

DB2 LOBS: UNSTRUCTURED DATA MANAGEMENT



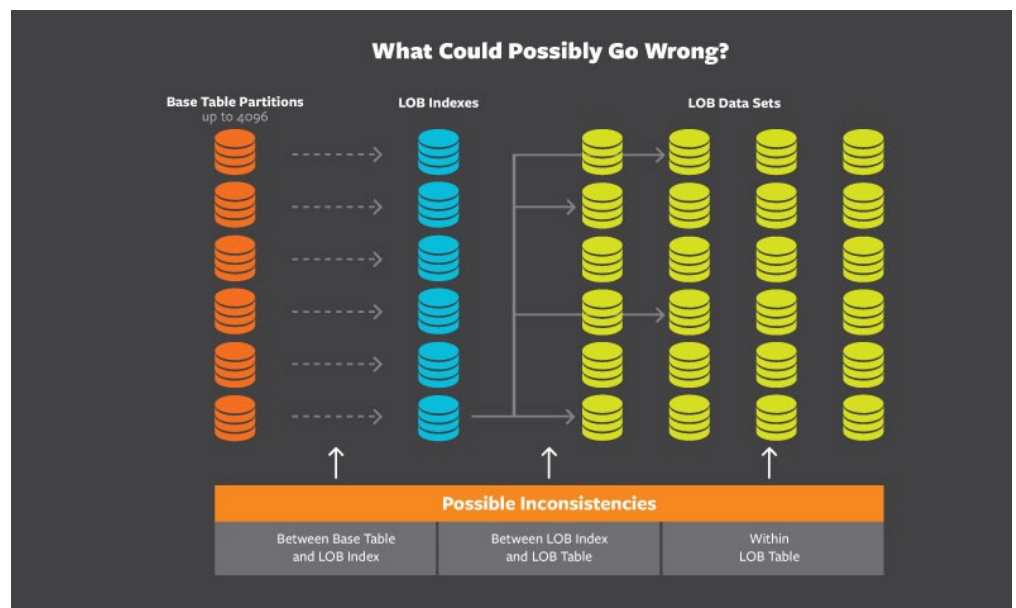
We all know that IBM DB2 has been around for a long time and was originally designed to house structured data, data that fits neatly in individual fields (e.g. name, address, phone number). In DB2

V6, in response to object-oriented databases, IBM then introduced LOBs (Large Objects) as containers to hold unstructured data.

For more than a decade, LOBs had been sporadically used throughout the industry as needed. With v10 however, LOBs are part of the catalog and available for all. It's a good thing too because [digital transformation](#) is driving explosive growth of unstructured data. According to IDC, [unstructured data now makes up 90 percent of digital data!](#)

Unfortunately, the implementation of DB2 LOBs is very complex with pointers between base tablespaces and LOBs, Indexes on LOBs, pointers between LOB indexes and LOB tables, and one LOB per LOB Column per partition! That means just a couple of LOBs could result in thousands of datasets and even more pointers to be concerned with in your unstructured data management process.

So what can go wrong? If normal DB2 indexes can be inconsistent with their associated Table (and hence the requirement for [CHECK INDEX](#) and [REBUILD INDEX](#)), the issues are multiplied for LOBs. There are a lot of pointers in LOBs and to prevent a disaster in your environment these pointers should be checked regularly.



DB2 LOB issues can be categorized in four groups:

1. The ROWID-Version number in the Base Table row may not be found in the LOB IX.
2. There may be entries in the LOB IX that are not referenced by any row in the Base Table.
3. The LOB data itself may not be where the LOB IX points.
4. There may be LOBs in the LOB TS that are not referenced by the LOB IX.

These groups ignore any structural problems in the LOB TS itself. Those show up when DB2 tries to process the LOB data. The most common abend is 00C90101.

To mitigate all these risks, there are three utilities that should be run regularly: [CHECK INDEX](#), [CHECK DATA](#), and [CHECK LOB](#). Unfortunately DBAs rarely, if ever, run these checks because of lack of resources and complexity. The common practice in DB2 data management is to wait for the phone to ring notifying you of a problem, and then run checks and diagnostics. With LOBs, that practice is a major risk to your business because LOBs are not regularly accessed. They work more like a repository for data that is rarely accessed. By the time an application finds a problem, your copies will be rolled off and your data will be unrecoverable! Often, LOBs house data held for legal or compliance reasons. Failure to regularly validate LOBs can create a real liability for your company.

According to IBM, "LOBs are different animals" so you have pay special attention to them. Taming these animals (i.e. this data) doesn't have to be hard, you just have to have the [right technology](#) to do

it!