hired at that time, and we organized between the two of us this game that we called the Query Game, where we'd think of different questions that needed to be expressed and we'd try to find out syntax to express them in. These are some original foils from back in those days that we put together to try and convince people of things. We called the notation SQUARE; it stands for Specifying Queries as Relational Expressions. We had this idea, that Codd had developed two languages, called the relational algebra and the relational calculus. In the

---

[16] IMS is hierarchical.

[17] C. Bachman. "The programmer as navigator" (Turing Award lecture) *CACM 16*, 11 (November 1973) pages 653-658.

[18] "Data Models: Data Structure Set versus Relational" Supplement to *Proc. ACM SIGMOD Workshop on Data Description, Access and Control*, Ann Arbor, Michigan (May 1974).

[19] CICS stands for Customer Information Control System, IBM's TP monitor, or framework for writing online transaction-processing applications.

relational algebra, the basic objects were tables, and you combined these tables with operations like joins and projections and things like that. The relational calculus was a kind of a strange mathematical notation with a lot of quantifiers in it. We thought that what we needed was a language that was different from either one of those, in which the basic objects that you worked on were sets of values, and the things you did to those sets of values were you mapped one set of values into another using some kind of a table. So we had the usual database of sales and departments and items being located on different floors and we would take a value like two and map it through this notation into the departments that were on that floor, and then we'd map it again into the items that were sold by those departments. We would try to show that this mapping notation was simpler than some of the complex ways that you'd have to express this query in relational calculus, or of course far worse, using something like CODASYL.

So that was where this idea called SQUARE came from, and that was what Ray and I were working on when we transferred to San Jose in 1973, along with Leonard and Frank and Vera Watson and Robin Williams, who all came to San Jose at the same time. Jim Gray had come out the year earlier because he liked it on the west coast. Franco and Mike followed, I believe, in the following year, in 1974. So that was what was happening in Yorktown during the same period of time that Irv was working with Ted Codd at San Jose.

Mike Blasgen: That's great; I'm learning all kinds of things I didn't know.

Something that Irv mentioned was that there was a number of us who had an association with the University of California at Berkeley, and it is an amazingly large number. You wouldn't guess it – well, maybe it's because of geography. It's Irv, and Bruce [Lindsay], and Paul [McJones], and me, and Mario [Schkolnick], and Bob Selinger later, Bob Yost, and of course Jim Gray, who's actually a McKay fellow at the University of California at Berkeley right as we speak, is that right?

Jim Gray: As we speak, until midnight. *[laughter]*

Mike Blasgen: May 31 is his last day.

In case anyone is interested, here is the 1968 General Catalog for the University of California at Berkeley. That happened to be the year I taught at Berkeley. My name's not in here. Butler Lampson's name is in here, as teaching a course in operating systems.

Bruce Lindsay: I took that course.

Mario Schkolnick: I have heard rumors that you could flunk this course just by having grammatical typos in your reports. I was very sensitive to this, having just arrived from Chile to study at Berkeley.

Franco Putzolu: Do you know when INGRES started?

Mike Blasgen: I actually have that here, but I don't know the answer: about the same time. I went to Berkeley at the beginning of 1975. Gene Wong was my advisor when I was at Berkeley, Wong was one of the developers. Wong had a particular optimization procedure that he was advocating, and INGRES implemented it. Stonebraker had developed QUEL. So QUEL was mapped to this trick which I don't actually remember and which is not the fundamental contribution that INGRES made to the world.

Irv Traiger: It was to optimize based on how the query was doing dynamically, right?

Mike Blasgen: Well, it was a specific technique …

Raymond Lorie: Single-variable query.

Mike Blasgen: That's right, it was a single-variable trick. I went to see that in 1975 and it was running. You could type QUEL into a UFI-like thing. They supported only query – there was no possibility of update. I guess you could have multiusers given that it was a timesharing system. It ran on a PDP-11/45.

Jim Gray: In about 1972 Stonebraker got a grant to do a geo-query database system. It was going to be used for studies of urban planning. The project did do some geographic database stuff, but fairly quickly it gravitated to building a relational database system. The result was the INGRES system[20]. INGRES started in about 1972 and a whole series of things spun off from that: Ingres[21], Britton-Lee, and Sybase.

Hostility developed between the San Jose IBM group and the Berkeley group because they were working on very, very similar things and had very, very similar ideas. Almost everybody was young and insecure (untenured), so there was a lot of concern about the priority of publishing. As a consequence we came to the conclusion that the best thing was not to talk to each other. Every time we talked, papers would appear that reflected the conversations without attribution. Occasionally people would go back and forth; Randy Katz was in both camps. We occasionally had summer students come to IBM and occasionally we would all give talks but always very carefully. In the chron file

---

[20] M. Stonebraker, E. Wong, P. Kreps, and G. Held. "The Design and Implementation of INGRES" *ACM TODS 1*, 3 (September 1976) pages 189-222.

[21] The company was first called Relational Technology Inc., and was then renamed Ingres Corporation. ASK bought Ingres, and was itself bought by Computer Associates International, Inc.

there are letters from Stonebraker saying, "Thanks for pointing out that in paragraph so-and-so of paper such-and-such we forget to cite ???". Of course this was not one-sided. The Berkeley folks thought the IBM guys were ripping off ideas from the INGRES project. We had a strained relationship[22].

Mike Blasgen: I actually personally have fairly fond memories of the relationship. But I know that lots of others like Frank and many others have bad feelings about it because apparently ideas were being taken from us and used by them without any credit.

Jim Gray: And conversely.

Franco Putzolu: Vice versa.

Mike Blasgen: OK, and vice versa. But I always heard the accusation the other way. *[laughter]*

But I personally had only good interactions with – well Gene Wong was my research advisor and was one of the key players in this thing. John Paul Jacob organized an event at the Catholic University in Rio in 1975 I would guess, the summer of 1975: it might have been the summer of 1976. Sharon and I went down to Rio, which was a really nice trip, we stopped in other places in South America. At that thing was Mike Stonebraker staying there for a month, Dennis Tsichritzis and his wife from the University of Toronto, Sharon and I, and others. I don't remember who else from IBM was there; was anybody in this room there? Jim wasn't there. I was in Rio for maybe two weeks: one week by myself giving lectures at this conference they had, and one week with Sharon just fooling around and giving more lectures. We were kind of stuck there, the five of us: Dennis and his wife, Sharon and me, and Mike Stonebraker (who was single). And so we palled around together. And so I got to be like a friend of Mike's because I was stuck in this place far away where you had nothing to do except go drink, which we did a lot of. So I got very close personally with Mike; Mike has always treated me, I always thought, very nicely. 'Course I don't know: maybe he talks behind my back.

Jim Gray: The good news was you worked on B-trees; they didn't do B-trees. *[laughter]* I worked on locks and they didn't do locks, so I was also OK.

System R

Mike Blasgen: So now we've reached the ancillary stuff, the peripheral stuff, and now we have the kickoff of System R, which Don has already introduced with this task force and all this stuff that happened, and which I didn't know. I originally thought that this twentieth anniversary should be the twentieth anniversary of some particular event that occurred on some day. The day I was going to pick was the day that the project got named System R. It was full-fledged by then; then this chart that I had up here existed. Once there was a System R, all these names fell out: RDS, RSS. Actually, historically it may have been the other way: it may have been these names that lead to this name. That I believe was at the end of 1974; almost Christmas of 1974. Does somebody remember a better date than that? Irv, I know you were involved. I remember you and Frank were walking down the hall, talking about the name.

Irv Traiger: Leonard had ordered all of us to pick a name for this project. We just sort of shrugged off, "It's not important." He said, "It's important in terms of recognition to have a name." We would make attempts at coming up with a name over weeks. One was Rufus, which was Franco's dog.

Franco Putzolu: Rufus would have been a better name. It stands for Relational User Friendly Universal System.

Mike Blasgen: It would have been a better name.

C. Mohan: Later we actually had a project named Rufus. Kurt Shoens'[23] …

Irv Traiger: It was really hard.

Mike Blasgen: So it was named roughly at the end of 1974?

Irv Traiger: Don't remember.

Tom Price: Was that the time that Leonard made you guys all work on Christmas Eve? I heard a story once that he wouldn't let anybody off on Christmas Eve?

Irv Traiger: I think that was back in Yorktown.

Don Chamberlin: That was in Yorktown; yeah, I remember that. *[laughter]* This was the Friday before

[22] The 1988 ACM Software System Award was shared by System R (Donald Chamberlin, James Gray, Raymond Lorie, Gianfranco Putzolu, Patricia Selinger and Irving Traiger) and INGRES (Gerald Held, Michael Stonebraker and Eugene Wong).

[23] K. Shoens, A. Luniewski, P. Schwarz, J. Stamos, J. Thomas. "The Rufus System: Information Organization for Semi-Structured Data" *Proc. VLDB*, Dublin, Ireland (1993).

Christmas and the lab had some kind of a party with cookies and Santa Claus and music and everything down in the cafeteria and Leonard wanted to have some kind of technical meeting right through the whole thing. Leonard expected a lot of his people, but he also treated them well.

Mike Blasgen: Leonard was quite a character: a lot of fire and brimstone and vim and vigor and all those pairs of words. I remember probably in 1975 we went off to the beach at Pajaro Dunes and Leonard stood up and said, "OK, what are all the bad things that are going on in the department? What are all the bad things I'm doing?" And he made everybody say them. Everybody complained. And he wrote down this list of complaints. He didn't say anything. He just wrote down complaints. And then he said, "OK, shut up," and he talked for two hours without a break telling us, basically, everything we were complaining about was not correct. *[laughter]*

Management by consensus: I have decided; you concede. *[laughter]* It was so amazing; it was completely oblivious to him that he was doing this. It worked; it worked very well for him. In case you don't know, he's the Chief Operating Officer of Cadence. Cadence's number one customer is IBM. They sell electronic design tools for laying out circuits on chips.

By the way, this System R thing of course makes me put this [cartoon] up. I don't know when this picture was drawn; this is my favorite chart. This is a rabbit and a beaver talking, and behind them you can see Hoover Dam. The beaver is saying to the rabbit, "I didn't actually build it, but it was based on my idea." *[laughter]* So this little beaver is System R, because I don't think there is much code of System R left around; a little bit in SQL/DS I guess.

C. Mohan: Quite a bit, actually, especially the RSS.

Mike Blasgen: All right, the index component is still alive. *[laughter]* That's what I wrote, and the index component is still in the product, SQL/DS.

C. Mohan: All the shadow-paging stuff is there.

Mike Blasgen: Oh, the shadow page's still there? That's Raymond Lorie's stuff.

C. Mohan: Record management, all that stuff's still there.

Bruce Lindsay: Storage pool.

Brad Wade: Like to know if anybody can still understand it.

Pat Selinger: Mohan still reads it.

Mike Blasgen: You don't have to understand it; it just has to produce revenue and profit. It's a successful product today.

???: It supports a lot of us.

Mike Blasgen: Right. So …

Brad Wade: Before we leave naming, there was also the RDS and the RSS names. Of course Don was manager of the RDS before it was called RDS; Irv was manager of the RSS before it was called RSS. And they were carpooling and they came in one day and said, "OK, here are your names: Don and Irv, Data Organized Naturally, and I forget what Irv was for: Intermediate or Interactive Relational …

Mike Blasgen: Intermediate Retrieval Vehicle? How about that? Sounds good. No, there was the Peterlee Relational Test Vehicle, so V was already established as an acceptable term in Relational terminology. So it's just a question of putting the Vehicle in there somewhere.

So how about what sort of happened with System R. Irv and Don were the managers of the project. Why don't one of you volunteer to take us through the System R history?

Don Chamberlin: I think it's going to need both of us to do this. I'll give it a start.

This shouldn't be a monologue; please stand up and help me out here. As Irv said, there was a long period after Frank arrived in California when we had a lot of meetings and a lot of discussions and task forces and tried to organize an approach to take to this business. Interestingly enough, Ted Codd didn't participate in that as much as you might expect. He got off into natural language processing and wrote a very large APL program called Rendezvous[24, 25]. He really didn't get involved in the nuts and bolts of System R very much. I think he may have wanted to maintain a certain distance from it in case we didn't get it right. Which I think he would probably say we didn't.

Mike Blasgen: Oh, he has said that, many times.

Don Chamberlin: What came out of this was we got organized into two groups, a higher-level group which

---

[24] E.F. Codd. "Seven Steps to Rendezvous with the Casual User" *Proc. IFIP-TC2 Conference on Data Base Management*, Cargese, Corsica (April 1-5, 1974) pages 179-200.

[25] E.F. Codd, R.S. Arnold, J-M. Cadiou, C.L. Chang, N. Roussopoulos. RENDEZVOUS Version 1: An Experimental English Language Query Formulation System for Casual Users of Relational Data Bases. IBM Research Report RJ 2144. San Jose, California (January 1978).

ultimately was called the RDS[26] and which was interested mainly in language issues, and a lower-level group called the Research Storage System, which was interested more in physical data management issues. I can talk mainly about what was happening in the top half of the project in those days and I'm hoping that Irv and maybe some of the rest of you – Jim – will talk about what was happening in the bottom half.

What really happened in the early days was Irv's group began developing a new data management interface, with support for indexes, locking, logging, concurrency and transactions, and all those kinds of things. Meanwhile the language folks wanted to build a prototype of their language and they needed a base to build it on, and the RSS wasn't ready. The only thing we could get our hands on was something that Raymond Lorie had built at the Cambridge Scientific Center called XRM. So we built a prototype of our language on top of XRM in the early days; we called it Phase Zero[27]. Brad has a wonderful tape which many of you saw last night that represents a complete working prototype of SEQUEL in 1976 I believe, complete with integrity assertions, which have just now made it into the product twenty years later. *[laughter]* And we demonstrated that, or at least showed the tape, at the SIGMOD conference in, was it 1976?

Brad Wade: 1976.

Don Chamberlin: Hopefully today we'll get a chance to see that tape again. It's a wonderful tape; you get to see Brad with a handlebar mustache. Good stuff.

Franco Putzolu: Don, did you have a customer in New England?

Don Chamberlin: Yes, as a matter of fact, that was the most important outcome of the Phase Zero work, I think in my opinion. That's a kind of interesting story. Back in those days, there were a lot of problems with fuel shortages; OPEC had just raised the price of oil and the gasoline companies were hoarding it and there were lines at the gas stations. The MIT Sloan School of Management had some kind of a plan in New England where they got a grant to build something called the New England Energy Management Information System, or NEMIS, and they needed a database to keep track of how full the oil tanks were and things like that. So the Cambridge Scientific Center was kind of tight with San Jose Research, and they got their hands on this Phase Zero prototype and worked on it with the Sloan School of Management on this energy management system[28], but anyway, one of the students at MIT who was involved with this was somebody named Bob Selinger. And Bob, didn't you kind of get your fingers into Phase Zero and use it a little bit for something? As a result of this, Bob came out to San Jose as a summer student, because of the experience that he'd had with the Phase Zero prototype. When he came to San Jose, he met someone named Pat Griffiths[29]. That's how Bob came to IBM.

So I think the most important outcome of the Phase Zero prototype was … *[laughter]*

Pat Selinger: Did the energy management system ever get used? [???]

Bob Selinger: There were databases on it. I'm not sure they were widely used. Actually they used it as a database for building designs. They kept track of square footage, number of windows, and then they had some FORTRAN programs that ran on top of it. It bridged FORTRAN into, I think PL/1, to extract the data. It was pretty hokey.

Don Chamberlin: So what this language group wanted to do when we first got organized: we had started from this background of SQUARE, but we weren't very satisfied with it for several reasons. First of all, you couldn't type it on a keyboard because it had a lot of funny subscripts in it. So we began saying we'll adapt the SQUARE ideas to a more English keyword approach which is easier to type, because it was based on English structures. We called it Structured English Query Language and used the acronym SEQUEL for it. And we got to working on building a SEQUEL prototype on top of Raymond Lorie's access method called XRM.

At the time, we wanted to find out if this syntax was good for anything or not, so we had a linguist on our staff, for reasons that are kind of obscure. Her name was Phyllis Reisner, and what she liked to do was human-factors experiments. So she went down to San Jose State and recruited a bunch of San Jose State students to teach them the SEQUEL language and see if they could learn it. She did this for several months and wrote a paper about it, and gained recognition in the human-factors community for her work.[30, 31] I'm not sure if the results were very conclusive; it

---

[26] RDS stands for Relational Data System.

[27] M.M. Astrahan and D.D. Chamberlin. "Implementation of a structured English query language" *CACM 18*, 10 (October 1975), pages 580-588.

[28] J.J. Donovan, L.M. Gutentag, S.E. Madnick, and G.N. Smith. "An Application of a Generalized Management Information System to Energy Policy and Decision Making – The User's View" *Proc. NCC*, AFIPS Vol. 44 (1975) pages 681-686.

[29] Now Pat Selinger.

[30] P. Reisner, R. F. Boyce, and D.D. Chamberlin. "Human Factors Evaluation of Two Data Base Query Languages--

turned out that sure enough if you worked hard enough, you could teach SEQUEL to college students. *[laughter]* Most of the mistakes they made didn't really have anything to do with syntax. They made lots of mistakes – they wouldn't capitalize correctly, and things like that.

Looking back on it, I don't think the problem we thought we were solving was where we had the most impact. What we thought we were doing was making it possible for non-programmers to interact with databases. We thought that this was going to open up access to data to a whole new class of people who could do things that were never possible before because they didn't know how to program. This was before the days of graphical user interfaces which ultimately did make that sort of a revolution, and we didn't know anything about that, and so I don't think we impacted the world as much as we hoped we were going to in terms of making data accessible to non-programmers. It kind of took Apple to do that. The problem that we didn't think we were working on at all – at least, we didn't pay any attention to it – was how to embed query languages into host languages, or how to make a language that would serve as an interchange medium between different systems – those are the ways in which SQL ultimately turned out to be very successful, rather than as an end-user language for *ad hoc* users. So I think the problem that we solved wasn't really the problem that we thought we were solving at the time.

Anyway, we were working on this language, and we adapted it from SQUARE and turned it into English and then we started adding a bunch of things to it like GROUP BY that didn't really come out of the SQUARE heritage at all. So you couldn't really say it had much to do with SQUARE before we were done. Ray and I wrote some papers about this language in 1974. We wrote two papers: one on SEQUEL/DML[32] and one on SEQUEL/DDL[33]. We were cooperating very closely on this. The DML paper's authors were Chamberlin and Boyce; the DDL paper's authors were Boyce and Chamberlin, for no special reason; we just sort of split it up. We wanted to go to Stockholm that year because it was the year of the IFIP Congress in Stockholm. I had a ticket to Stockholm because of some work I'd done in Yorktown, so Ray submitted the DDL paper to the IFIP Congress in Stockholm, and the DML paper we submitted to SIGMOD. This is the cover page of the SEQUEL/DML paper. It was 24 pages long. These were twin papers in our original estimation. We wrote them together and thought they were of comparable value and impact. But what happened to them was quite different. The DDL paper got rejected by the IFIP Congress; Ray didn't get to go to Stockholm. I still have that paper in my drawer; it's never been published. The DML paper did get accepted at SIGMOD. Several years later I got a call from a guy named Larry Ellison who'd read that paper; he basically used some of the ideas from that paper to good advantage. *[laughter]* The latest incarnation of these ideas is longer than 24 pages long; it's the ISO standard for the SQL language, which was just described last week at SIGMOD by Nelson Mattos[34]. It's now about 1600 pages.

Jim Gray: It's two large binders over there [on the artifact table].

Mario Schkolnick: Don, I remember you used to tell that Larry Ellison had called you and asked for the error codes; what error codes would IBM be using? He wanted to be compatible.

Don Chamberlin: Larry called up. Larry's company in those days was not called Oracle. His company's gone through two changes of name. The original name was Software Development Laboratories. He had heard about the System R prototype and he wanted to make sure that his product was fully compatible with it, right down to the error code values. We went and asked Frank, "Can we give our error codes to this guy Ellison and Frank said, "No – those are IBM Confidential."

Franco Putzolu: That was the only part that was confidential.

Mike Blasgen: You know that whole thing is sort of interesting. When we submitted the TODS paper[35], one of

---

SQUARE and SEQUEL" *Proceedings of the AFIPS National Computer Conference*, Anaheim, CA (May 1975) page 447.

[31] P. Reisner. "Use of Psychological Experimentation as an Aid to Development of a Query Language" *IEEE Transactions on Software Engineering*, Vol. SE-3 (May 1977) page 218.

[32] D.D. Chamberlin and R.F. Boyce: "SEQUEL: A Structured English Query Language" *Proc. ACM SIGMOD Workshop on Data Description, Access and Control*, Ann Arbor, Michigan (May 1974) pages 249-264. Note that the reference in the TODS paper says SIGFIDET instead of SIGMOD.

[33] R.F. Boyce and D.D. Chamberlin, "Using a structured English query language as a data definition language" IBM Research Report RJ1318. San Jose, California (December 1973).

[34] Nelson Mattos. "An Overview of the Emerging Third-Generation SQL Standard" *SIGMOD '95 Tutorial*.

[35] M.M. Astrahan, M.W. Blasgen, D.D. Chamberlin, K.P. Eswaran, J.N. Gray, P.P. Griffiths, W.F. King, R.A. Lorie, P.R. McJones, J.W. Mehl, G.R. Putzolu, I.L. Traiger, B.W. Wade, and V. Watson. "System R: Relational Approach to

the referees said that we ought to include the SEQUEL BNF, which we did, but it wasn't in the paper that we originally submitted. Its inclusion was insisted on by a reviewer and demanded by the editor, and so we put it in even though we thought it was kind of … whatever. I think the common wisdom in the world for many years was that we shouldn't have done that; we should not have put it in because that was sort of too much detail, made it too easy for copycats to copy it. I'm not sure this is correct, but …

Jim Gray: What was it that you put in?

Mike Blasgen: BNF – the syntax. No, the semantics were in the paper; that wasn't changed – we always described it. But somehow the details of the syntax … Leonard, for example, many years later felt that was a *big* mistake; we never should have done it.

Franco Putzolu: Later on I thought that publishing everything was a big mistake.

Josephine Cheng: Only you should have patented it before you published.

Mike Blasgen: People should know that patents were basically prohibited. Patents at this time were prohibited by the company and the Supreme Court. Software patents.

Franco Putzolu: I remember until 1979 we were publishing everything that would come to our mind, either implemented or not implemented, or dreamed of; and then all of a sudden there was a barrier.

Mike Blasgen: Right, somehow we decided maybe we could make some money out of this thing. Actually, that's a compliment, right?

We put out a big press release in 1975 or so associated with the kicking off of this. And that was suppressed by GPD[36]. They wouldn't let us put out the press release; do you remember that?

Irv Traiger: I don't.

Mike Blasgen: We had a bunch of paper work. Actually Sharon got involved. *[laughter]* My wife was the lawyer and she helped them suppress it.

Bob Yost: Do you think this would have been anywhere near as successful if IBM had just held it inside? I don't think so. I don't think it would have gone anywhere near as far.

Franco Putzolu: Well I think the critical thing was the fact that it was adopted by SQL/DS and DB2 – not that much that it was popular in universities.

Mike Blasgen: I used to talk a lot about this. I was kind of a spokesman for System R for a long time and a lot of people inside IBM asked that question. My answer was exactly what Yost said, which is that if we had not published those papers it would have failed. Now the reason it would have failed is that IBM would have ignored it.

various: Yes.

Mike Blasgen: No, it's clear that if you could change history and not publish all those papers and know that you were getting SQL/DS and DB2 out, then we would have been better off not to have published the early papers. But I'm convinced that the only reason that anybody cared … well, Jolls will say something maybe about this. Actually it's too early for your time; your time will come. But I'm convinced that publication was the right thing to do. I know a lot about this because I worked on RISC. I was the manager of the 801 project, too. The 801 project did not publish anything, and it was much harder to get it out. It was much harder to get IBM to do something about it. We had to transfer it to Sun. SPARC was the first highly popular RISC, and it was only after Sun went to RISC that we could wake IBM up to the opportunity here.

Tom Price: Was it only after Ellison started doing Oracle that DB2 …

Mike Blasgen: No, Ellison was not a factor in SQL/DS and I don't know about DB2.

C. Mohan: No, I was told that SQL/DS came out after Oracle came out.

Mike Blasgen: Oh, that's true, but that just shows how long it takes IBM to do something.

Irv Traiger: So thinking back to the task force days of System R, which wasn't named System R yet, there was this notion of getting the Phase Zero prototype going, that Don talked about. It was understood that GAMMA-0 and XRM and other systems might not be the right platform. They all had a funny characteristic – all of them. None of them stored the values in the tuple. They all stored 32-bit things that would point to the values. This was in the days of small disks and small memory. The concept was that if somebody was a programmer or lived in Poughkeepsie, you didn't want to have to store "programmer" or "Poughkeepsie" more than once. You'd have these classes of names of

---

Database Management" *ACM TODS 1*, 2 (June 1976) pages 97-137.

[36] GPD stands for General Products Division, which operated the San Jose facility hosting the research lab.

things, like names of cities, or names of job titles, or things like that – people's names. You'd store pointers to an element in that class for these variable-length strings. All of them did this; all of them. RAM was binary; a tuple-id, and a pointer to a thing, and a pointer to another thing. If it fit, great, but very, very few things fit. It became clear pretty early that, what if you're just going after one tuple? You know, "Tell me about Mohan, in the Employee file". The overhead would be incredible because you'd be chasing this pointer and chasing that pointer, so why not just store the stuff right there, which was being done anyway in VSAM[37] and IMS and DBTG.

So we came to that realization pretty quickly, and then, again in task force mode, which can kind of wear you down after a while, we came to this other notion of an intermediate level called the SLI: the System Logical Interface. This would be set-oriented query, but I think only on one index and one field and one table. Somehow SEQUEL would translate down to these smaller set-oriented things and paste together complex queries. This idea was something that my group, which was just getting going, was going to work on while Don and Ray and Paul Fehder and Morton Astrahan worked on the Phase Zero prototype. But none of us really liked this SLI thing, so that kind of petered out.

Something else was going on around then that helped it peter out: I got a kind of co-conspirator, Franco. Franco was brought out from Yorktown as part of this Leonard Liu package deal with Gomory[38], on who would come out. He was not supposed to work on this System R stuff. Ed Altman was one of the principals in the Mike Senko department on the DIAM project, and he was becoming a second-line of various other groups in Computer Science. I think Franco was brought out to do a physical database design tool with Altman …

Franco Putzolu: Yes, something like that; I never figured it out.

Irv Traiger: … maybe C.P. Wang was heading it, who had come from that Senko group. It was very delicate how Leonard would balance the skills across the Altman bunch and the Frank King bunch, because he really didn't want to look like he was taking advantage of the old DIAM people and favoring Frank King. So some of the strong people who came out were directed to the Altman side, including Franco. One afternoon, Leonard said to me I should go talk to Franco, and I didn't know why he wanted me to do this; he was just being kind of coy. And it was clear that Franco was a very, very perceptive guy. He understood what database people were doing back then and he really cared about applications. He would read these weird little papers on …

Franco Putzolu: I studied IMS, among other things; I even installed IMS in Yorktown once.

Irv Traiger: So it occurred to me that this SLI thing really was a bad idea, and we just needed somebody with a bit more practical insight, so I talked to Frank King and said we ought to get Franco. But Frank King didn't want to touch this situation because of this balance of power thing. We somehow made it happen.

Kapali Eswaran was hired in around this time, and I believe he was maybe reporting to me, but helping these folks in the Phase Zero prototype, putting in consistency constraints and triggers, which as noted before have only recently made it into the IBM product line. There were other things going on, too. We were working on concurrency, trying to add that, because none of these early systems had concurrency. If they did, it was by accident. I had done some early stuff in GAMMA-0, Jim Gray was very interested and he was doing some things, and Raymond came up with this gleam of this idea of what became called predicate locks, where since you're querying sets, why not lock sets: the most natural thing you could imagine. And that was consistent with what we had finally after a struggle figured out about authorization and views. Instead of authorizing columns of a table, just make it a view of those columns and authorize that. Kapali heard about some of these predicate things, and he went off and worked on predicate locking as well, and we began to also understand that transactions were like logical units, like all or nothing.

Bruce Lindsay: There's a great line in this paper I found in Jim's box about predicate locking. Little paragraph says, "The overhead and complexity of constructing the predicate, testing the predicate, and scheduling the predicate terrifies both Morton and Franco. It merely scares the rest of us." *[laughter]*

Irv Traiger: There was one short period where we thought that predicate locks were the right approach and, although we weren't saying it, that would give you this notion of a serial schedule, you know, logically equivalent to a single-user system. But it wasn't real crisp yet. I remember another afternoon, I was sitting I think in Jim Gray's bean-bag chair. He had this small office and this huge bean-bag chair, and a regular chair. We were talking about what all of this meant. Marc Auslander was visiting you that day. He was just kind of wandering around, sort of looking over our shoulder, and suddenly Jim began to better understand what serial schedule meant, why it was important, and why maybe predicate locks had nothing to do with that. But they sort of helped us to get there. He referred to a paper by

---

[37] VSAM stands for Virtual Sequential Access Method.

[38] Ralph E. Gomory was Director of IBM Research from 1970 to 1986.

Donovan that had to do with something like serializability on cache operations. This was kind of a stretch, but that's where consistency and serializability happened, as I recall. It was there. Does that ring a bell at all, Jim?

Jim Gray: Yes. Also Karp & Miller's asynchronous communication[39]. They were interested in determinacy and we were just looking for consistency. We wanted *an* answer, they wanted the *right* answer.

Irv Traiger: That's where that happened. Versioning was … shadows were a very strong notion in XRM, as I recall, which Raymond brought to us from Cambridge. He had transferred out pretty early in the cycle. So is there anything else I can think of? Recovery, logging, we wanted to go for tuple-level locking, so we realized that shadows weren't going to cut it, because they were at a page level, so we introduced this intricate combination of record-level locking and logging and shadows, which actually works[40].

Franco Putzolu: Kind of. *[laughter]*

Irv Traiger: One of the very strong notions is that we wanted to support all the data models, so the RSS had links, basically pointer chains, and the idea was to support hierarchies, networks, and relational. Over time we gave up on this noble ambition, but it got resurrected again at Santa Teresa. It was very interesting, it was more this not knowing where data was going to go, wanting to build the universal low-level thing, whatever might be on top of you.

Franco Putzolu: Yeah, I would say that there were two really important points. One is that RSS just existed. We accept as a given that you have to split the system into a low-level component and a high-level component. But only the systems with System R ancestry have this clear split. Many of the other systems don't, and I think suffer from it.

Irv Traiger: Yes, it wasn't clear that you should split it, or how to split it. As I mentioned, we struggled ourselves with this SLI thing that was a kind of medium level.

Franco Putzolu: The split was done at the right level. And the other major point was the emphasis that multiple high-level subsystems would use the common low-level engine. This approach was tried by all the systems that came after System R; all these attempts failed.

Pat Selinger: I remember Franco that you had led at least one study that I recall on how to map IMS – every construct: logical deletes, and sparse indexing, and all kinds of different things – all into the RSS level.

Franco Putzolu: Yes, what a waste. *[laughter]*

John Nauman: But you kept doing it, Franco.

Jim Gray: So somebody can appreciate that all this was happening in the background of something called FS[41] – Future System – and the people in FS thought that it was an incredible waste of research energy to be working on this relational database stuff. They were working on GRID or some project that was much more sophisticated and advanced than anything we were doing – it had a GUI and was wonderful.

John Nauman: Actually, it was three systems: it was supposed to support relational, and the network CODASYL stuff, and flat files – it was going do everything. But that was all dropped.

C. Mohan: But part of that came in System/38, right? They allowed file system access as well as high-level query access to the same data, which is still there in AS/400.

Morning break

Mike Blasgen: This is a brochure of the [IBM San Jose Research] Computer Science Department that we put out I think at the end of 1978 or the beginning of 1979. It has a picture of the whole Computer Science Department in it – all the people who are in this room are in this picture. It was taken on the back steps outside, behind Building 28. Eric Carlson, T.C. Chen. It's interesting to see who has the longest … there's also a list of publications here. The answer to this question probably hasn't changed. It's probably the same this year as it was that year. It's the publications of the Computer Science Department 1977-1978. Now what name do you think has the most publications?

various: Astrahan. *[laughter]*

Mike Blasgen: To my pleasure, Blasgen has two, so I'm not in the noise. Chamberlin has: "Data Base System

---

[39] R.M. Karp and R.E. Miller, "Properties of a Model for Parallel Computation: Determinacy, Termination, and Queuing," *SIAM JAM 14*, 6 (June 1966), pages 1390-1411.

[40] J. Gray, P. McJones, M. Blasgen, R. Lorie, B. Lindsay, T. Price, F. Putzolu, I. Traiger, "The Recovery Manager of the System R Database Manager" *ACM Computing Surveys 13*, 2 (June 1981), pages 223-242.

[41] G. Radin and P.R. Schneider. An Architecture for an Extended Machine with Protected Addressing. IBM Poughkeepsie Laboratory Technical Report TR 00.2757 (May 21, 1976).

Authorizations" in *Foundations of Secure Computing*[42]. It's probably not even on his CV. Anyway, that's Chamberlin, Gray, Griffiths, Mresse, Traiger, and Wade. But the answer of course is Ron Fagin, with about ten publications. The theory guys always beat us systems guys. And Frank King wrote "The phonetic encoding of word-components for the computer input of Chinese characters". It's his only publication in here. So I'll pass this around and people can look at it. This is really a great little thing. It's got a lot of good stuff in it.

We're back on track. Unless there's some opposition, at least part of this afternoon's session is going to be devoted to more of this pre-1982 material. Any comments or suggestions? Because we're not done; in fact we haven't mentioned the joint studies; we haven't mentioned the interaction with Santa Teresa. And if we're going to lunch at noon, we only have 40 minutes. We have videotapes that we're going to show you. We'll show you a little snapshot of those just before we go to lunch. OK? Some of you may have seen little snapshots last night. We have videotapes from the era, from 1979, so you can see what you looked like if you were on tape.

This [begins showing System R slides] was the kind of talk that went around at the time on System R. Ease of use. See – we cover all the bases; everything important is there, right? Did we miss anything? We got PL/1, we got VM and MVS; there were no other operating systems of importance …

Mario Schkolnick: And it's all under ease of use.

Mike Blasgen: So of course we have all the people who make more than their manager. We only had one database. Here's our advertisement for SEQUEL. Here's "Find the names and salaries of employees who make more than their manager" – right there, number four or something. "Give a $1000 raise to all programmers in Department 50" – it used to be K55, is that what it was?

C. Mohan: It's still K55.

Mike Blasgen: We compile. Now one of the things I want to talk about is Raymond's discovery of compilation. He gave a talk in the cafeteria conference room in Building 28 once where he said we can compile and it will run faster. I remember that meeting very well because we basically threw out a huge pile of code and started all over again because we realized we were doing it wrong. And we had to change all our presentations. Like the TODS paper is all wrong, as an example, because we changed the way it worked.

Here's an application program. This is what SEQUEL looked like, in case people don't remember. The dollar signs were required because we had a scanner that scanned PL/1 and we didn't want to fully parse PL/1 so we just wanted to look for illegal delimiters. So we just tokenized it and looked for illegal symbols. The dollar sign is illegal in PL/1; you can't say $DECL. This is "Let a cursor be"; OPEN; FETCH; System R codes; the codes of course are what Ellison wanted. There's the same picture as you saw in the overheads. MVS still in parentheses. That dates it, because MVS came out of parentheses in 1979. Execution. Oh yes, so we load the … Performance … More nuts. Oh, transactions. Why did this project turn into a research project on transactions, and locking, and logging? Actually the fundamental contributions of the work turns out to be the stuff that hardly appears in the papers, which is all the work that Jim and his colleagues did.

Oh, here's a history of INGRES, PRTV, QBE, and all the System R … here's R* … so that dates this. This is actually a presentation I gave in Atlanta at SHARE right after the SQL/DS announcement. So here System R begins; TODS paper in 1976; first install – that was at Pratt & Whitney in 1977. First System R install was 1977. So here's the definition of all the papers that we wrote. We called it full function, but hardly the full 1600 pages that is required today.

Jim Gray: Oh, I like that slide.

Mike Blasgen: … Oh, here's the "Indian's salary's greater than the chief's salary". NULL. Support for NULL. I remember going to talk to Ted Codd about support for NULL. *[laughter]* We couldn't get together – the RSS and RDS couldn't get together on that, so finally the RSS said, "You guys decide; we'll support whatever." We supported a general NULL.

Franco Putzolu: I didn't like NULLs in those days.

Mike Blasgen: Right, you sure didn't. But we supported them; RSS supported NULLs, in a particular way. We allowed you to specify a NULL symbol, and then we would insert that wherever it was.

Jim Mehl: I still remember a memo that Franco wrote addressed to "NULL theologians". *[laughter]*

Mike Blasgen: The non-nerds in the room don't even know what we're talking about. If you don't know somebody's age …

Bruce Lindsay: I just found "Revisions to the NULL Memo".

Mike Blasgen: There were great controversies about NULL. Just to give you an example, if you don't know

[42] R. Demillo, D. Dobkin, A. Jones, and R. Lipton, editors. *Foundations of Secure Computing.* Academic Press, New York (1978) pages 39-56.

somebody's age, but you know a lot of other things about them and you want to record it in a database but you don't want to put in their age. So you leave it blank, but now you sort, say, on age. Where do you want that person's age to come out: at the beginning or the end or the middle? This is NULL theology. There are the people who want to eat the egg from the big end and people who want to eat it from the little end. And then there are people who want to eat it from the middle.

Oh, gee, views and authorization. That's good; didn't know we had all that. Can't read that. I think I had a big screen for this presentation. Oh, accounts. I went to accounts payable. We went to ET1 – Eagle Transaction 1, which is now called Debit-Credit and now called TPC/A, but it was at that time called Eagle Transaction 1.

C. Mohan: Where did the name come from?

Mike Blasgen: Eagle was an IMS successor; it was going to do everything. And they were very worried about path lengths. So there had been something in IMS called TP1. But TP1 was more of a general characterization; ET1 was a specific program. And then Jim wrote all this stuff down in an article that he published in Datamation. It had Anonymous et al. or something like that as the author[43].

Jim Gray: Actually in about 1972, General Automation beat IMS at the Bank of America for the automated teller system, and we saw the attack of the killer minis that were going to wipe out all the mainframes.[44] We had maybe three or four years before the company was going to go out of business. And that was some of the stuff that was driving FS; some of the stuff that was driving …; we thought the computer industry was going to change, really dramatically with the minis. And the benchmark that B of A used was canonized as TP 1, 2, 3, 4, 5, 6, 7. TP7 was a minibatch; TP1 was a very, very light debit-credit transaction; and we ended up just using TP7 and TP1. IMS Fast Path was built to run TP1 well and that was the whole reason for doing IMS Fast Path. TP1 was recoded – it was a set of IMS calls – it was recoded for the VSS interface, which was the early name – the Andy Heller name – for what became DB2 eventually. It was called Eagle in one of its many incarnations …

Mike Blasgen: To call Eagle the same as DB2 is misrepresenting history quite a bit. Eagle wasn't the focus on relational; it was kind of an IMS successor.

Franco Putzolu: Well, we all came from amoebas and …

Mike Blasgen: Oh, I see, it's all the DNA; a tree is the same as a human, right? With a few changes in the genes. Well, Eagle was a tree.

C. Mohan: Wasn't the data manager the same?

Josephine Cheng: Yes.

Jim Gray: It was the same project.

Mike Blasgen: OK, so Eagle Transaction rewrote …

Jim Gray: They rewrote in VSS terms and …

Mike Blasgen: Well, I believe it's possible, likely, that I'm the first person ever to write ET1 in relational. Because I decided we had to have it and everybody else was busy. At that time I was a second-line manager. So I wrote it. And I remember taking some liberties with it.

Tom Price: Everyone who's ever written it has probably taken some liberties.

Mike Blasgen: … It might well have been Bob Selinger that helped, I mean I know you were working on tracing. Did you wind up tracing ET1?

Bob Selinger: Well, we called it SR1; it was the relational equivalent of ET1.

Mike Blasgen: All right, the details, which we can talk about at lunch, had to do with whether you had to keep the account records sorted by account. Or whether you can just insert the records into the table and then, at the time you print the bill, sort them then. And I argued for sorting then, because relational doesn't really have a notion of sorting order anyway. But then there was this argument about whether it was equivalent or not, because IMS was keeping it, if you will, more organized.

So what have we got? OK, you get the idea. There was this presentation that we took everywhere. This went to six GUIDEs and thirteen SHAREs and database conferences all over the world. I have a proceedings of one of the last of the biggies, which was in 1983, at Wang Institute, the Eastern Computer Science Colloquium or something, and Jim and me and Mike Stonebraker and Ted Codd – oh, gee, it's just a long list of neat names … Non-procedural, lots of optimization. This was one of my favorite talks. It talked about how there were many ways to do these things and how you would need an optimizer to make your choice. … I love this: avoids exponential growth of optimization time, and N-way join by pruning. Aren't we good. And here's …

Bruce Lindsay: Eight megabyte database; whoa!

---

[43] Anon et al. "A measure of transaction processing power" *Datamation 31*, 7 (April 1, 1985) pages 112-118.

[44] J.R. Good. "Experience with a large distributed banking system" *IEEE Computer Society on Database Engineering 6*, 2 (June 1983).

Mike Blasgen: We had an eight megabyte database! *[laughter]* … OK, so you get the idea.

So I'll put in some of my own reminiscences here for a minute. At one point, Irv and Frank decided to change the management structure, and in something that I've never seen happen before, Irv worked for me and I became Irv's manager. We switched places. So I became the manager of the RSS and Irv became incredibly productive. *[laughter]* And I became incredibly non-productive. I spent a lot of time making those slides. And then Leonard was offered a job back working for Bob Evans in New York, and that was onward and upward. He became a director at that point. Frank was named the manager of the Computer Science Department. That left open the job of manager of database systems, which Frank had been, so I took that job. That was in, like, the end of 1977 probably. And I stayed in that job until July of 1979, when my wife was offered a job in Washington DC, and I thought that I should follow her, rather than be left behind and not have a marriage. So we moved to Washington, and Robin Williams succeeded me in the job of manager of Database Systems. And then shortly thereafter, Frank King left, also to go back to work for Bob Evans, and Abe Peled came. And then the department changed quite a bit; it grew, and it became much more diverse. We were just talking a minute ago about the fact that as System R became more successful, it accreted more. I remember Don Slutz, for example, was not working on System R in 1974 or something. But you joined, what the RDS in …?

Don Slutz: 1975.

Mike Blasgen: … 1975, right. You heard the story about Franco in 1974. I mean he was grabbed out of some other thing, and then Don Slutz. I think that by the time of the end, if you count, I mean Juan Rodriguez-Rosell was a performance analysis guy interested in whatever – operating systems performance – but we converted him into a System R performance person. And so by the time we were done we might have had as much as half the department working on – not necessarily System R, but things related to System R. Mario and Paolo Tiberio were working on a tool to do database design and that was a database design tool that was keyed off System R. Several of the projects that had preceded System R, sort of like EXPRESS[45], sort of became smaller projects; some of the people left.

I tell you, my artifact [the CS brochure] is winning. Because during the time I've been talking, it's gotten to the third person, and he's whispering, "Look at this!"

The joint studies were brilliant in the sense of forcing IBM to move; I'm not sure it was so important for what we learned. Frank had the idea of doing joint studies; actually, I think this was Ralph Gomory's idea, or Leonard's, or somebody's; Ralph's, I think, as much as anybody. So we established a relationship initially with Pratt & Whitney Aircraft, then with Upjohn Pharmaceuticals in Kalamazoo, Michigan – Pratt & Whitney Aircraft is in East Hartford, Connecticut. And finally, after about a one-year delay, a major relationship with Boeing Aircraft, in Seattle, Washington. We have the reports; in fact I have copies of reports and there are copies up there [on the artifact table]. I think Bob Yost brought every single report we got. It's not quite the SQL standard – I mean nothing is that big – but it's a lot of paperwork that was generated. I was thinking that maybe Don could talk about Pratt & Whitney and Upjohn. Are you the best person to talk about that?

Don Chamberlin: I'm one such person, and I'm sure there are others, too.

Pat Selinger: Jim Mehl was the chief contact for Boeing as I recall.

Don Chamberlin: Jim, do you want to tell us about Boeing?

Mike Blasgen: Well, historically, it came third.

Don Chamberlin: All right. The initial joint study was, as Mike says, at Pratt & Whitney in Hartford. I'm not sure any of these joint studies really exercised all of the neat stuff that we had to offer, like concurrency and transactions and locking and different degrees of consistency and all those different things – they really didn't care about any of that stuff. The applications initially in Pratt & Whitney – they were a manufacturer of jet engines, and they used System R for inventory control of their parts and supplies that they were using to build jet engines. The Upjohn Drug Company in Kalamazoo used System R to store the results of clinical experiments that they were using in support of their drug applications to the FDA. The thing that I remembered the most about these joint studies is that we got a lot of trips to beautiful Hartford, Connecticut and Kalamazoo, Michigan, and that was kind of neat. We got factory tours; we got a tour of the jet engine factory in Hartford; we got to go through all the machine shops and watch them building jet engines – that was kind of neat. They told us how at every stage of building the engine they would weigh it very carefully to see if they had left any extra parts inside.

Tom Price: I remember when we went to Pratt & Whitney the first time, we showed them all the system mods that they needed to put on their VM systems so that they could run it, and they already had system mods on all those same lines – local mods. It was a real mess.

[45] N.C. Shu, B.C. Housel, R.W. Taylor, S.P. Ghosh and V.Y. Lum. "EXPRESS: A Data EXtraction, Processing, and REStructuring System" *TODS 2*, 2 (June 1977) pages 134-174.

Don Chamberlin: I remember in particular a wonderful trip to Kalamazoo to visit the Upjohn people. They had a place that they called their homestead, which was a beautiful Victorian mansion on a huge plot of land outside Kalamazoo. It had a pond and a greenhouse and all sorts of very wonderful accommodations that they kept there for visitors. Tandem bicycles parked around for people to ride around and have a good time. They put us in a room where there was one whole wall completely filled with different kinds of liquor. We asked them if we could take home anything that we didn't drink. That was a nice trip. *[laughter]*

A bunch of things were happening at about this time that I think we ought to mention just in passing. One was that we had to change the name of our language from SEQUEL to SQL. And the reason that we had to do that was because of a legal challenge that came from a lawyer. Mike, you probably can help me out with this. I believe it was from the Hawker Siddeley Aircraft Company in Great Britain, that said SEQUEL was their registered trademark. We never found out what kind of an aircraft a SEQUEL was, but they said we couldn't use their name anymore, so we had to figure out what to do about that. I think I was the one who condensed all the vowels out of SEQUEL to turn it into SQL, based on the pattern of APL and languages that had three-lettered names that end in L. So that was how that happened.

A couple of other interesting things happened about that time, too. Our famous paper that got published in TODS: the second issue of TODS had a paper on System R. And there's a story about that, too. I want to prove something to you by showing you a foil here. If you've ever seen a reference to that paper – that TODS paper – it says the title of it – this is the famous fourteen-author paper; everybody that had ever attended any kind of a System R meeting was included as an author of this paper – so this is the cover page of the manuscript that we sent to TODS for the fourteen-author paper. If you've ever seen a reference to this paper, its title as it got published was "System R: Relational Approach to Database Management" – it didn't have the "A" in it. When we wrote the paper, it said, "System R: *A* Relational Approach to Database Management"; when it got published, the "A" went away. And the reason for that is because when the galley proofs came back from TODS, they sent them back to us for proof-reading, and all of the fourteen authors were alphabetical, and Astrahan of course was first, so a lot of our papers are Astrahan et al. The penalty that Morton had to pay for that was, he had to proof-read the galleys. So I gave the galleys to Morton, and this was a pretty long paper, he had a lot of proof-reading to do, and he was pretty busy. So he did a pretty good job of proof-reading, but he didn't proof-read the title. So that's what happened to the "A".

Another thing that happened at about that time was that some technical problems came up that got dealt with and solved, and I thought they were kind of interesting and somebody ought to talk about it. I think somebody ought to talk about the Halloween problem. Pat, you had a lot to do with the Halloween problem; do you want to talk about it?

Pat Selinger: It happened in I guess, was it 1976?

Don Chamberlin: 1976 or 1977.

Pat Selinger: I'm having a little trouble remembering this, but we had exercised the "person who earns more than their manager" query to death, and finally got to the point where the optimizer was choosing indexes sometimes to implement this query and it happened to think that the Salary index was a pretty good index to select for this. And having selected the Salary index for the first time in us testing out the optimizer, we ended up discovering that this query didn't stop. Because we were using the Salary index to go after the Employee table and we were also updating it, and Don Chamberlin kept getting more and more raises. Which made him very happy, but it made us optimizer folks a little bit uncomfortable. So Morton and I sat down and discovered this and analyzed what was going on, and came to one of your RDS meetings and it happened to be on Halloween. So we ended up telling the group about this and consulting the general wisdom to figure out what in the world we ought to be doing about this thing. As we talked about it, it came to be known as the Halloween problem. And I think it's still kept that title to this day.

Don Chamberlin: It's famous in the industry – everybody knows the Halloween problem. And it happened to be discovered on Halloween. The query was … they had submitted a statement that was supposed to give a ten percent raise to every employee who earned less than $25,000. This query ran successfully, with no errors, and if you examined the results, all the employees in the database earned $25,000, because it kept giving them a raise until they reached that level. So that was how the Halloween problem got its name.

Pat Selinger: An interesting footnote is that we just discovered another one of these as sort of a variation on that, in the latest work that we did having to do with referential integrity and things like that, where the referential integrity relationships were going to trigger off the same kind of non-stop behavior.

Mike Blasgen: It's interesting because all these odd-ball things had names: there were phantoms, and there were other things, and those had to do with names that were somehow representative of what you were observing, right? So the phantom was because it was something that was sort of there, but not there; the name was descriptive. And this was called the Halloween problem not because it surprises you, or it's spooky, or trick-or-treat or anything; this is because it happened to be discovered on Halloween day. But

I think most people think it's the other; I think most people think it's called Halloween because it's so surprising. But it's not.

Don Chamberlin: Here are a couple of more artifacts from the joint-study days. I wrote most of the manuals for the users at our various joint studies to use and we designed a nice logo. And Jean Chen helped us make these nice binders that a lot of you still have. When we went to Upjohn, like other places, they gave us a factory tour. We got a factory tour of Boeing; we got to see them putting together 747's and at Upjohn we got to see them making vitamin pills. They would give the vitamin pills away for free in the cafeteria and they all gave us this nice sign. This sign says, "Keep the quality up. W.E. Upjohn." And here's a picture of a guy squashing a pill with his thumb. It says, "Upjohn: originator of friable pills." "Friable" means sort of squashable. So W.E. Upjohn was the originator of friable pills and that's the heritage of the Upjohn Company, and that was apparently what he said. You know, Thomas Watson has a lot of famous sayings, you know, "Respect the individual" and stuff. What they say at Upjohn is "Keep the quality up." That was their slogan.

In talking about these users, we always remember Upjohn and Boeing and Pratt & Whitney, but I just wanted to mention the fact that we had a lot of other users, mainly inside of IBM, as well. IBM Owego was using System R to build attack helicopters I think. There were several groups at STL[46] that were using System R for different things; there was a guy there named Gary Haas. Gary and Frank Nargi??? – do you remember those guys?

Bruce Lindsay: SREDIT.

Don Chamberlin: Yes, they built a sort of early GUI on top of System R, called SREDIT[47]. At [IBM] Poughkeepsie, there were some folks using System R in a kind of a design-automation system. At Yorktown, there was a guy named Fred Damerau who was doing natural-language queries. He had a project named REQUEST that was based on System R. Most of the IBM Scientific Centers had System R installed. Alex Hurwitz installed it at Los Angeles; Yoichi Takao installed it at Tokyo; Jean-Jacques Daudenarde installed it at Paris.

Jean-Jacques Daudenarde: We also had a joint study with a company who was developing some helicopter design based on two-column relations in System R; exclusively two column relations.

Don Chamberlin: The system wound up being installed in Madrid, Heidelberg, Rome, Cambridge; Scientific Centers all over the world.

Bob Yost: Do you have any idea when the last System R site went down? I think Upjohn used System R even after SQL/DS came out because they didn't want to pay for it. If you used System R, then you came to have it free for a long period after that. So I think they used it for a long time.

Don Chamberlin: I don't know how long that went on.

Bob Yost: They were running that on a Model 145 with one megabyte of memory. I saw that and I said, "My God, I'm trying to put DB2 in my eight-megabyte machine and it hardly fits at all."

Don Chamberlin: So in looking back on this, one of the things that I marvel at is the impact that this work had when it was really, by today's standards, very small according to certain measures. And I brought along a couple of foils that kind of show you a measure of things. This is a profile that I drew up for Frank King one time that shows the number of people that were involved in System R at various stages of its life. From 1973 to 1978, this was kind of the profile. We started out at about ten or eleven people. And this was when the Phase Zero prototypes were installed, and this was the different installations of System R at the joint study sites. As you can see, we never had twenty people in System R; probably the average was around fifteen. And after 1977, it fell off to half that. So the area under that curve was not a lot of manpower by Microsoft standards. But we had a pretty big impact from it.

Bruce Lindsay: By Santa Teresa standards.

Don Chamberlin: As far as the code size of System R is concerned, believe it or not, it wasn't very big. The RDS was mostly written in PL/1 and had 38,000 lines of PL/1, and about 9,000 lines of assembler. Now 38,000 lines of code isn't a lot; I mean, it's pretty hard to find any kind of a product that small. The RSS was written in PL/S and had another 35,000 lines of code. So add this up, maybe 80,000 lines of code and that was System R for you. It's not a lot for what we were able to accomplish with it.

This was the size of the load map. The RSS took a megabyte and the RDS took another 1.2 megabytes on top of that. That's all there was in the systems that we installed at the joint study sites. So it wasn't very big, either in terms of its code size or the people that wrote it.

I kept track of the two lists, called the bug list and the wish list, during the time that we had the joint studies going. We would have quarterly reports at each of these joint studies and they would wish for things and I would write them on the wish list; we mostly didn't implement any of them. At the end of the project, I think there were

[46] STL stands for Santa Teresa Laboratory.

[47] Pronounced Shred-it.

something like 160 wishes that were open. The bugs we tried to fix, though, and I've got some statistics on this. This is where the bugs came from. Over the course of the project, we had 251 bug reports. We found most of them ourselves, but Boeing found a lot, too. These were the number of bugs that were found at different joint study sites. This is what happened to them. Out of those bugs, we fixed 71%, we couldn't reproduce quite a few, and some of them we rejected, and so on. So we did our best to …

Jim Gray: Ten percent are features, I heard.

Don Chamberlin: Yes, we declared some features. And this was the hero list. This is the people who fixed the most bugs in System R. And Wade is on the top of the hero list, and Don Slutz, Jim Mehl, Irv Traiger fixed a lot of bugs.

Mike Blasgen: The RSS didn't have any bugs. *[laughter]* No, it's true, the reason is because much of the RSS was written by Franco. No, it's really true; Franco never wrote a bug. Except for one, right, Bruce? Did you find one?

Bruce Lindsay: One.

Mike Blasgen: He wrote about half of RSS, and I think we found one bug. And that was after nine years.

Brad Wade: I remember index management, though, as being a trouble … *[laughter]*

Franco Putzolu: How does the wish list compare with what was implemented afterwards by other systems? Did you ever look at that?

Don Chamberlin: I haven't really looked back and analyzed that. I couldn't tell you about that.

Pat Selinger: Do you have a copy?

Don Chamberlin: I have a copy at home; I didn't bring it.

Bruce Lindsay: We should send it to the SQL 3 people. *[laughter]*

Roger Miller: Our wish-list is more like a database.

???: It fits in eight megabytes?

Roger Miller: No, it won't fit in eight megabytes.

Don Chamberlin: Raymond, did you want to talk some more about the compiler and interpreter issue?[48] You had a lot to do with that.

Raymond Lorie: I don't remember it. *[laughter]* I must say I tried to locate my foils that I used when we had a little meeting but my database system is not up to that task. I remember one meeting; I believe Don was there, and Irv, and Morton, I think, and Frank King, of course. I showed how a compiled SQL program would simply comprise a few assembler instructions on top of the RSS. It was amazingly short. It didn't stay that short, but short enough to pursue the idea. Now of course, because we started from an interpreter, we went all the way, and compiled into machine language. Later on, because I think Franco didn't like that, we came back to well prepared tables, rather than code; but of course the idea of compilation remained unchanged, because you do the optimizer once and you package the whole thing; then you invalidate the "code" if things change that impact the access strategy. So that was one of the contributions to the system. It was a good example of how you can change direction in a group and convince people to follow it. It was a good experience. It also brought me into the RDS, where I spent several years, which I enjoyed very much.

Don Chamberlin: I think that was one of the key things that made System R a success: Raymond's idea to compile rather than interpret our high-level language. Because that was the thing that was responsible for our performance, and performance was the thing we had to prove to get relational accepted. Everybody agreed it was sort of neat, but they didn't think it could perform. And Raymond was the man that made it perform.

Mike Blasgen: Actually, this is sort of the history. You know, the first FORTRAN compiler that was ever made was probably the best FORTRAN compiler in terms of producing the fewest instructions per FORTRAN statement. Because they spent so much effort to get it all just right. For example, the reason FORTRAN has a three-way branch – IF (ABC) 1,2,3 – is because the machine had a three-way branch, and that way they could generate that in a single instruction. It's like CAR and CDR; they worked very hard on performance, from the very beginning.

Let's see, in the TODS era, I gave talks at the TODS era too, and then we had pre-optimized packages – that was some idea that we had that we wouldn't have to go through optimization. But that was before compilation, correct?

Don Chamberlin: Yes.

---

[48] R.A. Lorie and B.W. Wade. "The Compilation of a High Level Data Language" IBM Research Report RJ2598. San Jose, California (August 1979).

Mike Blasgen: Before compilation we were already worried about performance, so we didn't believe that we would fully interpret, but we had this idea of pre-optimized packages, I'm not sure if it was all worked out.

Tom Price: It was like you only had to optimize it once, but then you were still interpreting.

Mike Blasgen: Right, you'd interpret the plan.

Franco Putzolu: Weren't pre-optimized packages more like views, special …

Mike Blasgen: Oh, were they?

Pat Selinger: Yes, I believe they were just for views.

Mike Blasgen: Oh, I see it was really like combining the two, doing composition in advance. But then I remember that after Raymond's talk, and we all decided that compiler was it, we had still the question of what to do for *ad hoc* query. Because Raymond had not proved anything about *ad hoc* query; he had proven something about canned transactions, that you want to compile canned transactions. So then the question was, "What about *ad hoc* query?", and then there was a bunch of work that Raymond and Pat, or I don't know who was involved in that, but I know Pat was involved – where you did measurements of interpretation of *ad hoc* query. Then the plan became, "We'll do both an interpreter and a compiler." And that seemed like it was going to be a mess, just in terms of keeping the semantics straight, because you have two implementations of the same thing. How are you going to make sure? So then Pat was able to conclude, right, that we could do both *ad hoc* and canned transactions with the compiler.

Pat Selinger: I think Don was involved in this, too.

Mario Schkolnick: There were lots of measurements Morton and I were doing on the number of pages touched when doing an *ad hoc* query. Once we found 92 pages were hit to save touching two pages at runtime. So I remember Franco picked up the code of the optimizer and said "Well, let's figure out for a simple query how to reduce the number of pages that the optimizer was touching." So there was all that work going on …

Franco Putzolu: Yes, Ray's proposal was good. Without it, we wouldn't be here today.

Mike Blasgen: I think there are many such things that happened, that without that happening, maybe we wouldn't be here today. But I think I agree that this was an absolutely critical one. When we did do the ET1, by the way, and we wrote down the debit-credit transaction and ran it and did all the path lengths, if you count I/O as one instruction, the path length for ET1 was about fifty thousand instructions, and that was state-of-the-art. So System R, thanks to the work of Raymond, became state-of-the-art in performance, that is, there was no database system that could run any faster than System R on canned transactions, light-weight transactions, which is what you would think would be the weak link in a relational database.

Notice it is really true; I hadn't realized this, but this project changed quite a bit from the original idea of SEQUEL, which had to with vastly broadening the audience of computers to include people looking up recipes and mechanics wanting to know how to change oil in the car that just rolled in, to ET1. It just did that in a couple of years, and we all just thought it was great. And then we were worried about FRRs and SRBs; you know we wanted to do SRB scheduling, because then we could get three instructions out of our path length. Oh, and Irv Traiger went through and did all kinds of work to get eight instructions out of this, and four instructions out of that. Do you remember all the work you did in the RSS?

Irv Traiger: It was Fast-Next.

Mike Blasgen: Yes, he did Fast-Next! FNEXT[49]. It just cached a bunch of stuff in YTABLE1, right, was what it did. And we had cache-invalidation tricks. But if you didn't invalidate the cache, we had Fast-Next, and that took, oh, thirty instructions out of a NEXT, and that was the key to survival.

Franco Putzolu: We got really carried away with machine language.

Mike Blasgen: You think that was a mistake?

Franco Putzolu: Yes; on the other hand, recently I've seen some systems doing that.

Spreading the word

Mike Blasgen: That was in my era, and I remember that Santa Teresa had come around yet again to deciding to use System R for something. They said well, they liked it but they didn't want to generate machine code. They just wanted to generate something slightly higher level; trivially higher level, like symbolic assembler rather than real machine code, and then interpret that, which was what actually happened. And we fought it because, well, you can decide why we fought it; I know why I fought it, because I didn't want them messing with anything. I thought it would just be an opportunity to not do it again, to not ship anything again. I have the dates for when they were going

[49] Pronounced Fuh-next.

to ship VS QUERY and DB2 in one of the early meetings, and it was all 3/79. March 1979 was the first customer ship. GA, what we call GA.

Jim Gray: It was the 811 architecture.

Mike Blasgen: For System R architecture shipment, and of course they wound up being about three years later.

Jim Gray: But the project was called 811 because it was the ship date: eleventh month of 1978. And I think [Mike] Saranga and I … I still haven't paid him off on the bet I had that they'd never ship anything.

Franco Putzolu: Was this VSS?

Jim Gray: Yes, well it turned into DB2. There was also hardware coming with it. It was the 811 architecture, so XA was part of the package.

Mike Blasgen: Thirty-one bit addressing.

Jim Gray: This was, "FS crashed; what are we going to do? So what we're going to do is go back and do a new database system for MVS. We're going to put in the XA architecture into the 370." This was not thirty-two bit addressing, this still twenty-four bit addressing. It was horizontal.

So something that hasn't come up is that Irv was loaned to Palo Alto, I think early on.

various: STL.

Mike Blasgen: No, twice; he was loaned twice.

Irv Traiger: First Leonard suggested that Franco and I start going up to Palo Alto, which was where we met Steve Weick, John Nauman, Bob Jackson, a bunch of other people on this huge FS project. I don't know if the interactions came to a whole lot. He was trying to get us educated about real systems.

Franco Putzolu: It was a strange interaction. This was a part of the big FS effort. We were supposed to be there as representatives of relational knowledge, relational wisdom. Of course, FS was covering everything in the world, so it had to cover relational. I was very uneasy because I didn't have much experience in database in these days. On the other hand, talking a little bit with these people …

Mike Blasgen: … made you feel a little better. *[laughter]*

Franco Putzolu: I knew I was kind of flaky, but these people, they were experts …

Mike Blasgen: I worked on FS for two or three years in Poughkeepsie in Building 77.

Franco Putzolu: I worked for a year in Mohansic in FS.

Mike Blasgen: I was on it for two years. I worked for Rich Oehler, who worked for Pete Schneider, who worked for George Radin, who worked for Dick Case, who worked for Bob Evans. Bob Evans you know had basically all the development in IBM. Because the System Development Division developed all systems. So everything except typewriters, everything except Tom [Price][50]. I left FS because it was just too complicated, too hard to understand. Jim wrote a real long paper and said, "It's real attractive, but don't do it." Something like that, I forget exactly. Whatever it is, don't do it.

Jim Mehl: The thing in Palo Alto: was that called the Dawn Treader project?

John Nauman: No, that was different; that was in Palo Alto, too, but that was a different project than this.

Mike Blasgen: I have a technical notebook – the sort of slightly hard cardboard cover thing that were issued – dated November, 1974 on the cover. I have notes from all my meetings of Palo Alto with all these people, and names of people. These names are all lost. There are a few that are around. Steve Weick was one of them.

Franco Putzolu: Yes, Irv and I went for a couple of months. I don't know why we stopped; did FS die?

Irv Traiger: It was sort of creeping along. But we just didn't connect with anybody really well. We had meetings and they'd kind of pat us on the head and smile and get back to their work.

Tom Price: I remember when FS did die, and there were rumors it was dead like a year before they actually stopped the project. It was so funny – everybody working on this stuff they knew was dead, but they had to keep working on it.

Mike Blasgen: There is by the way a book that's just come out. I know Pat has a copy. It's Emerson Pugh's *Building IBM*[51]. It starts with Herman Hollerith. Actually, the first sentence is about something that happened in 1889, which is the day that the first patents were issued on Hollerith tabulators. And in there, quite toward the back, is a section

---

[50] Tom Price later worked in the Office Products Division.

[51] Emerson Pugh. *Building IBM*. The MIT Press, Cambridge, Massachusetts (1995).

on FS, which says, "It was the most expensive development failure in the company's history."[52] It's not usual to see stuff like that written about FS. We all knew that, but nobody ever actually wanted to write it down. Except for Jim, who said, "Don't do it" in the first place.

C. Mohan: Actually, the author has said that he was given complete access to all the IBM archives without any restrictions on what he was allowed to say. He didn't have to clear the manuscript with IBM before he published it.

Mike Blasgen: You know Pugh is no longer an IBM employee. So he wrote this as an independent author. He wrote three earlier books: *Memories That Shaped an Industry*, *IBM's Early Computers*, and *IBM's 360 and Early 370 Systems*, the last two of which were part of the IBM Technical History Project. He was an IBM employee and he had people working for him and relationship with publishers. And then he left IBM, but he continued it independently, and as a result had a little more editorial freedom than he would have, had he been an employee.

Jim, I think it's lunch time; what's the story on lunch?

Raymond Lorie: Speaking about the compiler, I'd like to thank Brad for the tremendous job. If you really want to know something about 360 and how base registers work, you should ask him; I think he still remembers.

Franco Putzolu: Do you remember that?

Mike Blasgen: I do remember one story about Brad's base registers and whatever. It's that the assembler sequences were in our code. So it said, "Load R1, whatever", and this would then be processed by something that would turn it into real machine code. We did a code review at Santa Teresa, because Santa Teresa was thinking about taking our code. They came back and said there are many bugs in this code, sorry many defects. And what were the defects? We were a little bit surprised, because we thought we'd coded it pretty clean. They said, "Well, you have literal references to registers. In other words, we really generated "Load *one*".

Tom Price: You didn't use EQUs.

Mike Blasgen: Right, we didn't equate anything. *[laughter]* But that's because this was the back end of a compiler. I mean, of course the back end of a compiler – I mean somebody has to decide which register it is. They were all upset, and they counted, you know, four hundred of these …

Brad Wade: There were five hundred and seven defects per thousand lines in my code. *[laughter]*

Mike Blasgen: The rule at that time was, you know, point seven or something. And so Frank or one of us turned to the guy that was in charge of code quality, and said, "We'll just work this out." And the guy said, "We'll ship this code over my dead body." And we shipped it. *[laughter]*

C. Mohan: But in SQL/DS, right? Not DB2.

Mike Blasgen: Endicott shipped it, right.

Franco Putzolu: Didn't they rename everything? I remember that the RSS and RDS naming convention were quite liberal. It was either a Y or an X in front of a name. Didn't they rename everything when the code went to Endicott?

Bruce Lindsay: A01, A02, A03, yes, they recoded all the RDS modules in PL/S and they renamed all the modules with the Endicott naming convention, which was like seven characters for the product name and another character for the component, and the remaining character could be anything that you wanted. *[laughter]* I remember that I worked on Brad's authorization code in PL/1 and PL/S. I had a hard time when I worked on the PL/S version, because they used to have mnemonic names – at least three characters worth of mnemonic for what it was doing – and they had decided that the proper names were 01, 02, 03, 04. They never actually got up to 10, but …

Tom Price: So you had a cross reference between the old names and the new ones.

Bruce Lindsay: Yes, I had to do that. IBM coding conventions were quite something in those days.

Jim Gray: So what happened next?

Mike Blasgen: There are a few things I want to do after lunch, if we have to go to lunch now.

Jim Gray: We have to go to lunch now.

Mike Blasgen: One is, as we're leaving, Brad can turn on the videotape, and we can see some of the stuff that's on this tape. The other thing is that after lunch I'd like to talk a little bit about the relationship between San Jose Research and Santa Teresa at this crucial time which eventually led to DB2 and SQL/DS. Particularly SQL/DS, which, while it was finished by Endicott, was actually started in Santa Teresa, with the DOS DL/1 successor, at least that was my recollection of what happened. And Bob Jolls was the manager of that whole effort, and he's here, and nobody's here to contradict you. I think you can kind of make up the story, you can tell them how you caused it all to happen. And we'll do that after lunch. That's what happened at the beginning, say through 1982, which culminated eventually

[52] Pugh, page 309.

with SQL/DS and DB2. Then there's the stuff afterwards; we should be able to have plenty of time after that to do that. So we're going to find out why Larry Ellison became rich and why Jim Gray left IBM and why Bob Jolls is living in Chapel Hill, and where Mario's going next month. Lots of interesting things. And that will be after we run the tapes. Here we go, look at Don Chamberlin wearing Brad Wade's mustache. *[laughter]*

Lunch break

Mike Blasgen: We had originally planned in the afternoon to switch to the discussion of what happened after the whole Building 28 involvement; everything to do with research would be behind us and we'd go on to talk about products. Who won, share 10Ks and stuff like that. But we ran behind this morning, so what I thought we would do is discuss some of the things that occurred in the interaction between San Jose Research and other parts of IBM that were successful or failures in terms of moving this product out of the laboratory. I had earlier claimed that the publications had a lot to do with why IBM was willing to move this out of the laboratory and into products. Of course, the products that eventually obtained were SQL/DS, which is still a current product and still contains a lot of System R code, and DB2, which doesn't contain directly any System R code, but System R was a major influence on its design. And then there are lots of other System R derivatives, stuff from other companies. Jim would be able to tell us all about how Tandem took all our good ideas, if they did, or that Oracle did. But I'd particularly like to focus on the before-1981 steps that led to the first set of products out of IBM. First of all, there were the exchange of people between San Jose Research and the development group that was originally in Palo Alto and then moved to Santa Teresa when the Santa Teresa building was completed. What was the sequence of that? Did you take a year there, Irv?

Irv Traiger: It was Franco and I going up part time to Palo Alto, for FS. Both of us later had assignments at Santa Teresa Lab, during the initial work, and the key decision-making that led to DB2.

Franco Putzolu: Then Jim went.

Mike Blasgen: Then Jim went to Palo Alto, right, and you spent a year there.

Jim Gray: Right. And we moved to Santa Teresa in the process. So I ended up in Santa Teresa.

Mike Blasgen: So you were working on Eagle?

Jim Gray: Yes; it was called VSS at the time. John Nauman was part of the team, and Thomas Work was managing; [Steve] Weick was managing Thomas. Andy Heller was taking credit for it. In six months we were going to build a product which would replace IMS and do all of System R. Andy was even more aggressive in those days.

Mike Blasgen: I have an Andy Heller story about that. At the time that you were there I think was the time that Tom Price and I were writing a paper which we intended to publish and never did, called "How Database Systems Recover." We were going to try to document how recovery worked in several existing systems and then go on and speculate about generic stuff, which eventually led to those terms we used: "no-force", "steal/no-force", "no-steal/force"; all that stuff in part came out of work Tom and I were doing. We wanted to write down how VSS would work. So whenever Tom and I would see Andy in the Research building, we'd grab a hold of him and come in and say, "Well now, exactly how does it work in this situation?" Andy would say, "Blah, blah, blah", and we'd take notes, and then he'd leave. Then we'd sit around and try to write it down in such a way as you could understand it. And we never could. We'd get down to that point where suddenly it doesn't work anymore. And we'd wait until we saw Andy again and we'd bring him in, and we'd say this doesn't work. He's say, "No, no, no; that's not what I meant; it may be what I said." We must have met with him seven or eight times, and never were able to find out … however, I will say, seven or eight algorithms that don't work were invented. *[laughter]* It was fantastic.

Franco Putzolu: We eventually got into an agreement that if you wanted to talk to Andy, he had to write it down, in all the details. Otherwise we wouldn't talk with him.

John Nauman: That stopped the interactions. *[laughter]*

Jim Gray: So after I left, then Franco went. And Franco was there for I think about …

Franco Putzolu: Almost three years.

Jim Gray: Well, you were there for a year, and it was time for you to go back. He said, "It's time for me to go," and they said, "You can't go!" And he said, "Well, I'll make you a deal. If we don't have to go through the Santa Teresa process, then I'll stay. But here's the deal: no reports, no reviews. Once a month, we'll tell you what our status is." I may be missing something …

Franco Putzolu: Yes.

Jim Gray: And they said, "Go pound sand." And so he came back to Research. About a month or two later, all of a sudden, you were back in Santa Teresa again. And there was this team that involved [Don] Haderle and Bob Gumaer and …

Franco Putzolu: Let's see: that was 1977 to 1979, about three years.

Jim Gray: So 1978 is when …

Franco Putzolu: Yes, there were some minor conflicts with management on that. *[laughter]*

Jim Gray: Franco wanted to convert the eighteen-step process to a two- or three-step process.

Franco Putzolu: Well, Santa Teresa was really paralyzed. They had this big group and they were changing project names continuously. First it was called VSS, and then DS/1, and then Eagle, and then Ampersand, which was the only cute name, because Ampersand stands for a variable in the SCRIPT language. This was supposed to be the system to replace all database systems. It was going to replace IMS, provide new fancy interfaces, provide all sorts of compatibility. There were three components. There was System Services, and that is the only part that survived. Things like logging, recovery, and locking; it's the only component that survived in DB2. There was the Data Communication Component that totally went away.

Jim Gray: I/O Subsystems; buffer management.

Franco Putzolu: No, that was added later on. And then there was the Future Database System, which went through a number of incarnations. For a while it was Chris Date's extension of PL/1.

C. Mohan, Mike Blasgen: UDL.

Franco Putzolu: Think of it as a really low-impedance database system, using current Object terminology. And then for a while it was a system with many personalities: it had a hierarchical personality, a relational personality, a network personality; it had DL/1 sort of grafted into it. And this effort wasn't going too well. So when I joined Santa Teresa I decided that there was a clear need for a subsystem that would support all these external interfaces. I decided to work on a lower-level subsystem that would support all of these interfaces. This was an under-the-cover effort, because management was very intrusive in these days in Santa Teresa; they really wouldn't leave you much freedom. So this project was called Technology Evaluation. It didn't even have an IBM code name. I mean normally, any project in Santa Teresa would have a name of, I don't know, some pagan god, some beast, some blood-thirsty beast. But this thing was just called a Technology Evaluation, and only after a while we had a meeting and decided we could not have this name appearing in a product, and have customers seeing that they were using a Technology Evaluation database engine. It was simply renamed the Data Manager. Let's see; how many people were initially working on it? There were Josephine Cheng working on the cache, Dick Crus, Tim Malkemus, Sid Kornelis, Bob Gumaer, Ming Shan, and Jane Doughty, who eventually went to Sybase; she was the only one who got rich in the process.

Mike Blasgen: My impression at that time, with respect to the things that you were mentioning, that Santa Teresa was kind of doing their own thing; that System R, if it was relevant, well, it was a source of programmers. I mean, you could hire people like Franco. We negotiated; we got Bob Yost in return. I thought that was a good trade. I have a question: what did System R have to do with that?

Franco Putzolu: Well, the official position there for a long time was that System R did not count. What was important was to develop a network database system. Of all the interfaces – I mentioned five or six interfaces – that this ecumenical system was supposed to provide, the network interface was supposed to be the real interface; and it wasn't of course DBTG, it was something else.

C. Mohan: Was that because of Bob, what's his name?

Tom Price: Engles.

Franco Putzolu: Maybe it was Bob Engles; I don't remember.

Jim Gray: It was cultural: it was because of all that money coming in from IMS.

Franco Putzolu: IMS was supposed to be there as a graft. Just a piece of code to be grafted. So they let us work in peace for about three years. We knew that this subsystem was going to become part of a production system, so we tried to write good code, and unlike System R we had naming conventions; we tried to do decent software engineering.

Mike Blasgen: This led to a series of components that are now a part of DB2, is that right?

Franco Putzolu: Yes, I think most of it survives in DB2, and actually, I'd like to hear what happened to it after I left. *[laughter]*

Mike Blasgen: I would too, but let's finish this thing. So these components were developed. Those components had relatively little to do with System R, correct?

Franco Putzolu: Well, they were written assuming that System R interfaces would run on top of them. I was assuming that SQL would be one of the interfaces that were supposed to run on top of them, so it was also suitable for SQL.

The other major interface I was convinced would run on top of the Data Manager – or Technology Evaluation, same thing – was DL/1. And in fact we spent maybe thirty percent of the code, and of the time, writing special code in the Data Manager to support DL/1. There were all sorts of special DL/1 hooks. By 1979 we had a prototype running on VM. It was pretty functional, and we ran the first substantial tests on what later became DB2. These were the existing regression tests of DL/1 physical databases. It was quite interesting; it was running DL/1 pretty well.

IBM products

SQL/DS

Mike Blasgen: OK, now into this picture comes System R, which as you point out is basically irrelevant except for the fact that you thought maybe you'd do an implementation of SQL on this code that you were writing. But something happened, and suddenly it was OK to talk about System R in Santa Teresa.

Franco Putzolu: Bob Jolls should talk about this. I really don't know the background of this transition. The only thing I know is that as far as the high-level components are concerned, there were problems in execution of the plan, but I don't know what prompted the transition to SQL and real DB2.

Bob Jolls: I think the best way to talk about this is to make it personal, because that's really what it is for each of us. We all personally got involved in this in one way or another. My way was that I came out to Santa Teresa from New York in August of 1977. I don't think I was part of any New York Mafia coming out here like some of the rest of you. I came out to be manager of Database Design at Santa Teresa. I had some technical background and some business background and I was quite comfortable being able to do this job. I was actually worried a little more about being in California. You know, all the people that I knew on the east coast had all kinds of things to say about what it was like to live in California, and raise your children in California, and so on. So I came in sort of thinking of California as the land of alternative life styles, and found out that in fact it was the land of alternative data models. *[laughter]*

I had this group of people working for me, and there was a data model per person. *[laughter]* So like we had the DBTG folks – that was a small group. We had people that were so enamored with the data dictionary; they kind of thought if you just put all the information in the dictionary, you wouldn't need to bother with those databases anymore, it would just all be there in the dictionary, right? Then we had UDL; Franco mentioned that. That was the idea that relational might be OK as long as you put it in the guise of really long-lasting products like PL/1, COBOL, and FORTRAN, so long as you didn't have to talk about it separately. So I was confused, to say the least, by all these people with all these arguments.

And then I started to learn about System R from some of my people and from coming down and meeting with some of you all. So I started using System R, frankly, as the sort of benchmark. I'd ask, "How does this compare to what they're doing with System R?" and always would learn about all the problems with System R, or all the problems, let's say at the beginning with SQL. Sometimes we're a little too tied up I think in this whole discussion today with what happened to the code. In my mind, what happened to the code is obviously important to those of you who wrote parts of the code, but the real thing that we all did as an entire group of people was to make relational real and to make SQL real, and there can be lots of different implementations of that. I would always hear what all the limitations of SQL were, and problems would be stated that couldn't possibly be solved with SQL. And then if you could write down the question and get a proper answer with SQL syntax and semantics, then here's all the performance reasons why this can never work, and etc., etc., etc. I guess this is, I'd say myself and some of the other people who were involved in this, at least for me, my experience as a programmer helped me here because I kept realizing, "Gee, here's something that is actually operating, that's working every day in different environments. All the people with the concept of how things might work are making up lists of reasons about what's wrong with the thing that is working." And I think this is sometimes the way that we operate in Programming, or was back then.

So I guess the biggest friend that System R or SQL had in this process at Santa Teresa was Eagle. You might think of Eagle as a tremendous resource drain, and so on, and so on, and it was, but the fact that Santa Teresa was completely preoccupied with how to completely replace IMS and do something much better than IMS and do many of the things that were in FS and all this stuff, sort of kept all the guns away from us, while we looked at what would be the best way to do what we then called DS/2. In other words, they sort of carved out everything but MVS and gave it to my group and said, "Well you guys go figure out what to do for database for the VM and DOS environment, and we'll worry about the really big 'Production' problems," which was what Eagle was going to address. So I think it was the best friend of the whole process, in that we got to look at that in our own pace without an awful lot of help from White Plains or local management or Poughkeepsie or any of that. We were able to make a decision to go ahead and use System R as the basis for SQL/DS, and without that being a politically-incorrect decision. Which if we had tried to make that decision at that point in time to supplant a lot of the Eagle work with System R, I don't think it would have been possible for people to make that decision.

Mike Blasgen: My recollection is the midrange product that was in the field was DOS DL/1.

Bob Jolls: Yes.

Mike Blasgen: And fortunately for us DOS DL/1 was not considered to be a success. IMS, in contrast, was a big success.

Bob Jolls: Right.

Mike Blasgen: So it's very hard to take on IMS. But it was easy to take on DOS DL/1 because it was not thought of as being a good product. So when you got responsibility for the successor to DOS DL/1, much like Eagle was the successor to IMS, you didn't have as big a hurdle to get over.

Bob Jolls: Right, there wasn't a lot of baggage that came with it.

Jim Gray: I think ADABAS[53] was cleaning IBM's clock at that point.

Bob Jolls: Right.

Jim Gray: So the Europeans said, "We need a low-end database system." And the fascinating thing is that Jim Frame, who I think was the manager of Santa Teresa, did the standard thing: he put low-end database as "out plan." IBM Fall planning was built around the notion of in plan (funded) and out plan (unaffordable). You put your pet projects in plan, and put their pet projects out plan. You know that they'll give you more money to make you do what they want you to do. You put everything they don't want you to do in plan, and the stuff they want you to do out plan, and then you get more money. This was how the funding game was played. Endicott – maybe we're getting ahead of the story, but – Endicott just had its operating system effort canceled. It was going to do a unified VM and DOS. So there were seventy zombies wandering around without anything to do, and they said, "Gosh, there's this thing that Santa Teresa can't do; they're just too busy: this low-end database stuff. Now we don't know anything about low-end databases, but maybe we could learn, and it would be a job." So they bid on this low-end database stuff. Is that a …?

Bob Jolls: Yes, that's right, and I have a personal reaction to that one, too. It's very painful for me and my little cadre of people, because we had finally gotten on the bandwagon with System R, and we were saying, "Hey, we can make an interesting, successful product out of this." And because of all these management machinations that Jim's talking about, even though we had a lab of 1200 people, and this was – I don't know – a thirty-person effort, or something like this, it was "Out-plan". So, yes, it got all of a sudden migrated to [IBM] Endicott.[54]

That's when we came up with what we called Plan B. *[laughter]* Since our mission went away, we said, "OK, we don't have that mission anymore. We'll do the database part of Eagle. We came up with Plan B as saying if something perchance should happen to all the rest of Eagle, here's what we would do. Somehow again, talking about politically correct and incorrect, it wasn't OK to say that would happen, but it was OK to say, "Well if it did happen, here's what we'd do," and that's Plan B. So my team and I began working on Plan B, which was to essentially build DB2. There were a lot of debates within the group, and I think John Nauman can talk about this too, about how much code we should import from System R versus what we should develop ourselves, and there were a lot of good arguments on both sides of that, as there always are[55]. The surprise in the reverse direction happened with this one, frankly from my point of view. The surprise of SQL/DS was that it went out from under me, or us, I should say. The surprise of the MVS project was that it happened faster than I thought it would. In other words, Plan A collapsed, all right? Eagle collapsed, and all of a sudden, everyone turned to us and

---

[53] ADABAS is Software AG's database system.

[54] For details on the evolution of System R to SQL/DS, see:

Donald D. Chamberlin, Arthur M. Gilbert, and Robert A. Yost. "A History of System R and SQL/Data System" *Proc. VLDB*, Cannes, France (September 1981) pages 456-464.

[55] Bob Jolls notes: "Irv Traiger, since he was on assignment to Mike Saranga, had the opportunity to view these arguments from both the System R and the STL perspective. Perhaps he'll want to add his perspective to this discussion." Irv Traiger responds: "It was a tricky situation. I was obviously a Research Division guy, on assignment at STL for a year (which later turned into two years). So anything I said or did favoring System R, or feeding "intelligence"-type information back to Research, could obviously hurt my usefulness at STL. So I had made a decision early on to spend the year as a Saranga advocate, helping any way I could, and building credibility. There were several things I got involved in, but the biggest one was to help him understand how Eagle was doing, how serious some of the weaknesses were, how or if they could be improved, etc. So I'd work with some of those folks. And at the same time, I had gotten to know Bob quite well. We had a lot of discussions about Eagle, and also about the database work. Saranga made the decision to stop Eagle, which took a lot of courage. Once that decision was made, STL was focused on a couple of tough questions that I was able to help on. One was to help figure out what should be salvaged from all the system services part of Eagle. And of course the other was to get the database work moving quickly. As Bob mentioned, there were good arguments on both sides, about whether to use the RDS from System R, or the so-called LSS that was being designed at STL. But it was pretty clear that the LSS had quite a ways to go, and that they'd have the usual problems along the way when you build any complex system. And the RDS, with all of its warts, at least already existed, and could be improved over time. So I tried to help Bob and Mike from that perspective."

said, "OK, when can you ship this database product?" *[laughter]* And that's when we had to make some fairly hasty, difficult decisions on …

Franco Putzolu: When was it realized that Eagle wasn't going anywhere? I mean, is there a date that you can …

Bob Jolls: When was *it* realized that? Well, I don't know. I found the whole process very mysterious. I don't know how you – well, I guess I do know how you felt. *[laughter]* As a former programmer and a manager of development, I sort of knew how to ask questions of groups to find out whether they were on track or not. It was very obvious to me that that group was never on track. We had the same phenomenon at Santa Teresa with Andy [Heller] that you described, in that he was kind of the great guru of this whole project. In this case, it wasn't just some people trying to write down some algorithms; there was like one hundred fifty – two hundred – three hundred people programming based on these meetings that were sort of fly-ins. Andy would fly in; they'd have a meeting about how something was supposed to be designed. He'd leave and the ripple effect – three weeks later somebody's writing a module where there's really no overall design.

C. Mohan: Was he in Watson at that time?

Bob Jolls: He was in Austin a lot.

C. Mohan: No, was he working in Yorktown Heights, or where was he at that time?

Franco Putzolu: He was everywhere.

C. Mohan: Probably had a home location somewhere.

Mike Blasgen: He lived in New York.

Bob Jolls: Yes; he was all over the place. But he was at Santa Teresa I'd say one day a month – maybe two days a month – something like that. There'd be all these big meetings, and all these decisions made …

Franco Putzolu: Lots of shouting.

Bob Jolls: Yes, lots of shouting, screaming, and all this stuff.

Josephine Cheng: That's how we found out he was there.

Bob Jolls: Right; if you were anywhere near that wing, you could tell he was here. So I think if you asked anybody who was an experienced, professional programmer, they would have told you this thing was going to crater. But I think there were so many layers of management, and frankly there was so much politics around – you know, "What does Poughkeepsie think? What does Harrison think?" and all that stuff, that people just kind of blindly went out on – management people did. I think the one thing that, you know, Mike when he asked me to come and speak at this meeting, I told him and I told Jim, that I think the biggest thing I had going for me in this whole process was that I was politically naive. I wasn't from either the Poughkeepsie establishment or the Harrison establishment, or any of the old Santa Teresa establishment, so I could decide that we should make System R into a product based on what I thought was best for the company, the customers, et cetera, and it was actually pretty easy.

C. Mohan: Where were you before you came to STL? Which organization of IBM were you in?

Bob Jolls: I had worked for five years on a very large internal transaction processing application, as a programmer and manager. And then I'd worked for five years or so in Business Planning in Harrison, so I'd done product forecasting and that kind of thing.

Mike Blasgen: So I'll just tell a personal thing, because this is all very consistent with my view of what happened. My job was to make the sale, close the deal, right. System R, with all its warts, take it, love it all; love me, love my daughter. We met a lot with Bob, personally, not just Bob's group, I mean Bob as a single person. And he looked at System R and I guess he took it away and maybe we installed it, or you installed it; I don't remember exactly what happened. Anyway, it was for this DOS DL/1, this midrange database opportunity that Bob had been given responsibility for. And he came in one day to my office, or called me, or something, and said, "Well, we decided to use System R." I couldn't even think of that; the most I was hoping for was we'd have a meeting to discuss it again. *[laughter]* We hadn't been rejected. At that time, not being rejected was good; meets minimum. But this was way more than I ever expected, and he was serious. I thought, well, he doesn't even know what he's saying, but he did. He was actually going to take System R and ship it out of Santa Teresa as a midrange database product. I'm sure there were lots of things were going to be redone and modified.

Anyway, then there was this game you talked about – out-plan, and Endicott appeared on the scene. The thing that was so important about that is that Endicott didn't re-decide. They didn't say, "Oh, we have the midrange database mission; now let's go out and do a bunch of research to see what the market requires," because that would have been two more years of no progress. They actually said, "OK, now we've got to ship this code. What do we have to do to make this code ready to go?" That was pretty much the attitude of the Endicott team. Bob's team actually helped in this transition a lot, and we got very heavily involved. We were talking about sending off people to live in Endicott, and that was eventually not considered

to be a good idea. *[laughter]* Not considered salable. But those guys did not re-decide, and that saved two years.

Bob Yost: They basically sent two or three people here to live with us for a period of six weeks or so, and they took it all back.

Mike Blasgen: They took the RSS with almost no change; they took the RDS with almost no change, but they completely rewrote it – they transliterated it from PL/1 to PL/S.

Bob Yost: In fact they were under instructions not to do anything more than that.

Mike Blasgen: When I was living in Washington, I went up to Endicott to help them with some problems, and I remember that they were concerned that the bug curve wasn't dropping at the rate that they wanted it to be, and they were having performance problems and working set problems and all this stuff. It was touch-and-go near the end. And then there was a big announcement, which was at some conference in Atlanta or some place like that, I went to as part of the announcement roll-out. In fact a lot of these slides that you saw earlier were actually a part of a talk that I gave as one of the front-men for the announcement of that. I have a copy of the booklet that Ted Codd wrote, called *The Significance of the SQL/DS Announcement*[56], which he published in 1981. So I think once that transition to Endicott occurred, and Endicott didn't have any new ideas, it wasn't the data model of the week, the data language of the week, that they were just going to do it, then all that debate ended. And then we were able to take advantage of the situation that Bob just described, about what happened to Eagle, which was the cratering of Eagle. Would you like to be called on?

Shoot-out at the OK Corral

Jim Gray: The part I didn't understand is, there was something called QBE that came out?

Mike Blasgen: Oh, that's a whole other story. We have a whole session for query.

Jim Gray: No, it's before this.

Mike Blasgen: OK, let's go to that; that's a good point, you're right.

Jim Gray: There was QBE, and then VS/QUERY and the shoot-out at the OK Corral; all predates this …

Mike Blasgen: OK, let's do that. Brad, can you do QBE?

Brad Wade: I'm going to have to have quite considerable bit of help on this, because in time-honored tradition of IBM, we had the System R group working on relational database systems at San Jose, and we had Moshé Zloof working on Query by Example[57] – another relational database system – at Yorktown. So a question arose as to why do we have two groups in IBM on opposite coasts who are doing exactly the same thing. And exactly where Moshé and his group came from, how they got started, I know not.

Mike Blasgen: Actually, I know the answer to that, and I'll talk about it only because it's interesting. There was a group in Yorktown that was working on something called Programming by Example.[58] Their idea was that somehow you could do business data processing, like payroll, by giving an example of what you wanted, right, it was sort of like the output back. If you just showed a fancy piece of software what you wanted as the output, it would figure out how to get it.

Jim Gray: The RPG model.

Mike Blasgen: So it was, "Show me what you want as the output, and I'll figure out how to get it to you from the data that I know is available." That never came to anything. Actually, do you know what programming by example is? It really actually happened; it's VisiCalc. That's the closest thing you have to programming by example, and it's very close, it really is what they were thinking about. But they never got that, they never got VisiCalc. But they got something kind of like it, which was Query By Example. So not programming by example, but at least query by example. And the idea was you could draw a picture of the answer, and say, "This is the answer I want" and then it would actually figure out how to get that and give you the answer. So that was Query By Example; Moshé Zloof was the key guy, and Peter de Jong was the manager.

Brad Wade: And so, Query By Example was an early graphical interface, if you will, because it would draw a picture of the table on your screen, and if you wanted "Salary = 10000," you would write "10000" in the Salary column. And I guess if you wanted people with Salary

[56] E.F. Codd. "The Significance of the SQL/Data System Announcement" *Computerworld* (February 16, 1981).

[57] M.M. Zloof. "Query by Example" *Proc. NCC 44* (1975) pages 431-438.

[58] M.M. Zloof and S.P. de Jong. "The System for Business Automation (SBA): Programming Language" *CACM 20*, 6 (June 1977) pages 385-396.

greater than 10000, you'd write ">10000" in the Salary column. In place of SQL's SELECT clause, you'd write a "P." in the columns that you wanted the query to emit. I can vaguely remember playing the Query Game with some of Moshé's stuff, but my reaction is that simple things in Query by Example were simple, and complicated things were impossible; at least they were possible in SQL.

But anyway, in the time-honored tradition of IBM, we had two groups doing the same thing. The performance issue also raised its head. The shoot-out at the OK Corral comes from a direct head-to-head performance comparison.

Jim Gray: Brad, I think that before this, somebody from the field fell in love with QBE, and they were actually shipping it.

Jim Mehl: As an RPQ or something.

Jim Gray: And there were people in the field, and they loved it. They had stories of tape librarians who'd automated their tape library with it, and Gene Trivett was going around and fixing some of the performance problems, and it was popping up all over the planet. So it had a very loyal following. It was obvious to everybody that this did something wonderful. That this was an end-user program. So then the question became, "So why don't we cancel System R?" or "Why don't we grow this thing?" I think Bob [Jolls] must have been confronted with this question a lot.

Bob Jolls: The VS/QUERY group was constantly trying to get the best of both SQL and QBE, and so that came later. But certainly when I managed that group, we had to settle all that. But I never was involved in anything where people looked at QBE as a production replacement for SQL.

Jim Gray: So what was the purpose of the shoot-out?

Pat Selinger: I think it was driven by research management.

Mike Blasgen: I'll tell you what the shoot-out was. It was a very unpleasant and interesting confrontation. Gomory was worried about the fact that he was investing heavily in two projects that seemed to be doing roughly the same thing. One of our arguments was that we had all these smart people like Franco and Jim that had built the RSS, and the RSS was really good stuff, really good code, I mean it's in SQL/DS even today; the same code.

C. Mohan: Around when was this?

Mike Blasgen: I don't have the exact date, but around 1978, right? When did the actual shoot-out occur? 1978? Gomory asked Dick Case to do a review of the work. Dick Case included Ashok Chandra, who currently runs the Computer Science Department – he's the latest version of Frank King – and one other person, who were all disinterested people, but were technically capable. They went to Yorktown and learned all about QBE, and then they came to San Jose to learn all about System R, and I gave them my long lecture about how the lock manager works and how Compare-and-Swap could do locking, and we did it all right, and we knew how to do Compare-and-Swap-Double. Dick Case was really impressed, because he's probably the architect of Compare-and-Swap. Ashok was there, and there was this whole issue of whether you could do QBE on SQL. So that was a different approach, that was, if you will, our approach, which we wanted VS/QUERY to adopt, which was that you wouldn't write QBE on the RSS. That's multiple personalities. We wanted QBE as a graphical interface, emphasizing its strengths – the graphical stuff – and de-emphasizing its weaknesses, which are XRM, no multi-user, poor storage management, and poor performance. Because they were not a compiler, they were an interpreter only. So Irv Traiger did an enormous amount of work, three months work, to show detail-by-detail how you could map QBE to SQL. There was a controversy about whether the language QBE was actually well-specified. There was an ambiguity in it and I have notes from a meeting in which Peter de Jong said, "Well I can't prove you wrong but I'm sure you're wrong. I don't accept your claim that we're ambiguous, even though you've documented it eight ways to hell that it was ambiguous." Anyway, you showed that it could all be done. And so these guys were out there to evaluate it, and one of the issues was performance, because our claim was you would want to do it that way among other reasons because you'd get far better performance. The people back east said, "No, that's not right because it is *ad hoc* query and interpretation's fine for *ad hoc* query." They wanted to program directly to the RSS.

Brad Wade: Thank you, Mike. My recollections don't go anywhere near as deep. I do remember it came down to performance. We had System R running on one of the 3270's in the terminal room. We installed QBE on another user-id, and had it loaded up on another terminal there in the terminal room. We primed it; we typed in a SQL query to do something or other; I don't even remember what. We set up the QBE thing to do the same query, and, sophisticated processes of the era, we pressed the ENTER keys at the same time and held our stop-watches up to it to see which would come back first. I even forget whether it was thirty seconds, or a minute and thirty seconds – System R was back with the answer, so we looked over at the other terminal and – it's crashed. System R's star ascended fairly nicely after that.

Mike Blasgen: We let them make up some of the queries, because they had come …

Bruce Lindsay: You mean they didn't finish any queries?

Mike Blasgen: No, they did. There was a big performance difference. Even though we were compiling *ad hoc* queries …

Jim Gray: The most striking performance difference was that Brad could type, and I forget who the person for QBE was, but he typed with two fingers and lots of mistakes. *[laughter]* So when they'd say, "Next query," Brad would go "Whirrr." *[laughter]*

Mike Blasgen: Well, anyway, it turned out to be a dramatic performance difference, and you could hear the cheers in your brain, even if you couldn't hear them with your ears. Everybody was very elated, because the performance difference was so dramatic, even on the queries that de Jong had given to Ashok and Case to give, because he was hoping those were our weak underbelly, even on those we were dramatically faster, like a factor of ten I think in some cases.

???: This was on UFI[59]?

Mike Blasgen: Yes, we were using UFI against the QBE front-end.

Jim Gray: What was the OK Corral?

Mike Blasgen: The terminal room, where the RSS was developed.

Brad Wade: After the event, and our visitors had gone, someone – I do not remember who – made up a sign. I think the sign just said, "The OK Corral." And somebody stuck it on the door. There was some movie that I remember seeing when I was ten years old; some famous western gun-slinging shoot-out.

Irv Traiger: Wyatt Earp and Doc Holiday and Hirsh Cohen. *[laughter]*

Jim Gray: *High Noon*.

Mike Blasgen: So that's exactly what happened, and it was great, and we called it the OK Corral after that, always the shoot-out. What happened was QBE continued as an IUP[60], but it never really went anywhere; it was never significantly enhanced. Santa Teresa didn't want to take it. DP, which was the sales division that supported the IUPs, continued selling it but didn't invest in it.

Jim Gray: Didn't it get reimplemented as part of DB2?

various: QMF[61].

Mike Blasgen: Sorry. Yes.

Roger Miller: It didn't become a very popular usage even in QMF.

Roger Bamford: We cloned it.

Mike Blasgen: That's interesting.

Paul McJones: It's now called Microsoft Access. *[laughter]*

Mike Blasgen: Some version of that will probably win.

C. Mohan: But other companies are implementing it, right? Paradox and all sorts of guys are implementing it.

Roger Bamford: I think the original objections are true: anything complicated is impossible. Our customers are using SQL statements that go on for pages. In QBE that would put you against the wall??? … take all the joins and everything.

Mike Blasgen: So there was a report. What I remember about the report was a ringing endorsement of the RSS.

Jim Gray: The index component in particular. *[laughter]*

Mike Blasgen: I think they especially liked the locking; I think they loved the use of Compare-And-Swap-Double. *[laughter]* And then there was some ambiguous recommendation, I don't know what ever happened to it, but as far as I know, Zloof never got the RSS. Peter de Jong went to MIT and is still there.

Brad Wade: And Moshé; where is Moshé these days?

C. Mohan: HP Labs. He started his own company, that didn't fly, then he joined Ashton-Tate; now he's with HP Labs.

Jim Gray: And Peter is at Apollo doing transactions inside of CORBA; HP Apollo.

Mike Blasgen: What happened to Moshé Zloof is he went on to something called Office by Example. In fact that talk that I gave about the SQL/DS; the next speaker was Moshé about Office by Example – OBE. He got a whole group of

[59] UFI stands for User-Friendly Interface.

[60] IUP stands for Installed User Program.

[61] QMF stands for Query Management Facility.

about twenty-five young, real programmers – not designers – he was real careful about to hire programmers. So he had a bunch of people and he got them to give him his own building – the Bernen House at Yorktown, and he took it over, and he was going to build a PC product. Gomory supported it completely. And then one day, the management changed – I think it had to do with Birnbaum's departure and the reannointment of Herb Schorr, and then maybe Abe Peled on staff. Somehow that set of people ganged up and said, "We're not going to support that at the level it's being supported," which was thirty people times a hundred thousand – three million dollars a year. And Moshé didn't like this, and eventually left IBM. And so he formed his own company as a startup to do it – I guess it didn't work – went to Ashton-Tate; Ashton-Tate was bought by Borland, then he went to HP.

C. Mohan: He's now doing Rendering by Example[62] – easy-to-do programming by example. So maybe he's going back to the System for Business Automation kind of stuff.

Afternoon break

System R winds down

Mike Blasgen: … into Eagle, VSS, all those names, to DB2, to SQL/DS. What happened was, Leonard went to New York, I moved to Washington, D.C., and you saw the staffing curve; the staffing curve started to decline. New projects were started; we can talk about the new projects, but it's sort of beyond the scope. Frank [King] – it was actually a few weeks or so before he left to go back east to get a job himself – had a dinner to commemorate the contributions that everybody made. And I remember, I was sitting at my desk in 18th & K in Washington, D.C., and my phone rang; it was Frank, and he said, "I'd like you to come out to this dinner." And I said, "You mean, it's a spouse thing?" He said, "Yes," and I said, "You mean you're going to pay our way?" And he said, "Well, yes." And so all of us got together, even though I was far away, and had a dinner at La Hacienda, which is on Route 9, between Los Gatos and Saratoga.

Don Chamberlin: That was back in the days when IBM had money.

Mike Blasgen: And it was a very nice event; it was very well done. My guess is, this is probably Brad Wade's work anyway: all of us received this plaque – I think there are three or four of the plaques here today, with different names on them, of people who contributed to the project, showing a "join" – probably a QBE-style join. *[laughter]*

Don Chamberlin: That tape that we presented at SIGMOD in 1976, showing the Phase Zero prototype, and Brad demonstrating it … and he's demonstrating it by transferring employees from Evanston to Newburg. That's the Phase Zero database.

C. Mohan: So when was this dinner? Which year?

Don Chamberlin: Must have been in the spring of 1980, I would guess.

Mike Blasgen: Shortly after these movies that you just saw; three months after the tapes that you just saw.

C. Mohan: And that was in December 1979.

Mike Blasgen: This by the way is signed "Don Rosenheim, Lab Director." I ran into Don Rosenheim a week ago at the Los Gatos hardware store; he was shopping. I said, "Hi, Don; I haven't seen you in a long time." He said, "Are you still working?" *[laughter]* Of course I'm still working. That made me feel old. Anyway, I invited him, actually, to come down, but I didn't forcefully do it. I didn't keep calling him saying, "You've got to come." I told him he could just drop in.

Anyway, that was kind of the formal end of the project, and lots of the people went on to greater and greater things. There were new projects, and Paul McJones had been one of the first to escape the roost. Tom Price escaped. Bob Jolls, eventually realizing that he didn't want to live in California and work for a company that was based in New York, switched and started working for a California-based company. Brad and Don went off to work on text-processing[63]. I don't know what Franco did.

Franco Putzolu: I went to Tandem.

Mike Blasgen: Well, eventually, but what year? Not right away.

Franco Putzolu: Early 1981.

Mike Blasgen: OK, so pretty soon; shortly after Jim left. Jim, of course, went to Tandem. And a few of us stalwarts stayed behind in IBM, like Irv and me and Bruce Lindsay and Raymond Lorie and Pat Selinger. And so that's the end.

---

[62] R. Krishnamurthy and M.M. Zloof. "RBE: Rendering By Example" *Proc. Eleventh International Conference on Data Engineering* (March 1995) pages 288-297.

[63] Paul McJones went to Xerox in late 1976. Tom Price went to IBM Office Products Division, in Austin. Bob Jolls went to Tandem. Don Chamberlin and Brad Wade started the Janus project.

Now at this point many things happened. In fact, some of these are happening in parallel, but in the interest of organizing into before and after … So what I'm going to do is ask Pat Selinger to become the moderator for the rest of the afternoon.

Pat Selinger: Just as a parenthetical remark, Jim left IBM only because I wasn't around to stop him. I had gone into labor that day. That's the only reason you got away.

Mike Blasgen: Actually, I feel the same way, and I've said it many times. I was Jim's first-line manager, and then I became his second-line manager, and the whole time I was in those jobs, he stayed, even though he talked about the fact that it was too far to drive. But he stayed because he liked me so much, and as soon as I left, he said, "Pffft".

Jim Gray: That's right; that's how it was[64].

VS/QUERY (QMF)

Pat Selinger: So we've got about another decade and a half to go. We're in the middle of 1980 or so. We're going to finish up the VS/QUERY, and then John Nauman's going to take over the DB2 era. Bob? VS/QUERY.

Bob Jolls: Thanks, Pat. After Plan B became Plan A, I was asked to take a new job to manage VS/QUERY and languages at Santa Teresa. Maybe you can't be the outcast – the person in charge of the outplan – and then when they decide you were right, they can't leave you in charge anymore, it's too embarrassing. So I got asked to manage VS/QUERY, which was a project in deep trouble. VS/QUERY was the product that the Marketing Division had said for the last four or five years, "You know, we don't really understand all the things you people out in Santa Teresa are doing, but we want that one. Give us that one." And somehow the VS/QUERY group had managed, year after year, not to be able to deliver a product. And they were kind of caught in a bit of a trap: they would go meet with Moshé and other people back in Yorktown, and decide that QBE was wonderful and they needed to have all the QBE function that could be defined working in release one. And then because they knew SQL was important, they needed to have SQL in there, and they had a few other things in there as well. So they had about forty people, and a mountain of work to get done, and a belief that they had to get it all done for release one. So they would always have reasons – I think at one point, Pat was saying to me during the break – they had a list of problems with System R. You know, here's the reasons we can't get this done; System R has these problems, etc., etc. And when this group started reporting to me, I would ask questions like, "Well, so what happens when you go meet with the people in System R?" "Oh, we don't do that." So they had problems, but they weren't resolving the problems.

Pat Selinger: I think one of them was that every time there would be a disk error, it would stop and say, "YSYSTERR".

Jim Gray: We had seventeen hundred calls to SYSTERR. It was a big barrier.

Pat Selinger: We put Bruce Lindsay on this problem and he went and he counted them and changed them to some other error code. *[laughter]*

Bob Jolls: And they didn't tell Larry Ellison.

Jim Gray: They all called one routine, called PANIC, and then that routine called …

Bob Jolls: I'd say there was a sort of a syndrome that was not unlike all the people with the different data models. And I hope I don't sound cynical here, but I'd say that a lot of people got into a syndrome that it was OK not to make progress as long as you had a reason. And of course there were always lots of reasons, so not much progress was made. I'd say that folks on VS/QUERY felt trapped by the fact that they thought they had committed all these things to Marketing, and that Marketing had told them they had to have all these QBE functions, and so on, or they wouldn't "have a product." Finally, after lots of debate about this, and going to lots of meetings about what in QBE couldn't be translated through SQL, and therefore what functions would be missing that Marketing said they had to have, I was frankly confused by this, didn't really understand it. So I finally asked for a list of the ten things that wouldn't be available in release one, because they couldn't be translated through SQL. I then went and met with Marketing, and found out they didn't understand what those things were, either, therefore certainly couldn't demand that they be there. And we managed frankly to scope the thing down to the point where we could get a product out the door. Since I left IBM I don't know exactly how that product has done, but at least when it was released it was a pretty successful product.

Pat Selinger: It's a big money-maker.

Bob Jolls: So they had all the right ingredients for a good product: System R, some ideas from QBE, and perhaps from other things …

---

[64] Jim notes: "Just kidding. IBM was moving further south to Sky Ranch. The three hour/day commute from San Francisco was bad and getting worse."

Roger Miller: Bob, QMF is tremendously successful. We had a huge cost base, so we had to have a high price. Customers bought it, so it became very profitable.

Bob Jolls: Who says cost-plus pricing isn't the right way? I guess we had a lot of trials and tribulations over the years with the VS/QUERY group from a System R point of view. Frankly, I think those got ended pretty quickly and they wound up with a successful product.

C. Mohan: So this is different from QMF?

Bob Jolls: No, VS/QUERY was the code name; sorry: QMF.

Pat Selinger: And then we went on to DB2.

Bob Yost: Didn't you actually run that on a bastardized version of System R? For a long while, there was no DB2 to run on. You basically took some code, made it work over in your environment so that you could actually test the query stuff.

Bob Jolls: Right. I can't remember whether they shipped QMF with that black box old version of System R.

Bob Yost: That was just for testing purposes because the underlying system didn't exist at that point.

Bob Jolls: So we're going to have John tell us about DB2? Good.

DB2

John Nauman: I'm going to do a little Eagle bashing. When I first met Jim, he had just come up [to Palo Alto] and we were working on a project which did not at that point have a name. I don't think it was named Eagle when you got there.

Jim Gray: No, it was called VSS.

John Nauman: But one of the marketing guys was looking at a slide – not a slide, a poster; IBM was big on posters – and it was the Santa Teresa lab announcement with this eagle, just sort of soaring. And he looked at it and said, "That's what we'll call the project; we'll call it Eagle." So we'd been calling it something else – it might have been VSS – and what we had to do was go back through the whole document (by this time the specification was probably forty or fifty or eighty pages) and we had to replace all the … I don't think there was a global replace option in the editor we were using at the time, so we were using SCRIPT, so we put in &PROJVAR was the name of the project, so you'd fill in &PROJVAR and you'd get Eagle everywhere. So after about, probably, six months, after we'd moved into Santa Teresa and discovered there weren't any eagles, we decided to change the name again, and at that point we changed it to Ampersand because that just seemed to be a better name than trying to change the project name all the time; we figured no one would ever figure that one out.

Roger Miller: I thought that was the lawyers who came in and said, "That's a predatory bird and you can't use that name." One of those wonderful stories that sounds great, but it's probably not true.

John Nauman: It's untrue. We just got tired of trying to think up names and the marketing guy had left, so we decided to just call it Ampersand. Probably the thing that convinced me that the project was going to die was we'd been working on it at Santa Teresa for about a year and we were having regular review meetings of the document. We'd been doing this document since we were in Palo Alto, so it was more than a year since we'd been working on this document. The average meeting was: you'd go into the room to review the document – the specification – and people would start talking about how there were widows and orphans. Does everybody know what widows and orphans are? This was the topic of conversation: "There's a widow on this page; you've got to fix it." At that point, I said, "Nah, this is the wrong thing to do. We shouldn't probably be doing this. This project is doomed." And it was. We were trying to figure out how to go forward, and what to do about the database stuff. My recollection of this is we had decided that we were going to go relational, but how to do that was something we weren't sure of. And I remember a lot of meetings with Frank King.

Franco Putzolu: When was this?

John Nauman: This was 1978-1979. By then, Franco was there sort of working on the RSS replacement – data manager stuff – and we were wrestling with what to do about the upper areas – the relational data store part of the system. There were two camps. One camp was me and Don Haderle, and the other camp was Frank King and everybody from Research. We felt like the right thing to do was, since MVS was not the same as VM, it was going to be hard to fit this stuff into MVS efficiently, so we probably needed to go in and restructure this stuff a lot. The System R folks felt like it didn't really need that much optimization; it was probably going to be OK.

Jim Gray: Love my dog.

John Nauman: Love me; love my dog. I remember vividly the day that – in fact, I was working for Bob [Jolls] at the time – Bob came and said, "No, the answer is you're going to take this stuff from System R." And we said, "OK. If that's what you want to do, it's a business decision. Let's go

do it." So we started working on it. We spent a lot of time and we assembled a team which is – some of the people are here in the room – people who included: Jay Yothers, who isn't here in the room today; Josephine; Roger, who I hired out of another job – I think IBM was his fourth position and fourth company and he told me when I hired him that he wasn't sure that he was going to stick around all that long, but he wanted some experience with a big company, *[laughter]* and it was bigger then, I think, than it would be now.

Roger Miller: I'm not usually that forthcoming.

John Nauman: Let's see; we hired Mort – John Mortenson – John had already been working in the company, we brought him into the group. We hired Jerry Baker, who is probably the person who has made the most money off of this – next to Larry Ellison; he works for Larry; is he your boss?

Franco Putzolu: No, he's not my boss.

John Nauman: He's in one of the development organizations.

Franco Putzolu: Porting.

John Nauman: OK, that's what he did when he left; he went to Oracle to do UNIX ports, because he had a UNIX background coming out of University of Texas, and didn't like MVS that much. So he made lots of money; the rest of us sort of worked for the good of mankind I think.

Jim Gray: Thank you; we really appreciate it.

John Nauman: But actually we had a lot of fun. There were a lot of interesting things going on. Jim was still helping us out to some extent. Franco was helping us out a lot. Franco was wrestling with how to do DL/1 and SQL with the same underlying data structures, and later he tried to do it again, as I mentioned earlier. But it was a lot of fun. The reason I left – I left IBM in mid-1981 – Jim had just left and called me up and said, "Gee, why don't you come talk to Tandem; they're looking for somebody to manage one of their groups." And so I came up and talked to them. Jim had written a treatise called "MIPS Envy" – I'm sure some of you remember this – which was the reason that Jim purports that he left; I think there's probably some truth to it. When we were doing the DB2 stuff, we had a terminal room at the end of the hall that had six 3270 terminals in it. That was all we had, that was all the compute resource that we were allowed to do DB2. We gradually got more and more 3270's and put them in people's offices, and that was sort of a revolution. Nobody had terminals in their offices; it was terminal rooms that made sense.

Jim Gray: And you were only allowed to log on at certain times, right?

John Nauman: Terminals were expensive. Yes, you could log on on weekends, and you could log on before eight and after six. So Haderle and I and Baker and a few other people would come in at four o'clock in the morning, and you'd tune in the radio sometimes at four o'clock in the morning and you'd hear these whale sounds on the radio station. And we wondered what was going on; why were we here; what were we really trying to accomplish? I was frustrated by some of the same things that Jim was, but I was as frustrated by the fact that I'd been working on the project in 1981 for about four years – if you'd count the Eagle time; that doesn't count the FS time before that – and I could see it was just about done. Here it was, mid-1981, about six months away from shipment, so I figured it was OK for me to leave now, because everything was sort of tied up. So I told Don, "I'm getting out of here. You can take it from here; it's going to be OK." This was about the third time I'd done this to Don – left him with a project that we'd worked on. But this one took a little longer than the others to get finished.

So I went to Tandem, and then we recruited Franco to come to Tandem, and we recruited Mike [Pong] to come to Tandem, and we recruited a number of people to come to Tandem. We stole Don from Esvel.

Don Slutz: Not stole. I walked away.

John Nauman: We built up a pretty strong group at Tandem, which was a lot of fun also; I'll let somebody else talk about that. But the reason I left was I thought things were done, and I wanted to go someplace where it didn't take four years to get a product out.

Franco Putzolu: Yes, I had the same feeling.

John Nauman: And I think that was one of the reasons that Franco came and some of the other people joined. And we did get things done much more expeditiously at Tandem, and I think it was more fun for that reason. I worked at Tandem for four years, and in 1985, after I joined 3Com, I read about the full release of DB2. So I was off by about four years on how long it was going to take to get the full product out. And a lot of that I know, from talking to Don and Josephine and other people, was around just getting it to work in MVS, which was by no means a simple system, and I think we had all underestimated how complex that was going to be and how high the performance requirements were going to be.

Roger Miller: Because as soon as we started shipping it to the early customers in 1982, they started using a lot of four-letter words in their discussions. The first part was fun you know; "Oh my goodness, it's exciting." But right away,

there were all these things that we'd done and John said we'd never have to do this. Things like Cold Start: "Oh, no; databases should never have to Cold Start." It took us ten or fifteen person-years to put Cold Start in, after the fact. That was ugly, hard programming to put it in. It took a lot of expertise. We kept scratching our heads and saying, "Why would a customer ever want this stuff?" And the answers kept coming back, "The question is not why they need it, but why you don't have it." So we kept going through problem after problem, and we kept finding these bugs. Oh, Jerry Baker would just go nuts, because Jerry was a high-level language kind of guy, and he was working in the RDS, but he had these fragments, and he had this glue code, and Jerry didn't even want to know what they did: pasting them together, the gluing; we couldn't fix a lot of the fragments, so we kept having to patch and paste. So in November 1982 we shipped it to the six; we shipped it to eighteen about in March of 1983. The big event – Blow-up Announce – and here we go shipping to forty - fifty - sixty customers as we announce: June 7th, 1983. Things suddenly start taking off, and we're in early ship, and running into problem after problem. Sixteen megabytes of memory isn't much; every PC in the room has that much, right? But you only get eight of the sixteen, and eight megs, when you start running a lot of users below the line, sucks. It didn't do the job, and here XA came, and MVS XA 31-bit addressing, and a whole stack of new problems, and incompatibilities that we weren't very comfortable with or used to in MVS. Which of those services go up above the line? "We're not telling you" – kind of responses; there's no list of such changes.

Tom Price: Get the dump.

Roger Miller: Yes, yes, just try it; you'll like it. And so we kept twisting and pacing and it was excruciating. Every once in a while you'd go up and talk to the people in Research and they'd say, "Well gee, I don't understand; it worked when I left." It's been really gratifying to have Mohan come down and say, "Oh, you know, that really is hard" on occasion. It's not really trivial. Because then as we started building users, we finally went into Controlled Availability in September 1984; General Availability; and then clear out in April of 1985 – by that time we had Release 2 coded. Release 2 came out about a year later than that. In Release 2, we threw away the fragments and built a Structure Gen just as you folks were doing HOP[65] and started saying, "Ah, my goodness." Essentially, Release 2 was: go talk to those 250 early users, get the feedback, build it into the product, make them successful. We must have done something, because they've been popping and popping, but after that it gets a little less interesting.

Franco Putzolu: Can you say something about the dual-database strategy?

Roger Miller: You mean the dueling-database strategy?

Franco Putzolu: Was there much controversy inside IBM on the dual-database strategy?

Roger Miller: We've always had a kind of love-hate relationship with the folks in the tower next door, because IMS has almost always been in the tower next door. On the one hand, it's this tremendous heritage; and on the other, a customer often comes through the door and says, "Well, I have to choose IMS or I have to choose DB2; now which should I do?" And there's a fair amount of antagonism – well, just as competing projects, if you will – for resources.

C. Mohan: Somebody should say something about this statement that Frank King was supposed to have made in Australia which cost a lot of headaches. This was the IFIPS Congress or something, right? When he gave a talk on the state of relational or something, and he was supposed to have said this will …

Roger Miller: Oh yes, and this will kill IMS, essentially. Because all of our crew was painfully aware …

C. Mohan: Was this 1981; I forget the year. It was some IFIPS Congress where he gave …

Pat Selinger: It must have been 1980, because IFIPS were every two years.

Roger Miller: And the repercussions for us were minimal. We weren't announced. SQL/DS was about to come out, but SQL/DS wasn't about transaction processing. SQL/DS is VM, queryish, and not really large databases. Today's large databases are terabytes, and real live customers in lots of situations are building six - eight - ten terabyters. SQL/DS kind of runs out of gas in the ten - hundred megabyter range – gigabyte or two gigabyter – it's not a high-end database. We always wanted to scale into – oh, sixty-four gigabytes, that was one of our stupidities, that sixty-four gigabytes [per table] will be enough for a long time.

Franco Putzolu: I thought it was infinity at that time.

Roger Miller: Yes, you can't believe how many folks are really ticked at us for the sixty-four gigabyte limit. Every hard limit in the product, everything that's built around one byte, is wrong. Everything that's limited by two bytes is a problem, and most of the three, four, and even some six and eight-byte sizes. We've tried to remove limits when we could, where it wasn't five thousand lines of code. Everything about name length was wrong. Eighteen is a terrible number. Especially for VARCHARs. We learned

[65] Roger Miller notes: "HOP is the System R flavor of High Performance Optimizer, as I remember it."

these things and haven't been able to change them in a number of cases. And yet we've been very successful.

???: When DB2 for MVS came out, it wasn't billed as a transaction system either, right? It was a Decision Support System?

Roger Miller: Well, we had to be careful, because that's really right what Franco was asking about: dueling databases. We had to be real careful. We weren't solid. We weren't ready to take on IMS. In the best sort of situation, if somebody says, "What's the path length; I'm worried about costs." The answer is 2X, roughly.

Franco Putzolu: So when did the situation change? When was DB2 beginning to be accepted as something good for OLTP[66]?

Roger Miller: Release 2 was really when a lot of that came in, because Release 2 we made that two factor drop down into the one and a half range, and for low-volume transactions that turned out to be pretty acceptable. Because we delivered some flexibility: the ability to recompile instead of having people recode is a big difference. And so folks would be going through a project on IMS and discover, "Gee, I need to put in a couple of extra indexes. Oh, well, if I put in a couple of extra indexes, I can't use them really well; I have to recode to go down the path." That's not a very acceptable choice.

Mike Blasgen: I used to give this talk, five years before.

Roger Miller: It's funny, I was just looking through my DB2 materials, and the Version 1 General Information manual, and as I was watching the foils and saying, "That looks like a pretty faithful rendition; it's missing Don and his beard." But a lot of this material has not changed for two decades.

Bruce Lindsay: I think you're being a little bit self-serving or conservative to say that DB2 wasn't posed as a transaction-processing product because it didn't have the performance, because there were plenty of other people out there making pretty good money with worse performance. It's because IBM protects weak products; protects its own products. Admit it: IBM will not attack its own products, even when they're weak and there's better technology and they have it. Ask Mike about RISC. Ask everybody in here about relational.

Roger Miller: But there's a little of each. It's called, "Would I rather take it out of my left pocket and put it in my right, for …"

---

[66] OLTP stands for Online Transaction Processing.

Bruce Lindsay: No, there's a saying that expresses this very well about trying to protect weak products: "If your children are going to be eaten, the best thing is to eat them yourself."

Josephine Cheng: Bruce, that may be true in the past, but I think things have been changing. If you look at the investment that IBM has made on the new products like DB2 Client/Server, it is quite substantial.

Bruce Lindsay: It hasn't changed.

Pat Selinger: We're getting kind of far afield here.

Mike Blasgen: Frank King for example was hard over that you take System R as is. That was non-negotiable. Then he went away. So he became a non-factor in this. But he still played a role in certain policies, like you're saying he gave a talk in 1980, which was after he was gone, I think. One of the issues I was working on, even though I was in Washington, even though I had some job that had nothing to do with database, was that everybody had concluded that we would always have to support DL/1; we would always have to support the old programs. If you look in the Pratt & Whitney report, it says, "Number one objective is we have to have full support for IMS data and IMS programs." Now, when did that go away? Just because nobody could do it?

Roger Miller: Pretty much literally. Right, because we started coding to try to make it happen. And essentially, it came down to a couple of things. Performance: the closer you get to performance, the worse it looks. And the brain killer was you can never tell except during running of the yearly close that there are some things in there that you can't support, because supporting exactly one hundred per cent with complete fidelity to DL/1 was not possible. We got pretty close, but pretty close is never close enough.

Mike Blasgen: Even Frank King's position when he worked for Bob Evans was that we would have to do it.

Franco Putzolu: Oh, yes. That was a given.

Mike Blasgen: And yet what happened was we were saved by the fact that we couldn't. If we could've, we would've.

C. Mohan: I was told that the second release of DB2 was to have been DL/1 support, but it never happened.

Roger Miller: Exactly. We had the spec, we had the running code, we kept going, and we said, "Can we ship a product that is this way?" From IBM we just said, "Here's the product; we can ship it, but will customers accept it?" We tried it on a couple of customers, and "Heck, no!" was the kindest they said. It was nice to have customers who were honest with us.

Franco Putzolu: Was it performance, was it page-locking, was it functionality?

Roger Miller: Performance was a negative, but the primary issue was the inability to tell if the conversion would work. Remember, this is building the calls at runtime, and we had three or four per cent of the calls which were not going to work. There isn't any code inspector that could determine when they were using a function that would fail. So until you can get to one hundred per cent, it's not acceptable. Some of the Brand X vendors can get away with a little less than we can, but the real killer was: you could never tell when it was safe to convert.

Franco Putzolu: When did it die?

Roger Miller: Release 2, essentially. Because we had Release 2 of DB2, which was a 1986 GA. We switched over in about 1984 or 1985 to say we can't do DL/1, so let's go pedal to the metal. Let's support relational, and do what relational DBMS needs to make those customers successful.

Don Slutz: I'm not sure when it was, maybe Don [Chamberlin] can help, but Frank King had Bob Taylor and myself go off for a couple of months and look at supporting DL/1 calls – I think 1978 or 1979; do you know? I was working for you; did you know?

Roger Miller: The team that essentially built it for us were: Sid Kornelis, who came from IMS. Sid knew IMS backwards, forwards, left, and right.

Jim Gray: It had fifty thousand test cases …

Roger Miller: Yes, we had *the* IMS regression test bucket.

Jim Gray: … fifty thousand, which was a number that boggles the mind.

Roger Miller: We had Lloyd Harper, who had the long history of many, many products that never shipped. We had Bob Engles, doing the part working for Homer [Leonard], and Homer was right in line with Bob Evans, down saying, "Well, gee, this product is only a toy until it supports DL/1." They were going to get it to ship until we came to the realization that it didn't matter if we did or not; we couldn't sell it if it did.

Mike Blasgen: And none of the good guys won, right? *[laughter]*

Jim Gray: I think Oracle. *[laughter]*

???: No, he meant Tandem. *[laughter]*

Mike Blasgen: Oh, SQL, this is an SQL review. SQL won.

Josephine Cheng: Well, I was really fortunate. Once I graduated from school, I joined IBM, and I worked in the project Eagle. At that time there were walls and doors that kept everyone out. You know, in Santa Teresa, we have towers that are all connected. They specifically put in doors to separate all the connections to other buildings – you needed a special badge to get in. So I joined the project, and Franco was working diligently; Irv was on an assignment to STL also working very diligently. Occasionally I saw Jim [Gray] and also occasionally I heard loud voices shouting, and I knew it was Andy [Heller].

I worked on the Buffer Manager for about two years, and then I transferred to John Nauman's department. Let me share a couple of experiences I had on our productization of Research code.

Mike Blasgen: You're supposed to be kind.

Josephine Cheng: Yes. Maybe next time I won't be invited. *[laughter]* I always have to have many system R papers all over my desk to help me understand the code. For instance, there's the part that talks about PTREE nodes. The Parse Tree node has fields called P1, P2, P3, P4, P5. And so in order to know the meaning, you have to go and look at the reference: P1 has different meanings for different node types. If this is a column node, P1 must mean "pointer to the table node" and P2 means "pointer to the descriptor." I ended up having little pieces of paper hanging all over my office.

In one of the modules – I don't know who wrote that – it was about a semantic routine that checks on type compatibility. Franco wrote that? It had this interesting algorithm for checking type compatibility. You take "modulo" four and compare the result with the generic type to see if they are the same. You wondered, "Why modulo four?" It turned out that modulo four gives you four bits to use. I thought the designer must have thought that four bits was enough to cover any foreseeable type. If you have numeric types (two to the power four), you can have sixteen different numeric types; should be enough, right? Unfortunately, we have NULL and non-NULL, so that takes up two. Later on, we added Kanji support, which reserved double-byte NULL and double-byte non-NULL. So each numeric type took up four codes. Then we had INTEGER and SMALL INTEGER, which took up eight codes. When we productized the code, we had to add DECIMAL and FLOATING POINT. That took up all 16 codes. Unfortunately, once we got the code out the door, customers asked for short FLOAT instead of long FLOAT to save disk space, and some asked for 16-byte FLOAT and 31-byte DECIMAL for more precision. So much for modulo four!

As I mentioned before, I always admire all the System R people. I think they're great inventors, not only on algorithms and doing all the great work, but they are also very good in creativity, in inventing things, such as words like "Search Argument" making it into an adjective:

sargable[67] So when I talk to a customer, I always say, "This is a sargable predicate." They look at me and say, "What?"

Jim Gray: Everybody has sargable predicates now: Sybase and Oracle …

Josephine Cheng: Yes, and customers diligently look up the meaning in the dictionary. So the people who write the manuals – our ID[68] folks – they didn't like that. And they named it Type 1 predicate: that means it can be processed by the Data manager; Type 2 predicate: by RDS – I don't like that; I really want to call them sargable predicate.

Anyway, I had lots of fun looking at System R code and productizing it. As a matter of fact, when we finished our first driver, we felt such relief – if we had not taken the System R code, I don't think we would have made our DB2 Release 1.

Bob Yost: What were your instructions? Were you told to take basically the RDS, because you had another Data Manager? It wasn't going to be the one from System R, but you were taking the top half of the System R technology as sort of a blue-print. Were you inspired by it, or were you to look at the code and try to translate it? Now when they went to Endicott, they just said, "Take the code. Just translate it; don't even think about it." But you must have had different instructions.

Josephine Cheng: Well our instructions were to make it work. *[laughter]* The first thing that we did was to try to understand it, so I think I contacted every single person in the room: Mario, Pat, Don – trying to decipher and understand what it's trying to do. For our first release, we would translate all that code so it would work with our system code: the storage manager, and trace, accounting – you know, all the productization work. We tried also to add some features to the Release 1, but not really that many – only those that we needed for commercial use. So we added floating point, and decimal. At the time, I took Optimizer; Jerry Baker took ASLGEN; and Nick Nomm took ODEGEN. Back then machine time was more expensive than human salary, so we went to work on Saturdays and Sundays. The three of us were occupying the whole floor on Saturdays and Sundays. We did all the RDS work and Jerry had the idea that we should go and support symmetric view. It means when you select something from a view, you should get a failure code if you try to insert out of the scope of the view. So we also added the symmetric view function to Release 1. So the goal of Release 1 was essentially trying to make it work, add very minimum function, and get it out of the door as soon as possible.

John Nauman: It was a lot of the same stuff that was going on at Endicott; that is, there was a translation from PL/1 to PL/S; we had to do that.

Josephine Cheng: That was already done by Endicott.

John Nauman: So we had to do that, but that was done by Endicott, so we could take that. The amount of additional work was relatively small, except in the precompiler area, where we did, I think, quite a bit of things.

Roger Miller: Where we had to refit and refit and refit because we learned what the System R code needed. I built the PTREE many times.

Josephine Cheng: Anyway, that was a really fun experience.

C. Mohan: Did you watch the tapes in the process of doing this work?

Josephine Cheng: We did. We also had everybody from the RDS group come to STL and give us tutorials, and we videotaped those, too. The folk at Almaden have been very cooperative. Any questions that we asked, we always got the answer back, and anything that we needed help on, we always got immediately.

John Nauman: There was a lot of concern, I know, when we made the decision to go with the RDS stuff, that the System R people wouldn't be there, and that was one of the things that worried Don and me a lot in addition to the underlying MVS stuff, and that was never the case. I agree with what Josephine said: if we had gone off and just started from scratch to do our own thing, it may or may not have gotten out. The final part may or may not have gotten out in the same time frame, but it certainly would have taken a lot longer to get the prototype up, and approval for what we were doing. So it helped a lot, I think, in that area.

---

[67] The RSS supported a simple search capability: you could specify a "search argument" (SARG) of a simple comparison in canonical form (e.g., salary greater than constant) and the RSS would perform the search along a specified path. This offered performance advantage. To use this the higher level software had to recognize when SARGs could be used. If the SQL statement itself contained a predicate that could fit the SARG model, then the predicate was called *sargable*.

[68] ID stands for Information Developer.

System R folks leave the fold

Pat Selinger: Well, that discusses what happened with DB2. But there were a variety of folks who actually went off and did other things in other companies. I wanted to go through the story of Esvel and Tandem and Oracle. Don, do you want to talk a little bit about Esvel?

C. Mohan: The richer folk, eh? *[laughter]*

Esvel

Don Slutz: I guess it was early 1981 that Kapali [Eswaran] was talking about doing something. He had some deal going, and then it all kind of disappeared in February or March. Roger [Bamford] and I went out to lunch with some people he was trying to sell, and …

Roger Bamford: … MDS???, right?

Don Slutz: No, that was the second one. The first one … So that all went away, and then sometime in August or September of 1981, he had another one going with this company in Boston, MDS, who basically wanted a multi-user RSS. Ron Revelle, who left in 1980, went off to Britton-Lee, and was working on a database machine. He actually went there as a hardware guy, even though he had been a software guy. Actually he was with Kapali in System D[69], and so they knew each other pretty well[70]. He wanted to do more in the Britton-Lee machine: more hardware accelerator stuff, and Britton-Lee didn't want to, so he wanted to go do it somewhere else, and he got together with Kapali. So the original Esvel plan was to make a better Britton-Lee database machine. So Roger was going, and Ignatius [Ding]. So we got there in 1981, and … So we were pretty much staying with the same technology – write-ahead log; bulk-fetch, because it was a database machine, so it was really client-server driven. Data-flow in the optimizer, because view composition was too hard, and a bunch of other reasons. So about six months later, Ron was killed in an accident, and that ended the hardware work. So we went on for a while, another year or so, and got some venture money. I guess they got nothing, right? The venture people never got anything out of it.

Roger Bamford: Everybody got stiffed. *[laughter]*

Don Slutz: Roger left in late 1983, I think.

Roger Bamford: Yes, it must have been.

Don Slutz: Eventually, the RSS part of it, we delivered that on VM in nine months, starting with an empty office in Campbell. And then there was an MVS version a few months later.

Roger Bamford: You mean the ??? – yes, kind of the RSS equivalent.

Don Slutz: Right.

C. Mohan: That HP bought, right?

Don Slutz: HP bought that, and that became ALLBASE.

Jim Gray: Tektronix was an investor.

Don Slutz: So we made a contract with HP in early 1984, and then things changed a lot and a number of us left – six or so, and then another seven or eight – and HP picked it up with ALLBASE. Finally the company was sold to …

Jim Gray, C. Mohan: Cullinet. [IDMS/SQL]

Don Slutz: Franco came there in late 1983 or early 1984 for three months.

Franco Putzolu: And then I had some minor conflicts with Kapali.

Don Slutz: And when I decided to leave, I called John Nauman; actually, I sent my resume back to Tandem, and I sent one to Oracle – I never heard from Oracle.

Roger Bamford: I thought you interviewed with Oracle.

Don Slutz: Well, no after I took the job with John, Bob Miner called and he said, "We lost your resume, we're real interested. Come on up anyway." So I went up and talked to him for a while, and I spent a few hours with Larry [Ellison], which was interesting. We got to talking, and I mentioned that I'd been working on performance at one time. And that's when Oracle had been slammed on the

---

[69] S. Andler, I. Ding, K. Eswaran, C. Hauser, W. Kim, J. Mehl and R. Williams. "System D: A Distributed System for Availability" *Eighth International Conference on Very Large Data Bases*, Mexico City (September 8-10, 1982).

[70] Jim Mehl notes: "Ron Revelle may have briefly done some hardware investigations at the beginning of System D, but he was certainly not part of the software work that became known as System D." Don clarifies: "I recall that Ron was working on a processor at first (and I thought it was for System D). When System D started using Series 1's, Ron switched to work on the network connection stuff. When that was stopped I recall Ron quit soon after."

Wisconsin benchmark[71], and Larry all of a sudden stopped talking about interview stuff, and opens his big wood desk. He pulled out all these listings. He said, "We've fixed all that," and he showed me all the Wisconsin benchmark runs that he'd made. We went on and on … Then I said, "Well, I really decided to go to Tandem." He said, "Don't go there. Come here; get rich." *[laughter]*

Franco Putzolu: I interviewed with Ellison when I was leaving Esvel. He said, "If you come here, you won't have any problems about money anymore; I promise."

Jim Gray: John, do you want to do the Tandem story?

John Nauman: Sure.

Don Chamberlin: Don, wasn't there some kind of shareholder suit around Esvel at some point?

Don Slutz: Yes, that came later. The suit lasted for years and the final agreement included a gag order.

Mike Blasgen: I always figured that Franco went over there as some sort of double agent. Didn't he go in to Esvel and then come out dragging several people with him?

Jim Gray: Right, we almost lost Andrea Borr, but she decided against it at the last minute.

Don Slutz: You have to understand: I was car-pooling with Franco to Esvel, and he doesn't say much. Actually, he was thinking of going back to Tandem, and I was thinking of going to Tandem, and I don't think we even knew that …

Tandem

John Nauman: Franco worked for me at that time at Tandem, and I promise you we didn't send him anywhere as a double agent. It was really a dramatic loss for Tandem. I joined Tandem in 1981 right after Jim did. It was an interesting experience, because at the time, Tandem had a query product called ENFORM, and a file-system product called ENSCRIBE, and then they had a transaction monitoring facility, which was in the making: TMF. Jim and I worked a lot on the TMF stuff – getting that up and running; a little bit on ENSCRIBE. Jim also wrote the phone directory …

Jim Gray: TELE.

John Nauman: … at Tandem over there and I wrote the FULIST program. What he told me when I got there was you have to make some sort of contribution in terms of code, so I spent longer than I wished to learning about the vagaries of the [6520] terminal. What I hoped to do when I got there was to take a transaction system that they were starting to build and turn it into something that I thought we knew how to do based on what we'd done in System R: translate it into DB2. So through all the experiences with Eagle and all the underlying Lock Manager / Recovery Manager stuff, I thought there was a lot there we could learn from and do a lot better at Tandem, given their NonStop architecture.

Jim Gray: And we were convinced that IBM would *never ship*.

John Nauman: Right.

Jim Gray: Because, you know, the organization, I mean, I don't know if Plan B had turned into Plan A yet …

John Nauman: Yes, it had. But you were convinced it would never ship; I thought they would in six months, remember. You were a lot closer to being right than I was. We brought Franco over to write the underlying data stuff for the relational database system that we wanted to build. And during the next, probably, three years, we tried to put together a NonStop SQL group, which was what the product eventually ended up being called. It was awful because there was a competing product. There's the FS story at IBM; there was a product called Rainbow[72], and Rainbow was the follow-on system. Rainbow was everything you ever … that's where Jim actually worked when I joined. Very shortly after I joined, Jim left Rainbow, and came over and worked on the real system. But Rainbow was always pulling the resources away from what we were doing today, and looking at what we could do in six months, five years, ten years, sometime down the road. So this was a real problem. I can't remember the number of times Franco came into my

[71] D. Bitton and C. Turbyfill. "Benchmarking Database Systems, a Systematic Approach" *Proc. VLDB*, Florence, Italy (1983).

[72] Rainbow was an all-new system (architecture, operating system, database, etc.) intended to replace the T16. It was eventually canceled, and a project named Crystal was started to build a low-end, easy-to-use system. Crystal in turn was replaced by Catalyst, a highly-integrated, transaction-oriented, easy-to-use client/server (PC/T16) system. In about 1990, Catalyst was spun off as a new company named Cooperative Solutions founded by Dennis McEvoy, his wife Kim Worsencroft (who had been the Catalyst project leader and visionary), and several other Tandem folks. They eventually built a product called Ellipse based on OS/2 and Sybase. Cooperative Solutions was bought by Bachman Information Systems, which is now marketing Ellipse.

office and explained to me how this Rainbow nonsense had to stop. We had to get serious and do a product. In about 1983, we had finally gotten to the point that we had critical mass, and we actually started to make some really good progress, I think. Franco was working with Andrea Borr and a couple of other people …

Jim Gray: Louise Madrid had come in from Britton-Lee via Esvel.

John Nauman: She came in after Franco left and came back. But in any event, there was a critical mass team going on. Andrea moved over from TMF to work on the underlying file system stuff, and we were actually starting to make a lot of progress, and then Franco went away. This was …

Franco Putzolu: ???

John Nauman: Yes, but I didn't know that at the time. This was real unpleasant. And then Franco came back, and Don [Slutz] joined us, and then things got a lot better.

C. Mohan: Why did he leave?

John Nauman: He didn't like his manager.

Tom Price: Kapali was going to make him rich.

John Nauman: I think it was the lure of a start-up.

Franco Putzolu: Yes, that's true.

Jim Gray: No, it was more complicated than that. There was this presentation; I was sitting there in a big auditorium. Dennis McEvoy, who's now head of Engineering at Sybase, stood up and talked about how wonderful Rainbow was going to be. And right in the middle of that meeting, Franco got up. He was sitting in the middle of the auditorium; he waded through a whole aisle of people; he marched down the aisle; he marched out; he went over to Esvel and accepted a job. It finally got to him.

John Nauman: I left Tandem before the NonStop SQL stuff got done, but it did get done, and from everything I've heard, it was an outstanding effort. The one thing I learned out of both the efforts that I worked on at IBM around FS and the Rainbow effort, and in every company I've ever worked for, it's the same situation: there's always the next-generation product. There's always the next thing that's going to solve all your problems, and that you should invest all your resources in, and basically stop working on what you're working on right now. Don't worry about that generation machine. That's the past; you have to look to the future. I have never worked in a company where that's worked. It is always the case that what you've got now – this is true of System R, as well – we had System R; that was something we should be working with. We shouldn't have been worried so much about changing the way SYSGEN worked, and getting rid of it, and a whole new hardware architecture, and a whole new software architecture, and objects that supported either relational views or network views or hierarchical views or whatever you wanted. We should have been trying to take slightly smaller steps but toward a much more attainable goal. That's awful to say, because it's so appealing to look at something and say, "We can change the world. We can go do something that's really important." But the problem is that there are very few of those that succeed. You look at all the successful software products I can think of – whether they're from IBM or Tandem or Microsoft – and what they are is the second or third time around of a product. Not the first time out where it's the glowing thing that everybody loves.

Mike Blasgen: Actually, one of the reasons we're here is that System R was one of the few times when a new approach was appropriate.

John Nauman: But not as a product.

Mike Blasgen: Improving the old approach is almost always the right thing to do. Improving IMS is almost always the right thing to do. But once in a while, there is an opportunity to do something new, and this was it.

John Nauman: But the difference was that System R wasn't IBM's future – FS was IBM's future, and Rainbow was Tandem's future, and whenever you've got your whole company bet on something, you start to lose … I view System R as a very good idea that then made a lot of progress behind the scenes.

Mike Blasgen: I understand the distinction; you're right.

John Nauman: I worked at 3Com for a while, and at the time I worked there, Ethernet was just starting to become something that people recognized as being important. 3Com went off in a number of different directions, with these brave new systems, all of which caused problems in the company until they came back and focused on what their core business really was. It was Ethernet. Now they'd sort of invented that, and then carried it forward.[73] System R was something that got invented, but then got moved forward in a very logical, reasonable progression. It wasn't the thing that changed the world overnight. Those are the ones that I think are real dangerous. That's sort of my experience at Tandem. Franco, do you want to talk about how the NonStop SQL stuff actually got finished and released?

---

[73] Actually, Ethernet was invented at Xerox PARC.

Franco Putzolu: It got finished actually pretty well. Let's see: we started in 1984 and we went to Beta in 1987, which I think is pretty reasonable.

Jim Gray: Can I interrupt you there and say something? We said in 1984, "It's going to take three years or four years, and you're not going to have a product until 1987." And Dennis McEvoy said, "What? I'm going to have to wait until 1987 to have a SQL system? Forget it." And so we gave him the fifty per cent confidence schedule, and the ninety per cent confidence schedule. The fifty per cent confidence schedule was 1986, and the ninety per cent confidence schedule was 1987. And he said, "Oh, OK."

John Nauman: But Franco's approach was that every time I go and ask him how long it was going to take, "Franco, when are you going to be done?" "I'll be done when I'm done."

Franco Putzolu: Yes, but the claim was we had to be out in 1987. We were off by about three months, which is not too bad, and I think it was a good system as far as the low-level engine; perhaps I am biased because I worked on it. I think it's still the best engine around. I mean it's the real engine that scales up as much as you want, recovers from failures in seconds, now has all sorts of on-line utilities, has locking done right – no other system does locking right. *[laughter]* On the other hand, there are some minuses, of course. Functionality is minimal: really basic, basic SQL. We made some major mistakes and I was responsible for some of them. We didn't try to really stick to ANSI; we tried to make it integrated with GUARDIAN: our own naming; our own security model. I think I'm guilty for the naming part, and that was a major, major mistake. It was fun to work on it. Jim, do you want to say something on it?

Jim Gray: Yes. Probably the key things about it were that the first release was for OLTP; first release was for transaction processing, and that was about 1987. People then went on and did a parallel SQL that shipped in about 1989, and I think Mike Pong could talk about that. In the last four or five years, they've worked hard to make it on-line, high-availability. So the particular things they've done is to make it possible to add indices while the database is being updated; to reorganize the database while it's being updated or accessed. These are interesting algorithms. They don't have referential integrity; they don't have triggers; they don't have foreign keys; and so on. But on the other hand, they have a lot of things that are useful for day in, day out data storage and data retrieval. It's going to be interesting to see what happens next.

Franco Putzolu: The engine is good, and it's actually interesting that back in 1989 we had parallel execution, and for about three or four years Tandem didn't do anything with parallel execution, and then all of a sudden they discovered this big DSS[74] market. But it was kind of late; other people had made the same discovery by that time.

Jim Gray: Mike Pong, do you want to say something about …

Mike Pong: I joined the NonStop SQL project to work on the optimizer a couple months after it got started back in 1984. The part that I remembered most was that Franco had finished a large part of the executor when I joined. This stuck in my mind because everyone one else was still trying to figure out the complete design! In the first release, we did not take advantage of Tandem's parallel architecture for intra-query parallelism. Soon after we shipped the first release, another developer and I started to work on intra-query parallelism with the help of Jim and Franco. The design and implementation took about two years. When it was completed, we were very excited to actually see linear scale up and speed up for large queries. Unfortunately for Tandem, marketing for the feature did not exist until about two years ago.

Pat Selinger: Any more Tandem stories? Bob Jolls?

Bob Jolls: I can't add to that.

Pat Selinger: Perfect as is. Oracle's next then.

Oracle

Roger Bamford: Franco, do you want to start with Oracle?

Franco Putzolu: Well, I interviewed before you.

Roger Bamford: That's right. Well, you know, I was at Esvel, and I got kind of burned out, and I went back to IBM, which you may not know. I worked in the Scientific Center for a while.

C. Mohan: This was Palo Alto, is that right?

Roger Bamford: Yes, Palo Alto. Well, there was this guy there doing an expert system – Harry Rhinstein I think his name was. He built it first in APL, and he was porting it to Pascal and they really didn't know how to do factoring, so every routine was copied many, many times and there'd be a few lines of change in each copy.

So I was looking for a job, and went to the restaurant at the end of Page Mill Road – at Foothill Expressway, and I was meeting this head-hunter, and it was a woman. So I'm

[74] DSS stands for Decision-Support System.

looking around for a single woman standing around looking for somebody, and there was a woman standing around looking for somebody, so I started talking to her. It was an Oracle employee, and she goes, "Oh, do you know Jack Harper?" who, like many sales people, had a very brief employment at Esvel. He was at Oracle, and I started talking to her. Don [Slutz] had talked to Oracle and he said, "Check these guys out." So when I got back to my office, I thought, well maybe I'll give Oracle a call. So I called Information and got their number, and I was thinking of calling them – you know: working myself up for it – the phone rang. I picked it up, and it was Jenny Overstreet, who was the assistant to Larry Ellison, calling me up to see if I wanted to go up for an interview. This was in 1984 I guess, so I got on my motorcycle and I rode over to Oracle, which was only a little ways away, and I got rained on in the process. I had a nice interview with Bob and Larry. It struck me that Larry had a lot of charisma and energy, and definitely had the drive for success. Bob Miner was a really nice guy and very smart too. And so I went to work at Oracle. It was funny, because when I got there, I'd come from IBM and Esvel, where the customer's data's sacred. The first day, walking down the hall, Ed Oates, one of our early employees, said "Oh, so-and-so's database got hosed again." *[laughter]* So we sent them out the latest version of the system. The only way you could use Oracle at the time was you'd export all your data, every day, and then plan on having the database get hosed, and then you'd load it back in again. And they were really happy. I mean, customers didn't like this, but they didn't mind too much, because it was really being used not as a transaction processing system at all; it was being used as an early decision-support system. And the software was really simple. It was feature rich: there were lots of these nifty built-in functions. There were a lot of datatypes that were very useful. It was there. It was a language that IBM had endorsed, so Bob and Larry figured out it was going to be the next standard language. At the time that I joined they were embarking on this portability strategy, which actually made a lot of sense, because hardware was expensive in 1984, and by making the software portable, you could essentially commoditize hardware. Which is what Oracle did, and that created a lot of revenue potential for Oracle, because they got back the money that the customers were saving by going to open systems. It cost you some other stuff to go to open systems, as they later discovered.

Tom Price: It transferred money from the hardware vendors to Oracle.

Roger Bamford: Right; to Larry, in particular. At Oracle for a long time, there was a running joke of Stu Feigin, you know, one of the other early founders of Oracle; I guess he was the first employee. He'd always say when we'd go out to lunch and spend a bunch of money on lunch or dinner, "Well, it's only money, and it's only Larry's money." That used to be a running joke.

In terms of System R's influence on Oracle: some ideas came from Esvel, and some of those came from System R. But the original code they'd written was really like somebody had a paper that described the language, and they had a computer and nothing else. And you could kind of tell that it had been coded … I mean, all the data structures were like, "Well, there's this query block, and then the query block has a select part, and the select part … and it has a this and a that." There was a totally straightforward mapping from the language directly onto hardware; very little intermediate stuff. I mean, if there were some indexes, they would use them. Tom [Price] used to be working on these papers with all the different join strategies, analyzing them. They never read any of that stuff. It was what Larry called an AI optimizer, which is now called a rule-based optimizer. So it's actually quite a long time before we even had a cost-based optimizer.

Franco Putzolu: It's really true when you look at Oracle code that there is no System R origin.

Roger Bamford: No, they just ground right through it.

Mike Blasgen: It's also historically correct, because there was no access, first of all: they wouldn't have had access to the code. And it's in parallel; it's not like they succeeded in history; they didn't come second, they came at the same time.

Roger Bamford: Yes, actually Oracle had an earlier SQL product than IBM. IBM invented the language, but Oracle shipped it first.

Mike Blasgen: I don't know when the first Oracle code shipped.

Jim Gray: 1979?

Roger Bamford: Version 2. The first version of Oracle was Version 2, because they figured nobody would buy Version 1. *[laughter]* It's true; another brilliant move on Larry's part.

Brad Wade: Well, when was Ted Codd made an IBM Fellow?

Mike Blasgen: 1976.

Brad Wade: I remember the reception they had for him in the Building 28 Cafeteria. At that time he said, "It's the first time that I recall of someone being made an IBM Fellow for someone else's product." It was Oracle's.

Mike Blasgen: It was very early.

Roger Bamford: When I got there, they were on Version 3, which had been almost finished by a guy named Bruce Scott, who later went to Gupta; he wrote a lot of the expression-evaluation code in the first and second version; I guess in Version 3. Version 2 had been written in assembly language for PDP-11; Version 3 was written in C. He did that, and he wrote this really nice beautiful, compact code – very well structured; a lot of it's still around now. The next version I think worked really well, and accounted for a lot of the growth. Then we kind of went on from there. There was a decision support and distributed query – Version 5 – went out after that. And 6 was a rewrite for transaction processing.

Franco Putzolu: How much was rewritten in Version 6?

Roger Bamford: Well, kind of the equivalent of the RSS, so that would be about half. And it was all thrown out, and written again from scratch.

Jim Gray: The same data structures on disk though, right?

Roger Bamford: No, volume formats changed. Everything's completely different. Like rows in Versions 3 and 4 and 5 were concatenated in blocks – you know: byte, byte, byte, byte, byte, byte, byte … with no index or anything. So if you wanted row sequence number twelve, you'd start at the beginning of the block, and you'd start scanning over columns, and rows …; and eventually there'd you'd be, right where you were looking for. *[laughter]* So how do you update a row and make one of the columns bigger? Well, you shift the rest of the block to the right …

???: Oh, my god.

Roger Bamford: Right, so we changed that in Version 6. I was kind of the lead designer in 6. When you were saying about the SARGS … at the time, there was no abstraction between what was the RDS and the RSS. There was an interface, but it was violated all the time. One of the things that you would do is you'd be deep in the middle of some block, looking around, and you'd call back through these upper layers, and it would do some SQL thing, like a sub-query evaluation. And since Oracle had consistent reads, it was OK to do that, because you could be holding this block and it wasn't preventing somebody from changing it, because they'd get their own copy and just change that. That stuff we preserved, because it turned out to be OK. But the logging, and the recovery and the way consistent read itself worked and all the locking – basically everything to do with data management was replaced in 6. And then, since then on, we've just been building on that pretty much.

That's kind of the Oracle story. Does anybody have any questions?

Don Slutz: Larry started out kind of copying System R as-is. How long did he kind of think of going that way versus shooting ahead?

Roger Bamford: What do you mean?

Don Slutz: You know, adding more function beyond. He started out directly with System R.

Roger Bamford: Well they took the published SQL specification and they built that, and they added stuff that customers wanted.

C. Mohan: Even Version 1 you had more user-defined functions, and so on?

Roger Bamford: No, Version 2 was all assembly language; I don't know what was in there. But 3, yes, they built a bunch of stuff into that, more functions for this …

Franco Putzolu: When did they add the forms, you know forms tools for writing applications more easily?

Roger Bamford: Oh, yes: IAP, I think: Interactive something-or-other. There was a precursor of SQL*Forms, I think went out in 3 or 4; I think it went out in 3. Yes, they hired this guy – this is typical Oracle, actually – they hired this guy straight out of school; a smart guy; he'd done a little programming. And the first thing he did was the UFI thing, and then he built IAP, which is this forms-based application.

Bruce Lindsay: Like SREDIT?

Roger Bamford: There were blocks, and then there were tables; it was like a table editor with a lot of escapes for transitions from one table to another. Nobody at Oracle was held back by lack of experience. *[laughter]*

Mike Blasgen: I remember seeing the Oracle system running for the first time at some computer conference like SIGMOD or something. There was a demonstration area, and in a little booth was Larry Ellison and one other person, showing off their system. I introduced myself (Jim Mehl was with me) and Larry knew about System R and about our work and he gave me a little demo. I was impressed, because it was obviously simple, in the sense that … well, you'll see why in a minute. It seemed fast. He loaded the database, queried it, and updated it, all in a few seconds. It was – I don't know how many – maybe five-hundred records. And it loaded instantly.

Roger Bamford: What year was this?

Mike Blasgen: I don't remember; probably 1979 or 1980. The thing that impressed me the most was that it ran on a

little PDP-11. The machine looked to be the size of a carton of cigarettes. It must have been an LSI-11 version of the machine, if my recollection of the size is correct. And System R at the time in most of our joint studies and at IBM was running on 168s. Now a 168 is only maybe the power of a 486DX2 or something, but the fact of the matter is it was a huge machine which would probably not fit in this room.

Jim Gray: It was water-cooled.

Mike Blasgen: It was a huge computer. And Don [Chamberlin] was talking about, "Well System R wasn't so big; it was only 1.5 megabytes of code and 87 thousand lines of code." But it did in fact run on a computer that filled this room. And the little Oracle thing ran on a machine that was the size of a carton of cigarettes. I remember because it was right there, stuck sideways onto the shelf. It was up on a little shelf above the desk, attached to a glass teletype. And that was all that it needed, and it ran fast, and I thought, "Simple, fast, cheap; that's neat. People will buy it." Exactly for the application that Roger mentioned: the query application, for decision support.

And the rest

Intergalactic dataspeak: SQL standard, Open SQL, ODBC, DRDA

Pat Selinger: Jim, I have your name next to Sybase, Informix, DEC, Teradata, Ingres, Britton-Lee, and Microsoft.

Jim Gray: Gee, what to say? I'm not going to say anything about most of them. I think actually what I'd like to do is talk about the SQL standard …

various: No.

Jim Gray: I'll do it anyway! Here is the original SQL manual (from System R). Just about 40 pages in Courier 10 font with lots of error numbers and lots of white space. It was real simple. Relational was hot, so ANSI started up a Relational Database Task force to define a standard. There was a DBTG task force that had a CODASYL network data model, and they were trying to standardize the network data model, and Don Chamberlin talked about how much fun it was to study the network data model. There were these things called currency indicators, and people loved them. You would do a query and it would set a currency indicator, and then you could fetch the thing that was pointed to by one of these currency indicators. In SQL terms, for every table there was a cursor. You could say the magic word, and it would change the cursor for that table. It would also change the global cursor. Have I got it sort of right?

Don Chamberlin: Yes.

Jim Gray: But you couldn't have two cursors on a table. So if you wanted to join a table to itself, then you'd have to remember where you were, and then go get the other record. So the Database Task Group basically was in big trouble; nobody really wanted to standardize this thing. So it was a standard that was this zombie; it was wandering around; I guess it got standardized maybe in 1990 or something like that? About the same time the first SQL standard came out, there was sort of this *quid pro quo* that we'll standardize DBTG and relational at the same time. But there was this relational task group that was wondering around, and they were getting in deeper, and deeper, and deeper water; lots of deep water, right? They'd done their own database language. At a certain point, Phil Shaw showed up at one of these meetings and said, "You know, you could do this," and he handed them something that was approximately this [holds up IBM's early SQL manual]. This is again ten-point type, single-spaced now, instead of double-spaced. Still a lot of white paper. These people, who were in hopeless deep water, said, "You're right; we could do this, and this is the only way we're going to make progress," because they were not making progress in any other way. So they glommed on to the … and I think this was the design document that Don was chairman of this committee in IBM that was sort of the … you were the Pope of SQL, or something like that? Have I got it twisted?

Don Chamberlin: Bob Engles had a lot to do with this[75]. I believe that most of the words in that book were written by Bob Engles. And what Bob Engles did was to study System R and write a formal specification for exactly what it did, warts and all. So there were all sorts of peculiar rules that were non-orthogonal: you couldn't do a GROUP BY if you also did a UNION; things like that. And there's no special reason for any of those things, except somebody didn't have time to build them or something like that. *[laughter]* So Bob Engles studied System R, and he's very meticulous and very precise, and he wrote down exactly what it did in a very formal sort of way. I think that's the document that you're holding right there. And then the standards committee kind of blessed it and they said, "This will be our standard."

Jim Gray: Only way we can make progress.

Don Chamberlin: They kept all the warts, too. They didn't try to clean any of it up.

Jim Gray: Right. No discussion of how to spell NULL. Chamberlin came back from an IBM Santa Teresa meeting one day, and said, "We spent the entire day deciding how we should spell NULL. Should it be ABSENT or NOT KNOWN or NULL or ?" The ANSI SQL guys did not mess around like the Santa Teresa guys. They took this, and this is essentially SQL 1 [holding document] – the standard. And this is SQL '86, is that right? ANSI – the Americans proposed this standard, but the Americans are just part of the international organization. The international organization said, "We'll make you a deal: we'll swallow this piece of junk if you'll swallow our referential integrity design" (foreign keys). And so there was going to be an

---

[75] Bob Engles died June 22, 1995. Roger Miller notes: "Bob was the authority on SQL standards; he was the author of the original "SQL Control Document," which provided the foundation for the SQL ANSI/ISO Standards. He was the DB2 representative to the SQL Language Council since its inception, authoring many papers and articles, and providing consultations to the world-wide SQL community. He was the designer for many DB2 features, including referential integrity, code pages and character sets support, date time data support as well as the latest SQL '92 work. Throughout his career at IBM, and even recently as his illness progressed, he was an inspiration to many of us with his commitment to DB2. He was one of the key contributors to DB2's success and we will miss him."

appendix that came later, and the international standards body (ISO) would swallow this [SQL 1] if they would get to write the foreign key design. And so they wrote the foreign key design, which was basically SQL '89, so there's an addendum. Am I getting this right? Straighten me out if I've gotten it wrong.

So we're up to 1989; we've got something like this [demonstrates], and an addendum which was pretty short.

C. Mohan: He's eager to get to the next part. *[laughter]*

Jim Gray: And in fact, here is the whole enchilada [demonstrates]. And then we get to SQL 2, and here is SQL 2 [demonstrates], and it's a lot bigger. Actually, I don't have SQL 2 very easily; I apologize. But it's on this scale, OK? And what it has, is data definition; it has constraints; it has time; it has … what are some of the good things?

Bruce Lindsay: Outer join.

Jim Gray: Outer joins. Sort of more complete. But it's very big; it's order five hundred pages.

Bruce Lindsay: It comes in three languages, too.

Tom Price: Did any of the referential integrity make it into SQL 1?

Jim Gray: Well, it was SQL 1.1.

Tom Price: And is it close to what DB2 implemented, or is it different?

Jim Gray: It's Chris Date's design, is the way I think of it. You know, they have cascading, and RESTRICT, and …

So now the SQL committee has a life of its own, and it has SQL 3. Now this is the current enchilada that is SQL 3 [demonstrates a three-volume set of books]. And this, you have to appreciate, is nine-point type, and it is very, very dense, and it's just full of stuff. I think it's fair to say that most of us don't understand what's in there. I think Don Chamberlin maybe has spent a lot of … he understands pages of it, I'm sure. And they're now trying to take SQL 3, and break it into two parts: SQL 3 and SQL 4. SQL 3 is probably going to get approved somewhere in 1997? And SQL 4's up into the next millennium, which I really think is a nice way of describing where it is.

Something else that happened is ODBC[76]; I'm coming to the Microsoft part. Something else that happened is that while we – Don Slutz and I, and a fellow named Rao Yendluri – were at Tandem, we said, "We really have a serious problem. We've got this database engine; this thing that stores bytes. It remembers things. But getting stuff into this computer and getting stuff out of it is virtually impossible. We've got no tools; we need to get tools. We can't build tools; we don't build tools. We want everybody to build tools that go to our system. How are we going to get everybody to build tools that go to our system? Well, we need to have a standard way for people to get to our system, just like getting to Oracle or getting to Sybase. So Plan A: we'll pretend to be Oracle. Everybody's going to build tools to go to Oracle. But that's kind of embarrassing, because that sort of puts us at Oracle's mercy. We have to masquerade as Oracle, and they can do things to shaft us, and so on. Plus, their externals aren't public. Sybase in fact has something called Tabular Data Stream, and we could masquerade as Sybase, and be a system that eats Tabular Data Stream and spits out Tabular Data Stream." So we thought about that and said, "What the world really needs is a client/server standard," because the tools vendors want to have a standard that they can program against, and know that their tool will work with anything. So the tools guys want to be able to go to every database server, and the server guys want every tool to come to them. So we said, "What we need is an intergalactic dataspeak." An Esperanto that would go on the wire, that would allow everybody to talk to everybody. I believe at the same time, in this period, IBM folks had exactly the same problem. They said, "We've got great servers, no tools; we need intergalactic dataspeak." So Slutz and Rao Yendluri and I wrote a white paper called "Open SQL". We said, "What the world needs is Open SQL, which is a wire protocol: how to talk SQL across the wire; how to talk tables back." We talked this up, and we went to the Sybase guys, and the Sybase guys loved to talk to us. Every time we talked to them, a press release would come out, about how Sybase and Tandem were working together on this problem. No code came out, just Sybase press releases. And every time we met, they said, "If you give us a hundred thousand dollars, we'll give you some code." But it was really very strange to work with them.

Don Slutz: They said it had to be TDS[77], too.

Jim Gray: Right. And, "Incidentally, whatever it is, it's ours; we'll just standardize what we've got. We'll minimize the effort we have to put in." So at a certain point, we realized we were being had by Sybase; we were pretty slow. All of a sudden, the skies darkened with executives from Digital Equipment Corporation. A cloud of DEC vice-presidents appeared on our doorstep at Tandem. They had gone through the same thought process, and said, "Rats, we need Open SQL." So they said, "Everybody has this problem; we're going to publicize it," and they formed the SQL Access Group. Informix was a founding member. We put off founding the SQL Access Group for, I think about three months, while IBM decided whether they wanted to

[76] ODBC stands for Open Database Connectivity.

[77] TDS stands for Tabular Data Stream.

join or to compete with the SQL Access Group (they had a plan called DRDA[78]). In the end I think they said, "Oracle, Informix, Tandem, all these guys: they'll never make any progress." I'm putting words in their mouths, but I think they said, "We'll make a lot more progress ourselves," and in fact they made pretty good progress. They came up with something called DRDA, which was a competitor to what the SQL Access Group did. So the SQL Access Group ground and ground and ground, and produced something called a call-level interface, and tried to build on top of some international standards, and the net that came out of this is something that is called ODBC[79], which is sort of an implementation of this. It is the standard way for clients to talk to servers. So you send me some SQL; what it means is defined by those multi-volume books we just saw. And so this is sort of how you make SQL requests, and how you send stuff back. And the scary thing is that a lot of people are learning how to write this stuff. Learning to program in this thing is a real undertaking; I kind of worry. But the good news is that the only people who have to learn to program in that way are the people who write all the tools. So virtually all the tools vendors are making ODBC drivers, which is to say end-users draw stuff on the screen and you make circles and arrows and say things that are pretty visual. The tools translate the GUI into SQL statements, and they use that call library to ship requests down the wire to a server. The server does its thing; sends tables back, and the tables do stuff on the screen. ODBC is beginning to have stored procedures and various other things.

Bruce Lindsay: I'm really confused because ODBC is not a server protocol.

Jim Gray: Right, it's an API.

Don Slutz: There's no DRDA involved.

Bruce Lindsay: At the beginning you said you needed a standard way to put things on the network that will get to the server, and you don't care which server it's going to; it's on the network and it works. And ODBC is not that protocol.

Jim Gray: The tools vendors can write against this interface, and the tools vendors don't have to worry. Somehow, magically, bytes will get shipped down; bytes will get shipped back. And all the tools vendors run, of course, on ODBC platforms.

Tom Price: Well, there's ODBC drivers for things other than Microsoft.

Jim Gray: Right. The dual of what's happening is that one of the things in ODBC is that you can ask the guy at the other end, "Who are you?" Good answers to come back are, "I'm Oracle," or "I'm Sybase," or "I'm Microsoft SQL Server." The tools vendors negotiate and, if it's Microsoft SQL server, they do things special, and there is a transport that goes to Microsoft SQL Server. There's another transport that goes to Oracle; there's another transport that goes to Sybase. And Microsoft SQL Server and Sybase are very similar. So we're beginning to get intergalactic dataspeak. This hasn't solved Tandem's problem; Tandem ends up now having to masquerade as one of those three characters. At least it's solved the tool vendors' problems, which is that they have a standard programming interface. You're right, and in fact maybe we should now tell the DRDA story?

Pat Selinger: Go ahead.

Tom Price: IBM doesn't support ODBC yet, do they?

Jim Gray: Well they do in the UNIX world. The RS/6000 world supports ODBC. I don't know if there's an ODBC driver in the MVS world. I think there is in the AS/400 world.

Pat Selinger: Sure there is.

Don Slutz: In the SQL Access Group, IBM never joined, but ??? came, and they had him send Frank Pellow, who's IBM Toronto, so he was always there.

Tom Price: And if you have Sybase or Oracle, do they provide drivers, or do you have to get them from third parties?

Jim Gray: When ODBC first started shipping from Microsoft, they put in drivers for Oracle and for Microsoft SQL Server, which is to say Sybase, and a few others, and they began to get a lot of push-back from customers about the versions and so on. So I think at this point you actually have to get the driver from the provider; that Microsoft doesn't ship them, but you can download them.

Pat Selinger: IBM provides versions for themselves, and there're companies likes Q+E that have them.

Shel Finkelstein: The SQL standard decides how … the foundation part of the standard now has these things called parts, one of which is Call-Level Interface, which is awfully close to ODBC. So it's not just that ODBC is a Microsoft thing; ODBC is part of a standard.

---

[78] DRDA stands for Distributed Relational Database Architecture.

[79] Microsoft Corporation. *ODBC 2.0 Programmer's Reference and SDK Guide.* Microsoft Press (1994).

Jim Gray: And it's actually in the status of draft international standard, likely to get approved this year.

Shel Finkelstein: And there's also persistent stored modules. There's this new part proposed for temporal, plus there's a separate standard that's being worked out for multimedia. So what Jim has over there is just a small part of all the wonderful things that are going on. I got to go to Oklahoma City one week after the bombing for a SQL Standard meeting ???

Jim Gray: This is the SQL Reunion, and I think probably one of the important things to mention is how it's turned into intergalactic dataspeak. It's how clients talk to servers if they want to send structured data around. There is another intergalactic dataspeak called IDL, which is for remote procedure call, and a third one called HTTP, which in fact is being used for the Web and Mosaic, and it looks like HTTP is going to win in the end. The surprise for the future is HTTP wins.

So DRDA[80] is the approach that IBM took, rather than going with the SQL Access Group. It is much more concerned about what the on-the-wire protocol is. So it's what's called a formats-and-protocol. The message format's on the wire. What you say [gestures?] and the protocol: I say this, you say this. So we abbreviate that formats-and-protocols, or FAP. In fact, ODBC has no FAP; it's a procedure call, and then what happens underneath is a mystery, magic. In fact, what happens underneath is a driver from one or another vendor. This is a terrible situation unless there is only one kind of client, and only one version of each server, because then you just get the particular thing; otherwise you end up with an N-squared problem. One of the surprises to me, and I think to many people, is that the number of kinds of clients has dropped off, mostly because of the success of Windows. At any rate, DRDA is an IBM standard, and it's supported by DB2 and supported by the IBM products and …

Pat Selinger: And twenty other vendors.

Jim Gray: And twenty other vendors, that's right.

Roger Miller: And X/Open.

Bruce Lindsay: DRDA fits underneath ODBC. You could use it for the ODBC stack.

Jim Gray: Could be. It's interesting; my impression is that the twenty vendors all have been paid to support it. I talked to the people at Informix, and they said, "Yes, we support it because IBM paid us to support it."

Pat Selinger: I don't believe that's the case. That is not the case as far as I know.

C. Mohan: No.

Roger Miller: I'm pretty sure that we did not pay.

Jim Gray: OK.

Roger Miller: We made it as easy as possible. We gave classes in attractive places and provided consulting. We certainly worked hard to get vendors to use DRDA.

Jim Gray: But they had to write the code.

Roger Miller: We had a nominal license, a few thousand dollars, less than the class would have cost, to license some pieces of the code. But we worked to make DRDA easy to implement and probably twisted some arms, but I don't think we paid anybody.

Jim Gray: So do you think it's going to be successful? Is it going to be the intergalactic dataspeak? Is it going to be the FAP do you think?

Pat Selinger: Who knows? It's certainly gaining some popularity among people who are performance conscious.

Bruce Lindsay: That's the interesting thing about ODBC; it seems to have ignored the performance issues. It's a strictly dynamic interface; there's no way they're running static SQL through ODBC.

Jim Gray: Actually, it has stored procedures, so …

Bruce Lindsay: Well, stored procedures and bound procedures are not quite the same thing, but close enough.

Jim Gray: They're better. *[laughter]*

Teradata; Ingres family: Relational Technology, Britton-Lee, Sybase, Microsoft

Jim Gray: So, let's see. Teradata. So here's just some background. There was a guy by the name of Phil Neches at UCLA, and he said, "We ought to do parallelism on commodity hardware." He fell in with some people at a startup, and they started this company, and I guess in about 1984 shipped the first parallel SQL engine. It's very similar to the Tandem story: it's a sort of one-trick pony. It doesn't have referential integrity; it doesn't have all the fanciness. You give it SQL; it gives you back the answers, very fast, and presumably cheaply, because it's running on Intel processors and on commodity disks. Teradata got bought by

[80] IBM Corporation. *Distributed Relational Database Architecture Reference*, SC26-4651.

NCR; NCR got bought by AT&T; and AT&T last I heard …

Along those lines, there was this whole other development, which was the INGRES project at U.C. Berkeley. The INGRES project had a language called QUEL. They started a company that implemented QUEL. QUEL fought SQL tooth-and-nail, and explained how QUEL was better than SQL in many different ways, and in fact it is better at doing aggregates. There are lots of areas where QUEL is better. Some people at Ingres now feel that the reason that they were less than successful is because they fought SQL rather than embraced it, so this gave Oracle a chance to differentiate themselves. The fact is that …

Mike Blasgen: Just as a point of time: I had a conversation on the phone with Stonebraker while I was living in Washington, and I left Washington in June of 1983. So it's obviously prior to that. I said, "I think Oracle is going to do well." He said, "Why is that?" I said, "Because they are one of the few who support SQL besides IBM." He said, "Well that status won't last more than a few weeks. Everybody's on that; that's done." So by, I would say, the end of 1982 or the beginning of 1983, they were far over that; they had made that decision. I don't know when they shipped their first code.

Tom Price: Although the first code they shipped was SQL on top of QUEL …

Mike Blasgen: It was see-QUEL. *[laughter]* That's right.

Tom Price: Which made me nervous about buying it.

Jim Gray: And so there was that thread. And spun off from the INGRES project was a Britton-Lee group. And the Britton-Lee group included Paula Hawthorn and Bob Epstein and Mike Ubell and probably a lot of other people. And they built a database machine[81]. In that era, there was this whole notion that you could really do much better by building a special-purpose piece of hardware and a special-purpose operating system and then a database system. Build up from the bare metal and it's going to run a lot faster. I think Roger mentioned that that was part of the Esvel concept as well. Louise Madrid was another …

Roger Bamford: I think we really believed you could get the revenue for it; I don't think we really believed that it was cheaper to build; it was just easier to sell.

Jim Gray: And the performance would be better. I think that was one of the arguments, that you couldn't get good performance on general-purpose; the special-purpose would beat …

Don Slutz: Lot of special hardware.

Jim Gray: They had an accelerator, which was their hook … The Britton-Lee guys in turn spun off Sybase, and Sybase came out. The key thing about Sybase as far as I can tell was they ran on UNIX; they didn't use any of the UNIX services except a single process. They used raw disks; they took a single process and multithreaded it, and ran SQL inside of that, or actually ran DB-Library. They were not very SQL enthusiast or compliant. They had the QUEL tradition; they were from INGRES. So the key thing that allowed them to be successful is they had great performance. You would send one request in; it would work all inside of this process; no operating system dispatches, no operating system I/Os, just raw disk I/Os. So they were like a factor of three better than everybody else in terms of performance. They managed to establish themselves as the client/server, open, database thing.

Tom Price: They did a deal with Microsoft.

Jim Gray: And Microsoft took their code and sold it on OS/2. The reason for that was that about 1986, IBM was trying to take over the PC market, and they had their own operating system – OS/2 – they had their own hardware. Microsoft said that they had to somehow protect themselves against something called OS/2 Extended Edition. There was going to be this thing called OS/2, which was basic OS/2, and then Extended Edition, which was going to cost hardly anything more, was going to have a database system in it, and compilers, and query – QBE was going to be built into it, and all sorts of stuff. So Microsoft felt they had to have something like that. So they went to Sybase and said, "We'll get our SQL engine from the Sybase guys, and that will be our Microsoft Extended Edition." And Microsoft remarketed Sybase in the OS/2 world. The relations between Microsoft and Sybase were not warm or cordial. When it came time to port Sybase to NT, Sybase let Microsoft do the job. And then there was a divorce at some point, similar to the IBM divorce about OS/2, that IBM would do OS/2, and Microsoft would go its own way. There was a similar divorce vis-à-vis Microsoft, where Microsoft now owns the Sybase code, so the Microsoft SQL Server now is going its own way, and they've made it more SQL-compliant, and they're adding GUIs to it, and so on. It's now a major force in this whole database world. And the thing that's driving everybody crazy I believe in the database world is, this thing is very cheap. It's, order, five thousand dollars for a server, as opposed to a hundred thousand dollars for a server. This server is capable of doing hundreds of transactions a second. Scary. Pat, did I … ?

[81] R. Epstein and P. Hawthorn. "Design Decisions for the Intelligent Data Base Machine" *Proc. National Computer Conference*, Anaheim, California (May 1980).

Informix

Pat Selinger: Informix.

Jim Gray: Informix. I don't know much about Informix; I don't know what their history is, so I think I won't say much. Does anybody know any Informix history?

Tom Price: The only thing I've heard is it's rumored that their latest product is really quite good.

Jim Gray: It is. I actually know about their current products. I don't know about the history of it. I do know that when Tandem was remarketing a UNIX box, they went over and, I don't know, maybe Don [Slutz] was one of the people who went over and talked to the Informix guys.

Tom Price: I was part of that.

Jim Gray: You were? When you went, the rumors that came back were that they were not very informed about how to actually do things.

Tom Price: Yes, for instance they didn't know how to do record locking, so Franco told them something about it, and they apparently went off and did it. *[laughter]*

Mike Blasgen: Franco, raised in the IBM tradition where telling somebody how to do it poses no risks that they would actually do it. *[laughter]*

Jim Gray: We wanted the Informix guys to have a good product, because we were remarketing it. We were going to be reselling it, so we wanted it to be good. We thought it was our charter to help them out.

Mike Blasgen: Thank you all.

The end

Index

ADABAS • 31
ALLBASE • 44
Altman, Ed • 17
Ampersand • 29, 38
Annapurna • 6
ASK • 11
Astrahan, Morton
  first author • 22
  Phase Zero • 17
  remembered • 6
AT&T • 55
Auslander, Marc • 17

Bachman Information Systems • 45
Bachman, C.W. • 9, 10
Bacon, Glenn • 8
Baker, Jerry • 39, 43
Bank of America • 20
Berkeley
  alumni who worked on System R • 11
  INGRES project • 11, 55
Bjørner, Dines • 8
Blasgen, Mike
  implemented ET1 benchmark on System R • 20
  managed Database Systems • 21
  managed RSS • 21
  moved to Washington DC • 21
  Rio trip • 12
  sold System R to Santa Teresa • 32
  worked on FS • 26
  wrote System R index component • 13
Blasgen, Sharon (wife of Mike)
  moved to Washington, DC • 21
  Rio trip • 12
  suppressed press release • 16
Boeing Aircraft • 21
Borr, Andrea • 45
Boyce, Ray
  at Yorktown • 10
  Boyce-Codd Normal Form • 5
  remembered • 5
Britton-Lee, Incorporated
  database machine • 44
  origins • 55
  predecessor of Sybase • 55
  successor of Ingres • 11
B-trees • 12

Cambridge Scientific Center
  built NEMIS on Phase Zero prototype • 14
  developed RM, XRM • 9
  installed System R • 23
  origin of shadow pages • 18
  XRM used by Phase Zero prototype • 14
Case, Dick
  George Radin's boss • 26
  SQL/QBE shoot-out • 34
Catalyst • 45
Chamberlin, Don
  at Yorktown • 10
  Query Game • 10
  RDS • 13
  SEQUEL • 14
  SQUARE • 10
Chandra, Ashok • 34
CICS • 10
CODASYL • *See* DBTG
Codd, Ted
  1970 CACM paper • 8
  appointed IBM Fellow • 48
  Boyce-Codd Normal Form • 5
  Codd-Bachman debate • 9
  GAMMA-0 • 9
  NULL • 19
  provinces • 10
  System R • 13
  Yorktown symposium • 9
Compare-and-Swap • 34
compiling queries • 19, 24
Computer Associates International, Inc. • 11
Cooperative Solutions • 45
Crus, Dick • 29
Crystal • 45
Cullinet • 44

Damerau, Fred • 23
Database Task Group • *See* DBTG
Date, Chris • 52
Daudenarde, Jean-Jacques • 23
Dawn Treader • 26
DB2
  Bob Jolls • 31
  John Nauman • 38
  Josephine Cheng • 42
  OLTP • 41
  origins • 29
  Plan B • 31
  Roger Miller • 39
DB-Library • 55
DBTG
  Codd-Bachman debate • 9
  pointer-based implementation • 17
  pre-System R experience • 10
  standardization effort • 51
de Jong, Peter • 33
Deckert, Ken • 8
DIAM • 8
Digital Equipment Corporation • 52
Ding, Ignatius • 44
DOS DL/1 • 30, 32
Doughty, Jane • 29
DRDA • 53, 54
DSL/Alpha • 8

Eagle
  ET1 benchmark • 20
  Jim Gray's participation • 28
  origins • 20, 38
  Plan B • 31
  preoccupied management • 30
Eaton, Jim • 8

Ellipse • 45
Ellison, Larry
  early Oracle demo • 49
  early strategy • 49
  financial success • 48
  interviewed Don Slutz • 44
  interviewed Franco Putzolu • 45
  interviewed Roger Bamford • 48
  read SEQUEL paper • 15
  System R error codes • 15
Endicott
  naming conventions • 27
  SQL/DS • 31
Engles, Bob
  DB2 DL/1 support • 42
  remembered • 51
  SQL standard • 51
Epstein, Bob • 55
Esvel • 44
Eswaran, Kapali
  Esvel • 44
  hired at San Jose • 17
ET1 benchmark • 20
Evans, Bob • 26, 42
EXPRESS • 21

Fagin, Ron • 19
Fehder, Paul • 17
Feigin, Stu • 48
Feldman, Jerry • 9
FNEXT • 25
Frame, Jim • 31
FS
  attack of killer minis • 20
  concurrent with System R • 18
  it's real attractive, but don't do it • 26
  most expensive development failure in IBM's history • 26
  second-system effect • 46
  termination triggered DB2 • 26
FULIST • 45
Future System • See *FS*

GAMMA-0
  concurrency • 17
  early relational systems • 9
  origins • 8
  pointer-based implementation • 16
General Automation • 20
Gomory, Ralph • 17, 21, 34
Gray, Jim
  Anon et al. • 20
  assigned to Palo Alto • 28
  concurrency • 17
  it's real attractive, but don't do it • 26
  MIPS Envy • 39
  NonStop SQL • 47
  ODBC • 52
  Open SQL • 52
GRID • 18
Gumaer, Bob • 28

Haas, Gary • 23
Haderle, Don • 28, 38
Halloween problem • 22
Harper, Lloyd • 42
Harrison • 32
Hawker Siddeley Aircraft Company • 22
Hawthorn, Paula • 55
Heller, Andy
  Santa Teresa • 32
  VSS recovery • 28
Hoover Dam • 13
HP • 44
HTTP • 54
Hursley Prototype • 9
Hurwitz, Alex • 23

IAP • 49
IDL • 54
IDMS/SQL • 44
IMS
  at Yorktown • 10, 17
  attack of killer minis • 20
  cash cow • 29, 30
  contrasted with DBTG • 10
  emulation on RSS • 18
  ET1 benchmark • 20
  Frank King's 1980 speech • 40
  pointer-based implementation • 17
  predecessor of Ampersand • 29
  predecessor of Eagle • 20
  predecessor of VSS • 28
  Santa Teresa's plan to replace it • 30
  support in DB2 • 41
  the tower next door • 40
Informix • 56
Ingres
  predecessor of Britton-Lee • 55
  predecessor of Sybase • 55
INGRES
  origins • 11, 55
  strained relationship with System R • 12
intergalactic dataspeak • 52

Jackson, Bob • 26
Jacob, John Paul • 12
joint studies • 21
Jolls, Bob
  DB2 • 31
  SQL/DS • 30
  VS/QUERY • 37

Karp & Miller • 18
Katz, Randy • 11
King, Frances (wife of Frank) • 6
King, Frank
  1980 speech • 40
  at Yorktown • 5
  came to San Jose • 6
  IMS support in DB2 • 41
  joint studies • 21
  System R error codes • 15
Kornelis, Sid
  DB2 Data Manager • 29
  DB2 DL/1 support • 42

La Hacienda • 36
Leonard, Homer • 42
Lindsay, Bruce
  Endicott naming conventions • 27
  found Franco Putzolu's only RSS bug • 24
  SYSTERR • 37
Liu, Leonard
  against publishing SQL • 16
  brought Franco Putzolu to San Jose • 17
  came to San Jose • 11
  fire and brimstone • 13
  managed Computer Science Department • 9
  promoted to Director • 21
  recruited Don Chamberlin • 10
locks
  in DB2 • 29
  in Informix • 56
  in NonStop SQL • 47
  in Oracle • 49
  in RSS • 18
  predicate locks • 17
Lorie, Raymond
  compiling queries • 24
  RM and XRM • 9
  shadow paging • 13

Madrid, Louise
  at Britton-Lee • 55
  at Tandem • 46
Malkemus, Tim • 29
McCarthy, John • 6
McEvoy, Dennis • 46
MDS • 44
Microsoft
  ODBC • 53
  SQL Server • 55
Miner, Bob • 48
Mohansic • 26
Mortenson, John • 39
MVS
  DB2 • 38
  Esvel port • 44
  FS terminated • 26
  ODBC driver • 53
  System R port • 19

naming conventions • 27
NCR • 55
Neches, Phil • 54
New York Mafia • 30
Nomm, Nick • 43
NonStop SQL
  hindered by Rainbow • 45
  released • 46
NULL • 19

Oates, Ed • 48
ODBC • 52
Oehler, Rich • 26
OK Corral • 33
one-liners • 10
Open SQL • 52
Oracle Corporation
  Franco Putzolu's first interview • 45
  Jerry Baker • 39
  originally Software Development Laboratories • 15
  Roger Bamford • 47
  sargable predicates • 43
  shipped SQL before IBM • 48
  Wisconsin benchmark • 44
OS/2 • 45, 55

Pajaro Dunes • 13
patenting software • 16
Pellow, Frank • 53
Peterlee Science Center • 9
phantoms • 22
Phase Zero
  origins • 14, 17
  prototypes installed • 23
  videotape shown at SIGMOD '76 • 36
Pong, Mike • 47
Poughkeepsie
  early System R users • 23
  FS development • 26
  not involved in SQL/DS • 30
Pratt & Whitney Aircraft • 21
predicate locks • 17
provinces • 10
PTREE • 42
Pugh, Emerson • 26
Putzolu, Franco
  advised Informix • 56
  assigned to Palo Alto • 26
  assigned to Santa Teresa • 28
  came to San Jose • 17
  first Oracle interview • 45
  NULL theologians • 19
  returned to Tandem • 46
  studied IMS emulation on RSS • 18
  terrified of predicate locks • 17
  went to Esvel • 44
  went to Tandem • 36
  wrote one bug in RSS • 24

Q+E • 53
QBE
  reimplemented as QMF • 35
  shoot-out with System R • 33
QMF • 35, 37
QUEL
  origins • 11, 55
  tradition at Sybase • 55
  used by Ingres to implement SQL • 55
Query by Example • *See* QBE
Query Game
  and QBE • 34
  invented • 10
Query Management Facility • *See* QMF

Radin, George • 18, 26
Rainbow • 45
RDS (Relational Data System) • 13
Reisner, Phyllis • 14
Relational Data System (RDS) • 13
Relational Technology Inc. • 11

Rendezvous • 13
REQUEST • 23
Research Storage System (RSS) • 14
Revelle, Ron • 44
RM (Relational Memory) • 9
Rodriguez-Rosell, Juan • 21
Rogers, Terry • 9
Rosenheim, Don • 36
Rovner, Paul • 9
RSS (Research Storage System) • 14
Rufus • 12

Santa Teresa Lab
  Bob Jolls • 30
  coding standards • 27
  compiling SQL • 25
  early System R users • 23
  exchange of people with San Jose Research • 28
  Franco Putzolu • 29
  Irv Traiger • 31
  John Nauman • 38
  Josephine Cheng • 42
  Roger Miller • 39
  SQL/DS • 27
Saranga, Mike
  bet Jim Gray 811 project would ship • 26
  decided to stop Eagle • 31
sargable predicate • 43
Schkolnick, Mario • 21, 25
Schneider, Pete • 26
Scott, Bruce • 49
Selinger, Bob
  benchmarking • 20
  NEMIS • 14
Senko, Mike • 8, 17
SEQUEL renamed SQL • 22
Serendipity • 6
shadow paging • 18
Shan, Ming • 29
Shaw, Phil • 51
Shoens, Kurt • 12
SIGFIDET • 10
SIGMOD • 10
  Phase Zero video shown • 14
  SEQUEL/DML paper • 15
Simons, Andrew • 9
SLI • 17
Slutz, Don
  Esvel • 44
  joined RDS project • 21
  joined Tandem • 46
  Open SQL • 52
  Oracle interview • 44
Software Development Laboratories • 15
SQL Access Group • 52
SQL Language Council • 51
SQL standard • 51
SQL/DS origins • 30
SQUARE
  evolved into SEQUEL • 14
  origins • 10
SREDIT • 23
STL • *See* Santa Teresa Lab
Stonebraker, Mike
  developed QUEL • 11
  Ingres added SQL support • 55
  INGRES origins • 11
  Rio trip • 12
Sybase, Incorporated
  Dennis McEvoy • 46
  Jane Doughty • 29
  Open SQL • 52
  origins • 55
  sargable predicates • 43
  sold SQL Server to Microsoft • 55
  Tabular Data Stream • 52
System A • 10
System Logical Interface • 17
System R
  bug and wish lists • 23
  code size • 23
  origins • 9
Systems Department • 8, 9

Tabular Data Stream • 52
Takao, Yoichi • 23
Tandem Computers Incorporated • 45
TDS • 52
TELE • 45
Teradata • 54
Tiberio, Paolo • 21
Tilman, Peter • 9
TMF • 45
TODS paper • 15
Traiger, Irv
  assigned to Palo Alto • 26
  assigned to Santa Teresa Lab • 31
  early San Jose work • 8
  FNEXT • 25
  managed RSS • 13
  mapped QBE to SQL • 34
Trivett, Gene • 34
Tsichritzis, Dennis • 12

Ubell, Mike • 55
UDL • 29, 30
UFI • 35
UNIX
  IBM ODBC drivers • 53
  Jerry Baker • 39
  Sybase • 55
  Tandem • 56
Upjohn Pharmaceuticals • 21

VM+ • 6
VS/QUERY • 37

Wang, C.P. • 17
Watson Research Lab • *See* Yorktown
Watson, Vera • 5
Weick, Steve • 26, 28
White Plains • 30
Williams, Robin
  came to San Jose • 11
  managed Database Systems • 21
Wisconsin benchmark • 45

Wong, Gene • 11
Work, Thomas • 28
Worsencroft, Kim • 45

XRM
  developed at Cambridge Scientific Center • 9
  origin of shadow pages • 18
  pointer-based implementation • 16
  used by Phase Zero prototype • 14
  used by QBE • 34

Yendluri, Rao • 52
Yorktown
  early projects • 10
  early System R users • 23
  Office By Example • 36
  Programming by example • 33
  Query by Example • 33
  Ted Codd's symposium • 9
Yorktown Mafia • 6, 30
Yost, Bob • 29
Yothers, Jay • 38
YTABLE1 • 25

Zloof, Moshé • 33