Solution White Paper

A New Approach to Managing Unstructured Data

Manage unstructured data stored in IBM® DB2® LOBs for optimal data integrity

By **Craig S. Mullins**
President of Mullins Consulting, Inc.

Table of Contents

Executive Summary

As larger amounts and types of data are being used more commonly in business systems, relational database systems like IBM® DB2® are being used to store more unstructured data. This requires different data types called "large objects" (LOBs). However, LOBs cannot be managed the same way as traditional data. The old utilities and tools that have been in use for years do not work the same way for LOBs and it is easy to miss steps and cause data integrity problems with LOB data. Therefore, organizations are adopting complicated processes, or worse yet, doing nothing to manage LOBs.

Failure to adopt new policies and procedures when managing unstructured data can result in missing or inaccurate data. And the last thing you want to be faced with is bad data when you need it most. Instead of trying to use tools and processes that are not designed for this new type of data, **a better approach is to adopt modern tools that are built to work with unstructured data and DB2 LOBs, such as the automation provided with BMC's Next Generation Technology (NGT) and LOBMaster.**

Modern data and database infrastructure requires more than the ability to handle traditional data such as text and numbers. Today's applications and businesses rely on complex, unstructured data more than ever. Read on to learn more about the importance of unstructured data, how you can manage it in DB2, and how that impacts your database administration procedures.

UNSTRUCTURED DATA

The amount of data we generate is doubling in size every two years, with the total amount of data expected to reach 44 zettabytes by 2020 with 80 percent of that data being unstructured.[1]

2x
The amount of data generated every two years

44
zettabytes
Total amount of data expected by 2020

80%
percent of that data is unstructured

Unstructured data is a catch-all term used to describe any data that is not rigidly structured. This includes data that does not fit into a simple data type, like numbers, characters, dates, and time. Examples of unstructured data include contracts, policies, videos, MRI images, emails, application log files, customer support logs, and photos. In general, unstructured data is becoming more important as organizations find new ways to derive value from analytics. Furthermore, industry and governmental regulations stipulate how organizations must manage and maintain data, driving more focus on the data management needs of unstructured data.

Main Issues of Unstructured Data

- Cannot be easily organized
- Does not have a pre-defined data model
- Requires internal pointers and structures to implement
- Can be large and hard to manage

A new data type, LOB, short for “large object,” was created to bring this unstructured data into DB2, a very structured relational database management system, making the data more readily accessible when needed.

Although DB2 for z/OS users were slow to adopt LOBs in mainframe databases, regulatory requirements and the widespread adoption of big data have contributed to increased usage of LOBs. **Big data and analytics are driving organizations to accumulate and analyze more data, and more varied types of data, to gain better business insight**.

By storing unstructured data in DB2 LOBs, the data can be managed and administered along with your other mission-critical data—taking advantage of the capabilities of DB2, such as backup and recovery, database logging, and other built-in data management features of a DBMS that are not available if the unstructured data is stored elsewhere.

While there are many benefits to storing unstructured data in DB2, you must be willing to expand and change your data management processes because things won’t work exactly the same for LOBs as they do for traditional DB2 data.

[1] John Gantz and David Reinsel, The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East, IDC Research, December 2012.

DATA MANAGEMENT CHALLENGES

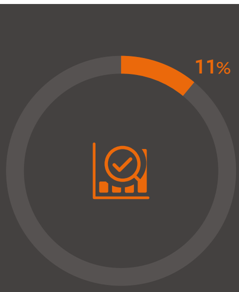
DBAs are charged with managing and protecting the data in production databases, but there are many issues being faced by modern applications that make it difficult for organizations to fully embrace new challenges like managing LOBs. Many DBAs don't know what's stored in their LOBs, how the data is stored in DB2, and how to properly manage them.

Many organizations do not know the extent to which LOBs are being used in their DB2 databases and applications. It is not uncommon for a DBA to believe that LOBs are a non-issue at their site, thinking that there are no LOBs in the production databases or at least that no sounding alarms of data issues are evidence of data integrity. But this is often a wrong assumption. I have talked to IT professionals who believe that LOBs are not an issue, but when a query is run on the DB2 Catalog looking for LOB columns, it is typical to find anywhere from a handful of LOBs to hundreds of them. Further inspection often reveals issues with the data integrity due to the complexity of the structure of the LOB.

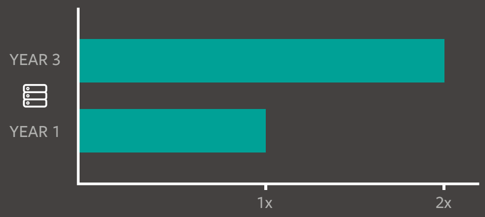
Another wrong assumption made is that LOBs are not accessed often enough to require active management. When LOBs exist in DB2 databases, it is important that they are managed and checked for integrity on a regular basis. Failure to do so will likely result in data integrity problems or lost data.

So how do problems like these and misinformation become so commonplace?

DBAs are dealing with managing extraordinary growth in the amount of data being stored (big data). The Bureau of Labor Statistics projects the growth rate in the number of DBAs at 11 percent between 2014 and 2024, but that is still less than 2 percent per year.[2] Contrast that with the amount of data doubling every two years and it is clear that there is a growing management problem. DBAs are challenged to support and administer their existing and new databases with fewer resources.



Projected growth rate in the number of DBAs between 2014 and 2024



Data is doubling every two years, but the growth rate in the number of DBAs cannot keep up

[2] Source: Bureau of Labor Statistics, https://www.bls.gov/ooh/computer-and-information-technology/database-administrators.htm

Compounding the issue is the rapid change in data and applications. **Agile development and DevOps practices diminish the oversight role that DBAs formerly exercised over production databases**. Sometimes multiple production changes are made to the same application and database during a single week. Some application and database changes can even get implemented without proper oversight or the requisite knowledge of the environment. When fewer IT personnel are involved in change management, institutional knowledge declines and problems arise.

An additional issue is a lack of education. While simply trying to maintain control of their environment, **many DBAs do not get adequate training that is necessary to ensure new structures' integrity and viability**. Without an understanding of how to manage them, DBAs either ignore LOBs or struggle to implement them using their existing knowledge and tactics, which are not applicable to these new data types.

The result is that organizations apply old, outdated techniques to the management of new, complex data—which simply will not work. Furthermore, it can place the organization in a precarious compliance or legal situation if not remedied. The next section offers a high-level examination of the technical challenges.

DB2 LOBS ARE DIFFERENT THAN MOST TYPES OF DB2 DATA

DB2 for z/OS supports unstructured data using three different data types: BLOB, CLOB, and DBCLOB—collectively known as LOBs. Together the various types of LOBs are used to store binary and character documents, thereby enabling DB2 to store and manage things like large text documents, photos, audio, and video.

LOBs are stored differently than traditional data and therefore must be managed differently. Just like other data, the LOB data must be defined for the table that will store it. But DB2 stores the actual LOB data in a different data set than the other data in the table.

Many DBAs assume that simply reorganizing LOB table spaces is sufficient for managing their LOB data. But this is not true. As with traditional DB2 table spaces and indexes, the REORG utility can be used to reclaim fragmented space and improve access performance. But this is only a portion of the management battle. To keep the LOB data aligned with the rest of the data, DB2 deploys pointers that connect the two data sets. As with any software that uses pointers, data integrity issues can arise when the pointers get corrupted.

There are many different ways that LOB data can get out of sync in your DB2 environment beyond just pointer corruption. Although an in-depth discussion of all of these issues is beyond the scope of this paper, a brief discussion of the other types of errors that can result is within the scope.

LOB Index Integrity Problems

Four types of integrity problems are possible with LOB indexes:

1. The ROWID-version number in the base table row may not be found in the LOB index.
2. There may be entries in the LOB index that are not referenced by any row in the base table.
3. The LOB data itself may not be where the LOB index points to.
4. There may be LOBs in the LOB table space that are not referenced by the LOB index.

CHECK DATA, CHECK LOB, and CHECK INDEX can be used to find and rectify these errors. But doing so in the wrong order can convert one type of error into another.

An index is required by DB2 to manage the LOB data separately from the other data in a base table. It is possible that the base table details are not found in the LOB index, causing problems when trying to access the LOB data. It is also possible that the LOB index contains entries that are not referenced by any row in the base table, which will cause DB2 to throw an error condition. Furthermore, the LOB data may not exist in the location that the LOB index points to, and there may be actual LOBs that are not referenced by the LOB index.

The size of LOBs makes them more difficult to manage as well. A single LOB can span multiple pages of physical storage. These pages must be managed by DB2, and problems can arise with page sequencing and how the LOB gets “put back together” when accessed.

The result is that these out-of-sync conditions can have dire consequences on your ability to access the LOB data. Depending upon your backup and recovery procedures and the frequency that you check the integrity of your LOB data, the consequences can range from simple intermittent errors to losing the LOB data entirely.

To ensure that problems do not arise requires an alert, trained, and responsive DBA staff armed with a modern toolset for managing LOBs. The DBAs must understand when and how LOBs are introduced, as well as how they work, to ensure that an appropriate plan is put in place. This means documenting LOBs as they are introduced, what the LOBs are used for, and a policy for managing those LOBs that includes backup, recovery, and integrity checking.

Without modern tools to assist the DBA, this requires far too much manual effort and additional time and resources. The DBAs are required to add additional processes to their already taxed procedures, including CHECK DATA, CHECK LOB, and CHECK INDEX. If problems are found, other utilities, including REPAIR and REBUILD INDEX, are required to fix the problems on a case-by-case basis—if they can be fixed. Complicating matters even more, these utilities must be run in the proper order and at the proper times to find and correct problems or you risk converting one type of problem into another.

Given the immense amount of work and increasing volume of data, there is simply not enough time in the day to work through this amount of complexity. **Organizations relying on LOB data in production databases should invest in modern tools, such as BMC’s Next Generation Technology and LOBMaster, that are designed specifically to manage unstructured LOB data safely, effectively, and automatically**.

DIFFERENCES POSE CHALLENGES

The challenges outlined in the previous section are not the only ones that are encountered as DBAs adopt LOBs to store unstructured data in their DB2 databases. Remember that LOB data is typically larger than traditional data; a single LOB can be up to 2GB—and that is just one column. There are nuances in the way that the normal DB2 utilities work with LOBs. They can work differently and can require different parameters rendering them more difficult to master and implement for LOB data. A recent LOB user reported that a reorg of a single LOB required more than two days to complete.

What Should You Do?

Simply implementing LOBs without implementing a strategy to support and maintain them is a recipe for a legal and compliance nightmare. This can range anywhere from experiencing performance issues to losing data that cannot be recovered.

So what can be done? The answer is *preparation*.

The first step is to **accept that old approaches will not work on LOBs**. DBAs must be trained to think differently about unstructured data, which requires education on LOBs and how to manage them. Make sure that your DBAs are properly educated on how LOBs work in DB2. Although LOBs have been supported in DB2 since version 6, many changes have been made that alter the way you might implement, access, and manage LOBs in your DB2 databases and applications. Without the proper knowledge of how LOBs work, there is no hope that the critical data stored in your DB2 LOBs will be accurate and correct when you need it.

Once DBAs have been trained, organizations should **put oversight processes and procedures in place that permit them to review systems with LOBs before they are rushed into production**. By reviewing and documenting all new LOBs early in the development process, your organization will not be blindsided, not knowing which applications require LOBs and for what purpose. Furthermore, it will give the DBA team time to prepare appropriate management procedures for all LOBs.

Resist the urge to push through production changes without proper oversight. Reviewing structural changes, such as those required to support LOBs, should not be treated lightly and may elongate project delivery timelines. But remember that failure to properly review and implement LOBs can result in lost data, which could mean fines or worse consequences, depending on what data is lost.

Implement regular integrity checking for all of your production databases with LOBs to ensure the data is sound—and that you quickly find any problems, making them easier to correct. A recent survey of LOB users revealed that only 18 percent of respondents believed they would ever be asked to prove data integrity on their unstructured data. Relying on the hope that no one will ask is simply not a viable solution for a digital business. However, this may require a labyrinth of utility jobs and it can be unrealistic to coordinate all the steps required for all your LOBs unless you rely on new, modern tools that understand how to manage and maintain LOBs.

Finally, automation is an important aspect of managing today's complex DB2 environment. **Adopting tools and processes that minimize manual involvement and intervention makes it easier and less error-prone for organizations to manage their DB2 environment, including unstructured data in LOB columns**. BMC's Next Generation Technology exploits automation to maintain data integrity and manage a modern DB2 implementation.

CONCLUSION

Using LOBs to store unstructured data in your DB2 databases is rapidly becoming more common. As organizations acquire and store more types of data, to be used for more analytical and operational purposes, every organization will have multiple LOBs that need to be properly created, managed, and maintained.

The administrative practices and procedures that have been used for years on traditional data will not work on unstructured data stored in LOBs. **Organizations should look for modern toolsets that have been designed to work with LOBs from the outset, such as Next Generation Technology and LOBMaster from BMC**. Failure to do so will result in data integrity issues and missing data—which no organization can afford.

FOR MORE INFORMATION

To learn more about data management solutions for DB2 on z/OS, please visit **bmc.com/it-solutions/database-management-db2-zos**

For Further Reading

1 “LOBs with DB2 for z/OS: Stronger and Faster,” SG24-7270, http://www.redbooks.ibm.com/redbooks/pdfs/sg247270.pdf

2 “DB2 10 for z/OS Technical Overview,” SG24-7892, http://www.redbooks.ibm.com/redbooks/pdfs/sg247892.pdf

3 “DB2 for z/OS LOBs Experiences & Best Practices,” 2013 Presentation by Haakon Roberts, IBM

4 “Large Objects in DB2 for z/OS: You better get used to them,” 2013 GSE Presentation by Francis Desiron, IBM

5 DB2 Developer’s Guide, 6th Edition, IBM Press, Craig S. Mullins

ABOUT THE AUTHOR

Craig S. Mullins is a data management strategist, researcher, and consultant. He is president and principal consultant of Mullins Consulting, Inc. and has been named by IBM as a DB1 Gold Consultant and an IBM Champion for Analytics. Craig has over three decades of experience in all facets of database systems development and has worked with DB2 since V1. You may know Craig from his popular books:

- DB2 Developer’s Guide, 6th Edition – Containing more than 1,500 pages of in-depth technical information on DB2 for z/OS
- Database Administration: The Complete Guide to DBA Practices and Procedures, 2nd Edition – The industry’s only comprehensive guide to heterogeneous database administration