



DB2 SQL Performance Analyzer Version 2.1

Functions and Features Overview

What does SQL Performance Analyzer do ?

- **Forecasts SQL performance:**
 - Response Times
 - CPU Times
 - I/O Counts
- **Presents query costs in terms of:**
 - Chargeback (monetary units)
 - Qunits (query CPU service units)

Slide - What does SQL Performance Analyzer do ?

SQL Performance Analyzer provides realistic cost estimates for DB2 queries, such as Select, Insert, Update, Delete, Join, Subquery, or Union.

It runs from any attach facility, for example, CICS, CAF, IMS, or RRS, using its Stored Procedure, and also operates in Batch, TSO and QMF modes.

It returns information in the form of Response Time, CPU, I/O, an internal value called Qunits (like service units), and monetary costs. The monetary value gives you an idea of what it costs each time a particular query is executed.

SQL Performance Analyzer is a blend of optimizer and database modeling technologies. If you specify the PRECISE YES parameter, SQL Performance Analyzer uses the optimizer's cost estimate to determine the CPU and Service Unit cost. With PRECISE NO, SQL/PA maintains sets of internal path lengths from benchmarks of each major release of DB2, measuring the various activities: Sync Read, Prefetch, Fetch Rows and so forth. From these, SQL/PA constructs an internal model of each SQL statement, using the optimizer's Filter Factors, if available. This blend of modeling and optimizer technology makes SQL Performance Analyzer a unique product.

What can SQL Performance Analyzer do for you?

- **Provide expert advice on how to improve your SQL**
- **Warn of long running queries**
- **Preempt costly QMF Governor cancels**
- **Illustrate the incremental components of cost**
- **Help design new DB2 queries**
- **Help tune production SQL via DBRM scans**
- **Evaluate future production volume performance**
- **Implement governing for any DB2 application**

Slide - What can SQL Performance Analyzer do for you? (1 of 3)

The SQL Advisor is a key feature of SQL Performance Analyzer. Embedded within the SQL/PA report structure, the SQL Advisor helps you write better, more efficient SQL.

The QTRACE report informs you where DB2 is spending its processing time: attach overhead, stage one, stage two, fetch rows, compression, log writes, and so forth. It lets you know if you are I/O or CPU bound.

The Advisor presents additional, practical observations, warnings and recommendations at three different user levels -- beginner, intermediate, and expert. The information it returns depends on the level you select with the ADVISOR NO|YES|ALL parm. It assumes you know the level of information you want.

It may inform you that your query is getting ready to switch from an index lookup to a table space scan. If you have a large table, you could use this knowledge to run a RUNSTATS. The Advisor can give you hints about restructuring a query for better results. It issues warnings and alerts to make you aware of special situations. It provides SQL statement and release-sensitive information, notes and recommendations, as well as guidelines and good news for the novice user.

Slide - What can SQL Performance Analyzer do for you? (2 of 3)

SQL Performance Analyzer is a great way to teach new application developers the proper way to code. The tool is equally useful at designing new applications using SQL, or at tuning existing production queries.

The SQL Performance Analyzer does far more than answer "What is the cost of the SQL statement?". While it does answer that question very well, the visualization of that answer and the analysis it provides, extend its use beyond simple costing. This slide illustrates some of the many problems that can be solved using SQL Performance Analyzer.

For one, the tool is very good at designing new applications with new SQL. You can build a prototype with it. In a TSO session, you can start with a query, get your cost, make modifications, and rerun it. The second run of SQL Performance Analyzer appends to the results file (ANLCOST.LOG). This makes looking at the differences very easy. You can see what you changed and the new results, so you have an easy way to prototype SQL.

SQL Performance Analyzer studies the indexes used and the number of columns matched, sorts requirements, searches for common coding errors (Select *), poor access paths (TScans), missing catalog stats, and more.

Slide - What can SQL Performance Analyzer do for you? (3 of 3)

In addition to helping you design new queries, SQL Performance Analyzer can also help you identify bad queries that you've coded. Let's say you have an application and a set of SQL that's taking far too long to run. You can take the DBRM module and run it through SQL/PA, and it will come back with the access paths, the costs and expert Advisor information to help you determine what is wrong.

The IBM DB2 Query Monitor is a tool that focuses on SQL operations and lets you monitor them as they flow through the system. Query Monitor can select and feed SQL statements to SQL/PA, invoking it directly under its TSO ISPF interface.

SQL Performance Analyzer can also be used as part of the QMF governor process. From QMF, the query is intercepted by SQL/PA, which evaluates each SQL statement: if it will run too long, SQL/PA lets you cancel execution before QMF starts running the SQL. Otherwise, it operates in the background, and the QMF user sees no message unless excessive run time is forecast.

In addition, you can call SQL Performance Analyzer as a stored procedure. Within any application you can determine the cost of the next query and decide whether or not you want to run it before continuing with the application.

What problems can you solve?

- **Tune long-running online queries or batch jobs**
- **Resolve database and index design problems**
- **Identify poor use of SQL predicates and clauses**
- **Prevent runaway queries before cancellation by governors**
- **Uncover changing data patterns affecting performance**
- **Eliminate prototyping and stress testing**
- **Build performance into applications during design phase**
- **Evaluate production problems using DRDA connectivity**
- **Assess production performance through DBRM analysis**

Slide - What problems can you solve? (1 of 2)

SQL Performance Analyzer provides visualization and the analysis that extends its use beyond simple costing. This slide illustrates some of the many problems that can be solved using it.

SQL Performance Analyzer determines which index is used, the number of columns matched, the index key column statistics, and sort requirements. It searches for common coding errors, poor access paths, table space scans, missing catalog statistics, and much more.

SQL Performance Analyzer gives you information on any alternative indexes that were not selected by the optimizer. This gives you the ability to say, "I have seven indexes defined on a particular table and the optimizer is picking only one of them. Why does the optimizer select that index rather than the others?"

You could determine the relative cost of the query using the other six indexes as well. If you find one that you think would be better, you can either rewrite your query so the optimizer uses it, or select a different access path to use the index you want via Hints. So, SQL Performance Analyzer can be used for more than getting simple costing information.

Slide - What problems can you solve? (2 of 2)

Building performance into applications during the design phase can save you money. If you use SQL Performance Analyzer during the design phase, you can avoid having to determine why your queries are not getting good results later. You could save not only the time to rewrite the application or query, but also the time spent determining the cause of the problem.

Users on one DB2 system can connect to another system using DRDA CONNECT, run SQL Performance Analyzer there, and return the results to the originating system. This is useful when you need to evaluate production problems (VIADRDA parm).

You can gather pertinent catalog statistics used by SQL Performance Analyzer from other DB2 systems (for example, production systems), and populate your test system catalog with them. This can assist you with assessing production performance, and can eliminate labor-intensive manual updates. The ANLSTAT program is part of SQL/PA and allows for easy migration of all the optimizer-sensitive size and cardinality stats from one DB2 subsystem to another. You can collect partition-level data as well as non-uniform distributions.

Runtime environment

- **Batch, as an MVS job**
- **TSO, under ISPF Panels**
- **QMF, as a governor intercept**
- **DB2 applications, as a stored procedure call**

Slide - Runtime environment

SQL Performance Analyzer runs in several environments. You can use it in Batch, interactively via TSO ISPF, through QMF (attaching to the governor exit), or develop your own governor using a stored procedure call. Batch users can also call SQL/PA behind IBM's Path Checker, which can weed out SQL for further analysis by Performance Analyzer.

The ANLPROC stored procedure is accessed when your application passes the SQL statement to SQL Performance Analyzer in a host variable, and the application then decides whether or not to execute the query, modify it, or reschedule it later, based upon the costing and warning flags returned from the SQL/PA stored procedure.

Both COBOL 2 and PL/I sample programs are provided (including source code and JCL to compile/link/bind) to help you design your own programs that can access the SQL/PA stored procedure.

The QMF intercept runs in the background and becomes active only if a query is about to run that exceeds your installation's run time expectations. Then it provides you a cost summary on the screen, and asks whether you want to cancel that query before it starts. This may avoid the query's probable termination later by QMF's Governor and a waste of run time resources.

How does SQL Performance Analyzer work?

- Parses the SQL and deciphers the predicates
- Invokes PREPARE and performs catalog lookup
- Evaluates the optimizer's chosen access path
- Uses extensive knowledge of DB2 for costing

→ Predicts the cost of queries before they run

Slide - How does SQL Performance Analyzer work? (1 of 2)

The first thing SQL Performance Analyzer does is parse the input source, the SQL. This identifies the key predicates and clauses that make up the SQL statement.

Next, the optimizer prepares and explains the query, normally into a generic and reusable PLAN_TABLE. This means that you do not need to have a PLAN_TABLE defined. You don't have to manage a series of plan tables that are used only for brief periods. DSN_STATEMENT_TABLES are also used, if defined.

The pertinent catalog statistics are gathered to describe the tables and indexes to be used in the access path and then a resource consumption estimate is created, using knowledge of the access path's characteristics and the statement's composition and function.

The resulting cost estimate is transformed into CPU, I/O Count, elapsed time and monetary units. Warnings are produced if the query exceeds any limits you have set. The monetary costs provide customization for currency and the currency symbol (\$, Euro, etc.)

Slide - How does SQL Performance Analyzer work? (2 of 2)

Because SQL Performance Analyzer supports DB2 V5 through V7, you can use it to analyze how things change from release to release. To get the best results, three things have to happen. First, the catalog has to have pretty good statistics. Second, before you perform analysis, the DB2 system must be at the level that you're going to run the query on. This might be something you would want to perform after you migrate to a new release on your test system. Third, your parameters need to characterize the run time environment you are expecting.

You can simulate other releases without moving the product. Note that when SQL/PA does a prepare, it uses the optimizer on the system to which you are connected. Other factors that SQL/PA considers include the catalog information, whether or not you are in data sharing, what type of DASD subsystem you have, and the MIPS ratings of the machine. These are all considerations in determining a realistic monetary cost. All of these factors are parameterized for easy entry and modification.

You will be asked to supply some of this information when you first install SQL/PA. You can update the settings as your environment changes. If you do not know the MIPS rating of the system, the Install Guide provides a table with this information and a method for obtaining ratings for new machines.

Input sources

- **Sequential data sets (PS, FB, columns 1-72)**
- **Partitioned data set members (PDS, FB, columns 1-72)**
- **DBRM library members OR the entire DBRM library (using DBRMKEY to filter which members to select)**
- **For QMF, direct terminal input**
- **For TSO, any edited or browsed data set**
- **For stored procedure, a host variable containing the SQL**

Slide - Input sources

Input to SQL Performance Analyzer can come from a variety of sources, such as a sequential data set, a member of a partitioned data set, a DBRM library, or a query stored in QMF. You can use any Fixed Block file that contains SQL.

In TSO, SQL Performance Analyzer uses a data set that you pass it. Normally, it is an editable data set (sequential or PDS member). With the stored procedure, you put the SQL into a large 32K varchar host variable. You can also use existing applications, for example a COBOL program, as input, and select the SQL by using the TSO edit command ANL (put PP-PP in the left line number fields to select the text for study). SQL Performance Analyzer is not programming language sensitive. It doesn't know you're giving it a COBOL or PL/I program. All it knows is that the input is from a file.

SQL Performance Analyzer puts you in edit mode so you can select the SQL statements you want to analyze. The SQL statements you select are extracted from the input file. You can select one or more statements to be analyzed, or submit the entire file. SQL/PA can also be invoked from IBM's DB2 Admin Tool, via the Launchpad, or from any TSO Edit or Browse session, using the "SPA" edit command.

In summary, there are many sources of input for SQL Performance Analyzer. The only requirement is that it be SQL code that DB2 understands. If you pass it an SQL statement that is not technically correct, SQL Performance Analyzer passes back the DB2 error code that it receives when it tries to do the prepare.

Types of reports

- **Query Limits Report**
- **Cost Summary Report**
- **Enhanced Explain Report ***
- **Detail Trace Report ***

** includes the SQL Advisor*

Slide -Types of reports (1 of 2)

The Query Limits Report gives you a list of SQL statements, along with each one's exceptions or cost overruns. It presents a one line summary of each SQL statement evaluated, by query number, showing warning flags for overruns, and the cost estimates for CPU time, Elapsed time, I/O time, Qunits and Monetary. The SQL that are outside your thresholds are highlighted so you can easily see the problem areas.

Consider starting with this level report so you can say, "Well, I've only got to worry about queries number 4 and 5". That way, you don't need to look at any of the rest of the information. You can now look closer at the other reports, concentrating and analyzing only queries 4 and 5, where your problems lie. These queries may also be isolated in a separate run for more detailed study. By ignoring the information for the queries you are not concerned about, you can focus on the ones you want to look at.

The Cost Summary Report is a little bit more detailed than the Query Limit Report. It shows the query, gives you the actual detailed cost, and includes all the cost values (CPU, ELAPSED, I/O, QUNITS and Monetary), and also displays and highlights any warnings of cost overruns in those categories.

Slide -Types of reports (2 of 2)

The Enhanced Explain Report is an English language explanation with a tremendous amount of detail. The report begins with information about the query, such as which matching index scans it selected, and the statistics from the catalog that describe the chosen access path. If it encounters any potentially serious problems, it gives you an alert or warning, identifying things you should look at.

One of the alerts, for example, is on a simple table space. If it finds a simple table space, it issues a warning, suggesting you consider putting this table into a segmented table space. It also tells you the reason for this recommendation. So, it not only makes a recommendation, but also tells you why.

If you have other problems, for example, if you use `SELECT *`, you get more information telling you that it's not a good thing to do, and encouraging you to select specific columns. It then explains what DB2 does when it sees the asterisk, and why selecting columns is a better approach.

Factors considered by SQL Performance Analyzer

- **Influence of DB2 catalog statistics**
- **Types and characteristics of predicates**
- **Hardware and software configuration**
- **User and installation parameters**
- **Predicate filter factors**
- **Optimizer cost estimates**
- **Release level of DB2**

Slide -Factors considered by SQL Performance Analyzer (1 of 2)

When you look at SQL Performance Analyzer output, it is important to understand that output is very dependent upon the catalog statistics. They must be reliable. It's the old "garbage in, garbage out" law. If you have bad statistics in the catalog, then you will obtain bad results.

If you have a predicate, for example, "Where column A In (a long list)", it will return a warning, telling you that this might not be the best thing to do, and why.

SQL Performance Analyzer looks at the hardware and software configuration. As previously mentioned, you specify the type of processor you're on, the MIPS ratings, the DASD subsystem, and various other parameters when installing and customizing the program.

There are also parameters that you can set and modify for each individual run. Let's say you are on a 3090 with 16 MIPS in a processor, and you are going to use one processor. What are your results? Now, if you want to explore running the same thing but on a zSeries 900 processor, you can enter the values for that processor and run it again.

Within a single run, if your SQL references the same table or index (or other stats) more than once, SQL Performance Analyzer goes to the catalog only once to pull all the statistics for all the objects involved. This is a performance design feature that is built into the product. It is designed to minimize catalog activity. The latest release increases the number of internally cached objects, from 100 to 250 or 500 each.

Slide -Factors considered by SQL Performance Analyzer (2 of 2)

The rest of the factors listed on this slide are relatively self-evident.

The release level of DB2 is not too important to SQL Performance Analyzer, in terms of being able to run new estimates against it, because the program is release independent. However, it uses a prepare to get the cost and, therefore, it uses the optimizer of the DB2 system it is running on. So, be aware that your results may or may not be valid due to changes that have happened in the optimizer for that particular release.

SQL/PA currently supports DB2 Versions 5, 6 and 7 simultaneously. You do not need to install another version of the product when you migrate from V6 to V7, for instance.

The DRDA Connect feature allows SQL/PA to run from one DB2 sub- system to another, but remember that SQL/PA must be installed on each machine. The BIND instructions detail how to make SQL/PA correctly correspond between the multiple copies.

Intangible factors

- **Uses benchmarked path lengths or optimizer's CPU for Costs**
- **Issues Warnings when installation thresholds are exceeded**
- **SQL Advisor spots problems, then recommends solutions**
- **Stored procedure provides ready-made governor for any DB2 application, from any DB2 attach facility (including PC, web)**
- **Enhanced Explain report provides simple summary of access path selected and its associated costs**
- **Detail Trace report provides in-depth analysis and cost breakdown, illustrating where the SQL is spending most of its time**

Slide - Intangible factors

You've seen many of the benefits you get from SQL Performance Analyzer: the expert SQL Advisor, the costs, the ability to use pretty much any input source, the ability to run it in batch or to run it interactively, etc. This page lists a series of intangible benefits. In addition to the Explain output from the optimizer, SQL Performance Analyzer uses extensive benchmark data collected on each release to project the actual costs against the proposed configuration described in parms.

This data provides a check and balance with what the optimizer generated, filling in the framework around the SQL execution (attach overhead, thread management, and so forth). It is more than just a problem notifier - it also recommends solutions and helps you solve the problems. If you look at other available cost estimator programs, they look at what the optimizer says it's going to do and then they return the cost, usually in Timerons (relative cost). The SQL Performance Analyzer additionally considers the factors that go on around that to give you the "real world" cost. It includes the additional costs of the overhead for doing an IMS attach, the overhead for managing a thread, the cost of writing out records on an insert, update or delete, etc. So you have the benefit of getting back a more complete cost.

As previously mentioned, you can use the stored procedure support to develop your own governor functionality in any application. The SQL Advisor provides three levels of support, depending upon your expertise - from novice to expert.

Built in flexibility

- You can vary the test catalog statistics via updates, inserts, or the RUNSTATS utility, OR migrate them from Production to Test using the ANLSTAT program
- You can emulate production volumes in test catalog and tune applications while still in early design stages
- You can vary the parameters
 - User parameters describe the application and select reporting levels
 - Installation parameters describe the configuration and warning levels

Slide - Built in flexibility

As a user, you have a lot of flexibility with SQL Performance Analyzer. You can vary the statistics in the catalog in many ways (see the User's Guide for suggestions). For example, production volumes can be simulated in the test catalog, allowing SQL Performance Analyzer to give an indication of performance after migration. In addition, you can simulate the growth projected by your business, so you can determine what performance will be like in a year, two years, or any time frame you choose. Sizing formulas are provided in the User's Guide to help you assess the initial size of new tables and indexes.

The configuration parameters allow you to describe any real or imagined configuration to SQL Performance Analyzer, which then projects performance onto that hypothetical system (it doesn't have to match the system you are running on, or even exist).

User parameters describe the DASD, DB2 release level, and other important factors, such as the level of costing to use (internal path lengths or optimizer), the level of Advice to show, which reports to generate, overrides for high level qualifier, route to another system via DRDA, scan DBRM library for member names matching a certain pattern, show alternate indexes available, etc. The tool is very flexible to your needs.

Summary

- **A truly predictive governor that handles both static and dynamic SQL**
- **Excellent prototyping tool and DB2 design aid**
- **Expert advisor educates the developers, from novice to expert level**
- **Query tuner, which is both release and SQL statement sensitive**
- **Support for all SQL statement forms: SELECT, INSERT, UPDATE, DELETE, JOIN, UNION, SUBQUERY, etc.**
- **Ability to be run from any z/OS or OS/390 environment: TSO, Batch, QMF, STP**

Slide - Summary

In summary, SQL Performance Analyzer is a tool that enables you to manage the performance of your DB2 applications.

SQL Performance Analyzer can help you design SQL statements and table/index structures, refine and review access paths, scan production SQL looking for problems, intercept resource-intensive QMF queries before they start, and much more.

Far beyond the intrinsic function of providing a cost for any query before it runs, it can also help you in many other ways, not the least of which is in getting the best possible performance from your present production SQL.

All estimates are done without actually executing the query. SQL Performance Analyzer does the prepare, but the query never consumes any resources related to going out and looking at data.

Finally, its Advisor can teach newer DB2 developers how to code SQL for optimum future benefits. It can also help experienced DB2 developers keep up with the changes occurring in DB2 from release to release.

SQL/PA interfaces with other IBM data management products: Path Checker, DB2 Admin Tool, DB2 Query Monitor and WebQuery Tool.