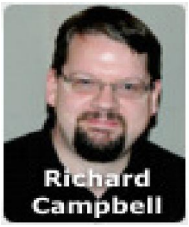
RunAs Radio

http://www.runasradio.com



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.

Text Transcript of Show #247
(Transcription services provided by PWOP Productions)

Jeremiah Peschka Talks to DBAs about ORMs!
January 18, 2012

[Music]

Brandon Wehn: 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 247 with guest Jeremiah Peschka, recorded Monday, January 16, 2012. RunAs Radio is produced each week by PWOP Productions, providing professional media and podcasting services online at www.pwop.com. You can follow the boys on Twitter at twitter.com/runasradio.

Richard Campbell: Thank you, Brandon. This is Richard Campbell. You're listening to RunAs Radio and we're having a crazy day. I have nothing to complain about actually. There's real winter in Vancouver. That's rare because we don't usually get snow. I thought we weren't going to get any this year, but we have it now and so I'm happily cozied up at home with the microphone in front of me and I've got Jeremiah Peschka on the line who is not at home but we'll get into that in a second. Let me introduce him for a second. Jeremiah Peschka helps developers, DBAs and engineers build faster, robust, and scalable solutions. Microsoft SQL Server is frequently a part of that solution, but other databases are also. He has worked as a database and emerging technology expert at Quest Software researching new trends and technologies around data storage. Before that Jeremiah worked across many industries as a system administrator, developer, and DBA, and has been involved in all aspects of application development and deployment. You're part of the Brent Ozar group these days, aren't you, Jeremiah?

Jeremiah Peschka: Yeah, yeah. I started Brent Ozar PLF with Brent, Kendra Little, and Tim Ford.

Richard Campbell: Nice. A fun bunch of folks. I've got Brent coming on the show in a couple of weeks so it's always fun to talk to you guys. You have a good time playing with SQL Server, that's for sure. But you're not a SQL Server exclusive kind of guy. What's your other favorite database?

Jeremiah Peschka: My other favorite database is actually Postgres and RIA. Postgres is just fantastic, and then RIA gives you a lot of flexibility and we talked a lot about that previously.

Richard Campbell: Yeah. I actually am a Postgres fan myself. I had a project and went into it somewhat tentatively and then came to really admire the little database that could.

Jeremiah Peschka: That's a fantastic way to describe it. They do such a great job of making a tool. It works and it works phenomenally and their default settings are really paranoid about data and you could tune it to be ridiculously fast and it's so well-documented. It reminds me about a SQL Server, you could just give a source code.

Richard Campbell: Yeah. I mean the big difference is, and I have found this true of every other database, SQL Server is the only database that I found that no matter how I craft a query to return a given set of results, they all get the same query plan. In Postgres and other databases too, how you craft the query matters. You have to look at the plan and tweak and tune, and you just don't do that in SQL Server anymore. I think we have forgotten that you can write -- I've seen some really nasty queries written that actually run great data plans.

Jeremiah Peschka: Yeah.

Richard Campbell: And then you could write a beautifully crafted query but because your data structures are a mess, the query plan is a disaster.

Jeremiah Peschka: Absolutely. I even took one of Microsoft's disaster queries to generate awful formulated subquery.

Richard Campbell: Yes.

Jeremiah Peschka: And I run it on SQL Server 2012 and it produced the correct execution plan and removed the correlated subquery from it.

Richard Campbell: You just look. You didn't mean to do that. Let me get the results you actually want without sucking and it just got to work.

Jeremiah Peschka: It has and this is the query from books online. It says this query will hurt your server. But no, it's beautiful now. Thanks query engine team.

Richard Campbell: Yeah, not anymore. Yeah, if you're not deeply immersed in databases, you don't necessarily know what we're talking about here because this has been the main job I think of DBAs for a long time other than the basic setup and turn down account separable and so forth tweaking and tuning queries to get good results. For the past few years SQL Server just took that job away. It writes better query plans you could ever hope to write.

Jeremiah Peschka: Yeah. Largely and for the most part you can install this from the same things with it, but a lot just comes down to will you rewrite it. Well, just break the stuff into steps instead of one sporty a thousand-line report.

Richard Campbell: Right. Yeah. Well, I always resisted the mother of all query approach to things.

Jeremiah Peschka: It's tough to resist it, but good for you for actually resisting that approach.

Richard Campbell: Just recognizing that that's fun for me and not fun for anybody else.

Jeremiah Peschka: Exactly.

Richard Campbell: That's the whole thing.

Jeremiah Peschka: Yeah, you've got to make sure you database is maintainable.

Richard Campbell: I wanted to talk to you about ORMs. Although really I mean ORM is an Object Relational Model. Right?

Jeremiah Peschka: So it does transfer a couple of things unfortunately.

Richard Campbell: Of course.

Jeremiah Peschka: There's also Object Relational Mapper which is the other sort of cared for which just allow things like NHibernate and LLBL Gen for Entity Framework too. They take your object schema and they map at your relational schema and they do some convoluted magic in between.

Richard Campbell: Now these are developer tools. Why do I as a DBA care about these things at all?

Jeremiah Peschka: So DBA care about this. This is actually how I got into database. The developer discovered ORMs and discovered what cool things databases could do. As a DBA we want to care about it because developers are going to be using these tools to ramp up their development time. Contact switching between languages and the expensive mentor rates,, they would just rather say person.five rather than actually, okay, well, I'm going to open and manage the Studio. How do I create the stored procedure again? Oh, yeah. You know, write the SQL statement. They just don't do this in a really easy way and it's going to automatically do a lot of work for them.

Richard Campbell: Well, and I think you've hit on the key point here which is generally speaking ORMs and stored procedures don't get along.

Jeremiah Peschka: Right.

Richard Campbell: The statement I've seen sitting in the DBA's seat is that the developer said "I don't want anymore stored procedures. I'm just going to use this ORM. It will write the SQL or me. Everything is going to be beautiful. Don't you worry, Mr. DBA man."

Jeremiah Peschka: That's usually the case, yeah, and that was my approach to them too when I first found them.

Richard Campbell: Yeah. It seems like a big lie.

Jeremiah Peschka: It is a big lie. But the nice thing is for most application they are really just pure OLTP. It's just give me a row. I'm going to write a row. It's just I'm going to delete this, I'm going to read this. It's _____ 07:08 kind of thing.

Richard Campbell: Right.

Jeremiah Peschka: So it doesn't really need the tuning that DBAs can provide, big reporting queries or multi-page, page out paging queries.

Richard Campbell: Yeah, there's not a lot of tweaking and tuning to a insert query.

Jeremiah Peschka: No, there really isn't. Occasionally you want to have better indexes of your foreign keys, but that's about it.

Richard Campbell: Right. But even then that's not something that necessarily needs stored procedure or anything like that. I see stored procedure, their value and more the abstraction layer that I can modify the data structure without having to change the code and as sort of a boundary layer for validation.

Jeremiah Peschka: Absolutely. Yeah. I always like to have my validation. As both the database guy and as one who write code, I like my validation to be in both the app and the database.

Richard Campbell: Right.

Jeremiah Peschka: And my app validates all rules for that data including the database rules and application rules and then any access into the database either in the stored procedure or however the data is getting in there, I validate all the universal data rules there so all my data is always correct.

Richard Campbell: Right. I mean if you're going to take away my stored procedure as a validation path, I guess I can stick with the declarative validations. Stick constraints on all my columns and things like that to basically force the failure of the query if you don't obey the rules.

Jeremiah Peschka: Yeah, as delay. The nice thing with ORM too is developers say, well, I don't have to write new stored procedures. I don't have to do all this extra work. At the same time, the back that we get is ORMs make it really easy to develop code quickly and then down the road once that schema is in place, once we know what the application is going to look like, we know what the data is doing in the app as well as in the data layer, they only can come back where we need to have validation and we did it as stored procedure and change the ORM to actually use stored procedure instead of just using plain old SQL.

Richard Campbell: And I think you're just implying on the crazy part. I think the thing that offends DBAs or database centric folks in general which is you're going to write code first, you're not even going to design a database. There will be no cocktail knocking here.

Jeremiah Peschka: I think a lot of developers really do like to converge and I like to think of it before you're going to write code I really do get to write for it. I think like I figure out all my nouns and my verbs, and I try to think of how they're all going to work together. But yes, ORM is typically used in a code-first environment which does drive the people absolutely...

Richard Campbell: Well, I mean there are two issues here. One is how often do we actually get to greenfield and up to a point where there's not a database already hanging around.

Jeremiah Peschka: Yeah. That's pretty rare like I said.

Richard Campbell: Yeah. And the second issue of course is I don't trust your fancy little tool to make databases that make any sense at al.

Jeremiah Peschka: There is that. One of the things that I think this a really good place for DBAs is to use both the domain level knowledge that they have and as the developers are building the application work with the developers to figure out how the data model is going to evolve. Even as the application model looks one way and the correct way _____ 10:37 should look at it in two different ways, it's really easy with an ORM to build that thing as to be a complete set of structures.

Richard Campbell: Right. I guess I can also imagine that you're going to end up with a hybrid. The customer tables and product tables are already going to exist, but there's going to be a few new tables that this new app is going to need and that can be generated by the ORM.

Jeremiah Peschka: Yeah, exactly. The ORM doesn't have to be responsible for all generations. Likewise, the DBA doesn't have to be responsible for all that stuff. DBAs have a lot that they have to do. To keep things running, they have to provide tuning. Sometimes they're called on for the data architecture, data modeling. There have a lot of stuff to do. Let's face it, a lot of these decisions more have come out of validating, yes, this is the same choice, or no, you shouldn't be storing Social Security Number as a _____ 11:20.

Richard Campbell: Is there any chance that an ORM is actually going to make that choice or that somehow the developer said that?

Jeremiah Peschka: I have never actually seen anyone do it. I have seen Social Security Numbers stored as integers before. So why are we adding them together? Is that what we have to do? We'll get married. Is that how we'll show them these numbers or get shipped? They're like, "Well, no." Like at the Char(9), yeah. This would be for real.

Richard Campbell: Yeah. It doesn't change. It's not mathematical. Just because it has numbers in it doesn't mean it's a number.

Jeremiah Peschka: Right. Yeah. That's where DBAs can get involved in this because, yeah, developers, yeah, they're picking on them, developers don't always make all the decisions. They usually don't.

Richard Campbell: Well, at least not intentionally anyway.

Jeremiah Peschka: Yeah, not intentionally.

Richard Campbell: I've got to imagine they make awful choices generally out of ignorance, not malice. That they just don't understand what the consequences of that decision is.

Jeremiah Peschka: That's certainly I think why most bad choices get made and that's where your DBA can come in and apply a lot of that knowledge. Well, I noticed SQL Server does it this way and I know that I'm going to take advantage of these indexing techniques or those querying techniques if you do it in a different way so let's use this data type instead of this first one.

Richard Campbell: Right.

Now I'm just trying to figure out what the workflow looks like for this. So the application developer is working with his ORM and this is actually generating into some kind of test database, a set of tables, and then as a DBA I can pick those tables up and look at them and then have a discussion with the developer?

Jeremiah Peschka: Yeah, that's definitely something you can do. Depending on the tools you use, if we're talking the .NET world obviously since we may have SQL Server it's pretty easy to even, you know, when you generate that schema from your ORM tool, you can even generate it into Visual Studio and then use the database developer tools to share that schema with the DBA. Usually they don't have to

look at your set of database. They could just look at your schema...

Richard Campbell: Right. And actually read through the schema which gets checked-in as part of the build anyway.

Jeremiah Peschka: Right. Exactly.

Richard Campbell: And if you haven't used that tool, it's awfully nice.

Jeremiah Peschka: Yeah. Actually this is sort of a plug for a great blog post. A guy named Jamie Thomson on SQL Blog wrote basically a mini book on working with Visual Studio 2010 database.

Richard Campbell: Wow.

Jeremiah Peschka: He put it out there for free for everyone to look at. There are a lot of tricks to it. I found that once you figure it out, they make life easy but initially just find the tricks. Or if you're angry enough to go search for blog posts, they're just irritating things about it.

Richard Campbell: So I'm thinking further along the workflow. So this is sort of initial iteration where they're generating subscripts that I can now read as a DBA and take a look at and talk to them about data types and maybe even -- can I alter those scripts in any way? Like how do I apply my changes and not have them wrecked by another iteration of the ORM?

Jeremiah Peschka: I think a point where it's a process, you know, like here in process where you do have to work with them and say, well, I want to make this change. I want you guys to take a look at what I'm doing. But here's what I've done. You don't need to change your ORM to reflect what I've got in here. A lot of it depends on how you're regenerating things.

Richard Campbell: Right.

Jeremiah Peschka: When you add a new column, if you don't have an existing database then they don't reach a whole database. Or if you are not aware of the changes in the database, you could get some odd things happening. So the server does comes down to ORM specific things where you may have to change a mapping file in NHibernate, Entity Framework so if you go into that TDML or change some XML most likely your LLBL Gen Pro you just point it back at your database and say, hey, I change some columns and make sure this still works and it will figure out how...

Richard Campbell: Take this back up.

Jeremiah Peschka: Yeah.

Richard Campbell: Yeah. I have a lot of respect for LLBL Gen. They've been doing this for a long time and they have I think a cleaner understanding of the DBA role and the dev role to let both parties play and have some of their knowledge of pure and essential location.

Jeremiah Peschka: You know, oddly enough I'm very spoiled because that's actually the first ORM I worked with and it really is the Cadillac of ORMs in many ways. They have features so a lot of people are now just adding it and it really does, like you said it really respects the role of you may have a data model or/and an architect and a DBA and developers. It allows all those people to play and look at the database in ways that make sense to them.

Richard Campbell: Make contributions that way. I mean does it make sense for the DBA to actually if I'm looking at I need to apply some constraints here, just come to the developer and say in your ORM indicate these constraints and they should show up in the database? I guess that depends on the ORM, right?

Jeremiah Peschka: Yeah. A lot of it does depend on the tools. One of the difficult things I think takes a lot of, one of stinking place that takes a lot of DBAs, there's a very little documentation out there for ORMs that's written from the perspective of, you know, all of this from standpoint of I'm a DBA, I have to have this tool in my environment.

Richard Campbell: Right.

Jeremiah Peschka: A lot of it is just being very, hey, come with me. And they're like, gosh, I just made NHibernate yesterday.

Richard Campbell: Yeah. And it's all automatically bad.

Jeremiah Peschka: Right. And it's not all that great as a tool. It's just kind of a crowbar. You can use it to pry things apart or you can break in the house with it.

Richard Campbell: Yeah, and it could be a little brute force at times. Can't it?

Jeremiah Peschka: Right. And a lot of it does come down to do things from a DBA's perspective talking to developers and making sure there's a lot of understanding on what's going on in the process and it does mean looking working them directly as distasteful as that may sound...

Richard Campbell: I've got to talk to the developer. Are you crazy?

Jeremiah Peschka: Yeah, I know. It's kind of like understanding at least a little bit of some of the things they try to do, some of the cost of what they're doing because there are some things that you're never going to notice and know or might do like the problem of yet the N + 1 select problem which is where you

might have to be doing something called lazy loading which is where your ORM just loads stuff when it actually gets called on the code. So if you wanted to display a person's order, you call back that person and then you look through their orders and then it's going to go and find all the orders after it finds the person. So you may get a lot of extra slots as it's being run and unless you're actually cautiously running SQL Server, go file it. Looking at what's happening in the ORM, it can be really tough to detect things like that.

Richard Campbell: Right, right. And I think what's scary here is at that point the developer is not necessarily sure when queries are running. Like I said as a DBA if I'm looking for a real slow running query, I could find it and then I can't find it. I can talk to a developer. He has no idea where that happened, who wrote that query because the ORM wrote it.

Jeremiah Peschka: Right, yeah. And as for clients before where you had to literally grab that text out of SQL Server profiler and search in the codebase and then of course you don't really know what's happening but then I try to reverse engineer that query as to where it might come from in the codebase and figure out which objects are involve and then go and track it down, that can be really tricky to do.

Richard Campbell: Yeah, no kidding and an interesting problem. The idea that you can't search necessarily in the query because the query is not actually in there, it's part of an ORM expression of retrieving data.

Jeremiah Peschka: Right. And one of the nice things with the people who sort of worked on in NHibernate, Ayende Rahein, he did NHibernate profiler which is a fantastic tool for developers to use and it's basically like a SQL Server profiler but it runs on your workstation and you just sort of add an extra reference in when you're in debug mode and it supplies all these profilers like so you can track down which queries are being generated and will write that back up to which files it's coming from. So from a developer's standpoint, you could run through a bunch of tests with any stuff enabled and then isolate the query that's actually being generated and causing the problem.

Richard Campbell: For sure. And I'll happily plug Ayende. I'm a huge fan. He's also made a profiler for Entity Framework as well.

Jeremiah Peschka: Yeah. I remember actually when he released that. Entity Framework perhaps sets me a little bit as far as ORMs go just because it's free. It's from Microsoft so a lot of people are using it.

Richard Campbell: Yes.

Jeremiah Peschka: Sort of have you ever got bias for buying big blue. Unfortunately Entity Framework is still really, really young and it does a lot of things that other people figure out ways on how long ago.

Richard Campbell: Right. So you're really embracing, yes, you got it for free but it was worth what you paid.

Jeremiah Peschka: In some instances, yes.

Richard Campbell: Yeah.

Jeremiah Peschka: They stick a lot of issues with it. There was an issue where it would generate parameters of random length so if I search for _____ 20:36, it would create bar chart for you and then if I search for day to create a bar chart for on top of that which is obvious to SQL Server people but which is not just obvious to anyone else, that's going to generate two execution plans so it's lots more memory being used.

Richard Campbell: Well, and you're just going to make a mess of caching and like you're just undermining SQL's ability to make things fast.

Jeremiah Peschka: Exactly. And yet _____ 20:57 has fixed that but you're still dealing with Ruby cycles out of a big company you rate.

Richard Campbell: Sure.

Jeremiah Peschka: It's not like open source projects and you're like, oh, oh, I can fix this right now and release a X.X.1.

Richard Campbell: No. It will come sometime later. Now is there a point in the tuning side? Once I get some iteration going and we start to do some generation, is there a point where I can start injecting stored procedures where I could see performance opportunities or safety involved?

Jeremiah Peschka: I would definitely suggest that to everyone. That once you start working with it, ORMs are great for rapid development.

Richard Campbell: Right.

Jeremiah Peschka: They tend to sort of generate, ah, I think this is an acceptable query type of querying.

Richard Campbell: Yes.

Jeremiah Peschka: It's like I have worked with SQL for four months, it's just getting a lot better. At that point, most ORMs even Entity Framework supports the ability to replace the selector, insert, update, and delete with the stored procedure so that way you can go in and you can replace that. You can provide

whatever tuning you want as a DBA in the stored procedure and then say okay guys, "Well, you've got this working. Your app works. Can we please make this change in your mapping to use this stored procedure for this object?" You wire that up and it works beautifully.

Richard Campbell: Yeah. It's an interesting point, the idea of let's have this rapid cycle of quick development where I don't have to constantly be pounding on stored procedure builders and so forth until the very end, until we're actually okay, we're ready for a build. Now we sort of move into that performance tuning stage or we care about those things, we got the feature set in place, now we talk about tuning and tweaking and can get more people involve to do higher performance or less resource intensive job.

Jeremiah Peschka: Exactly. I don't really need the tier. They look at my primary key, that's all is going to be as fast as its going to get.

Richard Campbell: Right, right.

Jeremiah Peschka: As one got selected. The nice thing about that stuff, you don't want to cache plans for that. You don't want to keep that around forever. It's just annoying to have to find out. But yeah, for other things when you have to do a search you may want to do it. You search in a certain way or there are some times when you have to conform to complex data structures in the business, one on one side, you frequently, oh, my God, you may have a product table, and the product table doesn't actually have a price in it because you have a price history table.

Richard Campbell: Of course.

Jeremiah Peschka: And the structure may exist or you may have price structures that vary from day to day and hour to hour if you have to look at the price of appropriately. Running that sort of query in ORM isn't always pleasant to write for a developer, but just go to the DBA and say, "Hey, I have this painful spot. Let's get a stored procedure for this one thing."

Richard Campbell: Now you keep implying a comfortable conversation back and forth between the developer and the DBA.

Jeremiah Peschka: I do keep implying that. I think it's really, really important for that relationship to be a go and not the adversarial.

Richard Campbell: But with your last statement there, you've implied the bi-directionality of it. There are going to be points where the ORM is going to be ugly working with a database and you can ask the DBA to solve that problem for you from a developer perspective, and there are going to be points where the ORM is making mistakes essentially that the DBA can see the developer probably can't see that he can push back to the developer to fix. This is a two-way street.

Jeremiah Peschka: Yeah, absolutely. This is the stuff I need to get to a two-way street and I think it's a good point to get a lot of that communication going between two teams who normally don't talk a lot.

Richard Campbell: Well, one of the reasons they've broken down is often the productivity of the developer working with the database is impaired because the DBA is awfully busy. There are an awful lot of queries to write.

Jeremiah Peschka: Yeah.

Richard Campbell: In some ways it seems like the way you presented this, here, I'm going to take all the queries off the table that we really don't need you to write so we could just focus on the few that I did need help with.

Jeremiah Peschka: Yeah, absolutely. I think that one of the most important things too with ORM is for people to communicate with DBAs, and for DBAs to understand developers aren't trying to get rid of DBAs. They're not trying to replace the DBAs for I'm clearly writing or clearly tuning. They're trying to get the garbage query away from the DBA and away from the developer. Nobody wants to write selected tables to pull all the columns back from a customer table. Nobody wants to write that. It's boring.

Richard Campbell: Yeah. It's nothing or the insert statement. I mean there are a lot of very simple queries here that the machine generates them. The funny part is I can imagine the machine is going to make fewer mistakes on those simple queries.

Jeremiah Peschka: Yeah, exactly. I automate everything I can that's boring because boring things, boilerplate things are where I make mistakes.

Richard Campbell: Right. Because I'm not paying full attention.

Jeremiah Peschka: Yeah, exactly. I don't pay attention, I get bored, I feel like going home and play MindTrap or _____ 25:42 or whatever, or all three at once on three separate computers.

Richard Campbell: Yes.

Jeremiah Peschka: I'm not thinking about where to put the comma.

Richard Campbell: So these profiling tools, these still are developer tools. It's not something that the DBA necessarily wants in their toolkit.

Jeremiah Peschka: Yeah, I wouldn't think so because you still have to be either running unit tests or writing some sort of test to generate, that's going to generate the activity, that's going to make that query have to sort of fire Visual Studio, open up your .NET project, run it, and run those profilers. Just because it's a profiler, it does look a lot like SQL Server profiler, it's still a developer tool. One of the things that a developer can do for DBA is to make life better. In your connection string, you can set an application name.

Richard Campbell: Right.

Jeremiah Peschka: And so set your application name and that really makes it easy for the DBA to look and then tune your queries.

Richard Campbell: But I also am coming from a DBAs perspective. I'm going to ask that developer to have that profiler available so that if I do dig a query out that I'm pretty unhappy with, he has a better chance of finding it with his profiler.

Jeremiah Peschka: Exactly. Yeah. It's a lot of communication that has to happen in here and if you actually start digging here, yeah, developers should be aware of the other tools that they can use to help them find the queries.

Richard Campbell: Sure. So Jeremiah, are there projects we should not use ORMs on?

Jeremiah Peschka: Probably yes. If you're doing complex reporting, so that's a definite no, don't go for ORM. ORMs are great when you're working with objects and you're dealing with all the wonderful things that Object-Oriented programming gives you, all that behavioral modeling.

Richard Campbell: Right.

Jeremiah Peschka: Anytime you're doing that, yes, ORM. Anytime you're doing a more along the lines of I'm doing massive data processing, I'm delivering the ports to users, I'm doing a number crunching in the database, we just need to have handcrafted higher performance SQL. You don't want a tool generator.

Richard Campbell: When you think about working with objects deeply, it's really the OLTP kind of stuff, entering orders, that interactive looking at data and modifying data quickly. Analytics is, well, I tend to shove that over into the OLAP space anyway but it's along those lines.

Jeremiah Peschka: Not everyone has OLAP expertise or the budget for OLAP consultants.

Richard Campbell: Sure.

Jeremiah Peschka: So as long as make do with relational warehousing but, yeah, anytime you're doing anything that's analytical, don't have your tools do that.

Richard Campbell: Yeah. So they definitely feel like there's a class of app. I mean I guess it's developer intensive apps where there's a lot of creation and recognizing also that I think developers are iterating faster these days. They're trying to ship stuff every couple of weeks.

Jeremiah Peschka: Yeah. Developers are definitely iterating very quickly. It can ship code. When that speed of feature development is the most important thing, that's we're ready to go. But a lot of it boils down to what's your feature. Jeff Atwood is fair when he said performance is a feature.

Richard Campbell: Yeah.

Jeremiah Peschka: You feel it. Stop looking at the ORM and start looking at how you can optimize that SQL. Most of the time though the feature is a new widget or a fancy thing.

Richard Campbell: Yeah. This is the difference between performance in execution and performance in delivering new code. I'm a guy who has seen a lot of Web performance tuning over the years and I'm a big believer in you've got to ship that before you possibly know what you need to tune.

Jeremiah Peschka: Absolutely.

Richard Campbell: It's only when it's exercised that you have a sense of the truth. If you spend all your time trying to tune in advance, you tune stuff that didn't need to be tuned and you waste a lot of time.

Jeremiah Peschka: Yeah. I definitely would say ship the code, figure out where it's slow. Shipping a new mapping can sometimes be as, you know, when you change from using dynamic code to using a stored procedure, sometimes it could be a simple uploading an XML file and you've changed it.

Richard Campbell: Yeah, interesting point that you're now actually breaking apart the compiled code and the mapping code with these mapping tools.

Jeremiah Peschka: Yeah. It depends on which one but you definitely can break that apart and maybe potentially update them independently of each other or even still you may have a mapping DLL and your behavior DLL from the outside of it so you could just upload a new DLL where you start the service and everything works.

Richard Campbell: Yeah. And I could see that because these are some things I definitely have done with stored procedures, create a sense of abstraction

between the application and database so that when we need to do decompose the sales database and the monthly tables, the developer don't have to change the code at all. We just made that invisible to him through stored procedures.

Jeremiah Peschka: Absolutely. That's really why I love stored procedures and reviews. I've just created a new service layer for people to call and it doesn't matter if it's an XML table or across 50 tables or one big denormalized table, nobody has to care.

Richard Campbell: You don't need to know. But the fact that I could probably still do a trick like that using an ORM just replacing out that mapping file, that from a database perspective is pretty compelling to me. I don't know if my developer will initially understand them. I have to talk to him about it, but it seems like a possibility at least.

Jeremiah Peschka: It's a distinct possibility. Again you hit on the big thing I like about ORM. It gets people talking which is never a bad thing in business.

Richard Campbell: Yeah, I tend to agree.

Jeremiah Peschka: And it gets people educated about where the strengths are so in that way you don't have to worry. Developers can say, aha, in this situation I need to leverage the database but in this situation I know I can use application code because it's going to be faster.

Richard Campbell: Right. Yeah, and if you eat of those conversations, I think everybody ends up respecting each other more and you get more work done in less time.

Jeremiah Peschka: Absolutely.

Richard Campbell: And never underestimate the value of pizza as a conversational lubricant.

Jeremiah Peschka: Exactly.

Richard Campbell: There are very few problems that cannot be solved with lunch.

Jeremiah Peschka: That is the truth I'd say.

Richard Campbell: Yeah, that's where I go. Hey Jeremiah, great to talk to you.

Jeremiah Peschka: Great to talk to you as well. Thank you much for your time.

Richard Campbell: And we'll point folks to your great blog post on ORMs for the DBA as well so they can go grab that. Thanks for coming on RunAs Radio.

Jeremiah Peschka: Thank you for having me. Yeah, it was a great time.

Richard Campbell: And I'll talk top next week on RunAs Radio.