



Craig S. Mullins

June / July 2005

[Return to Home Page](#)



zData Perspectives

by Craig S. Mullins

Twenty-Two Years of DB2

DB2 has been around for more than two decades now and it is amazing to consider how it has grown from its humble beginnings. Back in 1983, when DB2 was first unleashed on the world, IMS and IDMS were the ruling DBMS products and the mainframe was the undisputed platform for mission critical data processing work. A lot has changed since then.

First of all, the mainframe's position in the data center is no longer undisputed. The client/server wave of the 1980s and the web-centered development of the 1990s eroded the mainframe's base. Oh, don't get me wrong, many still understand that there is no better platform for delivering high

availability, unparalleled performance, and strong security than the mainframe – but these days a data center using only an IBM mainframe is rare indeed.

Appropriately enough, IBM responded to the changing landscape and adapted DB2 to multiple platforms. In addition to z/OS, DB2 now runs on Linux, Unix, Windows, VM, VSE, and OS/400, as well as the Palm and Windows CE PDA platforms. DB2 is essentially available everywhere!

Getting back to the history, IBM released DB2 for MVS in 1983, a year after releasing SQL/DS for VSE and VM. Before its release, IBM's internal code name for DB2 was Eagle. Like an eagle, DB2 has soared to great heights adding numerous features that we now take for granted.

During its V1 incarnation IBM released three point releases of DB2. In V1.2, IBM added the EXPLAIN command, a basic tuning tool that we all take for granted these days. Can you imagine trying to tune SQL without EXPLAIN capability? And V1.3 added temporal DATE and TIME data types. Being able to manage temporal data using native data types

greatly minimized the amount of code that we needed to write.

In 1987 IBM released OS/2 Extended Edition Database Manager: this is the code that eventually became DB2 for Linux, Unix, and Windows.

Next came V2 with its three point releases. Referential integrity bowed in V2.1. Before this the only way to assure data integrity was by coding it into application programs. Additionally, distributed database support and DRDA were introduced during the V2 timeframe, opening DB2 up across the network. Packages were introduced during this timeframe, too. Prior to that every DBRM had to be bound into a plan – and that made administering an online environment like CICS very difficult.

Somewhere during the time of V2 and V3, DB2 became accepted as a production OLTP DBMS. Prior to that, many IT professionals called it an Information Center DBMS. That was what we called data warehousing type work back then. Interestingly enough, today there are those that say DB2 is only good for OLTP, and not for the analytical processing

required of data warehousing and business intelligence. Of course, that isn't true.

DB2 V4 upped the availability game by introducing data sharing and Type 2 indexes. With the robust failover of data sharing and the removal of index locking as a problem, DB2 truly delivered performance and availability that was unparalleled in the market.

Right around the same time IBM re-branded all of its relational database products as DB2 and started supporting the Windows platform. Then came Data Joiner, allowing distributed joins across heterogeneous databases. Indeed, some have reported achieving better performance accessing Oracle through Data Joiner than by accessing Oracle through its native interfaces.

Now we're up to about 1996 and DB2 V5. Here is where IBM added help for those of us implementing VLDBs, including: support for table spaces up to a terabyte, online reorganization, statistics sampling, and more.

Putting procedural logic into the database became possible during the V4 through V6 span as DB2 added support for stored procedures, triggers, and user-defined functions. Using these facilities, programmers can create intricate, high performance applications using DB2 server-side logic. Such feats of programming acumen were un-thought of back in the prehistoric days of DB2 V1.

Not resting on their laurels IBM continued to stoke the fires of DB2 progress. Adding support for Java and XML, caching dynamic SQL, and extenders for image, audio, and other multimedia data ensured DB2 continued relevance as a modern development platform. With V8, DB2 has grown even further. IBM claims that there are more *new* lines of code in this latest and greatest release of DB2 than there were *total* in V1. With features like virtual storage constraint relief, long name support, online schema change, and multilevel security IBM has ensured that our favorite DBMS will be able to meet our needs today and into the future.

What is next? Well, we can be sure that each new version of DB2 will continue to add support for the

features we need to meet our ever-changing business requirements. IBM's on-going "On Demand" initiative will make DB2 easier to manage. But more features mean more complexity – and we'll all have to be prepared to meet the challenge of mastering this ever-expanding and growing thing we call DB2. I'm excited about the future of DB2 – and you should be, too!

From [zJournal](#), June / July 2005

© 2005 Craig S. Mullins, All rights reserved.

[Home](#).