Raghu Ramakrishnan Speaks Out

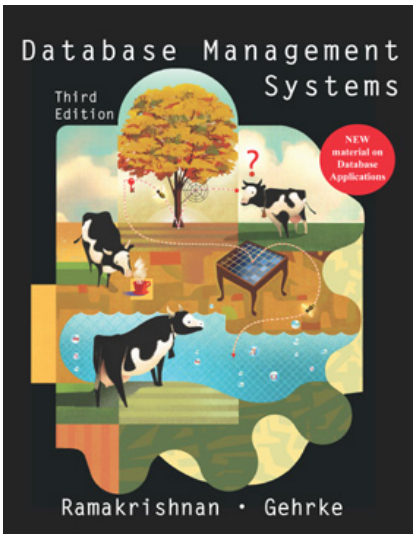
on Deductive Databases, What Lies Beyond Scalability, How He Burned Through $20M Briskly, Why We Should Reach Out to Policymakers, and More

by Marianne Winslett

Raghu Ramakrishnan
http://www.cs.wisc.edu/~raghu/

Welcome to this installment of ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are at the Department of Computer Science at the University of Illinois at Urbana-Champaign. I have here with me Raghu Ramakrishnan, who is a Professor of Computer Sciences at the University of Wisconsin-Madison. Raghu's research currently focuses on data retrieval and integration, analysis, and mining. Raghu was a founder of QUIQ, a company that developed a collaborative customer support facility. Raghu has received the National Science Foundation's Presidential Young Investigator award, a Packard Fellowship, and a SIGMOD Contributions Award for founding and maintaining DBWorld. Raghu is an ACM Fellow; he is a coauthor of the popular Database Management Systems *textbook; and he is also the current chair of ACM SIGMOD. Raghu's PhD is from the University of Texas at Austin. So, Raghu, welcome!*

Raghu, let's get the most important question out of the way first. What does the cover of your textbook, Database Management Systems*, mean?*



It's a depiction of how queries are processed. The cows are users, from Wisconsin of course, with the same number of cows as the number of editions. [Wisconsin is well known for its cheese.] The question marks are queries. When you have a query, the arrows show you what happens. You go to the root of a B-tree, and that directs you to a leaf, where you find a key. You take that key over to a relational table, to find the records you need. Then you put the records into a buffer pool, where you cook them, using relational operations. That, in a nutshell, is what database systems are all about: carrying out computation across memory hierarchies and moving data back and forth in a clever way.

You cut your teeth on research into deductive databases, with the Coral project. Deductive databases have been out of fashion for many years, but you had a paper in SIGCOMM 2005 entitled "Declarative Routing: Extensible Routing with Declarative Queries." Does this signal a renaissance for deductive database research?

The paper originated because Joe Hellerstein asked me to be involved in this work. Joe and the Telegraph group at Berkeley and at Intel have been looking extensively at how to use recursive rules inside of network routers, to do all kinds of cool things. Joe is fond of saying that maybe Oracle and the other database companies will support deductive databases, or perhaps Cisco will have a recursive query processing product soon.

As another example, Serge Abiteboul and his colleagues had a paper recently where recursion was driving an application [Serge Abiteboul, Zoë Abrams, Stefan Haar, and Tova Milo, "Diagnosis of asynchronous discrete event systems: Datalog to the rescue!", PODS 2005, pp. 358-367]. Their proposed system had some of the elements of the trust policy languages that you were talking about earlier, with distributed rules executing---essentially, Datalog with a fixed point. In that paper, essentially they were simulating a certain class of Petri nets of finite state machines in a highly distributed fashion.

Of course, the trust policy work that you and I talked about yesterday also uses Datalog. And when I was at Microsoft some time ago, someone there wanted a copy of Coral because they were using recursive rules in their debugging work. Their work involved graphical representations of programs, and they wanted to borrow an open source recursive language.

So it turns out that there are indeed a lot of recursive applications. Enough to sustain an industry? I don't know about that. It does seem that there is a growing resurgence of interest. Maybe that is because of the natural presence of recursion across nodes in the internet; maybe it is because of languages like XML (XPath has recursion); or maybe it is because of natural delegation in things like policies for security and access control. There is a lot of recursion latent in the world and I think we are finally coming to applications that are tapping into it.

Since we are talking about old things that are new again, let me ask about data cubes. Are they back in style again?

I would like to see them come back in style. If you look at data mining, traditionally what database people have brought to the table is scalability. Given an algorithm that works well with a small data set, database people have figured out how to make it work with much larger data sets. Database people have done this for clustering, classification, frequent item sets. But I think we have a whole lot more to bring to the table than just scalability. We understand notions like compositionality, and we understand things like how to structure a space for exploratory analysis. That is what the multidimensional data model and data cubes are all about. How do you take this set of concepts and combine them with predictive models or other kinds of tools borrowed from statistics and machine learning? In the case of machine learning, a standard research focus early on was how to take a given "case" and analyze it: should I give this person a loan or not? In the case of exploratory analysis, the emphasis has been not so much on an individual case, but rather on understanding the data set as a whole: in what parts of the world and in what periods of time did loan decisions have a certain bias? To answer these kinds of questions, you can employ the power of the exploratory analysis tools that we database people have developed, and combine them very profitably with the kinds of summarization and predictive tools that the machine learning and statistics communities have been developing. I think this could lead to a hybrid set

of tools, a hybrid set of approaches to exploratory analysis, that are conceptually more powerful. In other words, we would not just be scaling up existing tools developed in these related research communities, but instead developing a conceptually broader and entirely different class of hybrid tools for analyzing the enormous amounts of data coming at us.

It seems to me that you have now gone full circle in your research interests---back to where you started. Are you all done now, or are you going to go around the circle again?

Hopefully not the same circle.

A spiral?

Hopefully not a downward one. I enjoy going around and around in circles, I think, so I will probably do that for a while.

The 1989 Laguna Beach report on future directions in database research was quite negative regarding deductive database research. In hindsight, what do you think about such reports and their impact on deductive database research?

I must confess that back when I first came across the report, I was not a happy camper. Over the years, my reaction has mellowed. I think I do understand the value of having senior researchers in a field offer their opinions to help junior researchers in making their choices as to what to work on. But that said, I still have mixed feelings about the value of reports like this. On the whole, I would like to see emphasis on directions that people think have promise, rather than emphasis on things that one or another group of people feel ought not to be emphasized.

Why is that? I could see both of those playing a role: do this, don't do that.

When you think something is important enough to spend your time on, and have invested a significant fraction of your time and energy, I like to think that you have given enough thought to that choice. Now, if you think that something is *not* important, you have actively cast your vote by simply working on something else that you believe *is* important. By actively downplaying a topic that others might consider important, I think you could be doing a disservice to something that you have not thought about as deeply as those who are committed to that particular vision. So I would rather see you thoroughly focus your energies on promoting that which you have given the most thought to and invested the most in, because that is probably where you have the most insight and the most passion, and where you can have the most legitimate kind of influence.

How have database vendors made use of the many years of deductive database research in improving the performance of SQL3 queries that involve recursion?

I think that there are two areas, at least. First, the idea of magic sets rewriting has helped in dealing with correlated queries. When you have nested queries with correlation, magic sets rewriting turns out to be a very useful tool for dealing with correlation in a set-oriented fashion. I believe several venders have used this to get better performance on the TPC-D benchmark. Second, incremental view maintenance became a very hot area in the 90s. Virtually every vendor supports some version of this, and most of these techniques generalize and build upon some form of semi-naive evaluation. Both of these core techniques, magic sets rewriting and semi-naive evaluation, were developed initially in the deductive database community.

What do you think database theoreticians should be working on today?

In accordance with my earlier comments on how to answer questions like this, let me speak strictly from the perspective of the areas I have worked in. I have looked lately at information extraction from text. I think there is a lot of work to be done on handling data with associated uncertainty and handling probabilistic reasoning, not just in a query setting, but also synthesizing ideas from the statistical learning and the statistical analysis literatures. Fundamental practical problems like deduplication can have elegant mathematical frameworks, too. In fact, I think all of these topics do have active ongoing work of a theoretical nature. I think theoreticians have recognized that these are areas of opportunity, and they are working on them. Another broad area of opportunity is social network analysis, which seems to be popular in the web community. And simply because of the scale of the data, I figure that counts as databases, although perhaps it is not mainstream databases (yet!).

Raghu, during the course of your career, you have moved from pure theory to systems research, and from academia to a startup and back again. What led you to make these changes?

I have always had an interest in studying things that involve some kind of real application. Even in the earliest days, when my work was about recursive queries, it was to be included in an implemented system. Initially, when I was a graduate student, the target was the LDL system, and then later as I started my own research, it was the Coral system. So most of the theory I did was related to some ongoing practical effort that I was involved in.

Over the years, I have come to build and reason about systems more often, whenever an opportunity offers itself. But I don't see this as a change, really. While the percentages may have shifted over time, the basic interplay between systems and theory has been part of what I have done all along.

What about your startup company, QUIQ?

The startup was something I just could not resist. There was an opportunity to really do something outside of academia, to actually build a product that billions and billions of people would use. There was also an opportunity to work together with my brother, who had graduated with an MBA from Stanford: the opportunity to have a business person and a technical person work together.

Those were boom times, money seemed to grow on trees, and we actually raised about $20 million over the course of a couple of years. It was great fun building the company, and we did some pretty cool things similar to Yahoo Answers today. We were doing that for Ask Jeeves back in 1999, with a million different alerts being monitored continuously for Ask Jeeves alone. We had clients like Compaq, Sun, National Instruments. The idea of using mass collaboration in the context of technical support and question answering was a really very intriguing concept to try and take all the way to the point of commercialization. The end game was less enjoyable.

Tell us about the end game.

The bottom line is that we worked our way through $20 million a little more briskly than I would have liked. It took us more time than it should have to translate our technology into a coherent business strategy. The sad part is that we eventually did come up with that strategy. Compaq was using our product extensively at the end, they were paying us very significant money, and they were very happy with what they were getting in return. Several large companies were already our clients, but we were caught in a bind where we were running out of money at around

the same time that we were trying to close a bunch of deals. We were not able to close them fast enough, given the economic downswing then. And a few other things happened. We had a key deal with Apple which fell apart when the group at Apple got fired, after we had started working for them. And the venture capitalists---at that point, shall we say, they did not want to invest in that climate. And all said and done, we had to wind things down and sell the company at a level that made no one very happy. There were earlier opportunities when we could have exited much more gracefully and profitably.

While building the company was enormously exciting, on the downward path, we had to tighten our belts and let people go. Those were some of the hardest things I have done.

All in all, it was a unique experience and something that I think I wouldn't trade.

How was the transition back to academia?

Hard. I think that the venture capitalists would have like the first couple of grant proposals I wrote, but my peers didn't. My whole value system was skewed. I was looking for things that made business sense, as opposed to academic novelty. And it took me a while to readjust my senses.

Did you have the reverse problem when you first got to QUIQ---would NSF have liked your proposals?

I did tell you we blew through $20 million briskly, without making much money, so I will let you draw your own conclusions.

In the beginning, data mining research seemed to have a home in the database world, but now it seems to have evolved into a separate field with three major conferences of its own. What should the relationship be between the two fields?

I think it is healthy for the two fields to have a complementary relationship like they do now. Data mining involves more than databases; machine learning and statistics are integral components of data mining. It is good for the data mining community to have their own identity; to try and form a set of foundational ideas that are unique to data mining, that are at the intersection of and distinct from each of the contributing fields.

At the same time I very much hope that we continue to see data mining papers in the main database conferences, that we see database researchers at least attending the occasional data mining conference, and that we don't see a split between the communities. Obviously, this is going to be beneficial for data mining, but equally, I think, the database field in the future is going to be largely concerned with how to make sense of data. A lot of what is happening in data mining is very relevant to this goal. For example, if you are doing automatic tuning at virtually any level of the system, you can instrument your system and apply learning techniques to help in the tuning. What other techniques are appropriate in your setting? Who might you collaborate with? I would like to us take advantage of these synergies.

Raghu, recently you have become interested in data privacy and security. This kind of research brings in new kinds of considerations that database researchers are not used to thinking about: legal, ethical, and public policy. For example, if I publish a join algorithm that I say is 20% faster, no one really cares if I am wrong. But if I say I have a new data anonymization technique that preserves privacy 20% better, that is a scarier promise: the public is going to think that it

means something. Have you come to grips with this new dimension, and how are you going to handle it?

It's a very good question. I spent some time recently working on a paper on data mining and the law, together with Deirdre Mulligan, who is a professor of law at Berkeley, and Chris Clifton from Purdue. It was a real learning experience. We technical researchers are developing tools that can be used very invasively. I don't think the appropriate response to that is to say that we shouldn't develop such tools. Those tools are going to be developed, by someone else if not by us. At the same time, I don't think the appropriate response is to say, "Here's a technique, and all I did was develop it, and I am no longer responsible for what someone else may do with it."

Did you hear in the news today that the government has subpoenaed all Google searches for the past k *years?*

Yes. Google seems to be at the cutting edge of so many different legal questions.

But they are also about scale, just like we are.

They are about scale, they are about copyright. Consider the topic of databases on the web: ownership, legal rights, privacy---these are all related, you cannot entirely separate them. Ownership is at the heart of many of these questions. Whether you are talking about ownership of copyright, or thinking of ownership of my private information, this whole thing is a gray area on the web.

But to get back to your original question, I think we database researchers should be thinking a great deal about issues such as how to characterize the extent to which different tools can be used to achieve certain kinds of compromises of privacy and security in different settings. This doesn't mean we set policy, but we have to provide guidance to those who do set policy. And we have to provide tools to help enforce more flexible policies.

We need to be thinking about this in terms of the algorithms we develop, or the theorems we prove: don't just prove theorems about how fast something is, prove theorems about how far you can go in, e.g., cracking a certain kind of anonymization. Not only do we have this new agenda research-wise, I also think we have a burden to do something that we have traditionally not done. We need to be involved socially and legislatively, to influence people who set policy. We need to inform them, because they may not be fully aware of the kinds of risks associated with the data gathering that is going on today, and the tools that are available to subsequently analyze that data. We may not be aware of all the data gathering that is going on, but certainly we are aware of some of the possibilities. We are aware of many of the tools that are available or can be easily developed. I think we need to become more socially active.

Traditionally, we haven't been very good at pursuing that engagement. So how is it going to happen? Who is going to do it?

It is going to be us, although we have not taken on this particular gauntlet in the past---

I mean which *us?*

You and me: the people who have tenure in the universities, leadership positions in industry, us; not the young ones who have other concerns that are more immediately pressing for them. The

older people like us have both the time and the perspective to step back and look at not just the immediate research we do, but the consequences of that research. We ought to be doing that.

How did you end up in database research?

This is an interesting question. I went to the University of Texas and took a course on databases with Hank Korth. I really enjoyed it. I really don't know how I came to knock upon Avi Silberschatz's door, but I did, and we started working together. My first paper was on concurrent logic programs; it appeared in POPL, and I decided that it would not be my thesis area. Avi then pointed me to some work on data modeling. And then along came MCC and I found myself in a position where there was this entire new field of deductive databases emerging. I had the opportunity to work with some brilliant guys, excellent people, like Catriel Beeri, Francois Bancilhon, Carlo Zaniolo, and others. And I had exactly the right background; I knew about logic programs, I knew about databases. So I just jumped in there, and after that, I just kept going.

How was your schedule as a graduate student in Austin?

Well, that was back in the days when you had shared mainframes. Working during the day was painfully slow. Parking wasn't very good, either, so I took to working after dinner and going to sleep after breakfast. With that approach, I got excellent parking, and my machine was as fast as my machine is today. Some people did look at me squinty eyed because I ate cereal in the evenings. When eventually Bancilhon went back to France, his parting comment was, "Finally Raghu and I will be awake at the same time."

What is the spiciest food you have ever eaten?

I once made the mistake of going into a Thai restaurant and asking them to make it very spicy. I like spicy food, but never, ever will I go into a Thai restaurant and ask for very spicy again.

Can you describe it to us? How spicy was it?

Too painful to remember.

The past couple of decades have been hard for many professional associations, which have seen large drops in membership. Computer science associations have not been immune to this trend. What do you see as the role of ACM SIGMOD in the life of database researchers over the next ten years?

First of all, there is also an opportunity here, especially for information technology, for databases. Countries like India and China are exploding in terms of the number of professionals. Our enrollment of members from those countries is incredibly low. We need to be much more aggressive in getting members globally and retaining them beyond the first year.

And what will you offer to the people in India and China that will make them want to join?

Networking and awareness of what we as a community are doing. These are the same benefits that we get here: staying in the loop.

Can't they get that off the web?

I see the web as an opportunity. It used to be that you got the most benefit out of these kinds of memberships if you were in a position to physically attend the conferences, which were mostly in the US and occasionally in Europe. The web makes it possible to capture the conferences in video and make them available on the web, and this takes money. Part of that money comes from memberships. We have to figure out a business model, because at the end of the day ACM SIGMOD is a business. Our focus is not on profits, but we have to stay above water; we are not subsidized by anyone. We need to figure out how we can take all the things that are going on, such as conferences, symposiums, and other events, and make them accessible to people in our field, no matter where they are geographically. That's the benefit that SIGMOD members get, and hopefully they consider it worthwhile enough to pay for, and, in turn, to let us make those benefits generally available.

We also need to have our conferences in distant parts of the world. I think we are doing that. SIGMOD has made a very conscious effort to reach out and relocate elsewhere. I hope to see that continue. I think organizations like SIGMOD can also take a leadership role in bringing together communities outside of SIGMOD that share common interests, such as SIGIR, SIGKDD, and other SIGs where there would be benefit in researchers coming together. Maybe we can be creative and work together with those SIGs, to create opportunities for cross fertilization.

From among your past papers, do you have a favorite piece of work?

I have two. First, the magic sets paper I wrote is one of them, simply because I spent six months trying to figure out how to generalize the version that Bancilhon, Sagiv, Maier, and Ullman had written, and the solution came in the space of five minutes over coffee. There was an "Aha!" moment that I will always remember. As my other favorite, I did some work on dealing with sorted relations (sequences) that appeared in SSDBM and is relatively little known. It ended up influencing window functions in SQL99 quite significantly. I have always had an interest in streams and sequences, and this was the piece of work that had the most influence, so I have a fond spot for it.

If you magically had enough time to do one additional thing at work that you are not doing now, what would it be?

I would probably have lunch with my colleagues more often.

If you could change one thing about yourself as a computer scientist, what would it be?

I would take some smart pills.

People say that you have a great sense of humor. Do you have a joke to share with us?

I should tell you the story about Bill Gates and the Indian entrepreneur. Long ago, an Indian entrepreneur came to visit Redmond, and told Bill, "You really ought to go to India and hire people, because that is where the action is." But Bill wasn't interested. He said, "No, come with me." He gave the Indian guy a spade and asked him to dig. The entrepreneur dug, and dug, and dug. Up came a cable, and Bill said, "*Cable!* That's where the action is." (This was back during the time when Bill was trying to buy cable companies.) The Indian entrepreneur was very upset and went back home. Eventually, though, Bill saw the light, and went to India to recruit. (Microsoft was one of the very first US companies to recruit directly in India.) His Indian entrepreneur buddy gave Bill a spade, and said, "Dig." Bill had to dig, to return the favor. It's hot in India, and five minutes later, Gates was sweating. After ten minutes, fifteen minutes, he

was knee deep, and he's not exactly a spring chicken, so he was getting very tired. He looked up, and the Indian told him, "*Wireless!* That's where the action is."

Thank you, Raghu, for talking with us.

You are very welcome.