



**Craig S. Mullins**

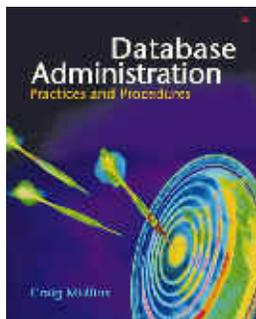
[Return to Home Page](#)

August 2004



## The DBA Corner

*by Craig S. Mullins*



### **Reviewing the Top DBA Excuses... and How to Overcome Them**

Let's face it; sometimes it is easier to give an excuse than it is to take the time to answer a technical question. Especially if it's the third time you've heard the same question in the last hour. And dealing with databases is complex and time-consuming, so questions are always coming up. And programmers are always coming to DBAs for help with technical problems. If you're reading this column right now you've probably been on the receiving or giving end of one of these excuses at one time in your life. Let's see how many of them you recognize.

The number one all-time DBA excuse can be broken down into two words – “it depends.” DBAs are notorious for answering every question with those same two words, “it depends.” And they are right, it **does** “depend,” but how does that answer help the situation? Oh, it might cause the programmer who deigned to ask the question to slink away, and maybe that is what the DBA was shooting for anyway. But they really shouldn't go away. If you are a

developer and the DBA ever says to you “it depends” do not take that for an answer. Ask the follow up question “on what?”

Any DBA worth their salary will be able to run down some of the issues involved in the matter and point you to the proper manual or code to help you figure out your dilemma. Which brings me to the second biggest DBA excuse, which is RTFM. As everyone knows, this stands for Read The Friendly Manual.

I know, fellow DBAs, it gets old very fast when people keep asking you questions that are covered in the software manuals. But DBAs know the DBMS manuals much better than the application programmers, end users, and systems analysts who are asking the questions. It really shouldn't be too hard to answer a simple question instead of barking out “RTFM” all the time. At least tell them in what chapter of TFM they can R it!

Another classic excuse is to blame the DBMS vendor. It is always easier to claim that some piece of sage advice being offered came from the vendor because it lends legitimacy to the advice while at the same time relieving the DBA of any burden of proof. You know the situation I'm talking about. You, a DB2 programmer encounter a problem and go to the DBA for guidance. And the DBA says something like “Well, IBM says to ...” Don't accept this type of excuse either. IBM is a company and it doesn't “say” anything. Same goes for Oracle and Microsoft and Sybase. It is better to know who at IBM said it and in what context. Or, perhaps no one “said” it but the DBA read it in an article or a manual or a book. That is great, but many times written advice is mangled and twisted when it is repeated orally. When this type of “excuse” is attempted you should obtain enough details from the DBA to seek out the source materials (or person) to make sure it truly applies in your situation.

Another popular excuse is the phrase “It is working as designed.” Well, that might be the case, but if the way it is working isn't what you need it to do, then you still have a problem. If the DBA tells you that it is working as designed, he is basically telling you that you do not understand the way the DBMS works. And that may be true, but then he should really try to help you find an alternate solution, that does what you want, while also working in the database “as designed.”

The final excuse I'll talk about here is over-reliance on company standards. Some DBA groups grunt out mounds of steaming standards and guidelines that they then attempt to enforce for every project. Standards can be useful, but they shouldn't become a crutch. When a specific standard ceases to make development and administration easier, or performance better, then it is time to make an exception to that standard. If you come up with an elegant solution that works better or performs better, don't just accept it when the DBA says, "that doesn't fit our standards." Make sure that you both understand each other's position. Perhaps there is a valid reason why the standard should apply – but make sure the DBA explains it to you if your solution really is easier. And as a programmer, make sure that you really have to deviate from a standard before making a big deal about it. That way, when you have a valid case, the DBA will be more apt to listen and compromise.

If I had a nickel for every time a DBA used the excuses in this article I'd be retired and living on an island somewhere. But you'd still be hearing the excuses! Now don't fret DBAs, because next month we'll tackle the top programmer excuses. Won't that be fun?

From [Database Trends and Applications](#), August 2004.

© 2004 Craig S. Mullins, All rights reserved.

[Home](#).