RDBMS Workshop: IBM

Moderator:
Burton Grad

Recorded: June 12, 2007
Mountain View, California

CHM Reference number: X4069.2007

Table of Contents

RDBMS Workshop: IBM

Conducted by Software Industry SIG – Oral History Project

Abstract: A number of ex-IBM employees covered a wide range of topics about how IBM Research developed SQL and System R and the role that IBM's Research Division played in getting IBM to finally announce and deliver relational database products: SQL/DS and DB2. There is a lengthy discussion of Query by Example (QBE) by its creator and a discussion of why it didn't get accepted as a product by any of the IBM operating divisions in spite of its strong support of the relational model developed by Ted Codd. The dominance of IMS in IBM's thinking and its impact on the lack of resources assigned to relational products, particularly the concern for possible erosion of IBM's enterprise MVS position, was explored. The focus in IBM Research of providing software which could be turned into products was examined and its effect in lessening the true research type of projects in the Research Division. Various methods for integrating research and development activities in IBM were also discussed. The discussion also highlights the problem of inter-organizational communications in an organization as large (and as balkanized) as IBM.

Participants:

Name	Affiliation
Burton Grad	Moderator
Michael Blasgen	IBM
Marilyn Bohl	IBM, Ingres
Peter Capek	Historian, IBM Research
Don Chamberlin	IBM
Sharon Codd	IBM
Chris Date	IBM
Don Haderle	IBM
Thomas Haigh	Historian, Wisconsin University
Bruce Lindsay	IBM
Jan Phillips	SI SIG, DEC
Jim Strickland	IBM
Moshe Zloof	IBM

Burt Grad: This session will review the history of relational database management within IBM, covering the things we haven't talked about already. It can be technical, it can be business, it can be mathematical. Since we haven't had a chance to get Moshe Zloof to talk about QBE [Query by Example] yet, I'm going to make sure we have time for him during this session. We're interested in the details of what was going on with the various products being developed and what happened to each one. I'd like to at least make sure we get up to DB2 and beyond if possible.

Don Haderle: I think we've got two timelines we've got to deal with. We've got one timeline which is what was going on in IBM Research and I think we have another timeline of what was going on in the product divisions, because we keep mixing these two things up.

Grad: I think that's an interesting point. Maybe we could do a rough timeline without going into detail as to what was going on in the different areas, in Research, in the product divisions, in marketing, and wherever else seems appropriate. What were the most important events, like what Luanne Johnson was talking about earlier. Let's do that first quickly.

IBM Research and the Relational Model

Bruce Lindsay: I think Don Chamberlin just covered it fairly accurately and comprehensively: the System R to SQL/DS transfer and then from SQL/DS back to DB2 on a mainframe. I think up to that stage of history, Don was accurate and correct in describing the steps.

Grad: Let me start the question on Research. You first have the work that Ted Codd did there, that's the initial research work I believe in the relational area. Is that a correct statement? And that's basically 1968 and 1969, with a paper published in 1970. Was there any other work in Research at that point in time on relational to any of your knowledge?

Chris Date: Yes, there were more Codd papers that were terribly influential. He did one on normalization and the one on data sublanguage Alpha and the one on relational completeness. Ted, in that period of about three years, produced five extraordinary papers and we're in this room right now because of what he did. The whole industry exists because of those papers.

Grad: That was the foundation work going on in Research that laid the basis for this other work.

Date: It was really just Ted by himself.

Lindsay: That was really Ted. There wasn't really a team behind this. This was Ted's own intellectual effort.

Date: It led up to, as I mentioned this morning, Ted proposing those multi-division projects within IBM and he started work on this thing called Gamma Zero, which I believe was a very strong input to the RSS because Irv Traiger was involved in both.

Michael Blasgen: In fact, Irv was the manager of the RSS at the beginning.

Grad: Let me get this right. What was Gamma Zero?

Date: That was Ted's name for essentially the storage engine, the low-level component.

Grad: That became the RSS project?

Date: I can't say it became it. Mike Blasgen can answer that better than I can, but they were very close. Irv Traiger worked with Ted in developing Gamma Zero and as Mike Blasgen just said, Irv then became the manager of the RSS.

Grad: So it was a separate step in the process. It wasn't simply that they renamed it to RSS or something like that.

Blasgen: No. The RSS was different but influenced by all that came before.

Lindsay: It was all new code, right?

Blasgen: XRM, a predecessor at Cambridge Science Center, might have been closer to Gamma Zero.

Lindsay: But there was no XRM code in RSS was there?

Blasgen: No, no. I'm just saying, if you're talking about Gamma Zero, Raymond Lorie was certainly influenced by this but that was in Cambridge, MA.

Grad: As I understand it, Lorie was at the Cambridge Science Center. He saw these papers and he saw the work being done on Gamma Zero.

Blasgen: Lorie's work there, pronounced "XRAM," but spelled XRM, was used by the Phase Zero prototype and was also used by Moshe and the QBE -project in Yorktown.

Sharon Codd: But I know before System R was assembled, Ted was working with a group and it must have been Traiger and Ken Decca.

Date: And Lorie came in after that.

Codd: This was before Leonard Liu came out to San Jose with Frank King and you, Don. When did you all come?

Don Chamberlin: It was 1973.

Blasgen: Raymond Lorie came to Cambridge at the same time.

Codd: Ted felt that when they put System R in place under King, they had taken away "his" project.

Grad: Doing the timeline, we have the Codd Papers, we have Gamma Zero, we have XRM which is done separately in Cambridge, and we now have the System R team. Is that now correct or have I missed any other step.

Codd: No.

Grad: And the System R team, the first project is to do what you call Phase Zero?

Chamberlin: That was half of what was going on. The other half was the development of the RSS which became the platform for our second prototype. Phase Zero was the first SQL implementation on top of XRM. In turn, RSS was the Relational Storage System which became the platform for the second SQL prototype, called System R.

[Ed note: In Workshop 1, RSS is identified as a project team ("a lower-level group"); the first group was called the RDS; seems like we are using "RSS" as a project id and as the name of a product/system.]

There was something very influential that was happening starting in 1976 that I'd like to call attention to, and that was a series of joint studies between the System R project and some key IBM customers. As soon as the real System R prototype, the RDS and the RSS together, became operational in late 1976 I believe, we installed it on an experimental basis in a dozen or so IBM locations and in three outside customers, which were the Pratt & Whitney Aircraft Company in Hartford, CT, the Upjohn Drug Company in Kalamazoo, MI, and in Boeing in Seattle. We met quarterly with each of these customers to evaluate how they were using it and to collect feedback, to try to correct problems and find out how the system should be improved.

This was a very influential process in the evolution of System R into a more robust platform that could be turned into a product.

Query by Example (QBE)

Codd: In terms of a timeline, we should also include Query by Example in 1973. Moshe, I think you should describe how you viewed it.

Moshe Zloof: At the time I got to IBM and you were at work, Don, we started the concept of QBE. The System R people first had Square and then SQL. I started QBE from the end-user point of view about the same time as the San Jose efforts got underway. QBE was a small effort. In 1975, our work was very similar to what they were doing with a larger group. We didn't have all the resources to do a really comprehensive job specifically in terms of optimization. We also did not do the RSS part, but instead used XRM. But we built a prototype that actually became an IUP. As all IBMers here know, IUP is an Installed User Program. The product division in Santa Teresa supported it, and we started selling the product directly from Research. Our system could be used in many applications; I had about 120 to 130 customers here -- personal finance and sales applications using QBE. So QBE wasn't just a query language. Later on it was put in an IBM product as just a query language but our IUP product was a full fledged system. It did data definition, it supported constraints. Chris Date wrote a whole chapter on it. It did data definition, constraints, query, integrity constraints, security, domains; all of that was in QBE, and in fact, customers used it for running their businesses. Bank of America was one of them; in Japan -- because it's visual -- they went crazy over it. I was invited to Japan many times to give talks. I translated all my papers into Japanese. But the main points that I want to make are the following. We were a small group so we weren't really concentrating on the lower part -- optimization, compilers and all that stuff. Now it's easy to say after so many years, but because of our egos, we really were not working together. Looking back, we should have worked together. When I left IBM, I formed a smaller company that I later sold to Ashton-Tate and Ashton-Tate had dBase and part of my job at Ashton-Tate was to translate QBE into SQL. So, if we had worked together, translated QBE into SQL, and put all the things that SQL has in terms of views and all of that stuff into QBE, that would have been the best solution. But we didn't do it at that time.

Grad: What did you consider the primary contribution of QBE?

Zloof: The primary contribution of QBE was that for the first time, operating people, even executives, were using 3270 terminals to directly interact with databases. SQL, as compared to DL/1, was at a much higher level, and it looked like English. But executives and people couldn't use it. In the IBM Research Division, there were about 80 internal applications in use for QBE -- things like keeping track of budgets, expenses, all of that, and they were being used by secretaries. So that was the contribution. In fact, QBE was the first system that had data in fields of forms. And later on, spreadsheets came into the game.

Grad: So the user interface for posting the queries was completely different-- it was a visual interface.

Zloof: Right. It was a visual interface not only for query, but for data definition. If they use it from A to Z, it means they have to create tables, they have to define tables, they have to define constraints, and they have to define security. All of that was part of QBE. The only thing is that at that time, in 1975, all those papers like Pat Selinger's on optimization, like what Bruce was talking about, is it a sort merge, is it hashing? Those papers weren't there. So we used brute force to carry out the queries. But there is something very important that I want to mention. Ted Codd defined the relational database as a pure thing: domains. When you do selections, you have to remove duplicates and so on. All of that basically caused performance issues. When you say you want to be truly relational, when you say "select" for example or "p-dot" in QBE, you have to remove duplicates. In the case of SQL, because they were competing, what was their thing to shoot at? It was IMS. So when I started QBE and they started their work, the people from the product division, the IMS people, basically said, "That's a beautiful concept but it's just a toy. You are showing me the 'join' for about 10 records, but what about if you have 100,000 records?" So, to show that they can really compete, they have to show performance, right? Therefore, they kind of tarnished a little bit of the original concept of Codd. For example, if you want to incorporate the concept of domains, it's a tremendous burden on performance. That's why they didn't have domains; we had domains in QBE because I worked very closely with Codd to make it purely relational and we paid for it. Basically, we paid in performance.

Grad: What machine were you implementing it on? S/370s?

Zloof: We ran on the S/370 under VM with XRM. We didn't do as much as System R took on. System R was a much larger group and they were mostly in Research. People wrote papers and from the papers, they generated the implementations. And then they re-did it and so on. We didn't have that luxury so people in my group, when they evaluated a query, they did it in a brute force way.

Grad: So what happened?

Zloof: The IUP wasn't supported because IUP is not something that IBM development supported. Its only support was my group, and we needed to move onto the next project, Office by Example. So eventually the IUP was discontinued. By the way, QBE is incorporated in Office by Example, and many of the concepts of OBE got into Microsoft Office. Actually, Access from Microsoft is purely QBE. They used to call it that when they started.

Thomas Haigh: I think it's still called Query by Example in the help files. I've got an outsider's question here to just make sure I understand what you're saying. QBE wasn't just a front end that would generate queries to be processed by another program? It was actually a whole self-contained database system?

Zloof: A whole self-contained system. The people in the divisions wanted to always say that this was an interface. But the QMF in DB2 was basically an interface, that's what they picked. But QBE was a full fledged system that people actually used in their business.

Haigh: And your explanation of why it happened like that was egos, right? You thought you could do a better database management system so you did.

Zloof: No. First of all, logistics was very important. We were in Yorktown Heights; they were in San Jose. Logistics was very important. Also, we had a small group, compact, and we knew what we were doing and we wanted to get to the customers as fast as possible. So we did it in one year and we got the product out; while they had to struggle with the divisions, they struggled and it took a long time to do the first full fledged IBM product, SQL/DS.

Lindsay: There was some comparison of the System R and QBE projects, at the level of usability, function and performance. I think one of the concerns, especially from the storage people, was that there was no program embedding through it. You can't write a program in QBE, that's business logic.

Zloof: That is a very good point. One of the major drawbacks of QBE was it was visual, two dimensional, while SQL can be a programming language--you can embed SQL within PL/1 or within C. We couldn't do that in QBE so QBE always gave the impression that it was not really a system, that it was just an interface.

Haderle: The system guys had very nicely taken SQL and neatly given us both an interactive version of it that you could enter on a screen and a compiled version of it so I could get compiler-like performance at runtime, which is what was needed for transaction and batch.

Grad: But again, in the 1970s, System R had not yet produced a product.

Haderle: Right. QBE had something in 1975 or 1976 aimed at the decision-support market for the VM operating system, but this was a secondary buy in any commercial house. If you went to Bank of America, they had their large enterprise systems and then they had the decision support and QBE which were sold only in the secondary market.

Competition from 4GL's

Grad: There was an issue raised this morning by Sharon about the competition from Nomad. There were all these 4GLs [Fourth Generation Languages] out there, which were selling significant numbers of systems: Focus, Ramis, and Nomad. Mark IV was still being sold. There were a dozen of these things. Why didn't someone in IBM see the opportunity for mid-sized machines?

Haderle: Remember, if it was Nomad, it was running on VM on top of IBM hardware. So B of A still ran on top of IBM hardware. IBM was capturing most of the available revenue.

Grad: Here's another point. IBM had a whole bunch of machines from the System/3 and System/32 up to the System/38, starting in the 1970s. Why is it that no one saw the opportunity of taking what you had done and putting it there? My point is, IBM had the S/32, S/34, S/36 and they were supporting lots of applications at the General Systems Division. They were selling tons of applications on those machines. I'm wondering why no one saw this opportunity. Maybe Research was blocked off and that may have been the problem. You raised the right point, Don. There was just no connection with the right people and I'm surprised DPD [Data Processing Division] or GSD [General Systems Division] didn't see that opportunity.

Haigh: I've got a related question. In terms of this easy to use end-user querying, I know SDC [Systems Development Corporation] attempted to sell some online timesharing systems with the same kind of end-user application where they could enter the data and then query online. This happened as early as the 1960s. The idea is that with Mark IV and Easytrieve you could tick off fields somehow, code a punch card, and then feed that in as your report template. So were you aware of those kinds of systems and what was it that was so much beyond them?

Zloof: It was really the relational model, because QBE was relationally complete, which means in their definition, you can write any query and get to any piece of data. And the systems you mentioned, they were not relationally complete.

Lindsay: I think there's a much, much more fundamental reason why the bigger systems have come to dominate over these other things like Mark IV and Nomad. This was recovery and concurrency control. Until the data storage system became reliable enough that you could throw away the paper, nobody was going to take it seriously. If you had to keep the paper, you are not going to spend another ten million dollars to buy a damned machine because you are still stuck with the paper. They would take the paper and enter it into the machine and maybe do some decision support. What really made these systems viable in the commercial world was the reliability that was built into the lower levels. It had very little to do with the relational model. Until you can believe in the computer enough to get rid of your paper, there's not a compelling business interest to spend a lot of money.

I want to go back to one little thing. I think there were some informal comparisons between QBE and System R and since you're interested in the oral history, perhaps Don can tell the story of the shootout at the OK Corral. Of the people that were there, some aren't still with us.

Audit of IBM Research Relational Database Projects

Grad: Don, you want to speak to that?

Chamberlin: In roughly 1977 or 1978, Research management at headquarters in Yorktown observed that they had two database projects -- one on the west coast and one on the east coast -- that were both doing related things. They wondered whether there shouldn't be more synergy between these two projects. So somebody was appointed to audit these two projects.

Zloof: Ashok Chandra and Dick Case.

Chamberlin: Right. They were appointed to audit these two projects.

Grad: Who were these two?

Blasgen: Dick Case was an executive who had made a major contribution to technical testimony on the IBM Justice Department lawsuit. That was his main job for maybe eight years, and then he was freed up and got involved with new things, like this.

Grad: I did a three hour oral history of Dick Case that is available on the Computer History website.

Blasgen: So Dick Case headed up this thing and he had Ashok Chandra who was a brilliant computer scientist from Yorktown who had no particular link to either project. They were independent people. Ashok was from the math department, Dick was from corporate headquarters. They were independent, real smart, and they came out and visited Yorktown and San Jose.

Chamberlin: Their job was to audit these projects and decide whether there should be more cooperation between the two projects and what form that cooperation should take. I think the conclusion of the audit, as I remember it -- other people may have different recollections -- was that the RSS was a more robust storage engine than XRM was.

Lindsay: In part because RSS consumed a lot more man-years to build.

Chamberlin: And the RDS had a lot of compiler-based optimization that was lacking in the QBE prototype. The conclusion said that the best way for these projects to cooperate was for QBE to be (sort of) put on top of the stack as a separate interface in parallel to SQL, so you could use either a two-dimensional interface like QBE or you could use a more linear interface like SQL on top of the technology stack that had been developed in System R. It really never came true, but I think that was the direction in which these guys recommended we move.

Grad: Moshe, do you remember the conclusion the same way?

Zloof: No, I don't remember the conclusion.   We had moved on to Office by Example. Eventually I left IBM because I wanted to start my own company, where I worked for two years. Then I sold it to Ashton-Tate for not much, like four million dollars; in those days it was a big thing. Then my task at Ashton-Tate was to put QBE on top of dBASE.

Grad: Who had the ownership? Did IBM own QBE?

Zloof: Yes, and they kept the code.   Later you could go to IBM and say you weren't doing anything with this and if they would give it to me I'll give you, say, five percent of the sales revenue.   But not when I left. It had never been done. So we had to start all over again because I couldn't have used the code since it was proprietary to IBM. We lost about a year to a year and a half in that process.

Grad: So you built a new product.

Zloof: Yes, we built a whole new thing from A to Z and then I sold it to Ashton-Tate. When I started, all these companies – Ashton-Tate, Microsoft -- they were very small. But soon they got so powerful that I couldn't compete anymore with no marketing and so on.   I had to sell, so I sold it and I joined Ashton-Tate. When I look back, maybe I could have come to IBM and offered to pay IBM royalties, or buy it, but I didn't and we lost some time which hurt us a lot in terms of timing.

Grad: Whether they would have listened to you or not is a whole different story.

Zloof: Probably not. In those days, why do we have to worry about the lawyers and so on, who cares about that?

Grad: Don, you had some comments?

IBM Development Database Projects

Haderle: Totally orthogonal in what was going on in Research was what was going on inside of IBM development. On the operating system side, they had three operating systems: DOS which was targeted at departmental system, VM, which was mostly for decision support, and the most important was MVS, which ran batch programs, transaction programming, and some measure of interactive processing. A key requirement for MVS [designed in 1972] was to run batch simultaneously with a transaction workload. Then in 1974-75, we were trying to mix interactive with batch and transaction; that was hard. The division we were working for, GPD, asked what we were going to do to support interactive query and dynamic decision support. Our answer was to take the System R prototype and simply resurface it into that environment. But that was not enough, so we took on the big job of 24x7, batch, transaction, manageability, etc. and that's how DB2 was born -- out of that environment. It did a little ad hoc query with lots of transaction and batch, and the VM one did lots of query with a little transaction and a little batch. The timeline was that the SQL/DS came out in about 1981, and DB2 came later with limited availability in 1983 on MVS and general availability in 1985, a phased rollout.

Haigh: So for those of us without a mainframe background, I know a little about systems like Oracle and Charles Bachman's IDS stuff. I know his original System/2 would basically tend to take the machine over. They weren't relying on the operating system for the virtual memory. They were doing their own internal allocation of priorities to different kinds of things and so on. So the database management system, to a large extent, was becoming an operating system. Now, when you talk about building the database capabilities into the operating system in this kind of mainframe environment, how much of that is relying on the virtual memory, storage allocation, and file management capabilities that are in the operating system? And how much new capability must be created?

Haderle: On day one, DB2 had no clout, so consequently it had to ride on top of the operating system and then within the constraints that they had for virtual management. We then had to go and build our own storage management on top of their storage management, our own memory management on top of their memory management, our own dispatcher on top of their dispatcher, and kind of work our way around the problem because we weren't important enough yet for it to go into the base operating system; we weren't well known enough to be able to go in there. So on day one, it was a built-on-top-and-work-your-way-around. As time went on -- over 10 years -- and as the business grew up, more of these primitives dropped down so that you could use what was there as native capability as opposed to have to build on top, which gave you better manageability, resource management, etc.

Haigh: And which 10 years are those?

Haderle: We never achieved that on VM, we never really achieved the level of that importance on VM. On the MVS side, 1988 was a seminal year for DB2. That was version 2, release 1. By then DB2 was only twice the cost of IMS, whereas when it came out on day one, it was more like six times the cost.

Grad: So was there a study of comparison data on the DB2 cost versus IMS? Was that a regular thing?

Haderle: We were not trying to replace IMS. It was not for that reason; people think it was for that. It wasn't to replace it; it was to be able (in somebody's mind) to be good enough for doing transaction processing. So if we were going to end up beating IDMS and Datacom/DB, our primary competitors at that point, not IMS, we had to be somewhat competitive to the other thing that we already had, which was IMS. So we did internal studies that we never put out, which said: "Here's how we compared against IMS," and then we spent years just trying to get the numbers down. So when version 2, release 1 came out in 1988, there was an article in *Computerworld* that said "Relational is now ready for primetime." It was simply because we had the features that were there for doing transaction processing. The customers came back and said it's competitive for that environment now. Good enough.

Grad: Let me ask you a question. SQL/DS runs on VM and DOS. What happened with that? I haven't heard that story. Did that succeed well? Was it selling well? What happened there? Who can tell me that?

Haderle: Well, I don't work for IBM anymore. In the mid 1980s, IBM brought out the 4300 series of S/370s that was going to sell billions in revenue. In fact, SQL/DS was targeted to that market.

Blasgen: The 4331 and 4341. Both were full S/370s with virtual memory, etc.

Haderle: And that was about 1985. At that point, the pricing on SQL/DS was based on how many of those suckers were going to sell. The business case showed that there were going to be gobs of these things. So they set the price very low, but they didn't sell gobs; they sold a few. Now the product was supposed to be self-funding for future development, so if you didn't sell a lot, your funding engine came short so when you didn't sell many of these 4381s, your funding declined.

Marilyn Bohl: Then they moved the development responsibility from Endicott up to Toronto and the people didn't go.

Haderle: They did that but their funding base just shrunk, it just shrunk like that because the platform they were running on kept shrinking and the money that they needed to self-fund wasn't there.

Grad: So IBM did not end up being competitive in anything other than essentially the MVS and its successor market. Is that a correct statement?

Lindsay: Or markets that nobody else was in, like the 4300 market.

Haderle: Or the System/38.

Grad: When I was at IBM, we were never going to support anybody else’s systems. Was there ever a discussion of supporting anything else like DEC or any other computers?

Haderle: Not in the 1970s and 1980s because IBM was hardware-driven, hardware-managed.

IBM Research Providing Basis for Products

Blasgen: I’d like to make some comments about this whole environment. As to the motivation of the Research Division, we were told rather clearly and unambiguously that we were successful only if we got our stuff out as a product. There was no point in having a Research Division, famous for its science and technology, without it being vital to IBM. That was the slogan: "famous and vital." Between the two, "vital" to IBM was like 40 times more important than Nobel prizes. Just look at the way the division spent the money. It was almost all on "vital" to IBM. Ralph Gomory became convinced that there was no way a 100 million dollar a year Research Division could be paid for out of "taxes" on the profit generating part of the company. It wouldn’t survive if it didn’t create products for IBM. It went to the extreme that we really felt that they had to take the product kind of as-is. I mean, we really felt that we would prove our strength by them accepting our work directly.

In the 1980s the Research Division in San Jose issued a memo that all software was to be written in the programming language PL/S [Programming Language/Systems was a machine oriented programming language based on PL/I]. That’s a pretty odd memo for a Research Division. Why insist on PL/S? Because we knew that the product development organizations would not take it if it was written in anything but PL/S. I was the manager of database systems for several years. System R was done and the joint studies were underway. So now this is choke time, this is the time to get the product out. We investigated every avenue: IUP, Type 3 or Type 2. We looked at everything, we met with everybody. We talked about Boeing releasing it. We did anything. By the way, System R, from the beginning, was envisioned as an MVS product. We knew that MVS was where all serious database stuff went. The attraction of VM was that it was IBM’s development platform: all software development in IBM took place on VM. So it was by far the easiest for us to do our first release on VM because that was the way we debugged it. Then we transported it over to MVS and used a lot of MVS services that are absent in VM, like shared memory. Boeing used our MVS version of System R.

But our first success was not in MVS. Of all things, it was in DOS, the one system we didn't support. It all started with Bob Jolls, a remarkable manager in Santa Teresa, who had been given the job of developing a replacement for DOS DL/1. DOS DL/I had not been a very successful product; it wasn’t quite compatible with IMS; it wasn’t quite good; it was built in Germany. He was given the job of figuring out what the new DOS DL-1 should be and he did an investigation which involved his people looking around for an idea. Jim Strickland is here; I think he was involved. They made various visits to San Jose Research and they also did other work; I don't know what all. And then all of a sudden, I got a call from Bob Jolls and he said "We've picked your product. We're going to base our offering on your stuff." This, of course, was just like ringing the bell at the top of this thing that you've tried to slam on for the last two years, Bang, Bang, Bing! It goes off. Within weeks, it seems to me, they made the decision not to do it in Santa Teresa. I think it had to do with resources and skills availability.

Lindsay: Some project had ended at Endicott and they had all these programmers available.

Blasgen: They said we're moving System R to Endicott; so we set up a whole world where we tried to support Endicott. Endicott immediately made the decision to rewrite the parts of System R that had been written in PL/I. So we worked very hard to help Endicott and they did bring it out on VM for the same reason as we had because it was already there. They also brought it out on DOS. Meanwhile, Santa Teresa was still plugging away on their projects, which Don knows much more about than I do. By the way, Don has mentioned several times that IBM had three operating systems. Actually, I think I can count about 15 because the only ones he's talking about are the 360/370 operating systems. We also had the 8100, System/7, System/3, System/36, System/38 and eventually, Stratus, which was called the System/88. And then we had a project in IBM to try to figure out how to get rid of all those machines, just like we'd had a project called S/360 to get rid of the 709, the 704 and 650 and 1401. It's like you learn from history that you don't learn from history. So we had these. Like the 8100 alone had two or three operating systems on it, as I recall. For some reason, the database systems we're discussing did not run on any of those smaller machines.

Supporting Other Platforms

Grad: That's one of the questions I'm raising. All these other people are coming into the DEC platform and into the other mini areas and IBM is not even looking at those as opportunities, which just amazes me.

Codd: Because IBM was a hardware company.

Grad: No, I don't think that is the whole answer. We were selling tons of System/3s and 32s and 34s and 36s and eventually S/38s. We sold millions of dollars worth of those machines. We really did. It was not a trivial market. But IBM screwed up in the mini market badly. We didn't understand DEC; we didn't understand the people using those machines; at least that was my belief.

Blasgen: I told this story upstairs, but we developed System R on a model 158, which would maybe fit in this room; depending on how many disks it had, it wouldn't even fit in the room. The first time I ever saw a sign that said "ORACLE" was at the National Computer

Conference in Las Vegas in 1980 or so. I knew about Larry Ellison, I knew about Oracle, but I'd never actually met anybody who worked there or seen a demo. But there they were at a small table at NCC running the Oracle software. Imagine my shock when I saw that their software was running on a carton of cigarettes. It was literally that size. I didn't know you could buy a computer that size anywhere in the world. It was a bottom of the line DEC PDP 11 and you could type SQL; I did it; with my own fingers I typed an SQL statement and it printed out the answer with 42 rows in the base table.

Grad: Michael, let me ask you a question. In IBM Research, what I saw in Yorktown and what was happening on the west coast was very much large machine focused. Is that a fair statement?

Blasgen: I think IBM was large machine focused.

Haderle: Those geographic areas you described are large machine focused. If you wanted a small machine, like the System/3 or the System/7, that started in San Jose and it moved down into Boca Raton and up into Rochester, Minnesota. So if you were on the east coast or west coast, you were large. You had to go down to either Florida to get real time, which was small, little dinky machines or up into Rochester.

Grad: The point I'm getting at is that maybe because of geographic location, maybe because of history, but IBM Research wasn't really focused on these other size machines, it looks like to me. Is that accurate or not?

Blasgen: No, but certainly the database and transaction processing applications were central because that's where we perceived the market to be, correctly or incorrectly.

Grad: Even though we saw these other guys selling it.

Lindsay: The small machines wouldn't have hosted the crazy, big, inefficient software we were writing in those days.

Haderle: And again, the System/7 was aimed at real time, it was a real time operating system. They were not interested in having "SELECT * FROM."

Lindsay: We wanted a big machine. We wanted a MIP [million instructions per second speed] and a meg [a megabyte of storage], I tell you.

Grad: I had a System/7 group located right here in Palo Alto, 120 people, to write system software and applications for the System/7. This was in the 1970s. We were right out here-- paying attention to you all, talking to you.

Lindsay: That machine never even got close to a MIP and a meg. That was sort of our dotted line.

Haigh: So if I can get the outsider's perspective again. It sounds like IBM was fairly terrible in terms of getting products into the marketplace. I was upstairs earlier with the Sybase, Informix, Oracle guys; they would write a system and then they'd port it like crazy and it would go out on all these machines and some worked and some didn't. What I'm hearing here is more that you have to find some sponsor group that have some hardware and an operating system and then persuade them that your thing is what they need. And then your success or failure just hinges on whether this group in IBM plays the politics and gets the machine out the door, and anyone ever buys it.

Codd: You are comparing a company that was predominantly selling hardware -- software was largely an afterthought, not entirely but largely -- to what were then startup companies whose entire reason for being was to enter the database market and capture this ad hoc capability. It's apples and oranges. And in IBM, different locations had what were known as "missions," and if it was not your mission to support System/38, you didn't think about it.

Haigh: It's kind of ironic that if you would want some application to support the broad range of machines, presumably by some point in history even a broad range of IBM machines, that you would do better finding an external product that had been ported rather than having IBM port some consistent thing across its own range of machines.

Haderle: There's an assumption there that all machines do the same. At least in 1970, you had a machine that did a very specific task, so you had a machine that did batch and that did not do transactions. You had one that did transactions and that did not do interactive. You had one that did interactive but did not do long running anything. You had the DEC machine that was good for little departmental thinga-ma-jiggy but was not doing big enterprise.

Lindsay: When you were upstairs with these guys, what was the platform or focus? Was it really VAX's or was it UNIX or VMS?

Haigh: Yes, those two operating systems.

Haderle: Which means they were small departmental locations that they were aimed at a very specific market.

Lindsay: But the point is that they were developing their software for pretty specific machines. They didn't develop for the S/360. They didn't develop for the Honeywell mainframe computers. They didn't develop for the Sperry/Univac. Those guys were in the mainframe business.

Bohl: But the needs of the users were a lot alike across those machines. In the case of SQL, we had meetings that included people from SQL/DS and people from System/38, System/36, and people from DB2 and from the PC environment, all talking about SQL. But when it came to implementation, the priorities for one IBM machine compared to the priorities of the team on another IBM machine were different.

Lindsay: I think those were probably false priorities. Don keeps mentioning driving stuff down into the operating system for the database. If you go and look at the database products today in the world, none of them place any significant requirements except for Oracle RAC on the operating system. Informix, they never told you to go patch your operating system to improve their system. With Sybase, it was the same thing.

Lindsay: It's criminal how long it took us to get our products out.

Enterprise versus Departmental Database Systems

Haderle: If I go back to that general statement there, from an IBM perspective in the 1970s you have this prototype that's out there, this System R prototype in 1976 or 1977, capable of being deployed. Meanwhile, on VM, you had Nomad, you had RAMIS, and you had TOTAL running. IBM is still getting the money coming off of the hardware revenue. Their primary database transaction system at this point in time is IMS and the DL/1 language. IBM is still investing in IMS to keep it vital. But after a couple of years, management sees that this is not playing well and they turn around to Bob Jolls and say, "Do something." He turns around and in a matter of about two years was able to standardize and convert from System R to SQL/DS and roll it out. Isn't that about what it took? From the time of the decision to go—it was about two years. This is the same story we heard from Moshe, which is given that you have nothing, it takes about two years. Meanwhile, the MVS thing was going along and they weren't going to go and deploy that thing unless it was robust because if it fell on its knees on day one, it would never get off of its knees again; that would be the end of it. So deploying into that enterprise environment, it's going to take you not two years; it's going to take you a longer period of time.

Lindsay: It's different deploying a product into the enterprise than it is into the department. If it's just to the department, it still survives in the company. You can't do that if you're going into the enterprise.

Bohl: The other thing was, IMS was coming out with Version 1.3. Jack Bertram was the head of GPD [General Products Division]. I was product manager for DB2. We would march into his office once every two months with our latest story about how many different sites we had DB2 in, but he only had one support organization in GPD and he wasn't going to let DB2 get out and in any way pull off resources that he might need to support IMS 1.3 deployment. So DB2's rollout was constrained to protect the customer service people for IMS.

Haderle: And I love Jack Bertram's, head of GPD, announcement of DB2 version 1, release 1. It went something like this: "We're putting out IMS version 1.3. It jumps over hoops. It is so good. It stays up forever. You can get it in any color. And by the way, we have this DB2 thing." Was that a correct version of the announcement?

Codd: We were also constrained in what we were able to tell customers because I was doing relational presentations vis-à-vis MVS in the International Briefing Center from 1981 to 1983, which was when they were full throttle on the DB2 product. I could never say that this is another DBMS. This was going to be the DBMS for decision support and it could run with IMS. But the fact of the matter is I remember discussing it with Chris [Date] and discussing it with Ted [Codd] and a few other people. I said who the hell cares? Let the product go out. The customers will figure out that it does the whole thing.

Haderle: The reason they didn't come out and say we had DB2 is that any time they announce something like that, it freezes the business. The IMS customers would not upgrade to the next version of IMS. They would say, "Oh, if we do that, then by God, there's this new thing that's going to come out that I should really be on." So from a marketing viewpoint, they always downplayed it, because they were afraid of freezing the upgrades to the new business.

Haigh: So in terms of this platform stuff, and the position of DB2, wasn't it about 1988 that they announced that DB2 was going to be a core component of OS/2's extended edition? How did that fit in with all this?

Lindsay: Well, initially the mission to port DB2 onto a PC-Intel base was given to Jerry Baker, who managed to get nothing done for about a year-and-a-half, and then he went on to Oracle.

Grad: And he couldn't be here today. I invited him.

Lindsay: That's my pretty clear recollection of that period. And so nothing really happened with relational on Intel for about a year after that.

Grad: What year?

Haderle: It had to be like 1984 or 1985.

Lindsay: And then a mission was set up in Austin to develop a database system from scratch. Naturally they couldn't use PL/S, since there was no compiler for that on the PC, so it had to be written from scratch, you see? It was in Austin. The Research Center was heavily involved with the technology that they would use, the recovery technology, concurrency control, B-trees; these were major area that I know we did a lot of work with the Austin team on "Here are the algorithms." That was a 16-bit database system for a 16-bit machine. And so that was a part of the extended edition: EE-something-or-other. Had a lot of initials in it. And so that consumed quite a lot of effort. And still no UNIX investment.

Grad: When does the IBM RISC [Reduced Instruction Set Computer] machine come out? 1986? Did that have DB2 support?

Haderle: Not from IBM. It was supported by Oracle.

Grad: Unbelievable.

Lindsay: Well, as I said there was no investment in UNIX software. Probably because we didn't have any UNIX products until the RISC Machine came out in 1986 or so.

Haderle: So IBM again aligned on hardware boundaries; the funding for the PC was coming out of PC group, and in fact, they asked us to put DB2 onto the PC and they said, "We'd give you funding for it," like "eight people to go and do it." And we said, "We can't do that with eight people." That's what Jerry was given, a mission to figure out what he could do with eight people. So they ended up with nothing, and then John Dash came in and said that he could do it. He left, too.

Lindsay: Same target, same destination.

Haderle: I want to give proper attribution here.

Bohl: That's right! <laughs>

Haderle: And then after they sunk the project and went to Oracle, the mission got moved back down to Austin where the PC group sub-funded themselves to go and do it with a small group, to create a non-competitive database on that platform, trying to utilize Research as much as possible to go and provide the labor. A lot of labor went into it. On the AIX [Advanced Interactive eXecutive] side, we have Jim Cannavino to blame on that platform (the RS/6000). He said he didn't care about the database. He was interested in the hardware, and so he funded outside of IBM rather than fund inside of IBM on that particular platform. That was Jim Cannavino.

Grad: Oracle on the platform, if it runs good, sells the hardware just as good as your stuff would.

Blasgen: I'd like to poke Don's memory about this, or Marilyn's. Help me remember the problems in supporting IBM's enterprise database in the 1970s. If we were going to move to DB2, what's involved in convincing them? I am confident it is hard. I remember that IMS had an error code, or return code, that comes out of IMS when it follows a dangling pointer that points at nothing. It's like a 44 or something. This started happening regularly on IMS. And so many customers were upset that the IBM CEO, I believe it was [Frank] Cary, instituted a rule that every time a customer encountered that bug, he had to be informed. Here he is running a $50 billion-a-year company and the developers had to call the CEO if a customer encounters this bug. Jim Frame's solution was to take every person that had an IQ above eight, and assign them to IMS maintenance.

So in particular, all the people with IQs over 200 were assigned there. Ron Obermarck, he was full-time chasing the bug that led to this particular shutdown of IMS. And that's what it takes to support enterprise. You have to take all your best people and put them in support. And the CEO is going to know every time you screw up. Well, this was not the path that a Research Division guy had been trained in, or was used to. But it was absolutely a requirement. I'm sure we do that with DB2 today.

Grad: You're all telling the story much more vividly than I've ever heard before about the errors, the judgment errors, that IBM was making during the 1970s, the 1980s, and literally onto the time that [Lou] Gerstner comes in. Cary and his two successors were fine gentlemen, but they did not know how to see what was happening around the companies. These stories are just wonderful.

IBM Database Accomplishments

Haderle: So again, that's a focus on the glass being half-empty, you know? If you focus on it being half-full, we went a long way. DB2 on MVS was a huge success, a multi-billion success.

Blasgen: In 1990 under John Akers, I think it was 1990, IBM turned in seven billion dollars in profit.

Grad: Correct, correct.

Blasgen: I mean, that's a phenomenal gross margin. That was with 40 percent SG&A [Selling, General and Administrative expenses].

Lindsay: I think the way to look at these decisions is it takes an especially wise and callous person to realize that if your children are going to be eaten, it's best to eat them yourself.

Grad: And no one wants to ever do that, do they? Whether you're in the transportation business or what-have-you, it’s so tough to replace your own products. Interestingly enough, IMS was the largest database management single seller, and it was certainly not over about 30 percent of the non-relational database market.

Blasgen: What year?

Grad: I was thinking of the 1980s, mid-1980s. It was very significant, but there were also a lot of significant software competitors.

Haderle: To IMS?

Grad: Yes. We had a half a dozen major software companies competing.

Haderle: Again, with IMS, in like about 1985, we had Datacom/DB, we had Cincom’s Total, IDMS, and Adabas and more. Total was drifting out at that time.

Codd: Right.

Haderle: They were really drifting out. So IDMS was the number one competitor, and Adabas.

Grad: Which of those companies still exist?

Haderle: Well, Total still exists. So does Model 204. Who cares?

Codd: Yes, but Cincom is a privately held company still to this day. And they did build a relational product, Supra.

Grad: Let’s keep going with IBM. Marilyn. Let’s talk about your picking up the DB2 stuff. You were responsible for that for a period of time here. When does your responsibility end on DB2?

Bohl: When I left IBM.

Grad: When?

Bohl: Late 1987.

Grad: And DB2 had been announced, was out there for some years. This is before the version that came out in 1988.

Bohl: Version 1.3 and 2.2 were in development.

Haderle: Correct.

Grad: Who took over on DB2 when you left?

Bohl: A performance guy from the East Coast.

Haderle: Yes, his name was Gary Ferdinand.

Selling and Supporting Relational DB2

Haigh: So here's my big picture question. Inside a company like IBM at that point in time, what does it mean to be in charge, specifically, of DB2? It's an important product, presumably bringing in lots and lots of money. Do you get to keep any of the money? Do you get a percentage? Do you just have to beg for all the resources? Do you get any kind of autonomy in terms of setting the direction of the product?

Zloof: All of the above.

Lindsay: None of the above.

Bohl: We had a lot of autonomy. We figured out what needed to be done.

Haderle: The features and functions that went into the product were totally determined by "the group." They were self-contained working for Marilyn. Even though they were announcing the thing as just for the decision support market, we knew that if we were going to survive, we had to be in the transaction and batch markets, even though we were not supposed to market the dang thing in those markets. So 80 percent of our energy was aimed at being transaction/batch capable, viable in that particular market. Because we knew our long-term survival was dependent on that. So it was totally self-determined.

Peter Capek: Was there any feedback from market success, development resources in a direct way?

Haderle: Oh, yes, we had early customer support programs linked. Gosh, run around-- we don't run around to every dog-- you know, the ten customers that were successful or not, and look at everything that we're doing and figure out how to make it more successful.

Bohl: Then there were the user groups.

Haderle: User groups. IDUG [International DB2 Users Group]; the DB2 user group came out right about that point in time.

Grad: Was there any SHARE [scientifically oriented user group using IBM mainframe computers], or Guide impact during that period?

Haderle: Yes, there were both SHARE and GUIDE [business-oriented user group using IBM mainframe computers]. They were less influential at that point in time; SHARE was aimed at the MVS users, and GUIDE was more for the DOS environment at that time. There was a small GUIDE database group and they kept on trying to drag us down that avenue. And the SHARE group was more transactional, where IMS was.

Bohl: We tracked revenue, and we did some tracking of what was the additional hardware revenue that came in when DB2 was installed. And we were going to Bertram to get our piece of the Division's (GPD) pie for the resources for the next year.

Haderle: You bet.

Bohl: We had a whole lot of that kind of analysis.

Haigh: Did you have your own sales and support people? Or were they sitting in some other parts of the company with a wall between you and them?

Haderle: The marketing division does all the sales.

Jim Strickland: There were two separate questions: sales; and support. Sales, the wall was there. That was the marketing division. Big wall. Tough to deal with. To the point where often we talked about having a DB2 sales force, but I don't think that ever came about.

Lindsay: That had been talked about.

Bohl: As to support, a unique thing of DB2 compared to IMS was in IMS we shipped source code. With DB2, we said we weren't shipping source code; we were only going to ship object code. At that point in time at IBM, support was a whole different division of the company. I mean, it didn't come together until, I don't know, the upper echelons of IBM management. But because we were going to ship only object code of DB2, I worked with that support division, and we got support people working, for example, in our test group long before the product shipped. Theoretically, it was so that they would be able to support DB2 when it came to the market. Since I didn't control the revenue or the costs of support, I'd have to work with my counterparts in that organization as to how many people they were willing to assign to DB2 and what their support revenue was going to be.

Strickland: That was the Field Engineering Division, and out of the old hardware background, field engineering fixed problems. They had the big lot on the customer's site and they had the database support for APARs [Automated Programming Analysis Reports] and so on. We had three levels of support: Level 3, 2 and 1. Level 3 -- remained in FE [Field Engineering] forever, I guess. The problems that could not be solved at Level 3, they'd go to 2; if they couldn't solve them, they'd go to 1. Level 1 support was always in development, and Level 2 support gradually moved from field engineering to development over the years. Is that a fair statement? Gradually we got closer and closer and put them in the same building, and finally we took them over and included their headcount in the development headcount.

Haderle: But we didn't have any trouble getting sales guys to be attentive to DB2. And Bruce hit the nail on the head in terms of "why" the attentiveness was there. With IMS in that era, application development times were long. In order to introduce a new database, or change an existing database, you had to do a dbgen, and you did that on Saturday. And when you wanted to go in and change the application, you had to do an applgen, and you did that on Saturday. So you got a one shot per week. It was kind of like 1971 compilation. You have only one shot each week to change the database, and to change the application. Then you ran your tests. Then you got another shot, the next weekend. The big winner that we were going after was that the guy could introduce a database, change a table type, introduce a new application, and with DB2 do it quickly, <snaps three times>. Snap and he was off and running. We were not targeting the big honker application, instead it was targeting the suggestion program, or something that wasn't exactly vital to you, but you could get it done rather quickly.

Bohl: And you'd sell DiskScribe.

Haderle: And you'd sell DiskScribe. And the sales guys saw this differentiation right away, because they had the application backlog demand, and they could satisfy it real quickly with this thing over here, and then put the stress on the robustness and the rest of that, and still maintain the IMS apps. So we weren't struggling at that era in trying to get sales guys, because they, "poof," saw this opportunity straight away.

Relational Competition to DB2

Grad: Were there any of the other mainframe manufacturers who had significantly relational products at that point in time?

Codd: The only one was Tandem.

Haderle: Yes, the only one was Tandem, and they didn't have a SQL-based product. I mean, at that time, in that 1983-1984 timeframe, they didn't have SQL.

Lindsay: VAX had database stuff.

Haderle: They did, but they were in the department, they weren't in the big enterprise side.

Lindsay: When DB2 came out of the box, we did not have good relational competition. Our primary competitor on the DB2 side was IDMS-R, because they slapped a relational interface on top.

Grad: It was a fake, I guess.

Haderle: There you go.

Bohl: Yes.

Haderle: And Datacom/DB, which was another slap-on the top of the thing. Those were the primary competitors.

Grad: Did any of the independent database manufacturers, systems manufacturers, software companies, did any of them produce a real relational product?

Haderle: Against MVS?

Bohl: Oracle ported version 10 to MVS.

Grad: Okay, so that's one. That's what I was looking for: to see if there were any of the other companies who were already in business going after the IBM MVS market. Again, they weren't eating their own children.

Haderle: Because who we were comparing ourselves against in those days was IDMS, Adabas, and Datacom/DB.

<u>IBM's AS/400</u>

Grad: One separate question, but related. This is about the time, in the late 1980s, that the AS/400 was coming out. The words I've been told is there was an integrated database management system along with the operating system.

Bohl: Right, yes.

Strickland: It was also always called relational, whether or not it was. It was always called relational, but it used the same interface they've been using for many years, RPG [Report Program Generator].

Grad: It came out of the RPG world?

Haderle: So we well know that the System R prototype had the distinction between the RDS layer, which was a query compiler, right? And the run-time, and then a storage layer, the RSS, which was the storage sub-system for doing storage and security, and all the kind of operation extensions.

Lindsay: Security was in the RDS.

Haderle: And on AS/400, all that RSS stuff was-- you couldn't do it above. You had to go in and do it at that next layer, right? You had to do it in the microcode there. Then they took the RDS layer of System R, and retooled that to run in their particular environment.

Lindsay: Are you sure they took the RDS, because I don't ever remember having any questions from them. Okay, so they asked you the questions? .

Haderle: Well, not really. But they took our PL/S version of RDS.

Grad: So in fact, they did use the results of the work you had done. That's what I'm asking.

Haderle: They did. The sequence is that we took RDS, a PL/I thing, and revamped that. And then for the S/38 they took that and revamped it again.

Grad: They sold 400,000 of those machines or more? Unbelievable.

Bohl: There was a group in IBM working on the SQL language; there was an AS/400 rep on that.

Lindsay: Well, the AS/400 had one of the strongest representatives on the SQL standards group.

<u>Kanji Implementation</u>

Strickland: Let me make a quick topic change, if I may, because this ended up being the last part of my career at IBM. And that's Kanji.

<groans from a few participants>

Strickland: By which we needed to get double-byte character set support into DB2 and the rest of IBM products so that we could support the Kanji character set-- the Chinese character set used in Japan. And my only comment is that was another case of trade-offs, or seeming trade-offs. The highest levels of management said, “We’ll do it.” And at the lower levels of management, nobody wanted to do it. I was hired to try to get that done, and to some extent-- well, to a great extent over a long period of time, we did get it done. My question is how’d we end up doing in Japan, because I really don’t know the answer to that?

Haderle: Great!

Grad: Did DB2 succeed well there?

Strickland: Excellent. We had to translate everything, of course, to Japanese. All the help screens and everything had to be translated to Japanese.

Haderle: We had sales in Japan before this was done, because they were still running everything in English anyway. They were used to it, so your work was to try to spur new sales, quite frankly.

Bohl: We had three guys come over, remember?

Haderle: Yes, they had three guys translate the manuals, right? And translate manuals as opposed to watching the screens. The bigger issue was really accepting the various character code sets, and then dealing with it.

Lindsay: Been there, done that!

Haderle: Yes, right.

Research and Development Working Together

Haigh: I’ve got a question. During the first half of this meeting, people over here talked [pointing at Lindsay, Blasgen, and the Research guys]. During the second half, people over here talked [pointing at Haderle, Bohl, the development people]. Was there any ongoing relationship? You know, to go through DB2, new versions, new capabilities with IBM Research as the product matured?

Codd: Yes.

Bohl: I was the product manager for DB2, and Pat Selinger was key in the lab. We worked closely, and started DBTI, the Database Technical Institute. The objective of that group was to get Research people and development people to talk about what were the priority items that needed research work on. We had joint people working on different topics.

Haderle: The reason you see Bruce over here is he was from Research. We were from the development side. Bruce was half the time in our shop, half the time in their shop; we were in their shop half the time. So there was a big cross between the two. In fact, on this row-level locking problem, we started the DBTI, and the first guy that we signed up was C. Mohan. He worked on the algorithms behind all this, so that when we introduced it as a product we have something that's going to last for more than two releases. We had lots of joint projects going on with hard problems.

Lindsay: Research had been, for perhaps too long, in the middle of developing IBM database products. As I mentioned, we were involved with the technical design of the OS/2 standard edition database. That mission eventually moved to Toronto, where they said, "Make it 32-bit and run on UNIX." So they finally got that mission and they did it. The Austin product they started with had a very sound storage level, RSS level implementation. Not quite so spiffy on the higher level query management. For about the five years we were working with them, we also started a new project just for Research. We said, "Let's start over and build a new database system." We built sort of a half-ass storage level. Focused very hard on the query level. And we had an up-and-running query level with a lot of new ideas in query re-write that Chris Date mentioned. We really formalized that, and made it a separate part of the optimization process. We got our support of SQL up to ANSI standard, because none of products were really very close at that point. Toronto realized that, "Gee, we're a little weak on this database stuff." I remember being in Toronto and saying, "Well, you know, we got this stuff in Research." I remember I said something very impolitic, in front of Janet Perna, their manager at that time, I said, "You know, we got five people that would be STSM's if they were in your division." They didn't have any STSM's in their division.

Zloof: What's an STSM?

Lindsay: STSM was a pretty high rank technical person.

Blasgen: Senior Technical Staff Member.

Lindsay: For a long time, it was about the highest technical rank they had in IBM.

Blasgen: Level 60. We used to only have level 59, "senior," right? Now they have an even higher level, called distinguished.

Lindsay: In Toronto I talked about our system which we called Starburst at that point. I said you should take our Starburst higher level, and they bit! It amazed me. They said, “Okay, do it!” And Research was involved. We did it. We took the seven people that were on the Starburst project and said, “Stop evolving Starburst.” Let’s do a heart/lung transplant on DB2 in Toronto. We did that for three years. And it really was a like heart/lung transplant. I mean, we ripped out a lot of the guts, and put in some equally bloody guts.

Haderle: But elaborate on Marilyn’s comment on this thing called the DBTI. This is the model that’s used inside of IBM now for how research and development should interact. Pat Selinger, Irv Traiger, and I put together a proposal, and we put individuals and resources into this. We then worked on joint programs. DBTI would point out that the thing that we have over here is what we need over there. Or how do we solve the technical problems that we needed to get solved. The funding level was 20-ish people.

Lindsay: Less than that.

Haderle: Okay? And so they would come together and work on these joint problems, and they were linked in with what was going to be in the product plan and the product needs. DBTI was a way to interact. And then we had the pure research going on, and that was a different interaction.

Grad: Bruce has made the point here, and I think to follow-up on your point before, Mike, that research was still the creature of what could be marketable, rather than pure research, or long-term research, without having a specific product in mind.

Lindsay: Well, yes and no. The Starburst project, half of which we put into the product, was envisioned by us as a long term research project. Another project of ours was a "database workbench." We said, “Wait, there’s a million ideas we want to try in database, but you know, we don’t really have a good malleable software environment to do it in. We’ll build one.” The good news is we built this thing, and it turned out to be useful for the product. The bad news is by the time we got through doing the port, we were all very tired, and we never got back to our "workbench."

Grad: You didn’t get to do the research that you were hoping to do with the "workbench," then?

Lindsay: Right.

Grad: We’re running out of time here, and we’re going to break in two minutes. One final question, and then I’m going to close the session.

Haigh: Did anything from IBM Research ever make its way into IMS? Or was that just stagnant technologically?

Haderle: From IBM Research, all of the instant recovery stuff that Jim Gray and Franco Putzolu were working on got into IMS, including the concurrency control, and real-time, and 24 by 7.

Blasgen: The RSS was finished before RDS, and two of the people who working on the RSS were now, in a sense, done. Franco and Jim took leaves from Research to join development to find out how development worked. This was maybe 1977.

Haderle: Irv Traiger came over at one point too.

Blasgen: You're right. Those were almost all the people, I mean, besides me, and I'd been promoted to my level of incompetence, so I couldn't go.

Grad: I'm going to have to draw the meeting to a close. It was a fascinating session. There's so much material we could have gone on for three more hours on this even without questions. Thank you for your help, Tom.