Taurus Software
Making Data Liquid

Top Archiving Tips

Top Archiving Tips

Taurus Software
Making Data Liquid

We all recognize that data goes through its own lifecycle – from creation and active use to its eventual disposal over time. During this cycle, as the data ages it becomes less and less valuable to the organization. Building and implementing a strategy to deal with historical information is complicated, but necessary. To help, we have put together a list of the most important components of a successful strategy to manage historical data.

What is it about managing historical data that is so challenging?

Data volumes have grown and continue to grow at a rapid pace. As storage costs go down, it is easier and more affordable to collect and store much more information than in the past.

Compliance, regulatory, and privacy requirements are more significant than in the past. Auditors may request financial or transactional information dating back seven or more years. Being unable to comply with requests for historical information can result in fines, penalties, or damage to the organization's reputation.

Keeping data has significant hidden costs. Large amounts of data degrade server performance and application performance. Slow systems and slow applications result in unhappy and frustrated users. Upgrades of the applications take longer and are more complicated. Database backup and recovery times increase due to increased data volumes. All of these factors have a ripple effect on the maintenance of development and QA environments, since most organizations use copies of production environments as a starting point for the data for both development and QA environments.

How do you keep what you need without impacting your legitimate historical data access needs? **Here are the tips:**

1

The historical data management process should mirror the data life cycle. For items that are accessed frequently — data that is typically less than 24 months old — keep these in production. For those items which are accessed sporadically (25–48 months), remove the entire object and archive it into an online historical database. Data that requires only minimal access — for audit purposes, for example — should be moved to some offline storage. For records older than regulatory requirements, the data should be decommissioned and deleted.

2

Set up continuous archiving cycles. The archiving process should be automatic and ongoing. Archiving is not a once a year process. The archiving process should be occurring on a weekly or monthly basis. Archiving should be taken as seriously as backup and maintenance cycles. All of these maintenance tasks are working towards a fast, responsive application and system.

3

Users should have easy access to historical information. Users will be more aggressive on the timeframes for archiving data if the archived data is readily accessible and can provide timely answers to business questions. Access to historical information can be more restricted, typically requiring read-only access to a limited number of users.

4

Find the right tools to move data. Of course, you could write a custom archiving program; but, using tools which provide automation and can detect data changes and differences in data structures would be much better. Features that are essential to a successful archiving solution include:

- The ability to retrieve and archive data from disparate databases, files, and cloud-based data sources;
- The ability to store data in self-describing files, which can be moved offline, but are easily available for subsequent retrieval;
- The ability to retrieve and restore archived data, even if there are data structure differences between the archived data and the structure to which the data will be restored;
- The ability to perform archiving tasks with little or no manual coding. This job should not require a custom written program.

5

Involve business stakeholders when defining the conditions that need to be met for data to be archived. It sounds simple: just archive things that are old — right? Let's take an example: We want to archive old orders. Would you want to archive an order if any of the items has a warranty which is still active? Or an open receivable balance or credit valance? If any items on the order have not shipped? What if some of the items on the order shipped outside of the specified date criteria? Business users need to set the standards and the threshold that define when it is acceptable to archive an order.

6

Learn and obey the rules. Industries and regulatory organizations set standards for maintaining records. These standards and regulations change them from time to time and are different from industry to industry. Be sure to check with your auditors and regulatory authorities to see what the standards are for storage of historical records.

7

Archive the entire record. If you archive all of an object, then no matter how the requirements change over time, you will be able to produce whatever information was available at the time the original data was recorded. For a sales order, that may mean archiving the order, the related customer data, order items, financial activity, and other related activity. If all of the data is archived, then all future retrieval and reporting requirements can easily be met.

8

Produce a list of those records that were rejected. For those records that didn't qualify, or that were rejected, you want to produce a list. Management or business stakeholders will want to review the list and perhaps correct the issues that kept the records from being archived. Often, a simple process issue can cause large numbers of records not to be archived. When the process issue is corrected, the issue will not recur on this or future archiving runs.

9

Consider moving and deleting in two steps. The typical approach is to move and delete data in the same step. Depending on when you archive — just during scheduled maintenance periods, or on a more frequent, regular basis — you may wish to consider not performing the move and the delete in the same step. Deleting large amounts of data can create inefficiencies and performance issues both in interactive use and in data retrieval. By separating these steps, you can first perform the archiving step, and then you can coordinate the data deletion with regular database maintenance windows.

10

Remember: restoring data is normally done selectively. Your compliance officers and financial staff will tell you that you should provide *only* the information requested during an audit. This means that you will probably not want to restore the entire set of data from the time frame, but instead restore a single customer's records, or a single type of transaction from the given time frame. Make sure that the retrieval process is flexible, yet robust enough to handle these requests.

11

Test, test, test, test. Too often, organizations focus their testing on ensuring that the right data has been archived, but spend little time testing the retrieval and restoring of the archived data. The whole cycle must be tested: archiving, restoring selected records, and ensuring that restored data gets re-archived during the next cycle. Additionally, it is important to develop a test plan with specific data in mind: data that should be archived or restored, as well as data that should not be archived or restored. You need to know which records and component data should be stored in or retrieved from the archive database. Automated tests can be crafted to perform such testing.

Your organization can have the best of both worlds. It can have both easy access important historical data *and* a fast and efficient production database and application by incorporating these tips to build a practical historical data management strategy.

Taurus has been Making Data Liquid since 1987 with over 700 customers world-wide. Our products are the fastest, easiest way to select, extract, manipulate, profile, clean and move data between platforms to synchronize applications, populate data warehouses, build data marts, migrate data cross-platform, and integrate production environments with Web Servers.
Learn more at https://taurus.com.

Taurus Software
Making Data Liquid