AS/400 SQL or Sequel? A Matter of Semantics?

By Kevin Montgomery

W *Editor's Note: SQL is a great tool in the hands of a competent user, and very dangerous in the hands of an incompetent user. I'll let you be the judge of which one I was…let me just say that I remember having to key a bunch of data into the F4072 once after I deleted another consultant's setup. (She was very nice about it, she just gave me 125 screen prints and said she needed them put back within 24 hours.)*

Introduction

Before we get into what an AS/400 SQL can and cannot do, let's talk about the name itself. SQL is an acronym for 'Structured Query Language,' also called Sequel, and there's the problem, because the terms are sometimes interchangeable. You can read the letters SQL and pronounce it 'ES QUE EL', or you can pronounce it 'SEQUEL,' as in, 'I program in Sequel.' So if you put on your resume, 'I run an SQL server,' and your prospective employer pronounces it 'Sequel,' he might just question your grammatical competence. It's much better to say, 'I run *three* SQL servers,' thereby eliminating the grammatical conflict while managing to look better in other obvious ways.

With that out of the way, this White Paper focuses on the benefits, disadvantages, and limitations of SQL language, and also introduces some ways that you can use SQL in a JDE environment.

Benefits of AS/400 SQL

SQL is a programming language like any other in what it can do, but different in some important ways. First of all, it's the easiest language by far to learn. There are only 6 main key words, and 3 'action' statements, INSERT, UPDATE, DELETE. You can get by for years using only about 4 key words: SELECT, FROM, WHERE, and GROUP BY. If you want to sort, insert an ORDER BY statement. Power users sometimes throw in a HAVING clause, but that's just to impress their colleagues. Naturally, there are some fancy things you can do, like calling a UNION statement for subtotals, using subqueries for filtering, and using built-in functions, but it's so simple to do the most complex things, that SQL has to be the most powerful language around.

SQL on the AS/400 also doesn't have to be compiled. It's command-driven, and it executes very fast. By the time you wrote an RPG program to look up a list of orders for a parent between two dates, or waited for a job queue to run your orldWorldWriter, you could have given your customer his answer and gone to lunch.

Disadvantages of SQL

You can wipe out Open Orders in three words if you forget the 'WHERE' statement, possibly affecting your performance review, if you ever get one. *Don't try this without adult supervision: DELETE FROM F4211. It's a pretty cool command.*

Limitations

SQL is a language that has been around for about 15 years. In that time, it's evolved like a snail (not very much and not very fast). The one big change was in the early 90's when someone invented the 'JOIN'

statement, which qualifies the way you match records between files. Unfortunately, AS/400 has evolved not like a snail, but like an ameba (even slower than a snail). There's no JOIN statement in AS/400 SQL. There's no 'ROUND' or 'RIGHT' statements, either. And don't even *try* to turn a string like '101.25' into a number. Date arithmetic has some good functions, but not like SQL Server, Access, or Crystal Reports, and everything except field names and commands is case sensitive.

Other SQL platforms let you create a table 'on the fly' with a 'Select * Into…' statement. Not the AS/400. You need to CRTDUPOBJ first or use the CREATE statement.

All SQL languages differ in some ways, but there's usually a way to get around the syntax. MS SQL server uses a 'CASE' statement to do a cross-tab query (months as columns, products as rows, get the quantities for each month). Microsoft Access uses a crazy 'Transform' statement; you create it with a wizard, and then look at the code, scratch your head, and say, 'Oh yeah, I get it now.' Not the AS/400. Can't do it. Use a DreamWriter.

Uses

In spite of limitations, there are things that AS/400 SQL includes that you might not expect. There's a substring function that can extract portions of a string from any position. There's no RIGHT function, but there's a STRIP function, which eliminates spaces. You can get the 'day' number of any date field, just as you can in Excel. You can determine the length of a string, and UPDATE statements are a little awkward, but still manageable.

With these functions plus the arsenal of normal SQL commands, you can do a lot of great things in AS/400 SQL and JDE. In the following examples, we'll see some ways to get around some of the limitations, and perhaps it will give you an idea of ways you can accomplish your job without excessive programming.

Beginning SQL

Most SQL users rely on the SELECT and WHERE clauses to give them quick info on what's going on. These key words retrieve field names to the screen, and the syntax can't hurt anything. This is where you should start. A simple SQL command could read records from the Price Adjustment File (F4072) and see if they're properly set up. The '*' means get all fields. Otherwise, list them out separated by a comma.

Start your SQL session with STRSQL from a command prompt, then type in:

SELECT * FROM F4072 WHERE ADAST = 'DISCOUNT'

Upper or lower case doesn't matter, except for the word 'DISCOUNT.' Also, the AS/400 doesn't like double quotes. All other SQL's will know what you mean if you mix caps or use double-quotes.

In a few seconds you'll see values on the screen. Hit the Page Down key to view more records. Pressing ENTER gets you back to an SQL command screen.

Feeling courageous? Change the ASDSV1 field to another value if it's 'AB.' No, don't ask another programmer. You can do it. Have confidence. Be *bold!!!* Solve problems. Re-do your resume first….

This Article Continues…

Subscribers, log in from our main search page to access the full article:

www.JDEtips.com/MyAccess.html

Not a Subscriber? Gain access to our full library of JDE topics:

www.JDEtips.com/JD-Edwards-Library