White Paper

metagroup.com • 800-945-META [6382]

January 2005

Emerging Opportunities in Database Information Life-Cycle Management

A META Group White Paper

Through 2007, data growth estimates will continue to tax the practical limits of relational database performance, maintenance capabilities, and time requirements. Although processing power doubles and disk prices are halved every 18 months, no such scalability model applies to database software.

META GROUP

Contents

Executive Overview

Corporate relational database management systems (RDBMSs) now manage more than 30% of business-critical data within the enterprise. IT organizations face continuing challenges in managing growing and increasingly complex, data-driven application environments. One option gaining awareness is a process known as information life-cycle management (ILM).

Clearly, marketing hype surrounding ILM as the “next big thing” has raised user’s awareness of ILM; however, there is a growing realization on the part of IT organizations that something must be done to manage data more efficiently. The continuing growth (estimated 45% CAGR) in data volume presents challenges to the efficient performance and maintenance of the database. Indeed, compliance with many of the new regulatory issues (e.g., SOX, HIPAA) should increase both the growth rate and retention requirement pressure. In many cases, these issues must be dealt with in the near term, forcing IT organizations to get serious about developing a database ILM strategy.

Database ILM is an emerging best practice that IT organizations are just beginning to address. It is important to separate the benefits of relational database ILM from the overall ILM market, which impacts file data, content management systems, e-mail systems, etc., as each one needs to balance the solution with the need. While there is certainly a great deal of vendor hype with respect to ILM, users must understand where real ROI can be quickly obtained. META Group believes database ILM is one area where ILM investment can produce quick benefits.

Introduction

Through 2007, data growth estimates will continue to tax the practical limits of relational database performance, maintenance capabilities, and time requirements. Although processing power doubles and disk prices are halved every 18 months, no such scalability model applies to database software. The degree of query complexity, physical design limitations, and CPU-bound vs. I/O-bound transactions precludes linear database performance improvement, even as processing power and additional storage spindles are added. In addition, database availability is impacted adversely as the database grows in volume. IT organizations need to develop some means of dealing with this growing problem.

One option gaining popularity is continuous archiving and deletion of dormant data from the primary application database. This process is known as information life-cycle management (ILM). Through 2006, organizations will begin to adopt

database ILM as a best practice, both from a cost savings perspective and as a means of maintaining acceptable performance and availability levels for critical applications, even as the volume of data continues to increase. Through 2007, we expect organizations to develop database ILM processes specific to individual, well-known (e.g., ERP, CRM) applications and then begin to implement database ILM more broadly as a best practice. In addition, by 2006, we expect an increasing number of application vendors to incorporate data archiving intelligence within their products via either primary development or the embedding of third-party archiving tools, complete with predefined data relationships and business rules. There remains a vast number of homegrown applications that will need to address the same issue and that exist across a variety of operating platforms from mainframe to distributed servers. Infrastructure planners will seek to avoid solutions requiring significant application or database changes if at all possible.

When assessing the challenges of database ILM, META Group recommends that organizations consider the following issues with respect to choosing a solution:

Solution Approach

Database ILM presents some interesting challenges due to the row-level nature of the archive versus file-level migrations used in traditional hierarchical storage management products. The need to access archived data while simultaneously improving database performance requires the solution to selectively migrate rows of data (and related items) from the production tables. This can be done only if the solution has strong capabilities to handle the definition and enforcement of data as well as criteria matching based on business rules that can be defined within the tool. In addition, users should recognize that application schemas and business rules change over time, affecting both the current production system and the archived environment (in case older data must be queried). Therefore, the solution must have a solid approach to handling such changes, so that consulting or lengthy and painful upgrade processes are minimized.

Legacy Impact

Database ILM is not a trivial undertaking. Archival and retention policies must be defined by the organization and specific anomalies must be addressed, including decisions such as how to handle the archived data (e.g., transparent access or controlled) and where to physically place archived data (e.g., separate tables, instance, near-line media). In addition, it is important to assess the likely impact (e.g., code changes, database physical design) on critical applications as time-to-value will play an important role in determining a proper solution.

Business Challenges to Life-Cycle Management

A business executive reading an article about ILM may be tempted to think it was going to revolutionize every aspect of computing. Clearly, there is the hype and then there is the reality.

ILM is not a software solution but predominantly a collection of processes complemented by software. If ever fully implemented, ILM could help organizations better manage their data (and therefore the IT infrastructure that manages the data) from the time it is created until it is no longer needed.

ILM is a combination of process and hardware and software infrastructure. On the hardware side, tiered storage infrastructures enable different services and service levels, offering potentially different cost structures. On the process side, data is categorized (e.g., by application) to better describe its specific value to the application and the organization. This process serves to match classifications of data with its proper tier of the storage infrastructure. Finally, there is a software infrastructure layered onto this environment to enable reporting, protection, and migration capabilities. Only when this, in total, is available can an implementation of ILM be fully developed. And, it is the combination of tiered storage, classified data, and software tools to manage protection, retention, migration, and so forth that provides the benefits of ILM.

There are many degrees to which an organization can attack the issue of exploding storage growth, and ILM is an ambitious concept. Like many large concepts touted in the past (e.g., enterprise data modeling), best practices are often “best” in the abstract. Often, the reality of a solution is more costly than the problem it was meant to solve. To truly do ILM, an organization must undertake a huge cultural, organizational, and development change. ILM processes should adhere to the following key characteristics:

- ***Central Management*** — By focusing on ILM, IT organizations must (in some cases) reorganize themselves to facilitate the new focus. This will likely mean centralized management groups with an end-to-end focus on data management versus infrastructure management.
- ***Heterogeneous Scope*** — The ILM strategy must recognize and adapt itself to the reality that data exists throughout the entire organization on a wide variety of computing platforms.
- ***Data Value Alignment*** — Across an organization, policies must be created to define the value of data, what its retention rules are, and who can access it. Then, the infrastructure (particularly storage) must be aligned to reflect that value.

META Group recommends that IT organizations address the issue of exploding data growth, not by undertaking a grandiose, enterprisewide initiative, but by applying these techniques to a few well-chosen applications as a start.

The higher the complexity of data management, the higher the likelihood of near-term ILM success. We would define success as the ability to show improvement across a number of key aspects in terms of both dollar cost savings and availability, recovery, and performance. RDBMS environments are the perfect starting point, because they support visible applications, are costly to maintain, and can quickly demonstrate success across all the previously mentioned success criteria and more. In addition, this approach does not require large organizational changes to facilitate and can address a near-term need with measurable impact and clearly defined goals. Once success is achieved, it can be internally marketed throughout the IT organization, and momentum for other ILM projects can be created and funding assigned.

The Growing Data Crisis

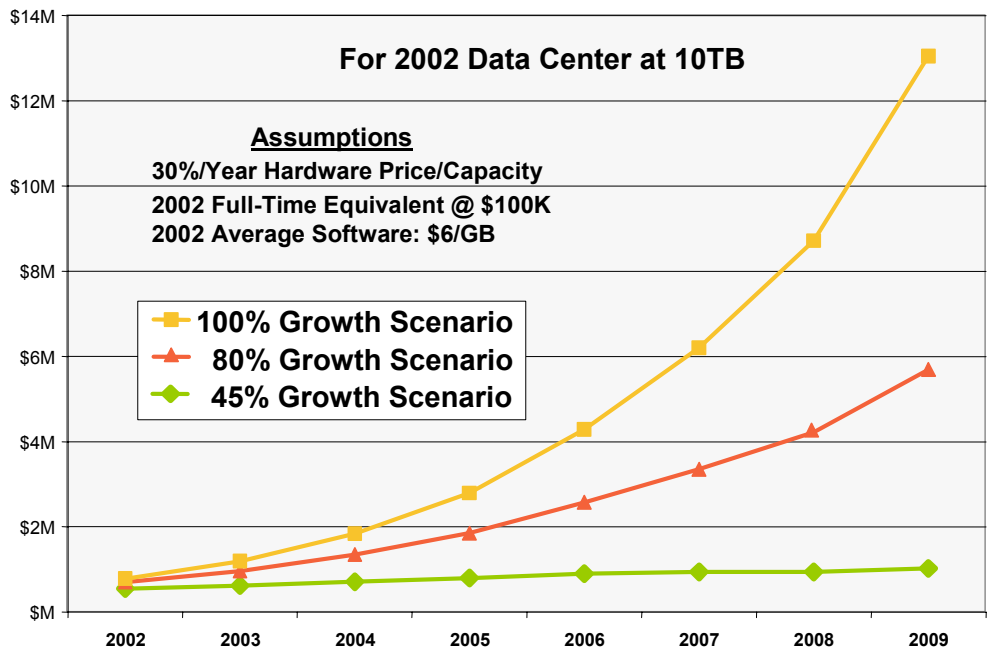
Society is becoming increasingly dependent on automated computer systems — not just to record a transaction but also to understand the transaction in its aggregate, as well as to better understand customer patterns and evaluate current performance metrics versus expected targets or to compare them with historical performance. According to a recent META Group client survey, 77% of the companies responding expected to capture more detailed data in 2004 than they did in 2003, and this trend is expected to continue through 2007. This detailed data is not even transactional; indeed, much of it is subtransactional in nature. This means that companies are now developing new classifications of data to capture, store, and analyze.

Exacerbating the issue of more and more data is the need to maintain duplicate application environments (e.g., quality assurance, high availability, disaster recovery) to address important operational requirements. The growth of data in the primary production database is now multiplied, and the impact can be noticeable both in hardware and in personnel costs. Indeed, the growth of "secondary" (i.e., local replication) storage requirements will exceed primary storage (i.e., new data created via business transactions) by a 7:1 ratio and may reach as high as a 15:1 ratio by 2008. In addition, new regulatory mandates (e.g., HIPAA, SOX) will increase the demand for and/or confusion over data retention and access to historical data as organizations struggle with regulatory compliance.

META Group estimates that storage costs now comprise 7% of the IT organizational budget. Even at a modest storage capacity compounded annual growth rate (CAGR) of 45% through 2009, the storage-related budget (hardware, personnel, and software) will grow at a CAGR of 9.3% (see Figure 1). Personnel and software costs are estimated to make up about half (46.2%) of that total budget. If an organization's storage growth rate is higher, their budget impact can be significant. This model assumes that an organization will continue to mature its storage management processes to achieve the level of automation and savings expected.

Our research indicates that most IT organizations rank storage cost reduction as only moderately important. This phenomenon is likely because planners tend to focus on contract bottom-line costs and often overlook the complete downstream cost impact. So, with disk and server/CPU prices falling an average of 33% annually, it is easy to see why only 39% of IT organizations have implemented any formal storage management policies. What remains a great concern, however, is the issue of business continuity, backup performance, and overall database performance, all of which are adversely affected by the unchecked growth in storage.

Figure 1 — Projected Storage Hardware, Software, and Personnel Costs



Source: META Group

The bottom line on storage growth is that, to control overall costs, we must control personnel and software costs. Unfortunately, as the amount of data a relational database manages grows, so does the need for more personnel to manage the database and storage environment. More data means more query tuning, longer backup and recovery times, and ultimately lower availability. As the database grows, it often needs more processing power, and since most database vendors charge based on the capacity of the server, the more expensive unchecked growth becomes.

Information Life-Cycle Management in a Relational Database Context

The concept of ILM predates the commercial introduction of the relational database. It was a time when personnel were cheap, and hardware and storage were expensive. It was a simpler time when a method known as hierarchical storage management (HSM) was utilized and based on a one-dimensional and very objective measure: How long had it been since the file was accessed? If the last accessed date fell outside the range, the file was copied off to a less costly mass storage system and a file stub left behind on the file system in case the file was ever called for, at which time the HSM tool would restore the file.

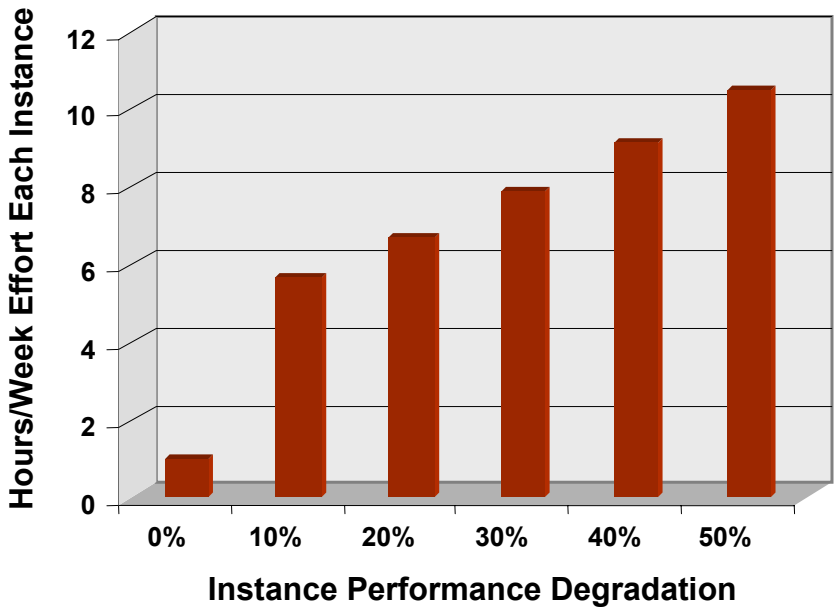
We now find ourselves in an era where storage devices and servers are declining in price year-over-year and are relatively cheap compared to personnel costs. Relational database management systems (RDBMSs) are much more complex environments than the simple flat-file formats of the past. Today's business environment and its reliance on information technology make one-dimensional archiving nearly impossible. Indeed, under new regulatory scrutiny, data that has long since lost its operational value may still contain information that can once again become extremely valuable due to potential penalties that could be levied as a result of more government/legal regulation.

RDBMS software adds complexity to the issue of ILM. Unlike flat files of the past, the RDBMS manages data at a much more atomic level. A single row within an RDBMS rarely contains all the information required to provide context to a business transaction. Rows are contained within tables that, in turn, are contained within tablespaces or segments, which ultimately map to one or more physical files across one or more physical disks. In addition, the RDBMS can maintain referential data relationships between table/entities, so the removal of a row from a parent table would force the RDBMS to also remove child rows in other referentially connected tables. Because the RDBMS adds this rich logical layer on

top of the actual physical file layer, it is easy to see why traditional file based HSM may not be sufficient for the RDBMS.

The RDBMS's query optimization engine determines the best (i.e., most efficient) way to retrieve the requested rows. To make this determination, the optimizer will look at a number of statistics to best determine if it should use an index, if it can eliminate data partitions, and if it should perform operations such as read-ahead prefetch I/O to speed the processing of a particular query. The amount of storage (i.e., number of physical data pages) that must be read, along with statistics such as column cardinality, plays a big part in determining the optimal access path. The access path and indeed overall database performance are directly affected by the amount of data that must be searched and operated on. As databases support higher volumes of data, the more maintenance it takes to keep the database running with acceptable performance. All this translates to higher personnel costs. Figure 2 illustrates the amount of effort required as the database performance degrades.

Figure 2 — Effort and Performance Degradation



Source: META Group

The database administrator would be the specific personnel asset affected here. This cost is not factored in to our previous storage budget estimates, but due to the higher salaries commanded by DBAs, it can increase the personnel-related costs by 20% or more. Indeed, annual management costs for a single database instance that is experiencing 10% degradation in its normal performance can result in as much as a 400% increase in management costs for that instance. Some typical tasks that affect the DBA's ability to manage more database instances as storage grows and performance declines are as follows:

- Responses to performance complaints
- Responses to out-of-space conditions
- Physical design changes (e.g., indexing, partitioning)
- SQL tuning efforts
- Configuration parameter changes
- Daily monitoring and interrupt-driven alerts

Methods of Database Information Life-Cycle Management

So if there are compelling budgetary reasons to pursue database ILM, how is this being addressed? Considering the complexity of RDBMS software, what approaches are available to address the modern demands to balance transaction processing, business analytics, and compliance concerns?

Database File-Level Archive Method

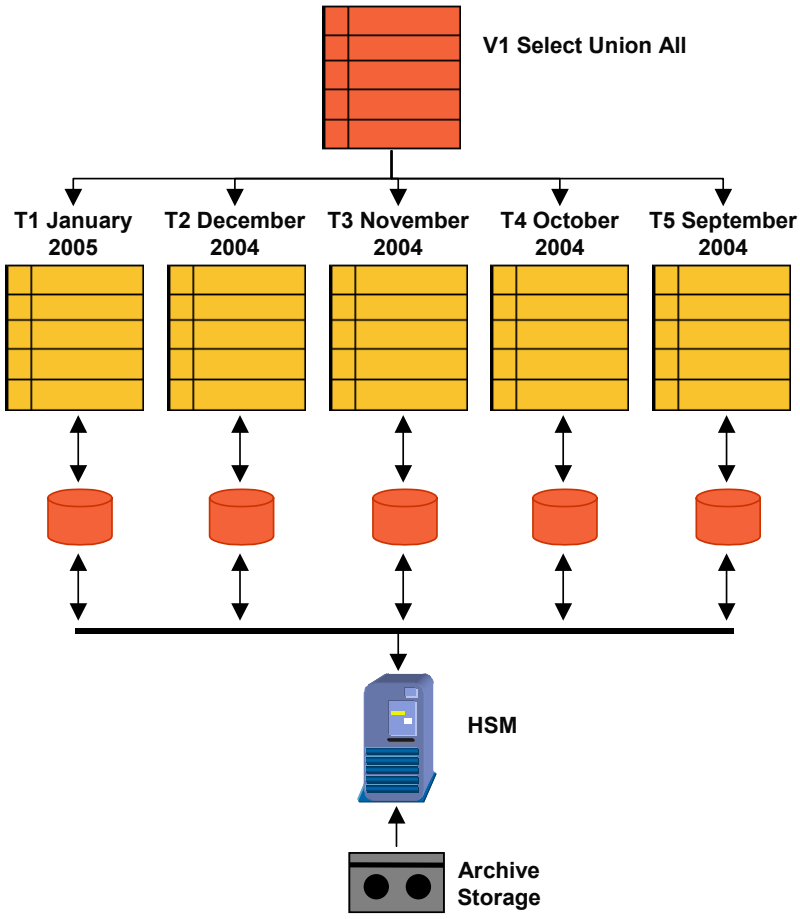
One method is to simply attempt to repurpose basic HSM technology for the database. This approach does require the DBA to essentially redesign the physical database schema so that the user creates a set of individual tables (see Figure 3), each containing a time-based segment of the entire table entity's data. When a time period is no longer useful, the HSM tool can archive the entire contents of the files assigned to that time-based segment of data. Some RDBMS products support actual data partitioning (e.g., Oracle, DB2 z/OS), which would simplify this process somewhat. Other RDBMS products could tie the individual time-based table images together through the use of a logical or partitioned view.

The problem with this approach is that it assumes data can easily be partitioned by time and that time is the only important dimension on which to assess the value of the data.

The impact on the physical design should also not be overlooked. Whenever an application requires a data structure change, that change must be replicated to all copies. The applications must now be modified to look for and update their data in

multiple places. The active data is in one structure, and the inactive data is in a second, third, etc. structure. This approach would require some development effort with the associated costs and risks. This is completely out of the question for purchased applications because the analysis, design, and coding would have to be repeated for each new release. In addition, such changes might not be covered by the application vendor's support agreement.

Figure 3 — Database File-Level Archive Method



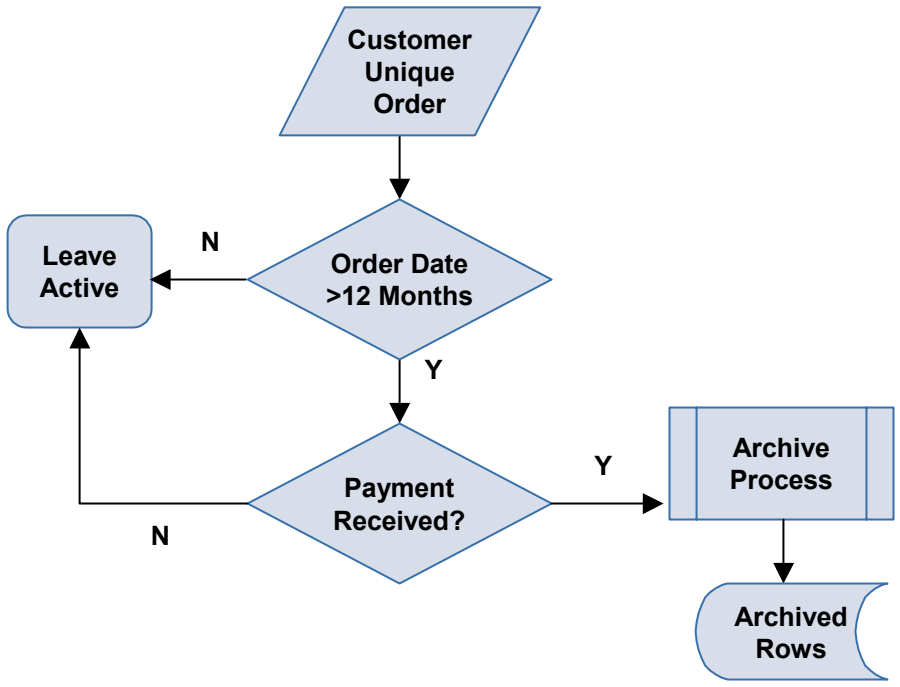
Source: META Group

The benefit of this approach is its rather simplistic approach, which could be implemented without a great deal of business re-engineering to understand the business rules automated within the application code.

Row Archive Method

The optimal approach to gaining value through database ILM is to do so at a row level. Using a multi-tiered, policy-based archiving method, the inactive data row is removed from the database for storage on less expensive or underutilized storage media. User-defined policies can be applied to address the more complex business/data relationship that the previously mentioned method could not accommodate. One example of a policy-driven, multidimensional archive process is illustrated in Figure 4. Rows can be migrated, over time, through multiple tiers of storage onto less expensive media. Initially, the data can be accessed on less expensive DASD, and as access declines further, rows are migrated to less expensive tape.

Figure 4 — Intelligent Row Archiving

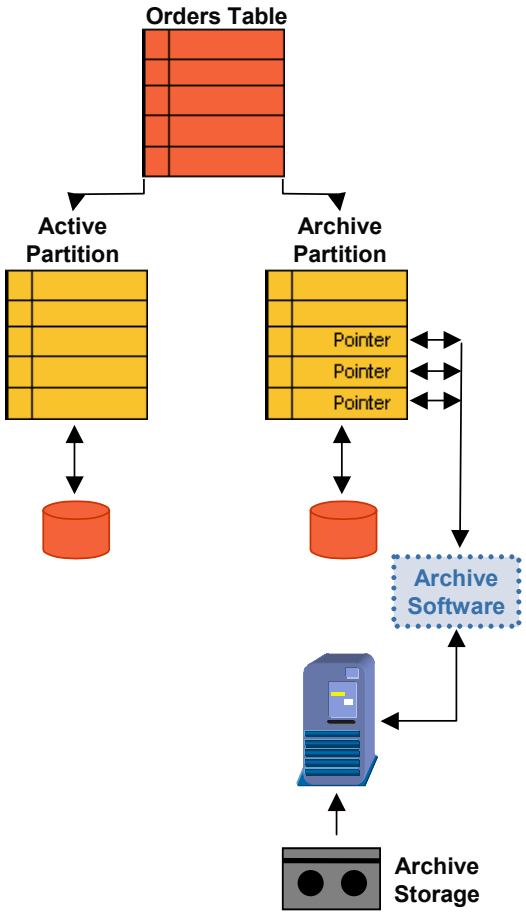


Source: META Group

In some implementations of this method, a “stub” is left in the database as a pointer in an “archive” partition (see Figure 5) to indicate where the actual data from the row is located. When an application requests data from the DBMS, the archive software intercepts the request and determines whether the data is active

or must be retrieved from storage. If the data is still active, the request is handled by the DBMS as always. If the data has been archived, the archive software determines on which tier the row is stored, retrieves the row, and passes it back to the DBMS for processing. Archived data would still be transparently available to application users.

Figure 5 — Direct Row Archive Method

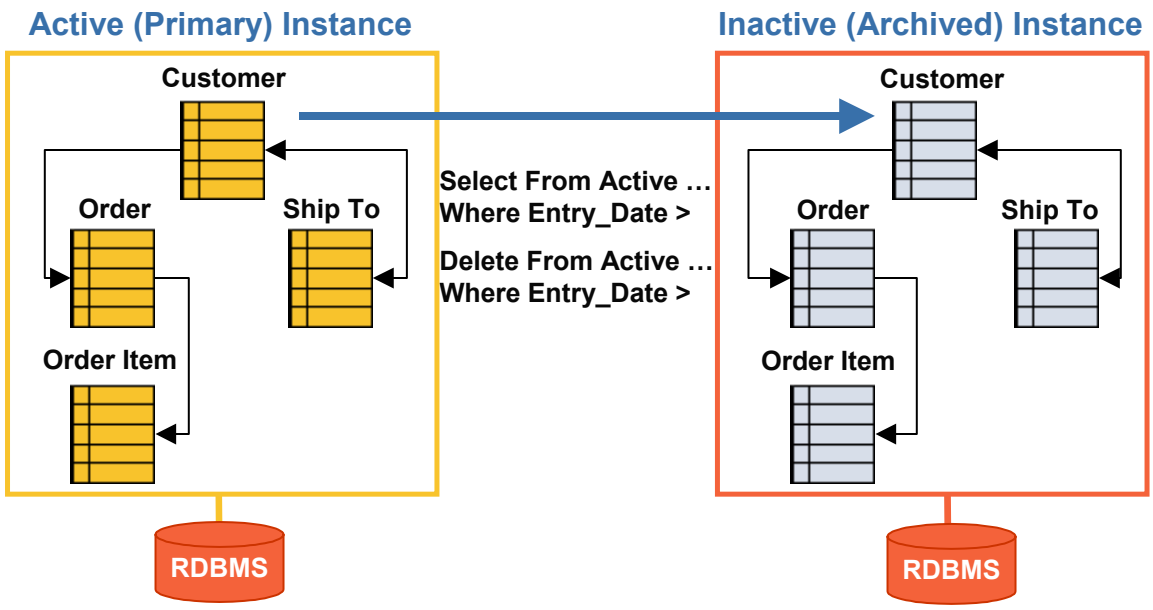


Source: META Group

Since the DBMS and the application are unaware that the row has been archived, all referential integrity (RI) is maintained whether enforced by the DBMS or through application code. This simplifies and greatly reduces the amount of analysis that is required to identify data that is eligible for archiving.

Another approach to implementing the row archive method is to actually remove rows that meet a specified set of archive criteria from the active instance. The rows are placed into a separate yet identical schema, either on another database instance or tables within the same instance (see Figure 6) or on a flat-file structure. Once the rows are in the archive instance, they are deleted from the active instance. The archive instance can use less expensive disk media.

Figure 6 — Active and Inactive Instances: Indirect Row Archive



Source: META Group

Eventually, rows from the archived instance or file can be migrated to cheaper tape or mass storage device. This approach has minimal impact on the database physical design, because the existing physical design is merely replicated for the archive structure.

Although we have discussed two approaches for implementing row archive capability, each approach has its own strengths and weaknesses. The first method (see Figure 5) requires no application changes and only minimal physical database design changes. In addition, it need not be aware of any database or application-enforced referential integrity constraints because the rows are never physically deleted from the active database. Instead, a pointer column is added,

and non-key data is no longer stored in the active database — thereby reducing the size of the active storage required. When archived rows are requested by the database, the archive software intercepts that call and retrieves it from tape or other mass storage device. Because rows are not physically deleted, however, the optimizer will continue to base its access path decisions as if all the rows remained in the database. This could potentially lead the optimizer to choose less than optimal access paths in some cases.

The second approach to row archiving (see Figure 6) not only requires the same planning with respect to archive policies, but also requires the referential integrity rules to be defined to it. In addition, a separate schema must be created and maintained. This method does benefit overall database performance; however, as the rows are moved to the archive instance, they are deleted from the production instance. While this does not save storage in the production instance (unlike the first approach), it does enable the optimizer to evaluate only active data (unless archive data is also requested), meaning more accurate access path decisions.

It takes a great deal of development time to determine what an application's business rules and data relationships are and then to codify those rules into database archiving software. The cost to do this for every application would be significant. To address that need, the marketplace has responded with application-specific database ILM solutions — currently limited to the better-known, large ERP applications. In these instances, the software vendors have done the work of understanding the application's schema and its business rules and have incorporated it into the product to significantly reduce the time to value for database ILM.

The benefits of the row archive method (either approach) are the enforcement of multidimensional business policy rules for archiving. In addition, it requires no (or minimal) application code changes, thereby reducing the implementation costs and risks. This method does, however, require more planning, especially in determining what the proper archive policies will be and then implementing and testing them within the chosen ILM software.

META Group's research estimates as much as 60% of the typical application's data has little or no access after 60 days. After one year, the data in a typical application has no access of any kind. A leaner active (i.e., production) database also means faster backups, table re-organizations, and recovery times. Space is reduced on the active partitions, so image copies and reorgs can be directed against the active partition only, which improves overall availability.

Production databases are replicated an average of seven times for a variety of reasons (e.g., QA, read-only reporting, testing). The smaller the production database, the smaller the secondary environments. All of this directly affects the CAGR of storage-related budgets.

Bottom Line

The impact of unchecked storage growth will have a significant impact on IT budgets regardless of the downward per-unit pricing trend. Although the concepts heralded by ILM are very valid, practical implementation is years away. Users should look to capitalize on near-term opportunities that can deliver the opportunities promised by ILM. Database environments are a great starting point because they are among the most expensive infrastructure pieces to maintain. By reducing storage growth within the database, the downstream impact on performance, availability, and recovery should easily justify the solution costs.

Charles Garry is a vice president and director in META Group's Technology Research Services organization. For additional information on this topic or other META Group offerings, contact info@metagroup.com.

About META Group

Return On Intelligence[SM]

META Group is a leading provider of information technology research, advisory services, and strategic consulting. Delivering objective and actionable guidance, META Group's experienced analysts and consultants are trusted advisors to IT and business executives around the world. Our unique collaborative models and dedicated customer service help clients be more efficient, effective, and timely in their use of IT to achieve their business goals. Visit metagroup.com for more details on our high-value approach.