

M. Michelle Buchecker, SAS Institute Inc.,
Chicago, IL

ABSTRACT

The SQL Query window is a point and click interface that enables you to easily access your data. As a component of SAS/ASSIST® software, it provides a syntax-free way of writing code and producing reports. You can also create new SAS data sets, both SAS data files and SAS data views, to be used later.

In this paper, working on a Microsoft Windows platform, you'll see how to join MVS DB2 tables without having to know the column names in advance. You'll also see how to create SAS data sets from the results and store those data on a Windows or MVS platform.

INTRODUCTION

This paper discusses an easy way to use data in a relational database to produce reports and SAS data sets. No previous SAS software knowledge is necessary since no code is being written. The SAS System will write it for you! Likewise, database knowledge is helpful but not required to derive the benefits discussed here.

The products shown in this paper are

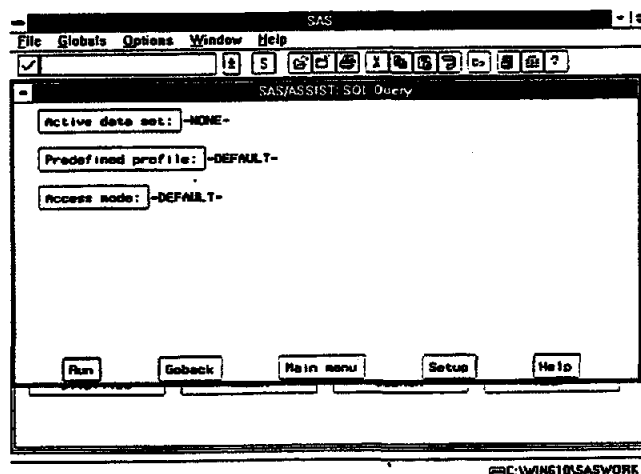
- base SAS software (obviously!)
- SAS/ASSIST® software
- SAS/CONNECT® software
- SAS/ACCESS® software.

While this paper shows examples of Windows to DB2, you are not limited to these. The concepts apply just as well from OS/2® to ORACLE, or from any operating system to any of these other databases:

- the SAS System
- ODBC
- SYBASE
- SQL/DS®
- Rdb
- DB2/2®
- INGRES
- INFORMIX.

INVOKING THE SQL QUERY WINDOW

To invoke the SQL Query window, from SAS/ASSIST select **Report Writing** → **Query** → **SQL Query**. This displays the main SQL Query window.



Display 1 Default SQL Query Window Primary Display

From here you may select

- a data set
- a predefined profile
- an access mode.

Currently, there is no specific SAS data set you want to access, nor do you have a profile created. Since there is no profile created yet, changing the access mode from the default (SAS) to DB2 will not give us the desired results. So let all fields here default and select **Run**.

DEFINING A PROFILE

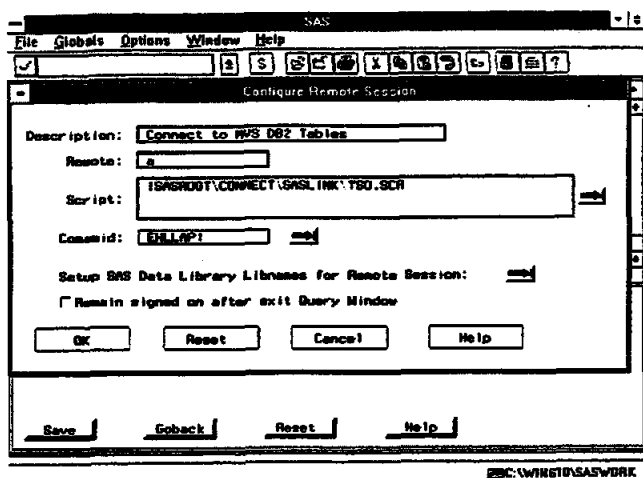
At the SQL Query Tables window, a list of the tables defined to your SASUSER library appears on the left side of the window. A list of all SAS data libraries currently defined to your SAS session appears in the middle of your window. Since you want to access DB2 tables instead, select **Profile** → **Set Preferences...** from the menu bar at the top of the window.

You can now create a new profile. This profile should contain

- the access mode (for example, DB2)
- remote session information
 - script name
 - remote session name
 - communications access method (for example, ehilapi)
- access mode options (for example, SSID).

Your access mode options differ depending on which database you choose.

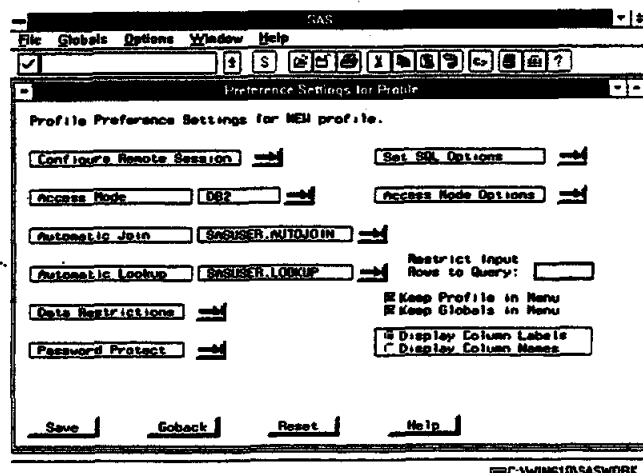
The remote session information uses SAS/CONNECT to establish a connection between your SAS session on your PC (aka "client") and, in this case, the mainframe (aka "server").



Display 2 Sample Remote Session Information

The script file TSO.SCR is a SAS Institute-supplied file that establishes a SAS connection to TSO with the assumption that you are already logged on to TSO and your TSO session is at a READY prompt.

Select **OK** to return to the Preference Settings for Profile window. Select **Access Mode** to specify the type of relational database you wish to access.



Display 3 Sample Access to DB2

You may want to override the DB2 defaults defined for

- system tables
- system columns
- SSID
- system views.

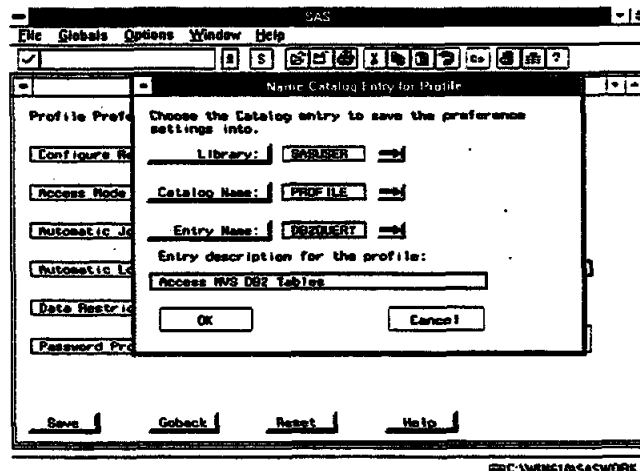
To change any one of these defaults, select **Access Mode Options**. Again, the Access Mode Options will differ depending on which relational database is selected. For example, if you select ORACLE as your Access Mode, your Access Mode Options are

- Username
- Password
- Path
- Synonyms Owner

- Access only tables owned by this username.

Once all of the desired information has been filled in, select **Save** to save this profile as a catalog entry. The default name for this profile is SASUSER.PROFILE.QUERY.QRYPROF. QRYPROF is automatically appended as the catalog type.

You should be as descriptive as possible when saving the profile. For example, save the profile as SASUSER.PROFILE.DB2QUERY.QRYPROF.



Display 4 Saving the Profile

After you select **OK** you automatically return to the Preference Settings for Profile window. A message displays stating that the profile has been saved as well as the name of the profile. Since the profile is now complete, select **Goback** to return to the SQL Query Tables window.

CHANGING TO A PROFILE

Even though a profile was just created, you still need to tell the SAS System to use this profile. To do so, select **Actions** → **Switch to New Profile** from the menu bar. This displays a list of currently defined SQL Query window profiles. Select the desired profile, then select **OK**.

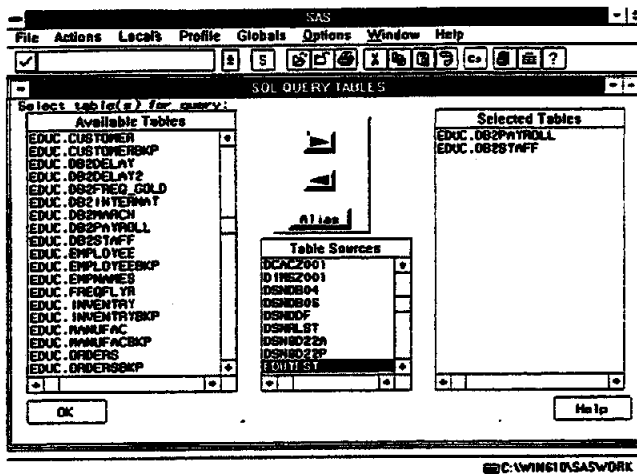
At this point SAS/CONNECT uses the script specified in the profile to establish the connection to your server. Any errors or warning messages are written to the log.

SELECTING DB2 TABLE(S)

Once your connection has been established, the SAS System on the PC sends code to the SAS System on the mainframe (aka a remote submit). This code uses the PROC SQL Pass-Through facility to query the DB2 SYSIBM SYSTABLES requesting the names of all of the DB2 databases. Once DB2 processes the query, the results are returned to the client (the PC) and displayed in the Tables Sources area of the SQL Query window.

Once you select your desired database, the SAS System remote submits another query to the DB2 SYSIBM SYSTABLES requesting the table names and creators (authorization ID) where the database name is the one you selected (and isn't that cool!). The results display in the Available Tables list in the SQL Query window.

Select the desired table (or tables) and click on the right arrow key. This 'moves' the tables into the Selected Tables list.



Display 5 Selecting DB2 Tables

After all of the desired tables are selected, select **OK**.

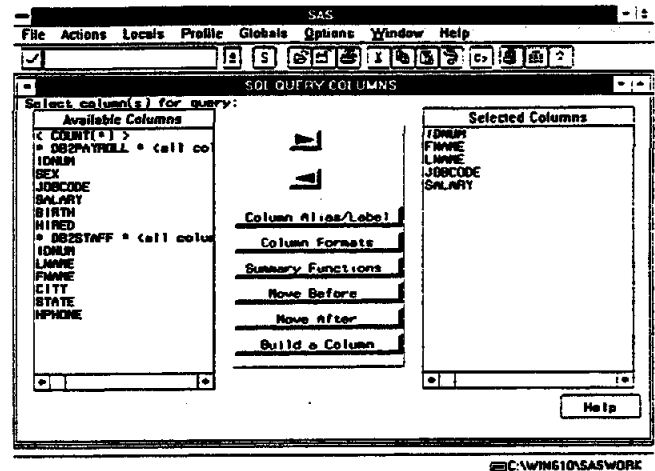
If you are the type of user who is not interested in the behind the scenes detail, at this point, magic happens and DB2 returns the column names of the selected tables to the SAS System. You may now skip the next paragraph!

Behind the scenes, the local SAS session sends a remote submit that

1. creates a SAS SQL view based on the first selected table.
2. issues a `create table` statement with the `like` keyword to copy the descriptor portion of the view (above) to a temporary SAS data file. Therefore, only the variable names and not the data reside in this temporary file.
3. downloads the temporary file to the local SAS session.
4. populates the SQL Query window with the variable names of the downloaded file.
5. erases the remote SAS data view and SAS data file.
6. starts the whole process again on the next selected DB2 table.

SELECTING DB2 COLUMNS

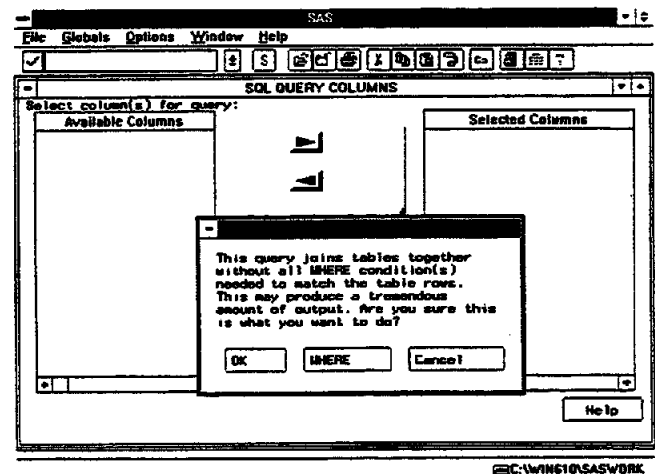
The column names are now displayed in the Available Columns area of the SQL Query Columns window. To create a report, first select the columns you want in the order that you want them. Then select the right arrow in the upper middle part of the display. This moves the columns into the Selected Columns area.



Display 6 Selecting DB2 Columns

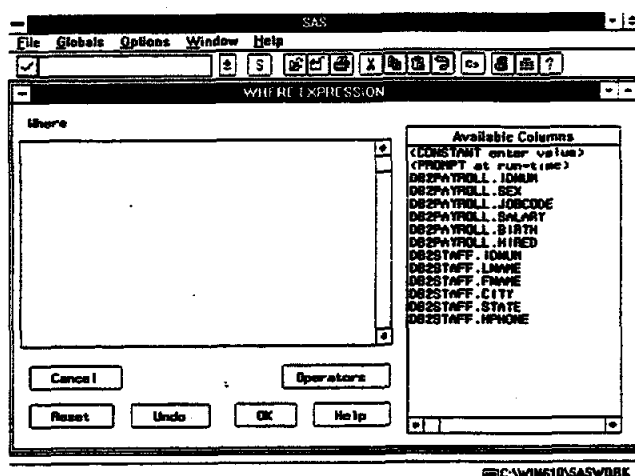
PRODUCING A REPORT

Good enough! Let's produce a report! Select **Actions** → **Run Query** → **Run Immediate** from the menu bar at the top of the display. Since more than one table is being queried, by default SQL joins every row in the first table with every row in the second table. This is bad since it takes up lots of unnecessary processing time. To help prevent you from making this mistake, the SQL Query window asks you if this is what you really want to do.



Display 7 Forgot to Join Tables with Where Clause Message

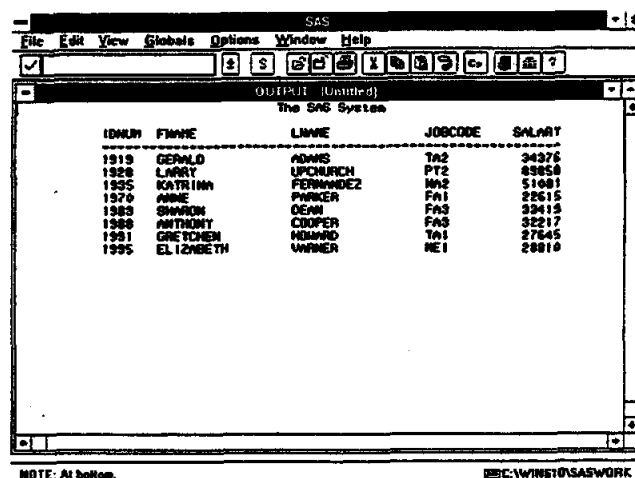
You can select **OK** which joins every row with every other row (which may be what you want). In most cases, however, you want to join the data together with a WHERE clause. When you select **WHERE**, the WHERE Expression window displays.



Display 8 WHERE Expression Window

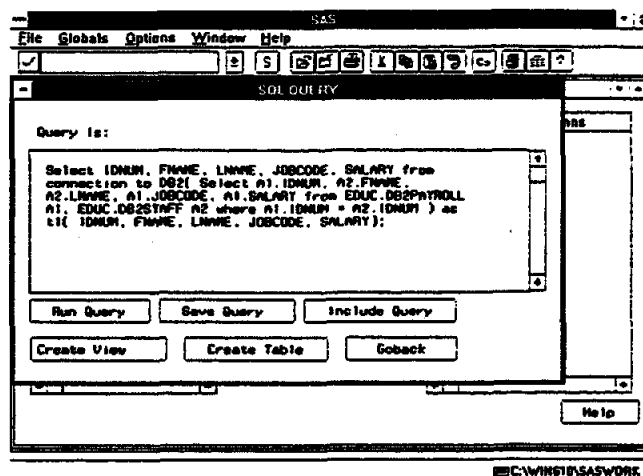
To join the tables, select the information that is common to both tables. In this case, IDNUM is common to both tables. Select DB2PAYROLL.IDNUM. A list of operators automatically appears. Select the operator EQ, then select the column DB2STAFF.IDNUM. When you are finished building your WHERE clause, select **OK**. This returns you to the Columns window.

Now let's try running the report again with **Actions** → **Run Query** → **Run Immediate**. Voila! A report magically displays in the Output window on the PC.



Display 9 Resulting Output

If you are interested in what is going on behind the scenes, return to the SQL Query window by selecting **File** → **End**. Next, select **Actions** → **Show Query...** from the menu bar. This displays the query that was submitted.



Display 10 Submitted Query

Notice that from this window you can

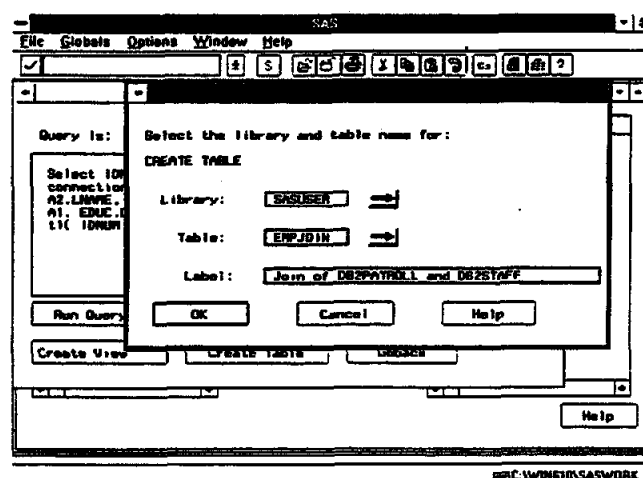
- run this query again
- save this query so you can run it at a later date
- include a previously saved query
- save the results of the query as a SAS data view
- save the results of the query as a SAS data file (table).

If you decide to create a SAS data view, this view will be stored on the server, in this case, the MVS session.

If you decide to create a SAS data file (table), the SQL Query window prompts you if you want to store it on the client or the server.

Example: Create a SAS data file on the client (Windows) named SASUSER.EMPJOIN (for EMPLOYEE JOIN) based on the the current query.

Select **Create Table** → **Create table on local host...**. A window displays prompting you for the library and table names. Optionally, you may supply a description for this table.



Display 11 Supplying Library and Table Names

After specifying the location for the new table, select **OK**. Behind the scenes, the SAS System remotely submits this query preceded by create table that creates a temporary SAS data file on the remote machine. Once this temporary file is created,

it is then downloaded to the local machine and given the name that you specified earlier. Finally, a verification message is written at the top of the window that the table was created and how many rows (observations) it contains.

You now have a local SAS data file of a merge of two DB2 tables without having to write any code!

EASY, BUT WAS IT EFFICIENT?

There is no easier way to get at data than to point and click. But since you have no control over the code and how efficient (or inefficient) it could be, is this really the best way to go? Yes!

- Since the SQL Query window is sending the code generated as a remote submit, the processing occurs where the data are stored.
- This code is using the PROC SQL Pass-Through facility, which means all of the processing is handled by DB2, not the SAS System. DB2 is more efficient than the SAS System when joining DB2 tables. Also, DB2 is more efficient when dealing with summarizing large volumes of DB2 data.

CONCLUSION

In today's world of many databases and many operating systems, the SQL Query window feature of SAS/ASSIST, along with SAS/ACCESS and SAS/CONNECT, provides an easy, integrated and efficient approach to access your data wherever and however they are stored. The SQL Query window also provides features not discussed here (such as summarizing and sorting your data), but are just as easy to use. By allowing you the flexibility of creating SAS data sets on either the local or remote machine, you can store your data where it makes the most sense to your organization. Furthermore, once your SAS data sets are created, you have the full functionality of the entire SAS System at your fingertips!

For more information on the SQL Query window, please refer to SAS® Technical Report P-254, *Using the SQL Query Window*, Release 6.08 (order #C59171).

SAS, SAS/ACCESS, SAS/ASSIST, and SAS/CONNECT are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. IBM, DB2, DB2/2, OS/2 and SQL/DS are registered trademarks or trademarks of International Business Machines Corporation. ORACLE is a registered trademark or trademark of Oracle Corporation. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.