



**Craig S. Mullins**

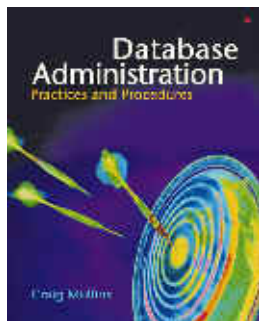
[Return to Home Page](#)

September 2004



## The DBA Corner

*by Craig S. Mullins*



### Programmers Use Excuses, Too

Last month, we examined some of the DBA excuses to avoid problems, work, and other bothersome issues. But application developers and programmers rely on their fair share of excuses, too. You've probably heard most of them.

The number one programmer excuse is to blame the DBMS. If you're a programmer, chances are that you've either said something like the following (or at least thought it): "There's something wrong with DB2 (or insert your favorite DBMS here)!" The basic mentality is that the DBMS is guilty until proven innocent; the programmer will never run up to the DBA and say, "There's a problem with this horrible code I wrote, can you help?" Blaming the DBMS is never a helpful strategy. In some rare instances, there will be a problem in the DBMS itself. Most "database problems" can be traced to programming

problems. By keeping this in mind, everyone will be better off; the problem will get fixed sooner, and you will not alienate your DBAs by constantly blaming the DBMS.

Another common excuse is known as Copied Code Syndrome. As most programmers know, copying working code from one program to another is an efficient way of developing programs quickly. But with database development, you have to be careful to make sure that what you copy is really what you need. The way this excuse works when the programmer is confronted with a problem in his program; he simply says, "That can't be a problem because I copied from another program, and it works." However, many things can go wrong with copied SQL. Maybe you copied something that is 95 percent of what you need, but you didn't modify the code for your purposes. Or maybe something is different about the rest of the code in your program that makes the copied code ineffective.

A corollary to Copied Code Syndrome is the "It worked yesterday" excuse. But today is another day, and if it ain't working today, it ain't working. Many things can change from day-to-day that can cause working code to become problematic -- not the least of which is that code itself. Programmers make so many changes to code as a requirement of their job that sometimes you can just forget that something did indeed change. Work on solutions to problems instead of excuses to deflect blame, which is counter-productive.

Yet another excuse bandied about by developers is the Better Mousetrap excuse. The best approach to developing programs using a relational database is to put as much work as possible into the SQL and let the DBMS optimize the access. But there is always that Wile E. Coyote developer who says, "But I can do that better in C or Java (or insert your favorite programming language here)." Doing it in SQL puts the work on the DBMS -- and there is a much better chance for the DBMS to be bug-free than whatever code you cobble together.

The final programmer excuse I'll mention can best be summarized as "There is always time to do it over later, but never enough time to do it right the first time." Usually this excuse comes to light when you hear that magic phrase, "It's too late in the project to re-write that." But the problem caused by the code continues to exist -- the programmer just wants some magic to occur to fix it that does not require coding changes. Won't happen! Sometimes the code has to change to solve the problem.

Research and understand any issue before you go running to the DBA. If your program fails, find the SQL code and reason, see if you can fix it yourself. If you don't understand something, read the manual before going to the DBA. There is no reason why everyone shouldn't have their own set of manuals; most of them can be downloaded for free from the DBMS vendor's Web site.

Let's all climb back into the trenches and vow to avoid using the excuses we discussed. It makes working with your databases a lot easier.

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

© 2004 Craig S. Mullins, All rights reserved.

[Home](#).