



First Look at DB2 9.7

New features improve performance and compatibility and are easy to implement

[Dave Beulke](#), Contributing writer, IBM Data Management magazine

Summary: A look at the new features in DB2 9.7 that improve performance and compatibility and are easy to implement

Date: 30 Jun 2009

Level: Introductory

Also available in: [Chinese](#) [Portuguese](#) [Spanish](#)

Activity: 1761 views

Comments: 0 ([Add comments](#))

★★★★★ Average rating (based on 8 votes)

The new release of IBM DB2 9.7 for Linux, UNIX, and Windows offers many advantages for better performance, better maintenance, and lower total cost of ownership, especially for large and robust databases. But after exploring the new version myself, what strikes me most about the features is how easy they are to implement.

The first feature that makes this release really stand out is index compression, which reduces both I/O and the size of the index structure. Through this new compression technology, DB2 is capable of achieving compression ratios as high as 50 percent or more. This index size reduction has big implications for dramatically reducing disk requirements for terabyte data warehouse implementations and for reducing overall I/O and response time for transaction systems. With this feature, DB2 is keeping well ahead of the competition.

The next new feature within DB2 is an improvement to its transaction concurrency model. This concurrency enhancement provides more granularity for the transactions. Within some high-performance systems, one transaction writing data and another transaction reading data can be blocked or delayed before the writer transaction is committed. This delay or blocking situation can cause some high-performance systems issues, so the DB2 team developed a new isolation level called Currently Committed that is now the default within the system.

The new Currently Committed isolation level allows writer transactions to continue processing while the other reader transactions within the system get their data. It works by retrieving the data row information that the reader transaction needs from the DB2 logs-usually from the log's memory cache, which is very fast. As a result, the reader transaction can continue processing without waiting for the writer transaction to be committed. This functionality provides more robust concurrency and implements it in a way that does not cause extra overhead for the

system or transaction processing.

New releases of DB2 almost always include SQL improvements, and version 9.7 has many SQL-related features that improve functionality, compatibility, and overall performance. Your application developers will like the new scalar functions, especially the date and time functions that will provide performance boosts for processing. There are too many to mention here, but your development group should immediately evaluate the extensive number of new SQL functionalities.

Next, a number of new phrases were added to the DB2 lexicon to make it more compatible with Oracle SQL syntax and PL/SQL. The improvements should enable you to migrate Oracle applications to DB2 in days rather than months. Plus, the ability to use the same SQL syntax will allow your development team to quickly shift to the DB2 platform with its self-tuning and automated memory management.

These are only some of the DB2 9.7 features; more information can be found at IBM.com. Many of these features were hot topics at the International DB2 User Group conference in Denver, held in May 2009. The conference included many great presentations by beta customers, consultants, and IBMers, talking through the internals as well as implementation and performance details. If you weren't able to attend, look at the presentation grid at IDUG.org and buy the conference CD.

About the author

David Beulke is former president of IDUG (www.idug.org) and a DB2 Data Champion. He has more than 22 years experience in architecture, design, and development of high-performance DB2 data warehouse and OLTP systems across all platforms.

[Trademarks](#) | [My developerWorks terms and conditions](#)