



**Craig S. Mullins**

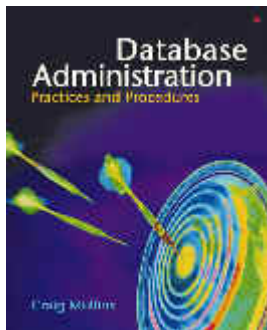
[Return to Home Page](#)

September 2006



## The DBA Corner

*by Craig S. Mullins*



### **DBA Change Management**

The only constant in today's complex business environment is change. Both businesses and individuals can find it difficult to cope with change. For individuals change usually involves taking on additional work – and who wants to do that if they can avoid it? For businesses,

change can be as simple as adapting to a new employee or as complex as merging with another company. At any rate, change requires our attention in order to ensure that business continues to be conducted properly.

There are many different aspects of managing change, particularly with respect to Information Technology. IT professionals are constantly adding, removing, and adjusting various aspects of the infrastructure: servers, networks, applications, middleware, and so on. With regard to databases, DBAs always need to be prepared to modify database structures to accommodate business changes.

Indeed, many factors conspire to force us into changing our database structures, including

- Changes to application programs that require additional or modified data elements
- Performance modifications and tweaks to make database applications run faster
- Regulatory changes that mandate storing new types of data or the same data for longer periods of time

- Changes to business practices that require new data or changes to existing data
- Technological changes that enable databases to store new types of data and more data than ever before

And change will not slow down any time soon, so we need to prepare solutions that enable us to better manage these inevitable changes. To successfully implement an effective database change management policy you will need to consider many factors including proactive change, intelligent automation, planning and impact analysis, standardization, and availability.

Proactive change, which can eliminate future problems, can be one of the most valuable types of change. The earlier in the development cycle that required changes can be identified and implemented, the lower the overall cost of the change will be.

Automating change management processes is important, too. With limited resources and growing workloads, automation can help to reduce the amount of time, effort and human error involved in effecting database change.

But simple automation is no longer sufficient. The impact of each change needs to be examined before implementation. A simple change in one area may cause a complex change in another. Intelligent automation of change management processes incorporating in-depth analysis can improve the quality of your database changes.

And speaking of analysis, an effective DBA will ensure that each change undergoes exhaustive up-front planning and analysis. Comprehensive impact and risk analysis is needed in order to gain a macro view of the change. The only way to know which change tactics are optimal is to conduct an in-depth impact analyses, and then choose the most effective, efficient, and worthwhile strategies.

Standardization is also important. With attrition, promotions and changing job roles a standardized process is needed to maintain productivity. An organized and thoroughly documented approach to completing a task reduces the learning curve, as well as training time.

Keep in mind, too that many database changes require down time to implement. Database structures may need to be taken offline in order to effect change. But high availability is required of most applications these days,

especially web-based applications. Reducing the downtime required to make a database change increases application availability. Indeed, all of the tactics discussed up to this point can help to reduce downtime when databases are being changed.

Keeping in mind everything discussed to this point, we turn our attention to the DBA. Even though the DBA is the custodian of database changes, he usually is not the one to request a change; that is typically done by the application owner or business user. There are times, too, when the DBA will request changes, for example, to address performance problems or to utilize new features or technologies. Regardless of who requests the change, the DBA is charged with carrying it out and ensuring that all changes are performed successfully and with no impact to data integrity.

Without a robust, time-tested database change procedure, the DBA will encounter difficulties. Why? Well, most DBMS products do not support fast and efficient database structure changes. Each DBMS provides differing levels of database change functionality. But none easily supports every type of change that might be required. One quick example: most DBMSs today do not enable a column to be

added easily to the middle of an existing row. To accomplish such a task, the DBA must drop the table and recreate it with the new column added. But when the table is dropped the data is deleted, too. So the DBA must first unload the data. But what about the indexes on the table? They would have been dropped when the table was dropped, so unless the DBA knows this and recreates the indexes too, performance will suffer. The same is true for constraints, authorizations, and so on – when the table was dropped all structures related to the table were also dropped. And this is but one example of a simple change that becomes difficult to implement and manage.

Database change quickly can monopolize a DBA's time. Preparing for change and implementing the proper tools, techniques, and procedures for database change management is one of the most important jobs undertaken by DBAs today.

© 2006 Mullins Consulting, Inc. All rights reserved.

[Home](#).