

SQL – Should You or Shouldn't You?



By Alison Butterill

Have you ever spent hours coding an RPG or COBOL process to handle the selection of records from a file? Or how about those search routines looking for a specific character string in a field? How about a routine for reading through records and accumulating a total? All of these functions and more can be done more easily by using some SQL instead of coding the logic in a high level language. Yet, it has taken a few years for SQL to gain momentum in the System i community. SQL is certainly the industry standard query language. It is also IBM's standard language for accessing DB2 databases.

If it allows you to create this complicated code more easily, why doesn't everyone use SQL? Frankly, I believe it is the history of application design on the System i family of servers and it is the familiarity factor that many traditional developers have trouble hurdling. To a lesser degree, it is certainly a statement of having to pay an additional charge to get the SQL development tools and pre-compilers necessary for using SQL on your System i machine.

From a historical perspective, we live with the heritage of traditional System i application code. On the S/36 and the S/38, there was no SQL for data access. The only option was to use traditional READ and WRITE statements. This was the same in the initial days of the AS/400. The only query tool was Query/400. Millions of lines of code have been written using these native facilities—language constructs for data manipulation and Query/400.

Familiarity with these native functions means that application architects have built their designs around using these facilities.



New TUG Booth at IBM i5 Roadshow

IBM graciously allowed TUG to "show our colours" at the recent Power Up Roadshow in Markham. Thanks IBM! We appreciate the support you give to your local users groups.

The lineup of events continues... See the IBM Event Website for more details:
<https://www-926.ibm.com/events/ca/canada.nsf/en>



Customers and IBMers (including, at far right: Kathy Gregson System i Advocate & Programs Manager IBM Canada System i) visit the TUG booth at the IBM Power Up Roadshow in Markham

Native READs and WRITEs are optimized for single record access. SQL is optimized for retrieving and using groups of records, not single record selection. Designing an application to use SQL requires some effort to learn new techniques and methods. If we're talking about re-designing a traditional application, there is significant work to re-engineer the application code—changing from single record retrieval to maximizing the use of result sets. Many architects and application designers feel that it is much easier to leave things the way that they are.

“Nah, we don't need to use SQL,” I hear you say. And yet, most of us are already using it, albeit, without knowing it. Tool vendors use SQL under the covers, just disguising it by a friendlier user interface. Business intelligence tools and those that do management reporting are typically using inquiries built with SQL, to retrieve pertinent data from the database. These vendors have chosen SQL because of its ability to do not only retrieval, but manipulation during the retrieval process. In many cases, time and system resource can be saved by executing data access using an SQL statement.

IBM uses SQL in its tools also. For example, iSeries Access for Web uses SQL under the covers to do data access. SQL is the basis for IBM's Query Manager Facility (QMF) product. QMF should not be confused with Query/400. QMF allows the user to exploit almost all of the strengths of SQL. Even better, QMF allows substitution variables, so when a QMF query is called from an application or program, some flexibility is gained by passing parameters at initiation time.


In the past, SQL on the System i has not always had favourable press. In the beginning, SQL was ported to the AS/400 and implemented in release V1.1.3. The press went crazy talking about the poor performance of SQL on the AS/400. In the early 90's, both IBM and AS/400 programmers had a lot to learn about SQL. AS/400 programmers were not knowledgeable about how to exploit the strengths of SQL and the ported version of the SQL environment was not the most efficient. Since those early days, IBM has significantly improved the efficiency of the SQL environment and optimized the performance of the SQL. The bad reputation, once somewhat deserved, is no longer appropriate. At the same time, programmers are becoming more skilled at the proper use and coding of SQL. Performance is no longer an issue.

Finally, SQL for i5/OS is a bit of a rebel in the database access world. SQL on all other platforms is both the data manipulation language and the data definition language for relational database implementations. On the System i, SQL can coexist with native function. You can use SQL data manipulation against files that were created with DDS. You can use SQL data definition language to create tables and then use native reads and writes to access the data in those tables. This dual definition and manipulation strategy is unique in the industry. From a System i developer perspective, it means that the move to SQL can be done gradually.

Continuing in its rebellious implementation, SQL on the System i comes in two pieces:

- 1.) There is the development environment, which does require a specific license—and has a cost that is in addition to the cost of the database which comes embedded in i5/OS.
- 2.) Then there is the runtime environment, the cost of which is included in i5/OS.

This means that a development shop can purchase one license for the complete SQL Development Kit. They can use it to create applications and then deploy those applications to any System i box, knowing that the code will execute. As I said earlier, this is quite different from other SQL implementations in the market. In most other SQL implementations, you need a license for each machine on which you intend to run the finished application.

So, back to the initial question: “SQL, Should you or shouldn't you?” The answer is that you should use SQL. It is where IBM is investing its development dollars. SQL is both the industry standard and the IBM standard. There will be many more functions and features being added via SQL to other products, such as iSeries Access and QMF. It is the way of the future—we need to get ready! 

Alison Butterill is an IBM Certified Consulting I/T Specialist. Her current position is System i Application Development Offering and Strategy manager for the System i Brand Team. She can be reached at butteril@ca.ibm.com.

Don't let this be you!



Cost Effective Continuity Solutions

- ◆ AS/400, iSeries & i5
- ◆ Wintel & Linux
- ◆ Unix
- ◆ AIX

Business Continuity Solutions, as well as
Disaster Recovery Hot-Site



905-841-9891
sales@dynamiccdr.com
www.dynamiccdr.com