

Tools for a Comprehensive DB2 Environment

BY CRAIG S. MULLINS

DB2, as offered by IBM, is a complete, full-function RDBMS. An organization can install and use DB2 as delivered, but will quickly recognize that the functionality needed to adequately support large-scale DB2 development is not provided sufficiently by DB2 alone.

The administration and maintenance of DB2 applications is time-consuming if you use the standard features of DB2. Fortunately, many tools that enhance the functionality of DB2 are available from third party vendors. These tools ease the administrative burden and reduce the possibilities of database error.

The true need for these tools becomes apparent simply by reviewing the sheer number of products that are available. Most DB2 shops implement at least one or more add-on tools for DB2. Of these, QMF and DB2-PM are among the most popular provided by IBM. Many more tools from other vendors fill market niches not adequately supported by IBM. Following is a rundown of the categories of products.

DB2 Object Altering Tools

DB2 provides the capability to modify the structure of existing objects using the ALTER DDL statement. The ALTER statement, however, is a functionally crippled statement. You should be able to alter all of the parameters that can be specified for an object when it is created, but DB2 does not support this. For example, you can add columns to an

existing table (only at the end), but you can never remove columns from a table. The table must be dropped, then re-created without the columns targeted for removal.

Another problem that DBAs encounter in modifying DB2 objects is the cascading drop effect. If a change to a tablespace mandates its being dropped and re-created (for example, changing the limit keys of a partitioned tablespace), all dependent objects are dropped when the tablespace is dropped. This includes all tables in the tablespace, all indexes on the tables, all primary and foreign keys, any related synonyms and views, and all authorization.

Ensuring that the DDL is issued after the modification to reverse the effects of cascading drops can be a tedious, complex and error-prone procedure.

Additionally, many types of DB2 object alteration cannot be performed using the generic DB2 ALTER statement. For example, you:

- Cannot change the name of a database, table, alias, view, column, tablespace, or index;
- Cannot change the database in which the tablespace exists;
- Cannot change the number of tablespace partitions;
- Cannot remove a tablespace or index partition;
- Cannot change tablespace type, for example changing a simple tablespace to a segmented or partitioned tablespace;
- Cannot copy primary and foreign keys using CREATE LIKE;

this command creates a new table based on only the columns of another table;

- Cannot move a table from one tablespace to another;
- Cannot rearrange column ordering;
- Cannot change a column's data type and length;
- Cannot remove columns from a table;
- Cannot change the primary key without dropping and adding the primary key;
- Cannot add to a table a column specified as NOT NULL;
- Cannot add any columns to a table defined with an EDITPROC;
- Cannot change a table's EDITPROC or a column's VALIDPROC;
- Cannot create a view based on another view;
- Cannot add columns to a view or remove columns from a view;
- Cannot change the SELECT statement on which the view is based;
- Cannot create an index based on another index;
- Cannot change the indexing columns;
- Cannot change the partitioning information;
- Cannot change the uniqueness specification of an index;
- Cannot change the clustering of an index;
- Cannot change the number of index subpages;
- Cannot change the index order (ascending or descending);
- Cannot create an alias based on another alias;

- Cannot change the location of the alias; and
- Cannot change the table on which the alias is based;

This list helps provide the justification needed to obtain an alter tool (and this is not an exhaustive list). An alter tool provides an integrated environment for altering DB2 objects. The burden of ensuring that a change to a DB2 object does not cause other implicit changes is moved from the DBA to the tool.

At a minimum, an alter tool should:

- Maintain tables easily without manually coding SQL;
- Retain or reapply all dependent objects and security affected by the requested alter if a drop is required;
- Navigate hierarchically from object to object, making alterations as you go;
- Provide panel-driven modification showing before and after definitions of the DB2 objects before the changes are applied;
- Batch requested changes into a group that execute in the foreground or the background;
- Analyze changes to ensure that the requested alterations do not violate any DB2 DDL rules. For example, if a series of changes is requested and one change causes a subsequent change to be invalid (an object is dropped, for instance), this should be flagged before execution;
- Have a controlled environment in which alters are executed; and
- Possess the capability to monitor changes as they are applied.

Auditing Tools

An audit is the examination of a practice to determine its correctness. DB2 auditing software therefore should help in monitoring the data control, data definition, and data integrity in the DB2 environment. Several mechanisms provided by DB2

enable the creation of an audit trail, but this trail can be difficult to follow.

The primary vehicle provided by DB2 for auditing is the audit trace. This feature enables DB2 to trace and record auditable activity initiated by specific users. When the DB2 audit trace is activated, the following type of information can be captured to the trace destination:

- Authorization failures;
- Grant and revoke SQL statements;
- DDL issued against auditable tables;
- DML issued against auditable tables;
- Bind requests involving auditable tables; and
- Authorization ID changes requested by the SET CURRENT SQLID statement.

Utility Executions

An auditable table is any table defined to DB2 with the AUDIT clause of the CREATE TABLE statement. There are three options for table auditing: NONE, CHANGES and ALL. Specifying AUDIT NONE, which is the default, disables table auditing such that the audit trace does not track that table. Specifying AUDIT CHANGES indicates that the first DELETE, INSERT, or UPDATE statement issued against that table in every application unit of work (COMMIT scope) is recorded. AUDIT ALL records the first DML statement of any type accessing this table in each application unit of work. Note, however, that this information is tracked only if the appropriate audit trace is activated.

This information is written to the output trace destination specified for the audit trace. DB2 trace records can be written to GTF, SMF, or an OP buffer. After the information has been written to the specified destination, the problem of how to read this informa-

tion still exists. If you have DB2-PM, you can run the appropriate audit reports, but even these can be insufficient for true auditing.

An audit tool should provide five important features that DB2's audit tracing capability does not. DB2 auditing requires a trace to be activated, and this can quickly become expensive if many tables must be audited. The first feature an auditing tool should provide is the capability to read the DB2 logs, which are always produced, and report on update activity as needed. This reduces overhead because it uses the regular processing features of DB2 rather than an additional tracing feature, which increases overhead.

The DB2 audit trace records a trace record only for the first statement in a unit of work. The second feature of the auditing tool is reporting all data modification from the DB2 logs.

The DB2 audit trace facility does not record the specifics of the data modification. The third feature of an auditing tool is reporting who (by authorization ID) makes each change, and also showing a before and after image of the changed data. The fourth feature the auditing tool should provide is the capability to report on the DB2 audit trace data if so desired. Finally, the auditing tool should provide both standard reports and the capability to create site-specific reports (either from the log or from the DB2 audit trace data). If your shop has strict auditing requirements, an auditing tool can be quite useful because of DB2's weak inherent auditing capabilities.

DB2 Catalog Query and Analysis Tools

The DB2 Catalog contains a wealth of information essential to the operation of DB2. Information about all DB2 objects, authority and recovery is stored and maintained in the DB2 Catalog.

This system catalog is composed of DB2 tables, and can be queried using SQL. The data returned by these queries provides a base of information for many DB2 monitoring and administrative tasks.

But coding SQL can be a time-consuming process. Often, you must combine information from multiple DB2 Catalog tables to provide the user with facts relevant for a particular task.

Add-on tools can ease the burden of developing DB2 Catalog queries. The basic feature common to all DB2 Catalog tools is the capability to request DB2 Catalog information using a screen-driven interface without coding SQL statements. Analysts can obtain rapid access to specific facts stored in the DB2 Catalog without the burden of coding (sometimes quite complex) SQL.

DB2 Catalog tools that provide only this level of capability are rudimentary tools at best. Most DB2 Catalog tools provide much more capability. Instead of merely enabling data access, many DB2 Catalog tools can do one or more of the following:

- Create syntactically correct DDL statements for all DB2 objects by reading the appropriate DB2 Catalog tables. These statements are generally executed immediately or saved in a sequential data set for future reference or use;
- Modify the "updateable" DB2 Catalog statistical columns using a non-SQL interface;
- Create syntactically correct DCL statements from the DB2 Catalog in the same way that DDL is generated;
- Perform "drop analysis" on a SQL DROP statement. This analysis determines the effect of the cascading drop by detailing all dependent objects and security that will be deleted as a result of executing the DROP;
- Provide a hierarchic listing of DB2 objects. For example, if a

specific table is chosen, the tool can migrate quickly up the hierarchy to show its tablespace and database, or down the hierarchy to show all dependent indexes, views, synonyms, aliases, referentially connected tables and plans;

- Create and drop DB2 objects, and grant and revoke DB2 security from a screen without coding SQL. Additionally, some tools log all drops and revokes such that they can be undone in the event of an inadvertent drop or revoke execution; and
- Operate on the DB2 Catalog or on a copy of the DB2 Catalog to reduce system-wide contention.

These features aid the DBA in performing his day-to-day duties. Furthermore, a catalog query tool can greatly diminish the amount of time required for a junior DBA to become a productive member of the DBA team.

Compression Tools

A standard tool for reducing DASD costs is the compression utility. This type of tool operates by applying an algorithm to the data in a table such that the data is encoded in a more compact area. By reducing the amount of area needed to store data, DASD costs are decreased. Compression tools must compress the data when it is added to the table and subsequently modified, then expand the data when it is later retrieved.

Third-party compression routines are usually specified for DB2 tables using the EDITPROC clause of the CREATE TABLE statement. The load module name for the compression routine is supplied as the parameter to the EDITPROC clause. A table must be dropped and re-created to apply an EDITPROC.

In general, a compression algorithm increases CPU costs while providing benefits in the areas of decreased DASD utilization and sometimes decreased I/

O costs. This tradeoff is not beneficial for all tables. For example, if a compression routine saves 30 percent on DASD costs but increases CPU without decreasing I/O, the tradeoff is probably not beneficial. A compression tool can decrease DASD by reducing the size of the rows to be stored. CPU use usually increases because additional processing is required to compress and expand the row. I/O costs, however, could decrease.

Prior to DB2 V2.3, compression was unavailable using standard DB2 features unless the user coded an algorithm. As of V2.3, DB2 provides a basic compression routine called DSN8HUFF. However, most third party compression tools provide more efficient compression algorithms and advanced analysis to determine the costs and benefits of compression for a specific table. This changed dramatically with DB2 V3. In fact, the internal compression capabilities of DB2 V3 will probably reduce the need for most third party compression tools unless there are other features that make the tool valuable. The major advantage of third party compression tools is that most of the vendors provide multiple compression algorithms for different types of data. DB2 provides only Ziv-Lempel compression.

Data Warehousing Tools

At times, multiple DBMSs co-exist in data processing shops. This is increasingly true as shops embark on client/server initiatives. Additionally, the same data may need to be stored in each of the DBMS products. In a multiple DBMS environment, the movement of data from DBMS to DBMS is a tedious task.

Data warehousing tools ease the burden because the tool understands the data format and environment of each DBMS it works with. The warehousing tool(s) that a shop chooses will

depend upon the following factors:

- How many DBMS products need to be supported?;
- To what extent is the data replicated across the DBMS products?;
- Does the data have to be synchronized across DBMS products?;
- Is the data static or dynamic?; and
- If it is dynamic, is it updated online, in batch; both?

The answers to these questions will help to determine the type of data warehousing tool necessary. Two basic types of data conversion tools are popular in the market today.

Replication tools extract data from external application systems and other databases for population into DB2 tables. This type of tool can extract data from VSAM, IMS, Oracle, flat files, and/or other structures and insert the data into DB2.

Propagation tools insert data from external applications and other database products into DB2 tables. A propagation tool is similar in function to a replication tool. Propagation tools, however, are active. They constantly capture updates made in the external system either for immediate application to DB2 tables or for subsequent batch updating. This differs from the extract tool, which captures entire data structures, not data modifications.

DB2-Related Client/Server Tools

Client/server processing has been very successful in recent years because it provides a flexible, distributed computing environment and decreases reliance on the mainframe. However, DB2 is a large participant in the client/server plans for many shops. Providing efficient access to large amounts of data, DB2 MVS can function as the ultimate database

server in a client/server environment.

This being the case, there are many tools on the market that can ease the burden of implementing and administering DB2 in a client/server environment. Middleware products and database gateways that sit between the client workstation and the mainframe enable access to DB2 as a server. These products can provide access to DB2 MVS as well as access to other server DBMS products (DB2/2, DB2/6000, Sybase SQL Server, Oracle, etc.). Additionally, many third party ODBC drivers are being made available to ease workstation access to mainframe DB2 data.

Another valid type of client/server tool is a 4GL programming environment that provides seamless access to DB2. These types of products typically split the application workload between the workstation and the server aiding the programmer to rapidly develop DB2 client/server applications.

Database Analysis Tools

DB2 does not provide an intelligent database analysis capability. Instead, a database administrator or performance analyst must keep a vigilant watch over DB2 objects using DB2 Catalog queries or a DB2 Catalog tool. This is not an optimal solution because it relies on human intervention for efficient database organization, opening up the possibility of human error, forgetting to monitor, and misinterpreting analyzed data.

Fortunately, database analysis tools can proactively and automatically monitor your DB2 environment. This monitoring can:

- Collect statistics for DB2 tablespaces and indexes. These statistics can be standard DB2 RUNSTATS information, extended statistics capturing

more information (for example, data set extents), or a combination of both;

- Read the VSAM data sets for the DB2 objects to capture current statistics, read RUNSTATS from the DB2 Catalog, read tables unique to the tool that captured the enhanced statistics, or any combination of these three;
- Set thresholds whereby the automatic scheduling of a utility is invoked based on current statistics; and
- Provide a series of canned reports detailing the potential problems for specific DB2 objects.

Repositories

A repository stores information about an organization's data assets. Repositories are used to store metadata, or data about data. They frequently are used to enhance the usefulness of DB2 application development.

In choosing a repository, base your decision on the metadata storage and retrieval needs of your entire organization, not just DB2. Typically, a repository can:

- Store information about the data, processes and environment of the organization;
- Support multiple ways of looking at the same data. An example of this concept is the three-schema approach, in which data is viewed at the conceptual, logical and physical levels;
- Store in-depth documentation as well as produce detail and management reports from that documentation;
- Support data model creation and administration. Integration with popular CASE tools is also an important evaluation criterion;
- Support change control;
- Enforce naming conventions; and

- Generate copy books from data element definitions.

These are some of the more common functions of a repository. When choosing a repository for DB2 development, the following features generally are desirable:

- The data stores used by the repository are in DB2 tables. This enables DB2 applications to directly read the data dictionary tables;
- The repository can directly read the DB2 Catalog or views on the DB2 Catalog. This ensures that the dictionary has current information on DB2 objects;
- If the repository does not directly read the DB2 Catalog, an interface is provided to ease the population of the repository using DB2 Catalog information; and
- The repository provides an interface to any modeling and design tools used for the generation of DB2 objects.

This section has provided only a brief overview of repositories. An extended discussion of repository technology is beyond the scope of this article.

Database Modeling and Design Tools

Database modeling and design tools do not have to be unique to DB2 design, although many are. Application development should be based on sound data and process models. The use of a tool to ensure this is a good practice.

Database modeling and design tools are often referred to as CASE tools. CASE, or Computer-Aided Software Engineering, is the process of automating the application development life cycle. A CASE tool, such as a data modeling tool, supports portions of that life cycle.

Many excellent database design and modeling tools are not specifically designed for DB2, but can be used to develop DB2 appli-

cations. Tools developed specifically to support DB2 development, however, add a dimension to the application development effort. They can significantly reduce the development timeframe by automating repetitive tasks and validating the models. If your organization decides to obtain a CASE tool that specifically supports DB2, look for one that can:

- Provide standard features of logical data modeling (such as entity-relationship diagramming and normalization);
- Create a physical data model geared to DB2. This model should support all features of DB2, such as the capability to depict all DB2 objects, referential integrity, VCAT and STOGROUP-defined tablespaces, and capacity planning;
- Provide an expert system to verify the accuracy of the physical data model and to suggest alternative solutions;
- Cross-reference the logical

model to the physical model, capturing text that supports physical design decisions such as denormalization and the choice of tablespace type;

- Automatically generate DB2-standard DDL to fully implement the database defined in the physical data model; and
- Interface with application development tools and repository products available to the organization.

DASD and Space Management Tools

DB2 provides basic statistics for space utilization in the DB2 Catalog, but the in-depth statistics required for both space management and performance tuning are woefully inadequate with several critical elements missing.

Chief among the missing elements of DASD space management in DB2 is the capability to monitor the space requirements of the underlying VSAM data sets.

Kelly Technical Services

Automotive and Technical Division
P.O. Box 667, Dearborn, Michigan 48121-0667
Phone: 313-336-3355 Fax: 313-271-7822 kelly73@ix.netcom.com

OBJECTIVE	To be the world's best technical staffing services company and to be recognized as the best by placing with our customers the most qualified technical professionals available today.		
COMPUTER SKILLS UTILIZED	<u>Languages</u> Assembler Fortran C, C++ HTML Clipper JAVA Cobol/II Progress Dibol Smalltalk Focus	<u>Databases</u> Access Oracle DB2 Paradox INFORMIX SAS INGRES Sybase	<u>Software</u> CASEtools Lotus Notes CICS NetWare Delphi PowerBuilder Ethernet Visual Basic IMS DB/DC
EXPERIENCE	Kelly Services Inc. Automotive and Technical Division Information Systems Staffing Experts Responsible for Staffing: <u>Position Types</u> Long-Term Contract • Contract-To-Hire • Direct Placement <u>Position Titles</u> • Programmer • Comm. Specialist • Network Engineer • Programmer/Analyst • Computer Operator • Network Administrator • Systems Analyst • OA Specialist • Appl. Developer • Software Engineer • DBA • Project/Team Leader	1946 to Present 1988 to Present	
ACHIEVEMENTS AND AWARDS	<ul style="list-style-type: none"> • 1200 Branch Offices Worldwide • \$2.7 billion in annual sales for 1995 • Preferred supplier to the automotive Big Three • Recipient of the Q-1 Quality Award • Blanket Vendor to Chrysler Corporation • Fortune 500 Company 		

An Equal Opportunity Employer

What's On Your Resume?

Contact the above Technical Branch for details on how we can connect you to our worldwide technical employment network.

READER SERVICE NO. 55

When these data sets go into secondary extents, performance suffers. Without a DASD management tool, the only way to monitor secondary extents is to periodically examine LISTCAT output. This is a tedious exercise. Additionally, the manner in which DB2 allocates space can result in the inefficient use of DASD. Often, space is allocated but DB2 does not use it. A DASD management tool is the only answer for ferreting out the amount of allocated space versus the amount of used space. DASD management tools often interface with other DB2 and DASD support tools such as standard MVS space management tools, database analysis tools, DB2 Catalog query and management tools and DB2 utility JCL generators.

DB2 Table Editors

The only method of updating DB2 data is with the SQL data manipulation language statements DELETE, INSERT and UPDATE. Because these SQL statements operate on data a set at a time, multiple rows—or even all of the rows—can be affected by a single SQL statement. Coding SQL statements for every data modification required during the application development and testing phase can be time-consuming.

A DB2 table editing tool reduces the time needed to make simple data alterations by providing full-screen edit capability for DB2 tables. The user specifies the table to edit and is placed in an edit session that resembles the ISPF editor. The data is presented to the user as a series of rows, with the columns separated by spaces. A header line indicates the column names. The data can be scrolled up and down as well as left and right. To change data, the user simply types over the current data.

This type of tool is ideal for

supporting the application development process. A programmer can make quick changes without coding SQL. Also, if properly implemented, a table editor can reduce the number of erroneous data modifications made by beginning SQL users.

Following are a few words of warning pertaining to the use of table editors. Remember that the table editor is issuing SQL in the background to implement the requested changes. This can cause a lag between the time the user updates the data and the time the data is committed. Table editor updates usually are committed only when the user requests that the data be saved or when the user backs out of the edit session without canceling.

Remember too that table editors can consume a vast amount of resources. Ensure that the tool can limit the number of rows to be read into the editing session. For example, can the tool set a filter such that only the rows meeting certain search criteria are read? Can a limit be set on the number of rows to be read into any one edit session? Without this capability, large tablespace scans can result.

Remember that a DB2 table editor should be used only in the testing environment. End users or programmers might request that a table editor be made available for production data modification. This should be avoided at all costs. The data in production tables is critical to the success of your organization, and should be treated with great care. Production data modification should be accomplished only with thoroughly tested SQL or production plans.

When a table editor is used, all columns are available for update. Thus, if a table editor is used to change production data, a simple mis-keying can cause unwanted updates. Native SQL

should be used if you must ensure that only certain columns are updated.

One final note: Tested SQL statements and application plans are characterized by their planned nature. The modification requests were well thought out and tested. This is not true for changes implemented through a table editor.

DB2 Object Migration Tools

DB2 does not provide a feature to migrate DB2 objects from one subsystem to another. This can be accomplished only by manually storing the CREATE DDL statements (and all subsequent ALTER statements) for future application in another system. Manual processes such as this are error-prone. Also, this process does not take into account the migration of table data and DB2 security.

DB2 object migration tools facilitate the quick migration of DB2 objects from one DB2 subsystem to another. They are similar to a table altering tool but have a minimal altering capability (some interface directly with an alter tool or are integrated into a single tool). The migration procedure usually is driven by ISPF panels that prompt the user for the objects to migrate.

Migration typically can be specified at any level. For example, if you request the migration of a specific database, you could also migrate all dependent objects and security. Minimal renaming capability is provided such that database names, authorization IDs, and other objects are renamed according to the standards of the receiving subsystem. When the parameters of the migration have been specified completely, the tool creates a job stream to implement the requested DB2 objects in the requested DB2 subsystem.

A migration tool reduces the

time required by database administrators to move DB2 databases from environment to environment (for example, from test to production). Quicker turnaround results in a more rapid response to user needs, thereby increasing the efficiency of your business. Typically, migration tools are the second DB2 tool that an organization acquires (right after a DB2 Catalog query product).

Operational Support Tools

Many avenues encompass operational support in a DB2 environment, ranging from standards and procedures to tools that guarantee smoother operation.

One type of operational support tool provides online access to DB2 standards and procedures. These tools are commonly populated with model DB2 standards and procedures that can be modified or extended. Tools of this nature are ideal for a shop with little DB2 experience that wants to launch a DB2 project. As the shop grows, the standards and procedures can grow with it.

Another type of product delivers online access to DB2 manuals. With this tool, you avoid the cost of purchasing DB2 manuals for all programmers, and DB2 information and error messages are always available online. In addition, analysts and DBAs who dial in to the mainframe from home can reference DB2 manuals online.

Standard batch DB2 programs run under the control of the TSO terminal monitor program, IKJEFT01. Another operational support tool provides a call attach interface that enables DB2 batch programs to run as a standard MVS batch job without the TSO TMP.

DB2, unlike IMS, provides no inherent capability for storing checkpoint information. Tools that store checkpoint information

which can be used by the program during a subsequent restart are useful for large batch DB2 applications issuing many COMMITs.

One final type of operational support tool assists in managing changes. Typically, these tools are integrated into a change control tool that manages program changes. Change control implemented for DB2 can involve version control, plan and package management, and ensuring that timestamp mismatches (SQLCODE -818) are avoided. Some tools can even control changes to DB2 objects.

PC-Based DB2 Products

Personal computers are everywhere nowadays. Most data processing professionals have one on their desk. Most end users do too! As such, the need to access DB2 from the PC is a viable one. Data requests originate from the PC workstation. The tools sends the requests to the mainframe for processing.

When processing is finished, the data is returned to the PC and formatted. These types of tools typically use a graphical user interface with pull-down menus and point-and-click functionality. These features are not available on mainframe products.

Another increasingly popular approach to developing DB2 applications is to create a similar environment on the PC. This can be done using a PC DBMS that works like DB2 and other similar PC products that mimic the mainframe (COBOL, IMS/TM, CICS, JCL, etc.)

Quite often, tools that can be used in a straight PC environment can also be used in a client/server environment.

Plan Analysis Tools

The development of SQL to access DB2 tables is the responsibility of an application development team. With SQL's flexibil-

ity, the same request can be made in different ways. Because some of these ways are inefficient, the performance of an application's SQL could fluctuate wildly unless it is analyzed by an expert before implementation.

The DB2 EXPLAIN command provides information about the access paths used by SQL queries by parsing SQL in application programs and placing encoded output into a DB2 PLAN_TABLE. To gauge efficiency, a DBA must decode the PLAN_TABLE data and determine if a more efficient access path is available.

SQL code reviews are required to ensure that optimal SQL design techniques are used. SQL code walkthroughs typically are performed by a DBA or someone with experience in SQL coding. This walkthrough must consist of reviews of the SQL statements, the selected access paths, and the program code in which the SQL is embedded. It also includes an evaluation of the RUNSTATS information to ascertain whether production-level statistics were used at the time of the EXPLAIN.

A line-by-line review of application source code and EXPLAIN output is tedious, prone to error, and can cause application backlogs. A plan analysis tool can greatly simplify this process by automating major portions of the code review process. A plan analysis tool typically can:

- Analyze the SQL in an application program, describing the access paths chosen in a graphic format, an English description, or both;
- Issue warnings when specific SQL constructs are encountered. For example, each time a sort is requested (by ORDER BY, GROUP BY, or DISTINCT) a message is presented informing the user of the requisite sort;
- Suggest alternative SQL solutions based on an "expert system" that reads SQL statements

and their corresponding PLAN_TABLE entries and poses alternate SQL options;

- Extend the rules used by the "expert system" to capture site-specific rules;
- Analyze at the subsystem, application, plan, package, or SQL statement level; and
- Store multiple versions of EXPLAIN output and create performance comparison and plan history reports.

Currently, no tool can analyze the performance of the COBOL code in which the SQL is embedded. For example, consider an application program that embeds a single SELECT inside a loop. The single SELECT requests a single row based on a predicate, checking for the primary key of that table. The primary key value is changed for each iteration of the loop such that the entire table is read from the lowest key value to the highest key value.

A plan analysis tool probably will not flag the SQL statement because the predicate value is for the primary key, which causes an indexed access. It could be more efficient to code a cursor, without a predicate, to retrieve every row of the table, then fetch each row one by one. This method might use sequential prefetch or query I/O parallelism, reducing I/O and elapsed time and thereby enhancing performance. This type of design problem can be caught only by a trained analyst during a code walkthrough. Although a plan analysis tool significantly reduces the effort involved in the code review process, it cannot eliminate it.

Two features that are required for any plan analysis tool are that it must be capable of interpreting standard DB2 EXPLAIN output and it must automatically scan application source code and PLAN_TABLEs, reporting on the selected access paths and the predicted performance.

Performance Monitors

Performance monitoring and tuning can be one of the most time-consuming tasks for large or critical DB2 applications. DB2 performance monitoring and analysis tools support many performance-oriented requests in many ways. For example, DB2 performance tools can operate:

- In the background mode as a batch job reporting on performance statistics written by the DB2 trace facility;
- In the foreground mode as an online monitor that either traps DB2 trace information using the instrumentation facility interface or captures information from DB2 control blocks as DB2 applications execute;
- By sampling the DB2 and user address spaces as the program runs and by capturing information about the performance of the job independent of DB2 traces;
- By capturing DB2 trace information and maintaining it in a history file (or table) for producing historical performance reports and for predicting performance trends;
- As a capacity planning device by giving the tool statistical information about a DB2 application and the environment in which it will operate; or
- As an after-the-fact analysis tool on a PC workstation for analyzing and graphing all aspects of DB2 application performance and system-wide DB2 performance.

DB2 performance tools support one or more of these features. The evaluation of DB2 performance monitors is a complex task. In some companies, more than one performance monitor is used at a single site.

DB2 Performance Enhancing Tools

Performance is an important facet of DB2 database administra-

tion. Many shops dedicate several analysts to tweaking and tuning SQL, DB2 and its environment, to elicit every performance enhancement possible. If your shop falls into this category, several tools on the market enhance the performance of DB2 by adding functionality directly to DB2. These DB2 performance tools can interact with the base code of DB2 and provide enhanced performance. Typically, these products take advantage of known DB2 shortcomings.

For example, products exist which:

- Enable DSNZPARMs to be changed without recycling DB2;
- Enhance the performance of reading a DB2 page; and
- Enhance DB2 bufferpool processing.

Care must be taken when evaluating DB2 performance tools. New releases of DB2 may reduce the need for these tools because functionality was added or a known shortcoming was corrected. However, this does not mean that you should not consider performance tools. They can pay for themselves after only a short period of time. Discarding the tool when DB2 supports its functionality is not a problem if the tool has already paid for itself in terms of better performance.

One final caution: Because these tools interact very closely with DB2, be careful when migrating to a new release of DB2 or a new release of the tool. Extra testing should be performed with these tools because of how they interact with DB2.

DB2 Programming and Development Tools

Many tools enhance the DB2 application development effort. These DB2 programming and development tools can:

- Enable the testing of SQL statements in a program editor as the programmer codes the SQL;
- Perform predictive performance

estimation for programs and SQL statements;

- Explain SQL statements in an edit session;
- Generate complete code from in-depth specifications. Some tools even generate SQL. When code generators are used, great care should be taken to ensure that the generated code is efficient before promoting it to production status; and
- Use 4GLs that interface to DB2 and extend the capabilities of SQL to include procedural functions (such as looping or row-at-a-time processing).

Due to the variable nature of the different types of DB2 programming tools, they should be evaluated case by case.

QMF Enhancement Tools

A special category of tool, supporting QMF instead of DB2, automatically creates COBOL programs from stored QMF queries. QMF provides a vehicle for the ad hoc development, storage, and execution of SQL statements. When an ad hoc query is developed, it often must be stored and periodically executed. This is possible with QMF, but QMF can execute only dynamic SQL. It does not support static SQL. A method of running critical stored queries using static SQL would be beneficial, because static SQL generally provides better performance than dynamic SQL.

QMF enhancement tools convert the queries, forms, and procs stored in QMF into static SQL statements embedded in a COBOL program. The COBOL program does all the data retrieval and formatting that is performed by QMF, providing the same report as QMF would. However, the report is now created using static SQL instead of dynamic SQL, thereby boosting performance.

Query Tools

DB2 provides the SPUFI

query tool bundled with the DBMS. Most organizations find SPUFI inadequate, however, in developing professional, formatted reports or complete applications. It can be inadequate also for inexperienced users or those who want to develop or execute ad hoc queries.

QMF addresses each of these deficiencies. The capability to format reports without programming is probably the greatest asset of QMF. This feature makes QMF ideal for use as an ad hoc query tool for users.

Another feature is the capability to develop data manipulation requests without using SQL. QMF provides QBE and Prompted Query in addition to SQL.

QBE, or Query By Example, is a language in itself. The user makes data manipulation requests graphically by coding keywords in the columns of a tabular representation of the table to be accessed.

Prompted Query builds a query by prompting the end user for information about the data to be retrieved. The user selects a menu option and Prompted Query asks a series of questions, the answers to which are used by QMF to build DML. Both QBE and Prompted Query build SQL "behind the scenes" based on the information provided by the end user.

QMF can also be used to build application systems. A QMF application accesses DB2 data in three ways:

- Using the QMF SAA Callable Interface from an application program;
- Using the QMF Command Interface (QMFCI) in a CLIST to access QMF; or
- Using a QMF procedure.

QMF provides many built-in features that can be used by application programs to reduce development cost and time. For example, QMF can display online reports that scroll not only up and down but also left and right. (Cod-

ing left and right scrolling in an application program is not a trivial task.) QMF can also issue the proper form of dynamic SQL, removing the burden of doing so from the novice programmer.

Another benefit of QMF is that you can use inherent QMF commands such as EXPORT, DRAW, and SET to accomplish tasks that are difficult to perform with a high-level language such as COBOL.

QMF, however, is not the only game in town. Other vendors provide different DB2 table query and reporting tools that can be used to enhance DB2's ad hoc query capabilities. Some of these products are similar in functionality to QMF, but provide additional capabilities. They can:

- Use static SQL rather than dynamic SQL for stored queries;
- Provide standard query formats and bundled reports;
- Provide access to other file formats such as VSAM data sets or IMS databases in conjunction with access to DB2 tables;
- Provide access from IMS/DC (QMF is supported in TSO and CICS only); and
- Execute DB2 commands from the query tool.

Tools that operate on workstations and PCs are becoming more popular than their mainframe counterparts (such as QMF). This is because the PC provides an environment that is more conducive to quickly creating a report from raw data. Using point-and-click, drag-and-drop technology greatly eases the report generation process.

Finally, 4GLs are gaining more and more popularity for accessing DB2 data. Though not a typical type of DB2 add-on tool, these products provide more functionality than a report writing tool, but with the GUI front-end that makes them easier to use than 3GL programming languages such as COBOL and C.

4GL tools typically work in one of three ways:

- Queries are developed using 4GL syntax, which then is converted "behind the scenes" into SQL queries;
- SQL is embedded in the 4GL code and executed much like SQL embedded in a 3GL; or
- A hybrid of these two methods, in which the executed SQL is either difficult or impossible to review.

Referential Integrity Tools

Referential Integrity (RI) has been available in DB2 since DB2 V2.1. However, it has always been difficult to administer and implement. RI tools eliminate the difficulty by:

- Analyzing data for both system and user managed referential integrity constraint violation;
- Executing faster than the IBM-provided CHECK utility;
- Enabling more efficient implementation of RI constraints; and
- Enabling additional types of RI to be supported; for example, analyzing primary keys for which no foreign keys exist and deleting the primary key row (pendant delete processing).

Security Tools

DB2 security is provided internally to DB2 with the GRANT and REVOKE data control language components of SQL. Using this mechanism, authorization is granted explicitly and implicitly to users of DB2. Authorization exits enable DB2 to communicate with RACF, IBM's mainframe security management package. This eases the administrative burden of DB2 security by enabling the corporate data security function to administer groups of users. DB2 authorization then is granted to the RACF groups, instead of individual userids. This decreases the volume of security requests that must be processed by DB2.

DB2's implementation of security has several problems. Paramount among these deficiencies is the effect of the cascading REVOKE. If an authority is revoked from one user who previously granted authority to other users, all dependent authorizations also are revoked.

This problem can be addressed by a DB2 security add-on tool. These tools typically analyze the effects of a REVOKE. These tools enable the user to revoke the authority and optionally reassign all dependent authority either by storing the appropriate GRANT statements to reapply the authorizations implicitly revoked or by revoking the authority and automatically reapplying all implicit revokes in the background.

These tools also provide other functions. Consider the administrative overhead when DB2 users are hired, quit, or are transferred. Security must be added or removed. A good security tool enables a user to issue a GRANT LIKE command, which can copy DB2 authority from one DB2 object to another or from one user to another. Suppose that a DBA is transferred to another department. A security tool can assign all of that DBA's authority to another user before revoking his authority. Or suppose that a new DB2 table is created for an existing DB2 application, and it requires the same users to access its data as can access the other tables in the application. This type of tool enables a user to copy all security from one table to the new table.

There is one other type of DB2 security product. Rather than augment DB2 security, however, this type of product replaces DB2 security with an external package. First let's discuss its benefits.

The primary benefit is the consolidation of security. If your organization uses a security package from another vendor rather than RACF for regular data secu-

urity, security administration for regular data security and DB2 security can be consolidated in a single unit. A second benefit is that the cascading revoke effect can be eliminated because MVS data security packages do not cascade security revocations.

The weaknesses of this type of tool, however, far outweigh the benefits. These tools do not conform to the rigorous definition of the relational model, which states that the DBMS must control security. Some do not provide all types of DB2 security. For example, INSTALLSYSADM is still required in DB2 for installation of DB2 and DB2 Catalog and Directory recovery.

Another weakness is that if the external security package fails, DB2 data is unprotected. Finally, these types of external security packages do not use supported DB2 exit control points. As such, they might be unable to provide support for new releases of DB2 in a timely fashion.

DB2 Online Standards Manuals

Products exist which provide "canned" standards for implementing, accessing and administering DB2 databases. These tools are particularly useful for shops new to DB2. By purchasing an online standards manual these shops can quickly come up-to-speed with DB2.

However, mature DB2 shops can also benefit from these types of products if the third party vendor automatically ships updates whenever IBM ships a new release of DB2. This can function as cheap training in the new DB2 release.

A product containing DB2 standards should:

- Provide online access via either the mainframe or a networked PC environment so all developers and DBAs can access the manual;
- Be extensible, so additional

- standards can be added; and
- Be modifiable, so the provided standards can be altered to suit prior shop standards (naming conventions, programming standards, etc.).

Testing Tools

Testing tools enable an application developer or quality assurance analyst to issue a battery of tests against a test base and analyze the results. Testing tools typically are used for all types of applications, but some have been specifically extended to support testing against DB2 tables.

Utility Enhancement Tools

The DB2 CHECK, COPY, LOAD, RECOVER, REORG and UNLOAD utilities are notorious for their inefficiency, sometimes requiring days instead of hours to operate on very large DB2 tables. However, these utilities are required to populate, administer and organize DB2 databases.

Several vendors provide support tools that replace the DB2 utilities and provide the same functionality more efficiently. For example, one vendor claims that its REORG utility executes six to 10 times faster than the DB2 REORG utility. These claims must be substantiated for the applications at your organization, but enough inefficiencies are designed into the IBM DB2 utilities to make this claim believable.

Before committing to an alternate utility tool, be sure that it conforms to the following requirements:

- Does not subvert the integrity of the data in the DB2 tables;
- Minimally provides the same features as the corresponding DB2 utility. For example, if the DB2 REORG utility can REORG both indexes and tablespaces, the enhanced REORG tool must be able to do the same;
- Does not subvert standard DB2 features, when possible. For example, DB2 image copies are maintained in the DB2 Catalog. The enhanced COPY tool, therefore, should store its image copy information there as well;
- Provides an execution time faster than the corresponding DB2 utility. For example, if the DB2 LOAD utility requires 20 minutes to load a table, the enhanced LOAD tool should load the same table in at least 10 minutes. (Of course, this should not be a hard-and-fast rule. Sometimes even a moderate increase in processing time is sufficient to cost-justify a third party utility tool.); and
- Corrects the deficiencies of the standard DB2 utilities, when possible. For example, the DB2 LOAD utility will not load data in sequence by the clustering index. The enhanced tool should provide this capability. Another example is the COPY utility. Until DB2 V2.3, the COPY utility could not take dual image copies with one pass. Tool vendors provided this capability and sold many copies of their enhanced COPY utility as a result.

When testing utility tools from different vendors, ensure that you are conducting fair tests. For example, always reload or recover prior to resting REORG utilities or you may skew your results due to different levels of tablespace organization. Additionally, always run the tests for each tool on the same object with the same amount of data.

Utility Management Tools

One last category of DB2 tool is the utility manager. This type of tool provides administrative support for the creation and execution of DB2 utility jobstreams. These utility generation and management tools can:

- Automatically generate DB2 utility parameters and JCL;

- Monitor DB2 utility jobs as they execute;
- Automatically schedule DB2 utilities when exceptions are triggered; and
- Restart utilities with a minimum of intervention. For example, if a utility cannot be restarted, the tool automatically issues a -TERM UTIL command and resubmits the utility.

Miscellaneous Tools

Many types of DB2 tools are available. The categories in this article cover the major types of DB2 tools, but not all tools can be easily pigeonholed. For example, consider a DB2 table space calculator. It reads table DDL and information on the number of rows in the table to estimate space requirements. A space calculator often is provided with another tool such as a DASD management tool or a database design and modeling tool.

Synopsis

Third party add-on tools can significantly improve the efficiency of DB2 application development. When evaluating products, look for features important to your organization. Consider adopting checklists for product comparisons based upon the features discussed in this article. And remember, although DB2 is a fantastic RDBMS, it leaves quite a bit to be desired in the administration, data access, performance monitoring and application development areas.

Craig Mullins is a senior technical advisor and team leader for analyst relations at PLATINUM technology, inc. His duties include interfacing with industry analyst groups, writing white papers, technical support via the Internet and speaking at user groups and industry conferences. Craig is also a frequent contributor to the *IDUG Solutions Journal*.