Craig S. Mullins

The Buffer Pool

The Hitch-Hiker's Guide to DB2
By Craig S. Mullins

DB2 is a complex piece of system software that can be very intimidating for users to learn and master. There are a lot of details, parameter, syntax choices, and manuals that need to be understood. Difficulties can arise for a number of different reasons. This hitchhiker's guide offers a handy roadmap to DB2 developers for where and how to find information, as well as clears up some confusing issues for DB2 users.

Basic Field Resources

As a technician using DB2 you need to make sure you are armed with the proper resources to utilize DB2 to its fullest capabilities. So, if you use DB2 as a developer, DBA, programmer, or analyst the first thing you should do is make sure you have a complete set of up-to-date manuals. Doing so is much easier than it used to be. IBM offers free downloads of all of the necessary DB2 manuals on its web site. IBM regularly updates these manuals – not just for new versions – but to add feature coverage, clarify explanations, and provide additional examples. Be sure to regularly check the IBM web site at http://www-306.ibm.com/software/data/db2/library/ for new manuals. The manuals are in Adobe Acrobat format, so you will need the free Adobe Acrobat Reader software which can be downloaded from the Adobe web site (http://www.adobe.com). The DB2 for z/OS V8 manuals are as follows:

Administration Guide

Application Programming Guide and Reference for Java

Application Programming and SQL Guide

Command Reference

Data Sharing: Planning and Administration

Diagnosis Guide and Reference

Diagnostic Quick Reference

Installation Guide

Messages and Codes

ODBC Guide and Reference

RACF Access Control Module Guide

Reference for Remote DRDA Requesters and Servers

Reference Summary

Release Planning Guide

SQL Reference

Utility Guide and Reference

Image, Audio, and Video Extenders Administration and Programming

Text Extender Administration and Programming

XML Extender Administration and Programming

What's New in DB2 Version 8?

Typically, programmers use the *SQL Reference* and the *DB2 Messages and Codes* manuals more than any of the others. The *SQL Reference* contains the syntax of every DB2 SQL statement; the *DB2 Messages and Codes* manual contains explanations of DB2 and SQL error messages. DBAs tend to use those two manuals as well, but I'd add the *Administration Guide*, *Command Reference*, and *Utility Guide & Reference* to this most regularly used list.

Of course, you should consider augmenting these manuals with IBM red books. IBM red books provide in-depth, detailed coverage of a specific technology topic. IBM publishes red books on multiple subjects, but there are many educational DB2 red books that can greatly assist DB2 technicians working to understand a feature or nuance of DB2. You can download IBM red books for

free in Adobe Acrobat format over the web at the following link: http://publib-b.boulder.ibm.com/Redbooks.nsf/Portals

You cannot easily digest every topic by reading a manual or a red book, though. Sometimes you just need to interact with someone else to gain the requisite understanding. Fortunately, there are many options for doing so. One of the best is the DB2 mailing list that is managed by IDUG. Also known as simply DB2-L or the DB2 Listserve, you can subscribe to it by sending a message to the subscription address, LISTSERV@WWW.IDUG.ORG. The message should read as follows:

SUBSCRIBE DB2-L

After issuing this command, the list server will send you a message asking you to confirm the subscription. Upon doing so, information will quickly begin flowing into your e-mail box (perhaps at a much quicker rate than you can reasonably digest). Literally, hundreds of messages may be sent to you every week. You can post messages asking questions, or read and answer the

questions of others. DB2-L is basically a big, online user group facilitated by e-mail.

To sign off of the newsgroup, send the following message to the same subscription address:

    SIGNOFF DB2-L

All of these commands, as well as many others, can be performed on the web, too. Simply access the DB2-L page at the following link and logon:

http://www.idugdb2-l.org/adminscripts/wa.exe?REPORT&z=3

In addition to a subscription address, mailing lists also have a posting address. This is the address to which mailing list posts must be sent. Never send subscription requests to the list's posting address. Correspondingly, never send a post to the subscription address. The posting address for the DB2-L mailing list is DB2-L@WWW.IDUG.ORG. When a message is sent to this address, it will automatically be forwarded to everyone currently subscribed to the list. Postings to the DB2 mailing list are archived so you can find old messages of interest that you might not have saved. Use this link to access the archive of past DB2-L postings:

Or you can send e-mail commands to navigate the archives. You can get a list of the available archive files by sending the following command to LISTSERV@IDUG.ORG:

    INDEX DB2-L

The files returned can be ordered using the following command:

    GET DB2-L LOGxxxx

The IDUG Insider is another way of discussing DB2 topics with knowledgeable DB2 professionals. IDUG Insider offers a technical library of DB2 information and a user discussion forum. It can be accessed over the IDUG web site. Other web sites, such as www.dbazine.com and www.searchdatabase.com also offer DB2 content and Q+A forums.

And don't forget to check out the IBM-sponsored DB2 Developer Works site. This site offers a vast array of DB2 technical articles, with new ones appearing all the time.

Check it out at
 http://www.ibm.com/software/data/developer .

Finally, be sure to foster face-to-face relationships at the annual IDUG conferences in your region. IDUG hosts annual events in North America, Canada, Europe, and Australia. And support your regional user groups. Participating in your local user group is the best way of building a network of helpful DB2 professionals to interact with, and to call upon for assistance.

SELECT Syntax

Now that you have your battle plan mapped out for learning about and keeping up to date with DB2 technology, it is time to hunker down and start using the best database in the world. Most of us usually start to use DB2 by writing SQL.

Most DB2 programmers think they know how to correctly code simple SQL SELECT statements. And they usually are correct, as long as you keep that adjective "simple" in the assertion. When the statement requires more than SELECT...FROM...WHERE though, problems can ensue.

The biggest SELECT problem encountered by DB2 users is almost always related to syntax. To paraphrase Mark Twain, sometimes what people think they know, just ain't so.

How can you find the proper syntax for SELECT statements? The DB2 SQL Reference manual contains all of the syntax for DB2 SQL, but query syntax is separated from the rest of the language. Typically, users go to Chapter 5 of the SQL Reference that contains syntax diagrams, semantic descriptions, rules, and examples of the use of DB2 SQL statements. SELECT INTO is there, but SELECT is not in this section. (Actually, IBM corrected this in the second edition of the SQL Reference for DB2 Version 7, August 2001. Now there is a section for SELECT in Chapter 5 that refers the reader back to Chapter 4.) Chapter 4 contains the detailed syntax information and usage details for DB2 queries using SELECT.

A further confusing aspect of DB2 SELECT is the breakdown of SELECT into three topics: fullselect, subselect, and select-statement. This causes many developers to confuse which query options are available to the SELECT statements they want to code.

The **select-statement** is the form of a query that can be directly specified in a DECLARE CURSOR statement, or prepared and then referenced in a DECLARE CURSOR statement. It is the thing most people think of when they think of SELECT in all its glory. If so desired, it can be issued interactively using SPUFI. The select-statement consists of a fullselect, and any of the following optional clauses: order-by, fetch-first, update, read-only, optimize-for, isolation and queryno.

A **fullselect** can be part of a select-statement, a CREATE VIEW statement, or an INSERT statement. This sometimes confuses folks as they try to put a FETCH FIRST n ROWS clause or an ORDER BY in a view or as part of an INSERT. That's not allowed! I recently had a conversation with a guy who swore that at one point he created a view using the WITH UR clause and that it worked. In fact, a fullselect does *not* allow any of the following clauses: ORDER BY, FOR READ ONLY, FOR FETCH ONLY, FOR UPDATE OF, OPTIMIZE FOR, WITH, QUERYNO, and FETCH FIRST. A fullselect specifies a result table – and none of these afore-mentioned clauses apply.

Finally, a **subselect** is a component of the fullselect. A subselect specifies a result table derived from the result of its first FROM clause. The derivation can be described as a sequence of operations in which the result of each operation is input for the next.

This is all a bit confusing. Think of it this way: in a subselect you specify the FROM to get the tables, the WHERE to get the conditions, GROUP BY to get aggregation, HAVING to get the conditions on the aggregated data, and the SELECT clause to get the actual columns. In a fullselect you add in the UNION to combine subselects and other fullselects. Finally, you add on any optional order-by, fetch-first, update, read-only, optimize-for, isolation and queryno clauses to get the select-statement.

Synopsis

It can be confusing to get your hands around the many options and statements available to you as a user of DB2 for z/OS. Hopefully this short hitch-hiker's guide has helped you to better navigate the confusing sea of DB2 development.

From *IDUG Solutions Journal*, Summer 2004.

Home.