



Craig S. Mullins

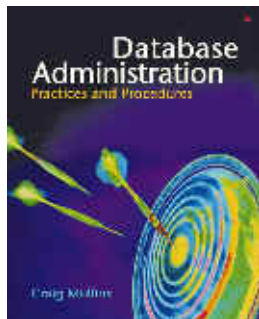
[Return to Home Page](#)

October 2005



The DBA Corner

by Craig S. Mullins



Database Archiving Solves Many Issues

With increased competition and reliance on sophisticated ERP, CRM and other business-critical applications, companies are storing more data than ever before in database management systems. Accumulating and managing more data can have a negative impact on application performance.

Applications can experience slower response times as queries have to slog through volumes of data to find the pieces of data being requested. Consider a query that deploys a full table scan. If the table has 1000 rows, that query will run faster than if the table has 100,000 rows-- even if the result set of the query is only 10 rows in both cases. Larger databases take longer to manage and maintain, resulting in costly downtime, causing application availability to suffer. In addition, it can take longer to spring back from a critical outage when both current and historical data reside in the same operational database and the DBA has much more data to recover.

So what can be done? A well-thought-out plan for archiving seldom-accessed data from production databases and storing it separately can help. Of course, this is not a simple task. Most applications have strategies for validating and updating data, but few provide a method for safely removing inactive data from production databases when it is no longer needed. Even if data could be safely archived, any subsequent need to access or restore the data would pose a major challenge. Having the capability to access archived data is essential.

Government regulations may require that data remain accessible long after your organization needs it to conduct ongoing business. Your company may need to access archived data to respond to an audit or discovery request. So, even though the data is removed from the production environment, it must remain accessible. Furthermore, the consistency of the data being archived is important. You cannot approach

archiving on a table- by- table basis but, instead, must focus on archiving referentially intact subsets of data. A capable database archiving solution must manage complex relational data and keep that data referentially intact.

Once the data is archived, technicians and business users need the capability to access archived data in its relational or business context without having to restore it. They also need the ability to selectively restore only the desired data. It should not be necessary to restore massive amounts of data to access the few rows required.

Database archiving involves planning, in-depth knowledge of the data and metadata, and a flexible mechanism for accessing the archived data. Furthermore, database archiving solutions must be scalable and support the most popular DBMSs and application suites, so you can use it to archive data from multiple, different DBMS products -- and restore it to a different DBMS if you so desire.

The best approach is to implement commercial off-the-shelf (COTS) database archiving software such as those from Princeton Softech and OuterBay. Most companies implement such database archiving solutions using a phased approach, with the overall corporate objective to employ a strategic database archiving solution across the enterprise. Once the archiving policies are established for a selected data- intensive application, much of the process is automated.

Some of the steps for implementing database archiving include: analyzing your business rules and objectives, archive requirements and policies, and access and storage requirements; identifying data that can be archived to meet business objectives; and verifying the archive process, including selectively archiving and removing data from a production database, and accessing and restoring it as needed.

Database archiving is recognized as a best-practice strategy for improving performance, increasing availability and lowering costs. Start taking control of continued data growth and plan for your database archival needs as an integral part of managing your enterprise application data today.

From [Database Trends and Applications](#), July 2005.

© 2005 Craig S. Mullins, All rights reserved.

[Home](#).