

JACKIE'S Forum

Database Performance Tip



Jackie Jansen

In the past I have written articles discussing the two database optimizers in DB2. The newest optimizer is the SQL Query Engine or SQE. The original optimizer is the Classic Query Engine or CQE. IBM's developers in Rochester have done a tremendous job in developing SQE. In nearly all cases you should be striving to have your database queries take advantage of SQE. The performance with SQE is usually considerably better than using CQE to execute the same query.

SQE was first introduced in V5R2 and every release since then has added to its functionality. Basically the query dispatcher analyzes your query looking for any requests that cannot be satisfied with SQE. In those cases the query is passed to CQE and executed as it always was in previous releases. Where possible the query is directed to SQE.

With V5R4 most queries should be using SQE. There are still a few items not supported. The two most common ones that I want to address in this column are querying logical files and the use of select/omit logical files. Some of the other items that still prevent SQE from running an SQL request are the use of National Language Sort Sequence, the UPPER, LOWER or the TRANSLATE function and User-Defined Table Functions (UDTFs).

If you query your data using OPNQRYF or Query/400 then your queries will not go down the SQE path regardless of the syntax. If you are using CLI, ODBC, JDBC or straight SQL then SQE is a viable option.

Logical Files

The most common reason for CQE to be chosen is when you are directly querying a logical file. You will often improve your performance if your query selects the underlying physical file instead. If the database index created with the logical file is

needed it is still available to the optimizer. There are advantages and techniques for replacing logical files with SQL views and/or indexes but that is the topic for a future column. Another alternative might be to create an SQL view to query in place of the logical file. SQL views can be used by SQE.

Select/Omit Logical Files

If you are querying a physical file that has a Select/Omit logical file over it then your query may start down the SQE route but as soon as the optimizer recognizes the existence of the SELECT/OMIT logical file the query is sent back to CQE. SQE doesn't understand derived logical files so when it finds a Select/Omit logical file that might help your performance the query is rerouted back to CQE for optimization. In a very few cases the Select/Omit logical may be beneficial to your query and CQE will outperform SQE. In the majority of cases, the derived logical isn't required for optimum query execution.

QAQQINI

To prevent the optimizer from returning to CQE when you have Select/Omit logicals you can use the "IGNORE_DERIVED_INDEX" option in the QAQQINI table.

The easiest way to create or modify a QAQQINI table is with iSeries Navigator. Within iSeries Navigator expand your system name, and then expand Databases. Right-click on your database (system name) and select 'Change Query Attributes'. This will allow you to easily create a new QAQQINI table or edit an existing one. You can also get to the 'Change Query Attributes' screen via 'Run SQL Scripts'.

SQE or CQE?

With V5R4 you can look at the SQL Plan Cache and see all the SQE queries that have been run recently. If your query doesn't appear it is likely because it ran under CQE.

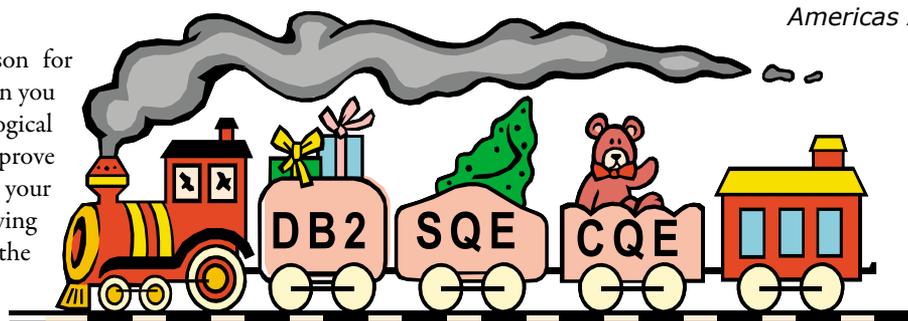
If you can't use the plan cache you can use the database monitor to analyze your SQE and CQE queries. With V5R4 the new analysis overview makes it very quick and easy to identify which queries use CQE and which ones use SQE. With earlier releases you need to query the database monitor tables. There are sample queries that can be downloaded from the IBM web site. One column to be aware of is QQC16. A 'Y' in this column means SQE and 'N' means CQE.

If you have a query that runs under SQE and you want to do a performance test against CQE you can use the UPPER clause. Since UPPER is not currently supported by SQE adding WHERE UPPER('n') = 'N' to your SELECT statement will force your query down the CQE path.

For those of you who take the time to read and implement changes based on this column you should have a nice Christmas present. Extra cycles are always welcome.



Jackie Jansen is a Senior Consulting IT Specialist. She currently works in the IBM Americas Advanced Technical Support Solutions Centre. Jackie is a frequent speaker at iSeries Technical Conferences and User Group meetings. Contact her at jjansen@ca.ibm.com



Happy New Year! Here is my New Year's present to you — how to free up cycles on your System i.