

Applied Rapid Development Techniques for Database Engineers

Whitepaper based on original
presentation by Dominic Delmolino,
commissioned by Red Gate Software

Applied Rapid Development Techniques for Database Engineers

In this article, Dominic Delmolino describes his experience in implementing a rapid refactoring and professional schema management process using standard software development techniques combined with built-in Oracle capabilities.

The presentation in a lot of ways comes from my evolution of understanding what a DBA is and what they do and I think it's because the DBA is in a difficult position in many organizations. First off, the title is suggestive of their role and I don't particularly like that, I don't particularly like the fact that they are described as an administrator. I think a DBA often takes a more active role in how data is used and shaped within an organization. So unfortunately because they wear a lot of hats they get shot at I think from a lot of different directions. They have an operations focus, most DBAs are assigned to a production operations organization; it is relatively rare in my experience to have a DBA outside of operations. So they're hit with the following things from the business and government owners: is the database backed up? Is it replicating to our DR site? Have you been maintaining space so we don't run out of space in our tables? We need to make another copy of it for some analysis purposes or testing so get on making a clone copy for me and make sure it's fast, so we're going to yell at you whenever there's something slow, it's to do with the database and you need to deal with it.

They have another role, and this might be the data governance type role where they are attached to or associated with the data architecture group in the companies and they are then required to help participate in producing a data model, ensuring that the database objects follow the naming standards, so that the data is discoverable within an organization. They may be responsible for maintaining a master data catalog so that you can understand, if we're representing people or customers in our organization what is the table name? What systems are those located in? And then always a never-ending battle against data quality, how do I ensure that what's going in the fields is what's supposed to be in the fields?

So it is not necessarily a surprise sometimes that the stuff on the left here, which is traditional database development type activities, are often get short shrift from the overworked and overwhelmed DBA:



A lot of times this is their last priority, "I'm here to support development, but I'm really getting hit on the stuff on the right, I'm measured on the stuff on the right and at the bottom, so I may not be the best person always to handle the stuff on the left."


So I often tell people I think one challenge to start off with that is when you have a DBA maybe you could consider splitting their functionality or their responsibilities among three different kinds of people. You have the administrator or the operator and often you want this to be the most conservative person in your organization, you want them to make sure the database is secure, reliable, running, it's up, it's backed up, but they're not necessarily the kind of person you want taking risks and figuring out new and interesting ways to store data and manipulate data.

You have the governor or the architect and this is the person who helps communicate out what the data assets are within the company. Who has access to the data; where can you find it; does it follow standards, so we can interoperate and interchange and often this is more of an architect and less of the hands-on type person.

Finally the area that I think we'd like to really focus on today is the database developer or what I call a database engineer.

Maybe it's really three people:

Developer / Engineer Governor / Architect Administrator / Operator

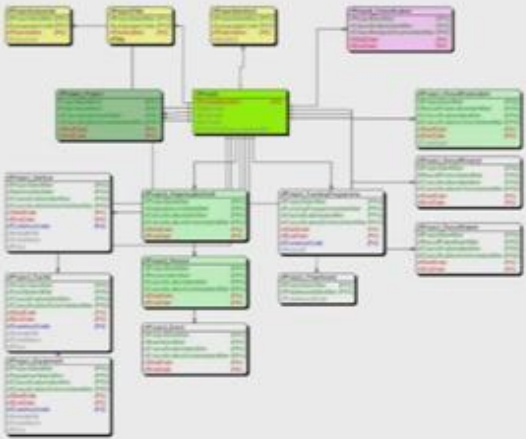


If you have a DBA assigned to your development project, how can you get what you need?

Or, how can you turn your Database **Administrator** into a Database **Engineer**?

I think when starting a development project it's really clear that you want the stuff on the left. You really would like support from someone who can ensure that you're getting database resources allocated, developed and provisioned as fast as possible so you can get on with the application development.

What I'd like to talk a little bit about is if you have a DBA assigned to your development team or your development project, how can you encourage them, how can you facilitate their ability to do that? Or what I like to say is how do I turn that DBA into a DBE, or turn a database administrator into a database engineer?



You're doing Agile development, so you don't have time to wait for a finalized enterprise data model.

So, how can you get started?

This goes along with the premise that many of you are probably starting down the path of Agile development so the traditional DBA type role where you'd sit down and wait for an enterprise data model, that's really not going to happen anymore, you just don't have time for that. You don't have time to wait around for a whole enterprise data model to get finished before you can start producing working systems and working code. So my challenge to DBAs is: I know you have to do that, but how do you get started?

You've got a brand new database that you've given to a development team; well a real good question is who creates the necessary database objects? Do your developers go in and do it? Do you submit a request to a DBA group that goes and does it? Do I have to produce a model? Do I have to have a plan before I do it? So a lot of people start there, they say "Well what's your data model and what kind of modeling tool are you using?"

When you have a brand-new database, who should create the necessary application support objects?

How should they do it?

1. **Model**

What kinds of modeling tool should they use?

```
ActiveRecord::Schema.define do
  create_table :employees do |t|
    t.column :name, :string, :null => false
    t.column :birthdate, :date
    t.column :email, :string
  end
  add_index :employees, :name, :unique
end
```

What I've seen in many cases, some of you may recognize this, most DBAs probably don't, but what I'm showing you on the right is a data model. This is an active record pattern from the Ruby on Rails echo system and this is an ability for a Ruby programmer to essentially define a table. They define an employees table, they allocate columns to it, they add indexes to it, they have a relatively rich language in the active record pattern in which they have an ability to actually go and define database objects and I think I found on many smaller teams that this is what is happening. They're going off and they're doing this in a code method that they're more comfortable with.

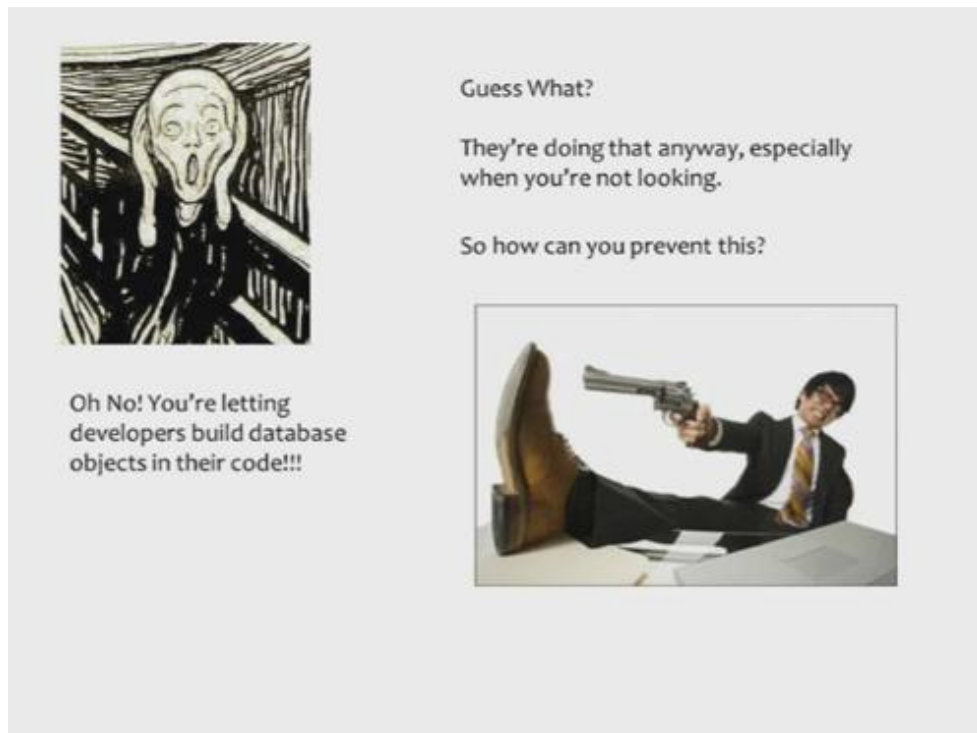
So, I know that if you're a DBA you're probably looking like the following picture here and wondering "Oh my gosh, we really can't let that happen, we can't let developers create tables like that, they don't know what they're doing, they're going to make mistakes, they don't know how to do this, that or the other thing."



Oh No! You're letting developers build database objects in their code!!!



The challenge though is they're probably going to find a way to do that anyway, especially when they've been given access to a system and you haven't really had time to check in on them. So really, what I'd like to say is let's assume that is happening and then figure out a way to prevent them from shooting themselves in the foot.



Guess What?

They're doing that anyway, especially when you're not looking.

So how can you prevent this?

Oh No! You're letting developers build database objects in their code!!!

Because quite frankly the knowledge you bring as a DBA or a database engineer is how can I prevent problems from occurring? How do I ensure that someone, when they do define database objects, they at least do it in a way that is going to be useful and has good performance and won't have a problem when it starts to need to scale?

So, one of the things I would love to have is a social development database. This is probably because I fell asleep at the computer and I had a dream that I was logged into my database, but I also had my Facebook page open on screen and I saw these kind of status messages go by:

First things first – turn on the social database:



Dominic Delmolino created the **scott.emp** table.



Chris Li added the **ename** **varchar2(10)** field to the **scott.emp** table.



Kevin Patch created the **scott.hireemployee** stored procedure.



Brian Taylor likes **Dominic Delmolino's scott.emp** table

I actually don't know of any tool that really does this, but in thinking through it, I really like the idea that as I'm logged into a database and I'm allowing people to create objects I'm going to really speed up development and I'm letting lots of different people create the database objects they need to in support of their application modules or functionality. I'd really like to see it as it is happening and this isn't necessarily an odd concept, many version control systems, and especially the distributed ones, will show what's going on in terms of people checking in code or doing their work.

On the bottom right here I've got a screenshot of GitHub for Mac and you can see here, this is the history of check-ins for a particular project. It is the master branch in this case and you can see everything that someone checks in, who did it, what their comment was when they did it, what time they did it, what the hash code is at that change.

First things first – turn on the social database:



Dominic Delmolino created the **scott.emp** table.



Chris Li added the **ename** **varchar2(10)** field to the **scott.emp** table.



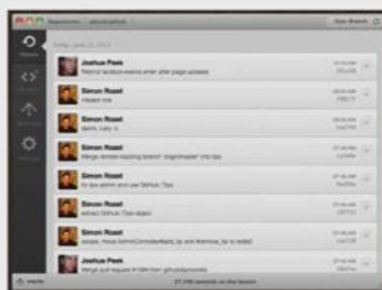
Kevin Patch created the **scott.hireemployee** stored procedure.



Brian Taylor likes **Dominic Delmolino's scott.emp** table

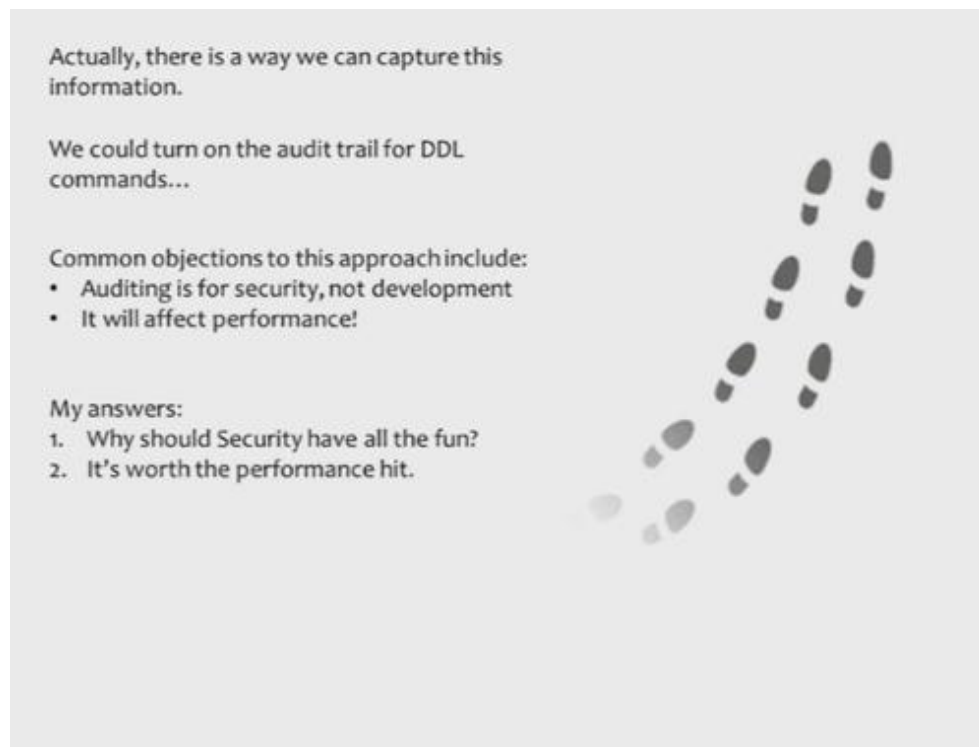
I don't know of any tool that actually does this.

But some version control systems provide this kind of view of what is being changed in a code base:



So the idea that a database system is similar to a code project in that I can track changes, I can see when people are doing things, is to me very useful. It's a repeatable paradigm, that's what they're doing in the code world, so I'd love to be able to do this in the database world. I don't think it's an unreasonable request to be able to as a DBA watch when people are making structural changes in real-time basis.

I mentioned that I don't think there's any particular tool that does this. I proposed this in a couple of places and had people intrigued by it and I've often told people "If you build this I will buy it," but I haven't found any yet to date. But the nice thing is there actually is a way to capture that information. Databases for the most part have built-in audit capabilities. Now it's funny, most people think that the audit trail is used for security people, it's to control access to data, but quite frankly, you can actually turn it on and have it record every structural change to your database. I often get objections to this approach, usually people say "Well no, no we can't touch the audit trail, the audit trail is used by our security organizations who ensure that no one logs in as Sys DBA or that no one accesses employee salaries, so audit is only for security, we don't use it for developers' convenience. Not only that, if you turn this on we're going to get a real performance hit, every time someone creates a database object it's going to log to the audit table and we can't afford that kind of overhead."



Actually, there is a way we can capture this information.

We could turn on the audit trail for DDL commands...

Common objections to this approach include:

- Auditing is for security, not development
- It will affect performance!

My answers:

1. Why should Security have all the fun?
2. It's worth the performance hit.

The slide features a graphic of a footprint trail on the right side, consisting of several dark, irregular shapes arranged in a path that moves from the bottom left towards the top right.

I don't understand that. I mentioned earlier that I believe data should be exploited by lots of different people, so if there's an ability to capture the data I want and I'd like to use it, why do security get all the fun watching what is going on in the system? As a development DBA or as a DBA responsible for developing the system I'd love to know how many tables they created this week and are they doing it on a regular basis.

I have a funny comment on the performance bit, in my opinion there shouldn't be an issue with the fact that there is a performance hit every time I create a database object and the reason I say that is because if I have a transaction that as a result of its transaction creates five tables and then drops them, it's not very efficient, it's not a very good use of the database structure to use the data dictionary essentially as a transaction capability. I want the ability to almost penalize people if they're doing high volume DDL, so my opinion that the performance hit is actually a good thing, one of the things I like to do in development environments is make bad practices painful and in my opinion a development process that relies on rapid DDL should be painful, I don't want that occurring in my system. So I think the performance hit is worth the benefit.

The other reason I really like it is when it is turned on I get this feeling of all seeing, all knowing and I'd done this at many of my sites. When I was the director of database development at Network Solutions, I had a team of about 16 database developers underneath me who were supporting four or five development teams and as they were working I would relatively routinely query the audit log see who was working on what. I would go over to their desks and I would say "Hey, I saw that you are working on the products table." They would jump out of their chair and say "How did you know that?" I would often just tell them I know everything and then they'd get all mystique and in awe about my capabilities. But I was just watching the audit log. One of the things I also liked about it was I could tell if two people were trying to touch the same object or if they were touching an object which I didn't think was in their area of responsibility, I was curious, "Why did you need to touch that package? I thought that we weren't using that package."

Why? Because with DDL auditing turned on, you can see **everything**.

For free, without any coding, you get a timestamped record of every structural change made to the database.

You can see:

- Operating system username of person making change (includes Windows username if coming from a client machine)
- Oracle username
- Host / Terminal (useful to see where people are working from)
- Timestamp (actually a DATE)
- Action / Action Name (operation performed)
- Owner / Object Name (object worked on)
- New Owner / New Name (captures rename and dependent objects for constraints)
- SQL Text (extended only) also captures some recursive SQL (for example index creation for primary keys)



It really encouraged discussion about what was going on in the system because you could almost interpret and anticipate what people were doing. And in the case where you don't have database engineers doing this under your direction, the developers are doing this, I found it fascinating to be able to watch them and say "Hey, I noticed that you created the table, but you forgot to put the primary key on it," or "I noticed that you added this index, you added a couple of single column indexes, perhaps you were trying to query and needed a multicolumn index." So really as a database engineering manager the ability to watch what was going on to me was very powerful. Unfortunately I haven't seen good tool support for this yet, but it's not all that hard to do since it's in a table, you can obviously query, you could define a report in the SQL Developer for example, to have this event information available.

In the above image I mention all of the different objects of data elements that you get with it. You can actually get the Windows user name if they're coming from a Windows client machine. It sometimes depends on the connection technology, if they're using SQL Net you get a little more than just the JDBC thin connections, but in general you get what user they were connected as, what host they were coming from, when they did in action, what they did if the creator dropped an object, what object were they working on, did they rename something if they're doing new names and stuff. Even if you turn on extended tracing you can get the index creation for primary key recursive stuff. So if I create a table defined with a primary key sometimes you'll see the index creations that follow on from that. I really believe this is a useful capability to have in terms of what you can see, what's going on in your environment.

Just for grins and chuckles, it's not just an Oracle thing, SQL Server, and this is dating me a little bit, but it had DDL triggers since 2005 and I've got a reference here to a DDL audit solution presentation at one of the PASS conferences where they talk about how you could use this capability in SQL Server, so I don't see any barrier to "I'm on SQL Server, I can't do this." The same with MySQL, MySQL has a binary log that you can use to capture what's going on on the system. DB₂ has this capability too. I really like the fact that at least in Oracle that it's table viewable. I have found places where you have to talk to the security organization, in particular around Oracle's best practices often are to make the audit log not in the table in the database because they want to make sure the DBAs can't change the data that is in there. The good news is I think if you turn it on to OS and XML it's only a view, it actually reads out from the file system, so the files are written out of the file system, the XML audit log is written and it might be in a secure location where the DBA can't delete the files, but the v\$ view will read those files and allow you to see what happened, but it's not a table so you can't go in and change it. So I think there are ways to skin this cat that allow best practice security in terms of the audit log, as well as giving me the information I'd like as a database development manager.

No, it's not just an Oracle thing...

SQL Server 2005 added DDL triggers, see
<http://www.sqldbatips.com/presentations/PASS2006.ppt>
for "Building a DDL Audit Solution using SQL Server 2005"

MySQL Binary Log, see
http://pooteeweet.org/files/phptek06/database_schema_deployment.pdf

DB2 db2audit

Best part about Oracle's solution is that all data is in a simple table (and you can create an XML log too!)

Think through some of the possibilities.

Now, imagine the possibilities...

Development	Integration	Production
<ul style="list-style-type: none">• See how much progress you're making on defining the data structure• See who is working on which parts of the "model"• Determine when you need to refresh your reverse-engineered model and run compliance checks• <i>Get a list of what's been changed since the last release</i>	<ul style="list-style-type: none">• See if anyone is making changes that didn't come from your build• Uncover potential performance issues (transactions that do DDL?)• <i>Ensure that everything you've built got installed</i>	<ul style="list-style-type: none">• Make sure no one is changing production• <i>Ensure that everything you've built got installed</i>

Now that I have got for every database that I'm working with, I have an actual log in the database of what was done to that database over time in terms of its structural changes. Now in the development organization, I find this extremely valuable.

On a periodic basis I can look at how fast we're doing or how much work we've done in defining the data structures that people are using. I can go in and say "Have we finished all the tables we said we were going to for this sprint?", "Who is working on them?", "When did they do them?" I can see who is working on which parts of the data model, if I've got a larger data model, and who is working on which pieces. I may use this as an understanding of ways to keep my data model fresh. So for example, I might reverse engineer stuff on a periodic basis into something like SQL Developer Data Modeler so that I am continually maintaining the data model or ERwin.

I constantly reverse engineer my databases into my models so I can visualize and draw them for the people that like to have the data models checked into the corporate data model. I also like the capability here to use the automated capabilities in things like SQL Developer Data Modeler to ensure compliance. One place I've seen where this really works well is when I've told people "Well why don't we let the developers create the tables or the database engineers create the tables and then we'll periodically assess whether or not they're complying with our naming schemas." The data governance organization I've talked with say "No, no, we can't do that, we have to validate that they are adhering to naming schemas before they can get started with development." I asked them "Well, why is that? If the data, naming schemas are clear and they're easy for anyone to understand, then if I funnel them through one organization I've created a bottleneck, I've created a delay gate." If I can give them to everybody and say we are going to measure how well you adhere to them then I've immediately increased my productivity there. I let all my developers do their work and then on a periodic basis I assess their compliance with the naming schemas. I can do that in SQL Developer Data Modeler by reverse engineering into the model and running the compliance reports, and to me that's a great use of a data governance person's time. Their job is to measure and assess compliance with standards, understand exceptions to standards and basically provide them with the tools to do this in a much more rapid fashion.

Finally, probably the most important thing is if we're doing a continuous integration or release cycle, how can I tell what's been changed since the last time we did a release? This is where the audit log comes in real handy, I can basically say show me every database change since this time period, I'd like to know exactly what's been changed since the last time we did a release. What I would do at Network Solutions is I'd print that list and I'd sit down with my engineers and I would walk through and say "Does this need to go? Does this need to go? Does this need to go? Does this need to go? What was this for?" We basically justified every object and figured out whether or not it needed to be sent to the integration environment. Then if I have this in my integration environment I like the fact that I can determine where changes came from. So in theory, I would do a build, I would create scripts, and I can do that in a variety of ways, and I'll talk about that in a little bit. Let's say I've created scripts that can be used to install changes into integration and I hand them over to someone else to run in an independent validation fashion, so somebody else runs them. Then I log into that system and I look at the audit log and I say did all of my changes come from my scripts or were there changes that came from somewhere else, and if so where?

I talked to some of my friends who are really good at continuous integration and delivery and they'll tell me that continuous delivery and continuous integration only really works well when the environments are the same: when production looks like development then I can continuously deploy to it. The fact that I allow it to diverge significantly, may cause failures in my continuous build process and I want to avoid those. I also think in the integration environment where we start to run some tests, I can start to see potential performance issues related to a frequent DDL. If someone has built a process, maybe in Ruby, that is constantly building and dropping tables to do something, I'll uncover that here in integration because I'm tracking that what I would call painful activity.

Finally, I'd like to make sure that I then do a double-check that everything I think was going to be installed actually gets installed. This is the one place where I can get that list. You can do an awful lot of stuff with Diff stuff as well and I'll talk about that too, but I like having a thing I can print out, I can always query, I can always get a record of what was installed when, at this day, at this time, these objects were created.

The last one is production and this is a fascinating one for me because I've found that probably 10% of the production problems I've seen are often related to someone went in and dropped an index or created a table or did something, altered the system, and there's really no record of it. That's really not the fault of anyone in particular, it may be they were doing it in response to a request, but the bottom line is we should know everything that is getting created in production, and it should only be related to a release or a particular known requirement. So when you're doing a root cause analysis of what occurred in the production database and you'd like to trace back to any potential changes, this is why I really like the DDL in the structural audit log. I can see in production that yesterday Jim logged in and created an index on that table, I can definitely see that. Then also when we are doing a production deployment, it's not similar to integration, did everything I want to get installed in production actually go ahead and get installed? So I like having that audit log for capability for production as well.

So people tell me that looks great, but you really haven't told me anything about how this makes me be more Agile. You're getting a good view of what's going on, perhaps in the environment you may be buying lots more people to do development, you are becoming more Agile.

But are there other things I can do? How can I do this continuous integration or deployment that you talked about here and more to the point can I make sure that what I'm doing is in alignment with what my developer is doing? They're doing a rapid development methodology, how do I align and start speaking in a similar language so they don't laugh me out of the room when I go in to talk to them? Are there tools that support that for me?

That's great, but it doesn't look agile enough to me...

How can I do continuous integration / deployment?

More to the point, can I align my tools and processes with the rapid methodology the developers are using?

Are there tools which can support what I need to do?



This really gets down to the fact that in many cases database development has some unique characteristics that are different from simple code. The bottom line is we basically have three things we want to keep synchronized: we have our database, we have our database of records that we're doing development against, we have a data model that we have a requirement to produce for our data governance organization and then we have build and deployment scripts that we want to have reflect the database as it is today and the set of changes that we've made to that database over time. So we really want to ensure that all three of these areas are in sync, so that if someone said to me "We've lost the database", I should be able to say "Well go to the deployment server or the build server and pull out all the scripts and rebuild it." That ought to be a no-brainer but it will need to be synchronized. Or someone says "I looked at the data model and I don't see this data element", that should reflect what's in the database as well. All three of these things should be in locked step in terms of what they reflect of the environment.

Fundamentally, we have three things we have to keep synchronized:

Few toolsets can handle all three:

- Toad and Toad Data Modeler, combined with Schema Compare and Toad VCS plugins
- SQL Developer and SQL Developer Data Modeler combined with SQL Developer Source Code extensions (Subversion, Dimensions) and Database Diff

Some tools can do two of them very well:

- Erwin - Database and Data Model
- dbMaestro - Database and SCM
- Liquibase - Database and SCM
- sqlplus - Database and SCM
- Schema Compare for Oracle - Database and SCM

Oddly, two-way Database to Data Model synchronization is hard.

Also oddly, given the number of comparison tools, creation of Scripts for SCM is hard.



One challenge I've found though is there are very few tools that handle all of these together. Toad and Toad Data Modeler with Schema Compare and the Toad source code control plugins comes close, it's not an entirely integrated environment but in general within one set of tools you could potentially do all three. SQL Developer and SQL Developer Data Modeler combined with the SQL Developer source code extensions like Subversion and Dimensions and the Database Diff capability will help you as well.

Some tools do two of them really well. ERwin will do the data model and database. dbMaestro which is a small company in Israel will do source code control on the database and source code control management. There is an Open Source toolkit called Liquibase that can be used to build deployment scripts. You can always use good old SQL Plus and Red Gate makes [Schema Compare for Oracle](#) which will help you keep the database in sync and [generate scripts for deployment](#) as well.

One thing I've found is that database to data model synchronization is really hard two-way in particular for the reverse engineering that I like to do from the database into the data model. Most data modeling tools assume a waterfall approach in which you complete your data model and then deploy and they are relatively weak at integrating back end changes from a database that's changed underneath a model. I've seen some neat things from SQL Developer Data Modeler for this where they can do these kind of comparisons and show you which side of the system is out of sync, but it's a little clunky I think, I'd like it to be better. Also what I found is given the number of comparison tools that are out there, the creation of scripts that can be sent into or fed into a build process is weak. I've found that most comparison tools believe that they generate and make changes immediately; they don't make the changes that can then be checked into Source Code Control so that we have a better picture of how we deploy changes.

They say “Go into the comparison tool and apply the changes in the other database so you get them in sync.” So I’ve been on a crusade for the past two or three years to get these comparison tools to produce script output that can be more amenable or useful when building a deployment or a build process and I’ll talk a little bit about that as well.

Most interestingly, lately I had a Tweet conversation with Chris Rice who is the development manager for SQL Developer and they made a change to SQL Developer 3.1 in which when you do the schema compares it used to produce one long script to change everything and they’ve modified it now so that it generates a script per object and per change, similar to the way it does on unloads. So what I like about that is I can see object by object what’s been changed, I can check those into Source Code Control, I can do a little more granular control instead of having one monolithic script. That’s a good change, they just added that to 3.1 SQL Developer, so check that out, I like that a lot.

I guess my point about this is, and this gets back to some of my laziness comments from earlier, I like to generate this stuff as much as possible. I like to drive off the audit log to see what’s been changed or I like to use the comparison tool to see what’s been changed. I really want to assist the ability to push changes into the upper level environments through automation. When I started at Network Solutions literally when we were done coding up the database to deploy to integration testing was another set of hand coded scripts to deploy those changes and I think that’s just inefficient. I want my folks focusing on development, not on deployment scripts, those should be automated. The other reason I like [automating the generation of deployment scripts](#) is I can actually inject features into those scripts. So for example, when I generate a deployment script I might inject the Source Code Control tags that I’m going to put into my post script control system. So I can standardize how my scripts work and that’s what I get with a generation tool, I can possibly also standardize how I inject or require certain things to be in every script.

So I’ve got my doughnut line here and it talks about this automation thing and basically what I’m going to put here is things I think you should be demanding from your tools backups, that data modeling tools should be much more robust at handling reverse engineering on an iterative basis. Your tools need to become as Agile as the method you’re using and that is when you’re doing incremental refactoring of a database, how do I reflect those refactored changes into my data model? I guess the joke you’ll hear from many places is that the data model is out of date the day it’s done. I don’t like that; I think a data model is useful, it can be very important for folks who are trying to get an understanding and an overview of the system.

Use the generation and automation features of your tools:

1. Data Modeling Tools
 - Reverse Engineering
 - Standards Compliance Reporting
2. Database Developer Tools
 - Visibility and Alerting
 - Continual, Easy Diff
 - Script generation → Migrations and Refactoring
3. Learn other tools
 - Git / Subversion
 - Liquibase / Ant / Maven
 - Puppet / Chef
 - Ruby / Rake



So how do I make it continually up to date? How do I make sure that it's always showing what's accurately in the system? I want my data modeling tools to be much better at reverse engineering and what I call standards compliance reporting. How do I use a data modeling infrastructure to describe the main rules and naming standards and use that to examine databases so that I can keep them in compliance and do that very rapidly. I shouldn't have to send a script to a data governance group and wait for them to review it by hand or by eye and see if I'm following the standard convention. The tools will do that, let's make them do that and use them for that fashion. The second area is the database developer tools and here we look at things like Toad and SQL Developer and [Schema Compare](#) and how do I make sure those tools are providing me with the things I'd like to have and do they have this visibility in alerting me to what's going on? That social database aspect that I mentioned before where I can see that people are making changes to the system, can my tool do that in a collaborative way? Can I have a little window on the right that shows me what's going on in the system? "Jim logged on, he's running a query, he created a table," and can I subscribe to the flow of what Sally is doing in the database and how can I really watch what's going on so that I'm fully aware of how the system is evolving? I want that from my developer tools.

I think Diff should be something you do every day. I used to do this at Network Solutions, I would [Diff the databases](#) on a constant basis and I would not allow changes or deltas, the fact that a system was different from another one, I wanted to eliminate every possible area of difference that could creep in. I would use the same table space names in each environment, I would auto-allocate storage crosses in every environment. What can I do to ensure that the environments are the same? When doing partitioning I want the tables partitioned in development the same way they are in QA and the same way they are in production.

Obviously the partitions in production will be much larger, but if I tested a thing in development with ten partitions, I want production to have ten partitions. To me the way this can be done is if Diff programs are easy to use and [Schema Compare](#) is a great example of this. You should run it regularly.

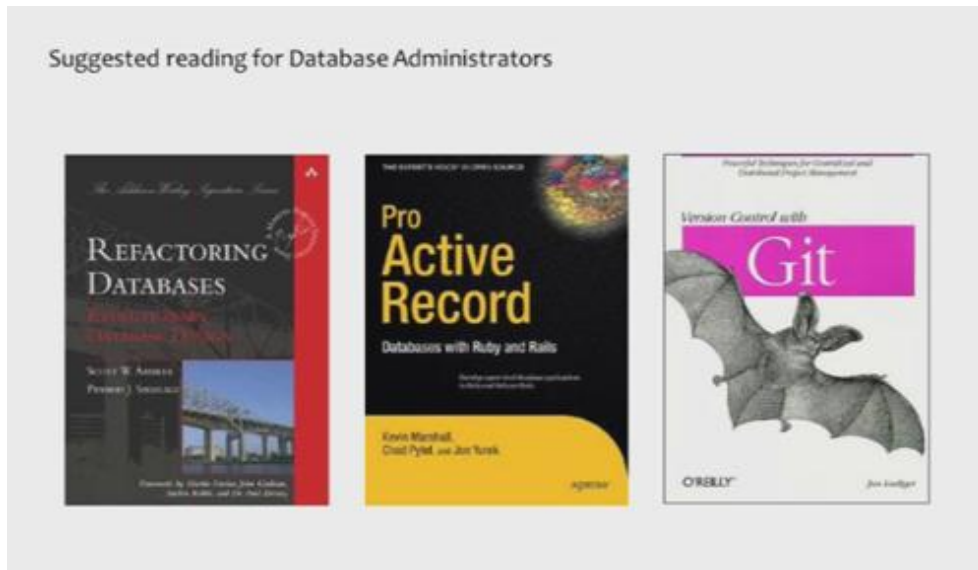
Now this does get into the issue of access. I've had many organizations say "Well, we can't allow developers to access production," or "We can't allow them to access QA," and Diff by its very nature requires access to the whole system, you need to examine the data dictionary from the whole system to compare. It might be possible, I know [Schema Compare](#) does this, where you can take a snapshot baseline of production and compare against that. So there are ways to do this that may assuage some of the security concerns, but at the end of the day I think it is tables stick for me, that your system should be exactly the same, index names the same, column orders the same. There really are no good reasons why they should be any different. I've worked in systems where they've said "Oh it's okay that the column order is different in production than in development," and then lo and behold someone does a Select* and the column order is different. They'll say "Oh, they shouldn't have done that," or "It shouldn't have made a difference," and to me if someone says that minor difference isn't importance, I can guarantee that six months from now something will happen and that was the difference that was an issue. So I think there is no reason why you shouldn't be constantly doing Diffs and it should be very easy to do.

The last one here is the script generation. I think you ought to be able to take a checkpoint of where you are since the last time you generated scripts and say what is the set of changes I need to promote to other people or other environments so that I can synchronize or check-in and merge what changes I've done. It's an interesting concept and I didn't talk about it a heck of a lot, but whether or not I allow developers to have their own databases, or I have a centralized database that everyone works against, I don't particularly care one way or another. However, if I'm going to allow everybody to have a private database I need this ability to determine what changes they've made to it and merge it back into a central database, just like code merging. So I think this is an evolving area for database development.

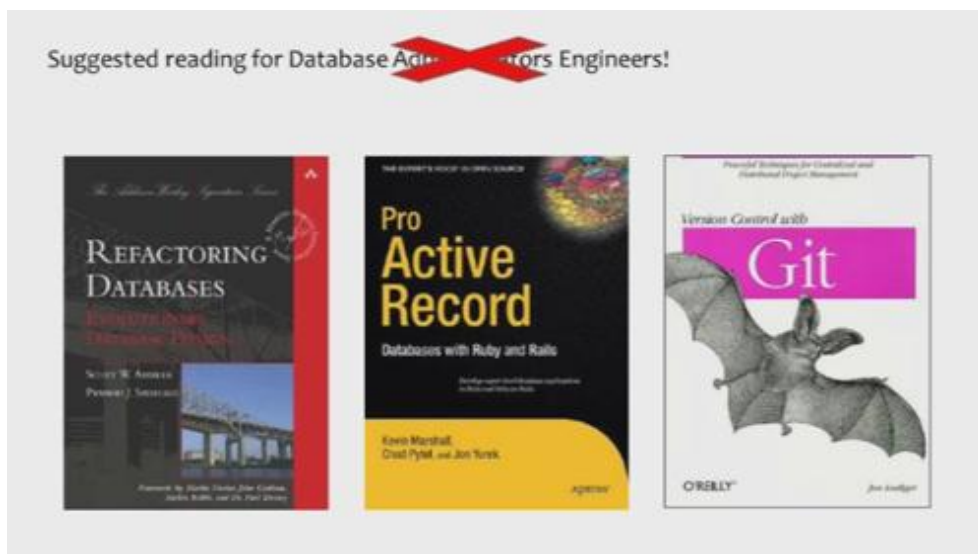
I talked with Bryn Llewellyn from Oracle Development about addition based redefinition and how they're looking at maybe including some of this capability in there as well, the merging capability and the branching capability. So I'm excited to see that Oracle is thinking along these lines as well.

I think it behooves us as database professionals to start learning the language and tools that our corresponding developers are using. If they're using Git or Subversion for Source Code Control and they want database scripts checked into there, we need to learn those things; we need to get smart on how those work. The good news is they're Open Source, you can download them, there are books on them, and this isn't hard stuff. I encourage everyone to vote for the Git plugin for SQL Developer, I want to make sure that people are really trying to add that capability into the tools that are there. I look at the other tools as well, like Liquibase and Ant and Maven for building.

I get familiar with things like Puppet and Chef for automated deployment and I also really think as a DBA it's interesting to learn about Ruby and Rake because of the active record schema definition paradigm and the database migrations capability that Rake brings to the table.



Above is some suggested reading for database administrators. The book on the left probably began my journey into understanding how I could do Agile database development. If you're familiar with the concept of refactoring in Agile, I think this is about the only book out there that talks about how to refactor a database so that you can constantly add changes to it. It's written by Scott Ambler and Pramod Sadalage and Scott Ambler is one of the folks around the Agile manifesto, so I think this is quite a good book to read and understand. The other two books are examples of things that we need to get smarter on, that is the active record paradigm, database development with Ruby and Rails, and then of course Git is what I'm seeing as replacing a lot of the Subversion stuff.



As I mentioned before, it's not database administrators, it's really database engineers.

Q+A Session

- Q. If you're suggesting the database and data model be kept synchronized, how do you reconcile it with allowing the developers to dream up DB objects as they go in their development environment? Who is supposed to have the responsibility of maintaining the data model?**
- A. That's a good question. One of the things I like about Agile Development is its collaborative nature and discussion. At Network Solutions the database engineers and developers would actually sit down and discuss what they were trying to accomplish with the data. What I like to do in that case is experiment a bit, how do I think about how I'm going to build the database structures and then reverse engineer. So it's really an iterative, collaborative approach. At the end of the day it got interesting, my job with developers and Ruby developers got to the point where they were like "If you can create tables fast enough for us, we don't have to go ahead and do it, we trust you to do it." At first I really didn't clamp down, I said "You go ahead and get started, but I'm a lot faster at this." It really came down to the fact that I was able to develop tables faster, changed the handover from control back over to the database professionals, but I was able to show them a lot by basically saying "Go ahead and do it, I'll reverse engineer and show you it, I'll show you what's wrong with it, I'll show you things you've missed," and after a while they were like "Oh this is just too hard, if you can do it faster for me please just go ahead and do it." The real issue became the fact that they then trusted me to do that control and make sure it was done correctly.
- Q. If you're only catching non-compliance or undesirable code at the integration stage it's more than likely that a good chunk of already written functionality is now dependent on that. Thus, the developers often force through what they want to the detriment of the integrity of the production systems. As they're not responsible for those they often don't care. How do you suggest this be avoided?**
- A. I think I talked about having that audit capability in development and honestly I don't want to wait until integration to check the compliance. What I talked about with the data governance organization is literally we would get all the databases wide open for you to run your compliance checks and I think you should do those daily. When you talk to some people doing continuous integration, continuous development, they want to do a build daily. We may not do a build daily, but I encourage compliance checking daily. So first thing in the morning, how many things were created yesterday and they all compliant and if they aren't, the good news with the audit trail is I know who did it and hopefully I can pick up the phone and talk to them about it, or better yet walk down the hall or walk around the corner and say "Hey wait, this table is not going to make it to integration, you can't do it this way." So I encourage that level of examination and visibility early on, which is funny, because many of the Agile developers will tell you they love collaboration and visibility except when it comes to other people looking at their code. But the really good mature developers really

welcome the fact that “Oh, you caught a problem early for me, thank you. What should I have done differently?” So I encourage that checking very early on, not waiting until integration.

Q. Is there a graphical deployment tool that can deploy the new code changes from a build to a development database?

A. I don't know of one. If there is one, if someone knows of one, please let me know, I would love to evaluate one to see that. I think it's a really unique and interesting area of research that could be done, but I don't know of one. I've been doing this by scripts and by hand, although the Diff stuff is getting better and I'm getting to the point where I'm hoping to abandon my script based approach in favor of something like Schema Compare or SQL Developer Diff which will really net it out for me as to what's going to happen in terms of the deployment. Where it's going to check it in, it's going to kick off a Maven or an Ant job to do the build. I'm excited that hopefully that's coming soon; I just don't see it there yet.

Q. How would you manage test data?

A. That's a really good question. Test data is a very interesting topic in several areas. I know that the Red Gate tools will synchronize data in addition to structure. You want to look for tools that can do that capability. It can be expensive and it's interesting to describe what kind of test data you're looking at. I know also that SQL Developer had PL/SQL unit testing where it can do scaffold work, stand up a table, populate it with data, run tests that have a set of assertions and then tear it down. In my environment at Network Solutions we would actually do a couple of things for testing, among them we would copy into the QA environment full copy production data and that was really to just get values in place. Then once that was in place they would run test data generation scripts on top of that. So our process really was repeatable in that what we would do on a regular basis was refresh testing databases down work from production and then run repeatable scripts to create testing data. Generally, also that stuff was checked in with Source Code Control. So we would set up our special test cases on top of the production data and we did that obviously to get repeat table test cases, but we liked having the production data underneath it because of the data volume issues that I think really affect performance significantly and may cause you to change your data structures. So I'm a big fan of wherever possible having as much production data downward in other environments. There are issues around decrypting and encrypting in different environments. There's a company in California called Delphix that I've been watching with some interest because they have this capability of essentially staging a production volume in development but not [inaudible 0:53:02]. Kyle Hailey from OakTable works there. I've been impressed with what I think they can do, but I haven't had a chance to really drive into them a lot. I think that's basically my feeling on testing data.

Q. What about the issue of changing the structure while there is test data existing?

A. The question again is about changing structure in the face of existing data and that's an interesting concept because a lot of people forget this. A lot of developers will say "I just want to add this field," and I'd say "Okay, well, how should it be rendered or represented with this new column? Is it a default value? Is it generated off something that's in the row? So I need to probably manually create those data migration scripts that will allow you to transform the test data from one release to a new structure. Of course this depends a lot on the extent to which the structure has changed. A lot of times I've found too, especially if you're doing this, this is why production volume in test environments, I have to deal with that production volume. Can I pre-convert data, can I look at historical data that isn't going to be changed and set up a staging area where I have lots of time to deploy a data transformation script so that the data has been transferred before I have to do my cut over or my deployment. So I think you do have to concern yourself with that and identify structural changes that then require changes to data in place. There are also new structures that require data themselves where you might stand up a new reference table and have a list of 100 new reference values and those insert statements have to be captured and checked in to Source Code Control and used as deployment when standing up that new table. So I think it's not only test data, but I actually look at this more from the perspective of managing the production data volumes. How do I deal with the fact that there's existing data, how do I have to transform it and then is there anything I can possibly do to pre-deploy? One of the things I like about this process is the ability to zero downtime type deployments, I want to deploy frequently, I want to deploy often, so how do I do that in such a way that I don't have to take downtime? A lot of that means, especially pre-existing data, that I have to do pre-deployment of data conversions, data transformation.

- - - - -

Dominic Delmolino

Dominic Delmolino is a former founding member of Oracle's System Performance Group, Director of Database Architecture and Development at Network Solutions from 2002-2007 and is currently Vice President of Data Architecture and Engineering at [Agilex Technologies](#).