



RUNAS RADIO



<http://www.runasradio.com>



Richard
Campbell

RunAs Radio is a weekly Internet Audio Talk Show for IT Professionals working with Microsoft products. The full range of IT topics is covered from a Microsoft-centric viewpoint.



Greg
Hughes

Text Transcript of Show #0104
(Transcription services provided by [PWOP Productions](#))



Kim Tripp on the Roles of Developers and DBAs with the Database!
April 8, 2009



[Music]

Brandon Wenn: From runasradio.com, you're listening to RunAs Radio, the Internet audio talk show for IT professionals with Richard Campbell and Greg Hughes. This is Brandon Wenn, announcing show #104, with guest Kim Tripp, recorded Thursday, March 12, 2009. RunAs Radio is produced each week by PWOP Productions, providing professional media and podcasting services online at pwop.com. You can follow the boys on Twitter at twitter.com/runasradio.

Richard Campbell: You're listening to RunAs Radio. I'm your host Richard Campbell and with me as always my co-host and newly minted pilot friend, Greg Hughes.

Greg Hughes: Hey Richard, how are you?

Richard Campbell: Congratulations, man.

Greg Hughes: Thanks. Yeah, I finished my private pilot certificate.

Kim Tripp: You got your private pilot license, Greg?

Greg Hughes: Yeah, I can fly you around in the air now.

Richard Campbell: Right on. Congratulations.

Greg Hughes: Yeah, it's actually a lot of fun. I started the private pilot training down here, you know I live in Portland Oregon area, down here in Hillsborough in mid September and actually finished it last week, so early March, first week of March was when I finished it all.

Richard Campbell: Wow.

Kim Tripp: Can I ask what you did for your long distance solo flight, your first long distance solo flight?

Greg Hughes: Sure. On my first solo, I flew down to Lebanon, Oregon which nobody, I'm sure, has ever heard of and back, and in my first long distance solo flight I flew from Hillsborough down to Newport, Oregon which is southwest of here on the coast, and then from there flew up the coast on a beautiful day. It was literally 70 degrees up in the air in January, and flew all the way up to Astoria which is at the mouth of the Columbia River right there right at the topmost corner of Oregon and from there back to the Portland area. So it was a lot of fun.

Kim Tripp: Hillsborough was one of my stops on my long distance solo flight so...

Greg Hughes: Oh cool.

Kim Tripp: Yeah, so I know it well.

Richard Campbell: And for our listeners who haven't guess yet, our guest today is none other than Kim Tripp, needs no introduction, friend of the show, my dear, dear friend and of course immediately we're just in conversation because we never get the chance to talk to each other near enough.

Kim Tripp: So I know it's pathetic.

Greg Hughes: It's good to talk to you again, Kim.

Kim Tripp: You know, I've actually got my log book out. I'm actually finding when I flew to Hoquiam, oh it was Hoquiam, I was thinking...

Richard Campbell: Oh, it's funny.

Kim Tripp: Sorry, I'll shut this, but yeah. Anyway, you guys catch up, I'll be quite for another minute or two, I'm sorry.

Richard Campbell: Let's do the show.

Greg Hughes: The next time we get together, we'll have to geek out on pilot stuff.

Kim Tripp: Yeah, absolutely. You know what, I'm not current. I haven't flown. Nobody would give me a plane at all but it's such a great hobby. I mean, it's just so fascinating to learn all the radio-speak and everything you need to do to learn where you can fly, where you can't fly, different altitudes...

Greg Hughes: Sure.

Kim Tripp: What happens when you have to fly around CTAC, I mean I've got some great stories but I haven't piloted a plane since '95 so it's been many, many moons.

Richard Campbell: Wow.

Greg Hughes: Next time you're in the Portland area, make sure you let me know and we'll fly around for awhile.

Kim Tripp: Oh, totally. I would love to go. What are you flying then? Cessna 172?

Greg Hughes: I've learned on a 150.

Kim Tripp: Yeah, okay.

Greg Hughes: And now it's a Warrior so 150...



Kim Tripp: Ah, Piper Cherokee Warrior. Those are great planes, low-winged planes. I flew those too. I was flying those and 172s.

Greg Hughes: Yeah, so a whole lot of fun. Definitely the freedom is pretty darn cool.

Kim Tripp: Oh, it's awesome, yeah. Oh, we could do a whole show. There's probably a whole bunch of other geeky people that are all certified and love to fly as well.

Greg Hughes: Yeah, that would be interesting, but in the meanwhile...

Kim Tripp: Sorry.

Richard Campbell: Kim, the reason we're doing the show actually is a bunch of emails. You and I were involved in around this whole state of affairs with developers and DBA's these days and you seem a little surprised. Weren't you -- tell me your version of the story here.

Kim Tripp: Well, okay. So Timmy Huckaby sent an email because he was going to do an interview for a San Diego newspaper or magazine and just wanted to kind of chat a little bit about features that developers would really be excited about in SQL 2008.

Richard Campbell: Right.

Kim Tripp: At least that's what I took the email to be about. So we kind of started down the path talking about new developer features and I forgot how it kind of went off tangent but it kind of got into the, well, but does a developer really need to know all these features and does a developer really have to know SQL really well to leverage SQL as a backend and that kind of spiraled into a, well, wait a minute, whose job is it to do the database schema, the database tables, the indexes, the stored procedures. Is it the database administrator that has to write the code, to slam that code into production? Because obviously a DBA is the only one that should really be implementing the code, but is it really the DBA that writes the code or is it a specific person, a database developer that is knowledgeable about SQL that creates all these, or is it the app developer, and so that kind of spiraled into a huge discussion that must have had probably 20 emails back and forth and I ended up writing a blog post titled "Whose Job is It Anyway," and kind of went into do you think that there's the re-distinct roles here, do you think there's an app dev, a database developer, and the DBA or not, and it was great, I got great comments to it, not only was our thread fantastic but I really learned a lot

about how people are viewing kind of the world of SQL...

Richard Campbell: Sure.

Kim Tripp: And that actually spiraled into a whole bunch of other blog posts.

Richard Campbell: Not just SQL per se but this whole point of data storage in general is becoming more and more amorphous I think to developers where they're just saying it's just plumbing. I just say store this and it happens, I don't need to think past that.

Kim Tripp: Yeah. I had a couple of comments from folks that said, and they didn't believe this actually, they commented that they're just hearing a lot, that SQL was kind of a set it and forget it technology.

Richard Campbell: Right.

Kim Tripp: You know, you can slam the objects in there and SQL will just do its thing. From the last RunAs when we talked about indexes, I had a database developer who sent me an email, sorry, I had an involuntary DBA sent me an email that said, you know what, I don't know much about indexes but I decided to listen to your RunAs and I found that my databases don't have any indexes. Their developers had told him, you know, as this kind of part-time DBA that really wasn't even meant to learn SQL, they had told him, oh, SQL just does all that for me, I don't need to know about the indexes, I don't need to know any of that stuff, SQL just does it and I think with SQL as plumbing and set it and forget it technology and the SQL does it for me kind of mentality that's happening, we're seeing even bigger problems as databases try unsuccessfully to scale.

Richard Campbell: Yeah and of course this only ever hits after the app has shift and it gets busy and suddenly there are problems and it's funny how often developers just talk about the problem of SQL Server, how it's the source of all pain.

Kim Tripp: I totally agree.

Richard Campbell: And at the same time saying set it and forget it like what do you think was going to happen.

Kim Tripp: So I talked about database best practices, design best practices, indexes, but one thing that we've started to get even more involved in is telling people how important certain maintenance test are and when to do those maintenance test. In fact, oh God, I probably shouldn't embarrass myself and say this but I had this long blog post typed up



yesterday and I do have auto save on, but I accidentally reached for my mouse and hit the back button and I lost like an hour of my last post. Even though it had auto saved, when I hit back it overwrote the auto save.

Richard Campbell: Oh.

Greg Hughes: I've never done that before.

Kim Tripp: Oh man, I was so -- you should asked Paul, he's like Kimberly is so nice when she presents but when something happens and she's really irritated you should hear the things that come out of...

Greg Hughes: Yeah.

Kim Tripp: It was a very colorful moment or two in the house when I realized that I had lost like at least -- even with auto save I would have had everything but it must have been an hour or two so I've now started to rewrite that post and I've turned it into three and it's going to be about database maintenance, but I guess my point is I'm seeing a lot more of this SQL is just forming, we create it and it goes out there and then when people have problems it's a whole about, oh, SQL doesn't do this and SQL doesn't do that, and it's not set it and forget it technology, I mean there are some things that SQL does really well for you, but if you don't go and do kind of root cause analysis for why you've gotten in to some of the situations, somebody has quick fixes that I've seen as post to let's say huge transaction logs, you know, they're not solving the problem, they're quick fixing do this and it's, I don't want to phrase this, it's a developer just saying, okay, that's the cause of my problem, let me go on and put out some other fire and they're never really stepping back and saying why did I have this problem and what was it in SQL that I did wrong potentially that I could have avoided this altogether, or I'm not seeing that. I'm seeing like I'm having this problem, here's a quick fix, move on. I'm seeing this problem, quick fix, move on and it's really becoming, I think, problematic. People aren't putting enough effort into having a database focused person. They have DBAs and the DBAs might -- well, not every company has a fulltime DBA but the DBAs focused more on back-up, restore, and disaster recovery and security but I don't even know if all fulltime DBAs are learning as much as they need to know about maintenance. Maybe they're thinking of somebody else's job. I don't know, Richard, Greg, what do you guys think.

Greg Hughes: You know, the question that pops into my mind is where is the architect in this whole thing.

Kim Tripp: Well, you know, an architect that's another good question. Is that another role? Is there an architect that oversees the whole project that I would expect on bigger project? Yes, but again I'm seeing an app developer that just thinks about SQL as a storage area and doesn't think really that it matters how they define that storage areas, SQL Server is just going to deal with it, and then maybe there's some operational person that again maybe doesn't know SQL all that well and they're now having scalability problems because they've defined some apps that's not fully taking advantage of SQL Server or they've done things in the application that really aren't ideal in SQL Server but nobody has told them otherwise.

Greg Hughes: You know, there's a non-developer IT management guy, our recent guest who, Richard, you might tell me if I paraphrase accurately here, who said something along the lines of the difference between programming and using SQL Server is, you know, when you're a programmer you're telling the machine what to do, whereas when you are interacting with SQL Server you're telling it how to characterize or what information you want.

Richard Campbell: What your sort of desired outcome is.

Kim Tripp: Yeah, that's just the thing.

Greg Hughes: Right. You're asking it for output as oppose to telling it what to do, but Kim, I mean so I'm sort of translating what you're saying and trying to sort of match these things together. If you treat it just as a set it and forget it or a black box kind of throw something at it and have it come back, you simply assume that what it does it's going to do really well and do it in the most optimum way that may or may not be the case.

Kim Tripp: Well, yeah, okay. SQL Server has an incredibly intelligent optimizer. Don't get me wrong, but there are things that when you write T-SQL, if you don't write them optimally, there are things that they just won't do a good job with ever.

Greg Hughes: Oh sure.

Kim Tripp: Like one of my favorite really simple examples is isolating a column to one side of an expression. Like if you have a monthly salary column and somebody writes a query that says select star from employees where monthly salary times x 12 is greater than some value. The way that that works, and you might even argue that SQL team could have written this, but the way that that works is they look at that expression and they're going to evaluate monthly salary x 12 for every salary of the table so it's going to force a scan, it's not going to move the x 12 to the



other side and I know that's pure algebra, and yes the SQL team they're well beyond knowing basic algebra, but you have to think of an optimizer as -- and this is a great line to say but it has to come up with a good plan fast and what that means is that it has to be efficient, it has to be well written and it has to be tight, it has to not have a lot of extraneous code, and my reason to say that is as they would have put in all sorts of where clause rewrites and especially simple rewrites, you could end up bloating the optimizer with a whole bunch of stuff that normally it wouldn't have to use and you start making the process of optimization in and of itself and you could make the argument that I guess they could put in all sorts of rewriting and make optimization more expensive so that you could kind of almost write any SQL command and they figure out what you kind of wanted to do but it kind of defeats the purpose of a truly scalable environment. I think they'd be going the wrong direction doing that. They just make the optimizer so expensive. So I guess my point is if a developer that knows SQL knows to avoid these things, they can write better, more efficient, more scalable SQL that can really leverage all the things that an optimizer can do that is much more complex than that. Do you kind of see where I'm going with that?

Richard Campbell: You know, I immediately hit on this idea of is this how Cloud computing with data storage really comes into play? I don't want to know, I don't want to think about it. I just want to store my data and get it back and it will always be fast and the promise of Cloud computing is it will always scale. I just don't need to know anything.

Greg Hughes: Or is that the myth of Cloud computing?

Richard Campbell: Yeah. Well, there you go.

Kim Tripp: Yeah. You see I don't know enough to speak intelligently about how Cloud is being written and so forth, but it is a whole another interesting scenario of kind of set it and forget it...

Richard Campbell: Absolutely.

Kim Tripp: And it's even I'd say even more beyond that and that it's; I don't even want to manage it.

Greg Hughes: Right.

Kim Tripp: You know, it is not just a set it and forget it, it's set it, forget it, and not even manage it and somebody else is going to back it up, somebody is going to restore it, and in a lot of cases it might actually be a more experienced DBA that is managing your databases in the Cloud. There's no doubt about that, but I still think there's going to be

certain things, and this is something that I have chatted with a few folks about Cloud, you know, what is it that a DBA managing their data in the Cloud is going to be allowed to do and/or what do they need to do versus what the operators that are running the Cloud need to do, and this is a really big debate. I mean, this is a huge topic. It's what about performance tuning, like if I manage a SQL Server even though I don't know the databases, there are certain things that I as an administrator can start to see in terms of frequently executed queries, queries that were exhausting a lot of resources, and I can start to even run those through tools like DTA or use the DMBs like there's an index DMBs and I could start using those tools without knowing the database app itself just looking at the different activities that are going on and I could start to add indexes and I could start to do even more efficient kind of maintenance, but at the same time, and I have to argue both sides, there are things that the developer could have done better earlier on to have made some things even more scalable that there's no way that the DBA could do later to make things more efficient like my favorite...

Richard Campbell: True.

Kim Tripp: The canonical example here is primary keys that are GUIDs that are then your clustering key.

Richard Campbell: Right.

Kim Tripp: And I had this discussion last time too that I'm seeing this everywhere! Like every application that I'm kind of running into that's being developed by database developers that let's say aren't really, really deep in SQL, they create these clustering keys, these GUIDs, and the problem is it works fine when there's thousands of rows, even tens of thousands of rows, but as you get to millions of rows, as you get to an app that's in production, there's nothing that a DBA can do except maintenance and even then it's a temporary fix to the problem, and the problems are way worse than the just the fragmentation that's created.

Greg Hughes: Yeah.

Kim Tripp: It's all of the keys that are unnecessarily wide, all of the tables that are wasting tremendous amounts of space by having this really much wider than necessary keys.

Greg Hughes: That's the reason I asked the question where is the architect in this picture. I mean, so much can be saved and done through some basic requirements and planning and having the right people involved at the very beginning.



Kim Tripp: So this is interesting. I think you and I might be saying you're saying architect, I'm saying database developer.

Greg Hughes: Yeah.

Kim Tripp: I'm not sure that every project has all four people.

Greg Hughes: It doesn't have to be but somebody who is filling the architecture role, in other words the planning role in the design phase, how do we ensure that we're looking at these critical decisions and making these decisions in a smart way that takes into consideration, you know. So it's the strategic view of what is it that's being built, not just the tactical view. But you're right, it could be one person that could be doing a couple of those roles.

Kim Tripp: Yeah.

Richard Campbell: The interesting point here, I think, and why you're saying the architect is that ultimately the architect's responsibilities is to make sure he does bring all of the players together.

Kim Tripp: Yeah.

Greg Hughes: Right, right.

Richard Campbell: And get that information so I mean the reality is that if a developer goes out and says, oh, I want to use GUIDs, if the database guy has no idea until the app is deployed, then you get into a problem where if in the design meeting they said, well, we're going to use GUIDs, and the database guy goes, well, you know, there are some serious consequences to that.

Greg Hughes: Right.

Kim Tripp: I'm expecting the database developer to be that guy to tell the architect who basically say I want to use SQL because I'm thinking that a lot of architects won't know this level of depth. They might especially if they've started to see some of these problems in prior architectures and they certainly can move that forward, but to me is the database developer, and again some people don't have this distinct person so maybe it is the app dev that is the database developer or this is the DBA that is the database developer or it is the architect that is the database developer. What I'm finding though and I think this is the ultimate base of this discussion is I'm finding that nobody thinks their needs to be a database developer, that nobody thinks that the architecture in SQL has to be thought out like it just doesn't matter and that's what's maybe freaking me out, you know? It's like it's surprising me I guess, it's that that's the whole thing here. I'm really surprised at

how people are thinking that you just don't need to put the effort into the database schema and I'm not sure whose job it is to call this out and whose job it is to know early on because I don't think every project has all of these distinct people unless you're at the very high-end enterprise, but at the low-end I'm just seeing more of who cares, you know, it doesn't need to be done kind of an attitude and it's getting people into a lot of trouble in terms of scalability of their applications.

Greg Hughes: And you don't need necessarily to have all those people as individual people but each of those roles is critical to doing things well, I think.

Kim Tripp: Yeah, exactly.

Greg Hughes: So I mean, I have a perspective of building enterprise software and leading teams that are leading millions of end-users based on SQL enterprise applications in financial services, right. So I guess the way, as you're speaking I'm thinking about this, I think about this in terms of product development and I know that in an IT organization for internal development, a lot of times it start off as, well, we'll just put this together and make it work and then we'll sort of tweak it as time goes on. You don't get that luxury or maybe it's not a luxury, maybe that's actually a mistake to try to think that way, maybe people should be thinking in terms of product development which is I'm going to build something, I'm going to test it, it's going to be well-thought out and well-architected ahead of time because I'm going to hand this off to somebody and they have to be able to rely on it.

Kim Tripp: And that's something that I always try to get people to do, it's spend more time in design and pseudo code or prototyping certain design techniques because not every design technique works for every implementation.

Greg Hughes: Sure.

Kim Tripp: The different usage patterns warrant in some cases different design strategies and as a result of not thinking about it, you know, putting it together as you say and just making it work, you put all these code, all these stored procedure, used functions, triggers, on top of a schema that hasn't been well thought out and the problem that you have is you can't change it now because as soon as you change the schema, all that code you've written and all those stored procedure, triggers, used functions, will probably need to be rewritten...

Richard Campbell: Right.

Greg Hughes: Right.



Kim Tripp: And your ultimate problem is now you're in a, well, we just need to make it work and we just need to make it better than it is now when it won't really ever fully scale because there are some things that have done so poorly, so early, and they were so fundamentally wrong that there's just almost nothing you can do except re-architect.

Greg Hughes: If you're going to build a house, you have to have that strong foundation and it has to be properly designed and tested and everything has to be good about it before you start building the sticks on top of that to build the walls.

Kim Tripp: Yeah, absolutely.

Greg Hughes: Otherwise, they will all fall down. There's a really bad analogy but probably a cliché analogy. Richard, actually you know a lot about that, don't you?

Richard Campbell: Thanks, thanks for that.

Kim Tripp: What cliché analogy?

Greg Hughes: Foundations and building new houses on top...

Richard Campbell: Building new houses, yes. Kim and I are competing for the longest renovation in history.

Kim Tripp: Hey man, we're taking a year off.

Richard Campbell: I maybe two months away from finishing but that's a whole other story. You know, getting back to Huckaby's email, one of the things he said that I think really made you jump was this idea that the DBAs, that the guys who own the database are an obstacle to building applications, that developers are just going around them anyway they can.

Kim Tripp: Yeah, I think that's really interesting that developers are either having to fight with DBAs or having to -- this I think was also another issue right along that same line, having to go to a DBA to ask them to write a stored procedure so that they'll put it into place because they are the ones that know everything how SQL works, and in my experience I really think of the DBA as the person that does security, back-up restore, maintenance, disaster recovery, high availability. I don't think of the DBA as the ones that are creating the schema, that are writing all that T-SQL. A good DBA, I think, should know some T-SQL because they'll need it in scripting and in automation, but I wouldn't necessarily say that a DBA could write some 18-table join because they wouldn't

necessarily know the schema even well enough to do that.

Greg Hughes: Well, you wouldn't necessarily want them to do that from a separation of duty standpoint.

Kim Tripp: Yes. You see that was my thing, I don't really want them. They've got enough to do with lack of restore disaster recovery.

Greg Hughes: Well, and you handed them all the keys of the kingdom that creates some risks, some security risks as well.

Richard Campbell: But I do think there needs to be somebody, you know, there needs to be the role, that is the bridge between the owner who shifts operation at the database and what the developers need to get the database to do.

Kim Tripp: Totally.

Richard Campbell: Whether that happens to also be the DBA or separate person or it's even a developer. You know, I don't know where that person sits in but I think that too many organizations don't know who that person is.

Kim Tripp: Exactly, that's really fundamentally the question, it's who is that person and who are they putting the training and the effort into for somebody to actually know this stuff, because if they don't ultimately they're going to have bigger problems later in terms of the application scaling.

Richard Campbell: Well, I think the other side of this is that the work around that the developers are doing and the other topic that we haven't even touched on yet, I mentioned Cloud but the other one is the whole ORM system...

Kim Tripp: Yeah.

Richard Campbell: These tools that essentially they build their objects and say persist that to the database, and it will build whatever it needs to on the backend and certainly with no thought to schema per se. Developers are being productive with that because when they get into this obstacle scenario, they get into this situation where essentially I can't get my work done. I go and use this tool and it spits out the database stuff to do what I need so now I can get my work done and when we're finished building the app, that's when we'll engage the database guys and go, oh yeah, and here's your database to take care of.

Kim Tripp: Yeah.



Richard Campbell: You know the one with no primary keys, yeah, that one.

Kim Tripp: Well, you know, speaking of no primary keys, I don't know really much at all about the entity framework, but one of the things that I found in kind of trolling around about whose job it was and part of what started me on this whole rant, if you want to call it that, was that somebody had mentioned, and I'm embarrassed I don't remember who this is, I have to find this on my blog, but somebody had mentioned that the entity framework, when it creates a primary key, the default choice is to actually create the primary key over all of the not nullable columns of the table.

Richard Campbell: Awesome.

Kim Tripp: To just grab all of the not nullable columns and make them, as a unit, the primary key which I don't know if it also then makes it what is the default, the clustering key, but if it does, just that stuff, if there is non-clustered indexes, will cause the entire table to get rebuilt and all of the non-clustered indexes to get rebuilt and it will create probably one of the worse keys that you could ever have for a table let alone the primary key, and it's the absolute worst default I have ever heard. I mean, I created a blog post that would be called Seriously, Are You Kidding Me?

Richard Campbell: And this is from a product that is owned by the SQL Server team.

Greg Hughes: Yeah.

Kim Tripp: I don't know who owns specifically the entity framework but it's definitely an interesting discussion in and of itself.

Richard Campbell: Right.

Greg Hughes: Well, someone should fix that.

Kim Tripp: David Yack who is another RD...

Richard Campbell: Yes.

Kim Tripp: He posted on the blog, and again I apologize, I think the gentleman's name is Ben -- oh, Paul came in and wrote a post a note that is totally incorrect, I'm not even going to tell you what it said. I thought it was helping me, giving me the name, but no, of course not. But David Yack had posted on the gentleman's blog and said that you can deselect the default making it primary key which is good and I said yes, that's great, but how many people are really going to know how bad this really is. You know what I mean? If you're using this to

circumvent having to know the database, then you would almost expect, "almost" being the keyword here, that whatever this product is that's leveraging SQL would help you not completely soothe you in the foot, hurt you, you know what I mean?

Richard Campbell: Absolutely. I think it's exactly the battle we're up against. So that means there's a lot of misunderstanding here, we still have to surface all of it.

Kim Tripp: Well, I think there's misunderstanding in lots of products and I'm starting to blog, I'm blogging a lot more this year. If anybody has noticed, thank you, and I'm starting to put a lot more of kind of things I'm finding and issues I'm running into and why these things are bad, you know, kind of pros and cons so stay tune with that, I'm actually going to post a couple today, the blog post that I lost yesterday on issues I have found when I was surfing and recommendations that are really just not great recommendations.

Richard Campbell: All right, Kim, I think we're just about out of time. Any final...

Kim Tripp: Oh no!

Greg Hughes: Time flies, huh.

Richard Campbell: We could just keep going, you know.

Kim Tripp: This is another discussion I think we should have. I think this is a huge one and maybe we could even just have a discussion sometime about just top 10 things that database developers shouldn't do and...

Richard Campbell: Absolutely.

Greg Hughes: Oh, absolutely.

Kim Tripp: Things that we just keep finding as issues and these are things I'm blogging about as well but it could want a whole show.

Greg Hughes: Absolutely.

Richard Campbell: Well, I think we've got some developer topics here to drag over to .NET Rocks! as well.

Kim Tripp: Oh, definitely.

Greg Hughes: Great.

Richard Campbell: Kim, thanks so much for coming on the show again.



Kim Tripp on the Roles of Developers and DBAs with the Database!
April 8, 2009

Kim Tripp: Well, thank you guys.

Greg Hughes: Thanks Kim.

Richard Campbell: And we'll talk to you next week on RunAs Radio.