

IBM **Analyze+** for DB2 z/OS

Key Benefits

- *Recognize quality defects in SQL programming early in development*
- *Hand-over high performing applications to production*
- *Monitor production applications to identify performance problems*
- *Identify problems in the dynamic SQL of ERP packages*
- *Analyze dynamic statement cache*
- *Automate the DBA to eliminate otherwise manual work*
- *Exploit DB2 V9, e.g., evaluate XML*
- *Compare access paths*

Optimizing SQL Performance

IBM **Analyze** automates analysis and tuning of SQL, making the effort easy and efficient. Well-tuned and high-performing SQL statements equate to efficient CPU usage and improved response times. IBM **Analyze** provides automated and intelligent analysis and tuning of SQL statements, improving the overall performance of production applications. IBM **Analyze** eliminates many of the performance problems typically found in a production environment, by creating a simulated production environment for application testing. It detects poor performance early in the development process, saving valuable technical and CPU resources and associated cost. With IBM **Analyze**, your technical staff is empowered to work smarter—saving time, increasing productivity and maximizing the use of the shrinking batch window.

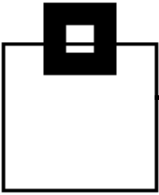
IBM **Analyze** fully exploits V8 and V9 of DB2 for z/OS; thereby providing you new capabilities to optimize your SQL workload. Its new technology internally uses V8 features like multi-row fetch and the new Explain tables, making it a milestone in speed, functionality, and usability.

It is a well-known fact that quality assurance – namely for performance reasons – saves \$\$\$ if performed as early as possible. In other words, this means QA is best when performed during the development process. IBM **Analyze+** achieves this goal by supplying application programmers with a functionality that checks for defects concerning performance or programming rules and gives them hints and tips to improve their SQL statements. IBM **Analyze+** makes this possible in a flexible way: interactively while coding SQL, during the compile procedure, or on the basis of programs already compiled.

To get the most out of automation, it is essential to tune production applications as well. As opposed to classical statement tuning that is addressed by the majority of well-known EXPLAIN tools that analyze SQL statements on a stand-alone basis, extended application tuning additionally reflects the actual usage of application systems by monitoring them in production. IBM **Analyze+** allows concentration on the worst SQL statements, i.e., highest resource consuming, and avoids the time spent analyzing SQL statements that have insignificant performance impact.

Another challenge in production is evaluation of access paths for packages and plans. With IBM **Analyze+**, you can evaluate access paths and do historical comparisons of them to see the results of binds and rebinds. This feature is supported by an interface IBM **ImpactExpert**.

Whether you are coding, handing over an application to QA, or tuning SQL already running in production, IBM **Analyze+** is the SQL optimization tool for you.



Technical Highlights

SQL Problems Solved:

- *Inefficient use of CPU resulting from inefficient coding of SQL applications and queries*

- *Poor response times resulting from inefficient coding of SQL queries*

- *Non-existing or ignored standards and rules for coding SQL applications and queries*

- *Poor access path selection in the production environment resulting from testing with incorrect/inconsistent catalog statistics in the test environment*

- *The EXPLAIN(YES) option isn't supported for triggers, but Analyze handles this*

V8 and V9 rule system—Best practices for coding SQL statements are delivered with optimization recommendations for V8 and V9 SQL features. Adapting these rules are easy so you can customize rule sets for programmers that are different than those for DBAs, for example.

Extended ISPF edit macro—An extended ISPF edit macro allows you to edit a statement, execute it, and Explain it. This easy to use macro has its own analyze, execute, and setup commands. The slick online interface presents the Explain results, rule violations, and recommendations exactly the same as when analyzing from the tool's interface. Use this extended ISPF edit macro to tune statements from directly within your application program while coding.

Dynamic Statement Cache capture—You can easily capture and filter the Dynamic Statement Cache online or using a batch job, which also allows aggregates like statements.

Single-point of control via DRDA connectivity—You can install **Analyze** on one system. Using the DRDA option, you can define the TCPIP address, run the bind, and test the DRDA connection. Then, on supporting panels, just type in the subsystem you want to switch to. Using DRDA connectivity, you must apply a PTF on the base subsystem only.

Global Explain Table Pool—Drastically speeds up dynamic Explains. Old concepts support one PLAN_TABLE per user for dynamic Explains, which is slow and the tables must be maintained and cleaned up. **Analyze** creates a pool of Explain tables at installation and automatically looks for a free table in the pool when a dynamic Explain is performed. **Analyze** also supports individual user PLAN_TABLEs, but the advantages of using the pool guarantee speed and eliminate the administration of user tables.

Supports statistics health checks—Included in the rule violations are statistics health checks. The foundation of the extensive rule system is based on IBM's own recommendations for good statistics. The statistics health check precisely pinpoints which objects would benefit from a RUNSTATS or those that otherwise require statistics housekeeping. When you do the Explain on a statement, the statistics health check is automatically performed.

Provides predicate reports—For each SQL statements, you can view all information pertaining to all predicates, including stage, type, filter factor, etc. The complete predicate text can be seen using the zoom command.

Extended Catalog Browser—Presents everything in the catalog related to an object including catalog history if you use RUNSTATS HISTORY option