

# The Eternal Donut of the Soul

## 1. Database Administration

[podcast@xor.com.au](mailto:podcast@xor.com.au)

What does it take to be a DBA, and an Oracle DBA at that, in today's programming environment, now seen as extreme / RAD environment? From eating boxes of donuts, intimidating developers, holding the keys to the database, to being arrogant, obnoxious and spiteful, the DBAs role is one which most don't understand. Are they the gods of the database world or a bunch of wannabe developers too stupid to work out a for-loop from an *if* statement?

In this podcast, we are going to look at the role of the DBA and try and understand what it takes to become an efficient and effective one.

First some background. I consider myself to be both a DBA and a Developer. I have had extensive experience working in both areas and so consider myself to be in a good position to see what its like being on both sides of the fence. This perspective as well as my exposure from working in large organizations to small ones, to now a startup, has given me a unique perspective on the evolving role that DBAs and Developers have. In particular, what role they should be moving to as technology changes. And this is the important point - as technology changes. Because as technology changes, the rules are changing.

So what is the fundamental role of a DBA? The following is a definition I coined ten years ago to address this and is I still believe is valid.

"To ensure that the database performs to its optimal, is fully secured and can be recovered in time of need."

From this core definition, many ideas about what a DBA really can now be formulated.

Now some history...

When it comes to looking at DBAs there are two extreme's. The first ( which most developers are confident the DBAs in their environment match) are the over-confident, power mad DBAs who slow down everything, enforce draconian change management procedures, treat with disdain any request given to them and demand donut food to do tasks. These DBAs came into being most likely because they can handle the stress, have a tinkering mindset and a knack for problem solving. They view any form of programming as a coding job (beneath them) and are annoyed by the antics of the developers who they view as inefficient SQL writing coders. . The power they have been given, or taken, has ensured that they live in their ivory tower free from the worry and concerns of the masses (mundane developers).

At the other end of the spectrum are the puppet DBAs. These are DBAs who have been given the position by developers, who needed to do something with the team members they couldn't trust to code, couldn't work with, and couldn't get rid of. Being under-skilled and undervalued their sole role is to make sure the backups run and they stay out of the way of developers who themselves control the design, performance and running of the database. Every now and then, the developers who are bored throw the DBAs a performance tuning bone for them to work on.

The point to note, is that if you see yourself at one end of the spectrum or the other, and this is the important – somewhere in between this spectrum (anywhere in between), then its too late. The culture of the organization is the problem and it needs to change.

To be an effective DBA, the culture of the organization needs to drastically change. Some fundamental concepts need to be changed as well. The models used in the past have failed because they do not take into account personalities, bureaucracy and the natural direction red tape takes in any organization (which is to gradually grow to the point of paralysis) They also don't consider changing technology, team members and group dynamics. Red tape is normally introduced as a protection measure. A person makes a mistake, so a procedure is introduced to ensure that mistake doesn't happen again. Sometimes the procedures are good to have; sometimes common sense would be quicker and better. So when a procedure is introduced, before it is implemented, the reasons why it is implemented should be reviewed.

Now lets side track for a moment and look at the word incompetent as I will be using it a number of times. The dictionary definition from [www.dictionary.com](http://www.dictionary.com) puts it as:

**Incompetent:** lacking qualification or ability; not having suitable or sufficient skill, knowledge, experience (note this definition is summarised).

So the term doesn't really have the negative connotations we are used to using the word. It really means the person needs more skill and ability.

Now back to the developer being incompetent and change management procedures. We should now ask, is introducing the procedure really going to make life better for everyone, or is it being introduced because the culture of the organization is afraid to realize that the problem was due to lack of skills and knowledge and are not willing to say that because of the conflict (read - potential personality clashes) that will result? Put in a procedure and everyone is blameless. We move on and hope it doesn't happen again, the procedure absolves the problem. Great in theory but all that happens is that the procedures grow and restrict the flexibility and adaptability of the organization. If technology didn't change - if the rules didn't change then this might be reasonable. But it is changing, it's a dynamic playing field and the procedures will get in the way and stifle the effectiveness of the group. If the procedure must go in, give it a six month life expectancy and review it at that time period to see if the reason for having it is still valid. Sure in some cases slowing down the process is effective (give me those cowboy spurs) but there has to be a change in the culture and realize why the rules and procedures are there and know when they are redundant.

But back to the original point of this discussion and that is, what needs to change in the DBA and Developer world? What cultural changes are needed? Lets throw some punches, get our hands dirty and raise six concepts:

1. That the DBA works closely together with the Developers, and this is within shouting distance.. Keeping them separated, by a partition, a wall, a floor, a building or even locating interstate is a recipe for disaster.  
Outsourcing the DBAs to another company is the worst thing an organization can do. If this was done to save money, then its apparent that the managers are incompetent and should be told so (on this point, another podcast will cover the topic of performance tuning an Oracle environment and why the greatest cause of any performance problems are that caused by management). Organisations that put a barrier up between the DBAs and the users are preventing communication, application development time and introducing hidden performance problems.
2. DBAs must be taught to have interpersonal skills. They must be able to talk to users. They need an attitude adjustment. DBAs can be their own worst enemy. By being arrogant, aloof or just plain incompetent, they can hide their lack of knowledge behind slick computer terms and change management procedures.
3. DBAs must be able to program. They must have a thorough understanding of the programming language the developers are using. They must have a very good understanding of computer human design concepts. They must know how the users use the application. Performance and tuning isn't just tweaking the engine, it's also looking at how the user drives the application. This is such a fundamental concept yet it is never seriously raised in tuning books – the mentality has always been to fix the SQL, which is just part of the issue. Its' not the whole issue. But because DBAs lack those skills for dealing with users, they are kept away from them (who knows what they might say to them)
4. DBAs must have business sense, and understand the needs of the organization and be able to balance the cost, deadlines of a project and work to them not against them. A DBA can be a manager's nightmare because they always slow things down, make life hard for everyone and use procedures as a justification for not achieving. They need to understand the business and look at the bigger picture as well. Of course if managers insist on separating the DBAs from the developers its no wonder they don't appreciate the big picture as they are kept away from being a part of it.
5. A good DBA is one who makes their position dispensable. A good DBA is lazy. But lazy in that they will make the database/computer do the work for them. They will spend 10 hours writing a program to monitor the system and report on it, just so that they can sit back and read the latest Dilbert cartoon and read up on the latest Oracle releases.
6. A good DBA is not only proactive; they know what that term means and actively implement the concepts behind it in their environment. They don't say it or preach it, they do it.

The comfort zone dilemma. The challenge is to realize that by our very nature we adopt a conservative approach. Using new technology is hard, risky and it can hurt the head just learning about it. It can be hard to motivate ourselves to learn to use it, we resist it. We sometimes wish it didn't change.

Technology is changing as rapidly as it did 10 and 20 years ago, and we must realize that concepts, ideas and assumptions once made (our belief system) needs to be challenged because its likely out of date. I would hope that the older generation DBAs would have learnt from this and taken a less orthodox and more adaptive approach based on experience, but what I see is often the opposite. They don't want change, hope things don't change and resist change or adopt change at a slow rate. (A sign of this is seen by the length of time it takes to upgrade to later releases). The younger generation just coming out of university, who have had to go through the changes are keen and eager, and this enthusiasm can be mistaken for ability. So the youngsters are given the reigns, without the experience and end up making the same mistakes and the cycle continues.

Now onto age and being a DBA. There is a myth about having to be young to program. This is leading to the culture of an organization fostering the view that management is the only progressive position to be in. To earn the extra money one has to manage. It's good that this myth is starting to crumble as organization start to realize this is not the case. It's also dangerous, as most people who are good at IT are lousy managers. But when it comes to managers and the problems they cause will be left to another podcast.

## **Conclusion**

The role of the DBA in most organizations must change. For this to happen, management must take responsibility and change the culture of the organization to ensure their role can change. The DBA must move out of isolation, be skilled so that they can communicate effectively and be given sufficient power and control to do their tasks. They need to be given more verbal say in the security of programs, be able to review code and work closely with the Developers in the very early stages of design.