

CHAPTER

3

Customizing Your Session and Using Advanced Features

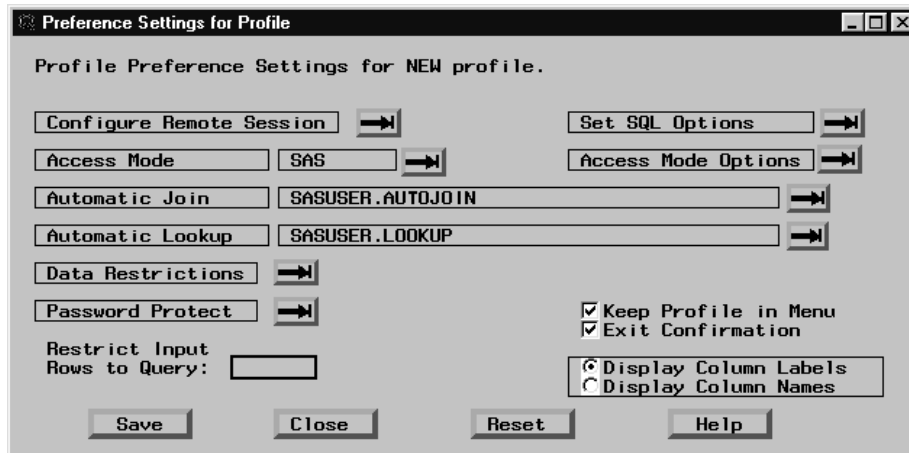
<i>Setting Your Profile</i>	73
<i>Configure Remote Session</i>	74
<i>Signing On to the Remote Host</i>	75
<i>Access Mode</i>	76
<i>Access Mode Options</i>	76
<i>Automatic Join</i>	77
<i>Available Table Sources</i>	78
<i>Available Tables</i>	78
<i>Saving Your Automatic Join Data Set</i>	78
<i>Updating Your Automatic Join Data Set</i>	78
<i>Creating an Automatic Join Data Set</i>	78
<i>Automatic Lookup</i>	81
<i>Data Restrictions</i>	81
<i>Password Protect</i>	81
<i>Restrict Input Rows to Query</i>	82
<i>Set SQL Options</i>	82
<i>Keep Profile in Menu</i>	83
<i>Exit Confirmation</i>	83
<i>Switching to Another Profile</i>	83
<i>Handling Missing Values</i>	83
<i>Defining a Format Outside the SQL Query Window</i>	84
<i>Using the Column Formats Window to List Your Format</i>	85
<i>Using the Format Procedure to Assign Lookup Values</i>	86
<i>Viewing Your Output</i>	86
<i>Changing Access Modes</i>	87
<i>ORACLE Access Mode Options</i>	87
<i>Creating a WHERE Expression</i>	87
<i>Viewing Your Query</i>	89
<i>Using SAS Data Sets to Store System Tables Information</i>	89
<i>Embedded Blanks in Column Names</i>	90
<i>Including Saved Queries</i>	90

Setting Your Profile

You can customize your SQL Query window sessions by specifying your own default settings and storing them in a profile. When you invoke the SQL Query window with the profile, your own preferences are automatically in effect. Your user-defined default settings are called preference settings. You can set up customized profiles for yourself or for a group of users. For example, a profile can be set up to restrict which table sources and tables will be available in a session of the SQL Query window.

Create a profile entry by selecting from the SQL Query Window PMENU

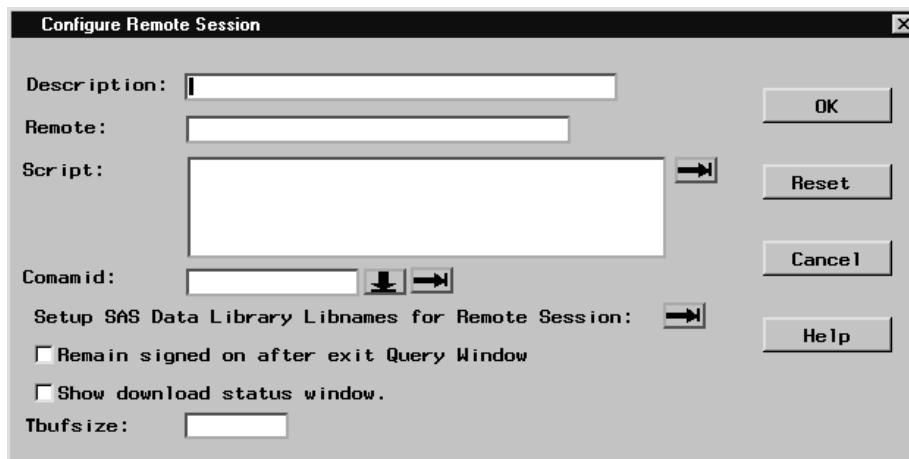
Profile ► Set Preferences



Configure Remote Session

Installations that license SAS/CONNECT software can use the SQL Query window to query tables or databases that are stored on remote hosts. To connect to a remote host you must first create an SQL Query window profile that contains information on the remote configuration.

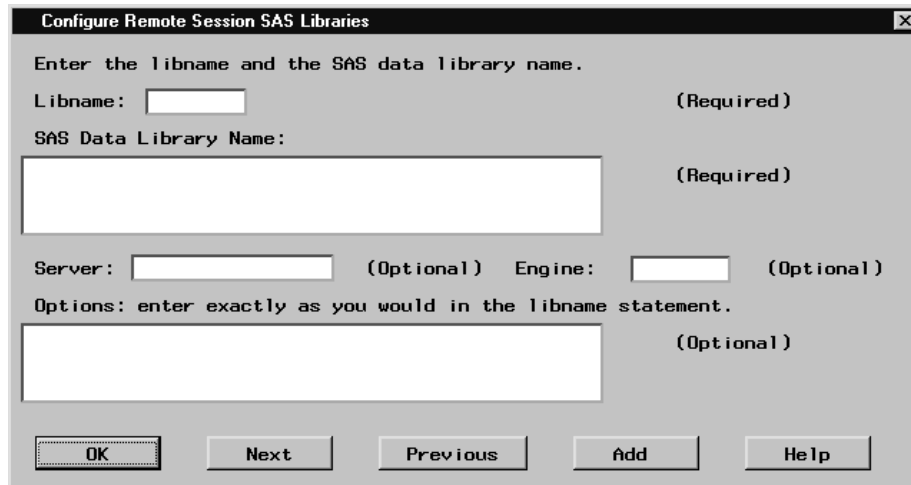
Select **Configure Remote Session** in the Preference Settings for Profile window.



Fill in the fields in this window with values that are appropriate for your site.

The **Description** field can contain any descriptive text you enter to describe the remote configuration.

Select the button next to **Setup SAS Data Library Libnames for Remote Session:** to enter the values that will be used to submit SAS statements remotely.



Configure Remote Session SAS Libraries

Enter the libname and the SAS data library name.

Libname: (Required)

SAS Data Library Name: (Required)

Server: (Optional) Engine: (Optional)

Options: enter exactly as you would in the libname statement. (Optional)

OK Next Previous Add Help

To query DBMS data through a SAS/ACCESS libname engine, enter the name of the libref you want to create in the **Libname** field. Enter the name of the SAS/ACCESS libname engine you want to use (usually the DBMS name) in the **Engine** field. Enter libname options required for the libref in the **Options:** field. In most cases, leave the **SAS Data Library Name** field empty. For more information on libname engines, see your SAS/ACCESS documentation.

When you have entered your values, select **OK** to return to the Configure Remote Session window. Select **OK** to return to the Profile Preference Settings window.

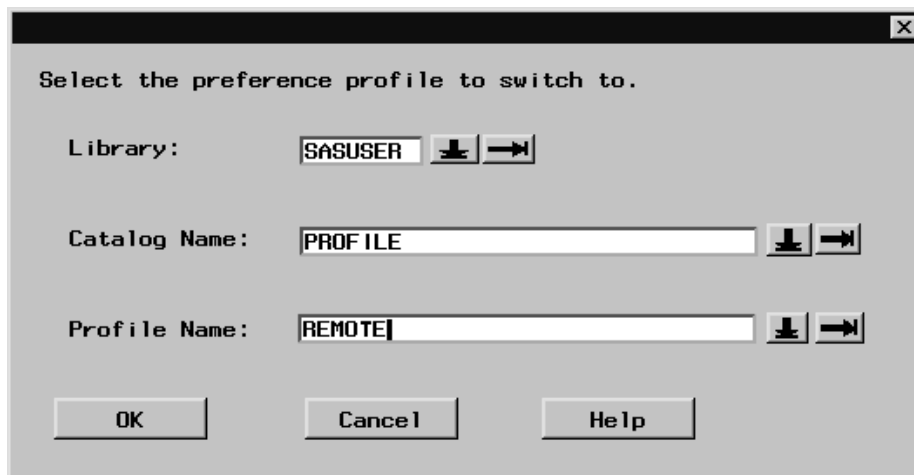
Use the other items in the Profile Preference Settings window to specify any other preference settings that you want to include in your profile.

Select the **Save** button to save the profile.

Signing On to the Remote Host

You can sign on to the remote host either when you invoke the SQL Query window, or during an SQL Query window session.

- To sign on to the remote host when you invoke the SQL Query window, specify the profile that contains the remote configuration information. The connection to the remote host is made automatically.
- To sign on to the remote host during an SQL Query window session, select **Switch to New Profile** from the **Tools** PMENU. The Switch to New Profile window appears.



Select the preference profile to switch to.

Library:

Catalog Name:

Profile Name:

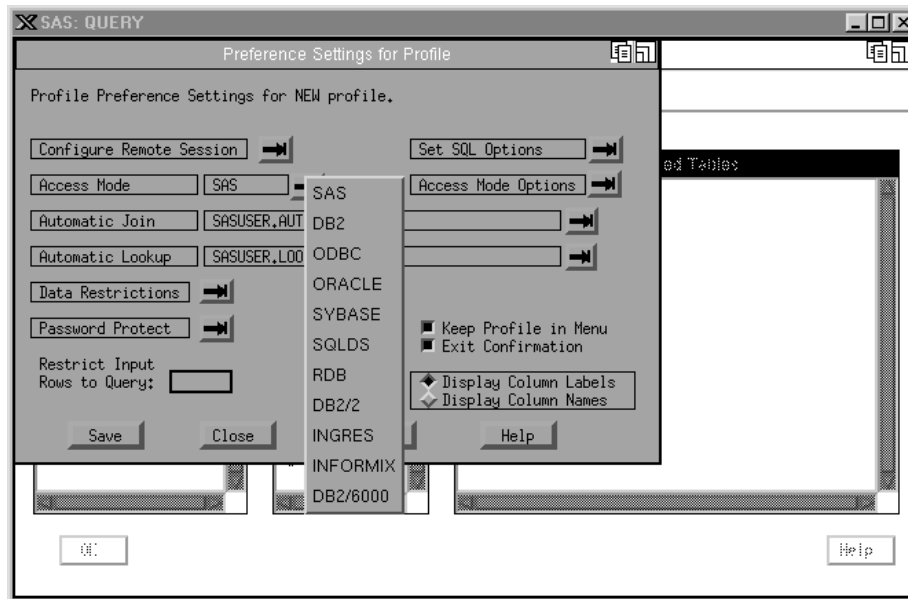
OK Cancel Help

Specify the library, catalog name, and profile name for the profile that contains your remote configuration information. Select **OK** to make the connection to the remote host. This example shows the default library and catalog names **SASUSER.PROFILE** and the profile name **REMOTE**.

- To remain signed on after exiting the SQL Query Window, select **Remain signed on after exit Query Window** in the **Configure Remote Session** window. If the remote host that is specified in your profile has already been signed onto during your SAS session, the SQL Query Window uses that connection to the remote host. You are not signed off from the remote host when you exit the SQL Query Window session.

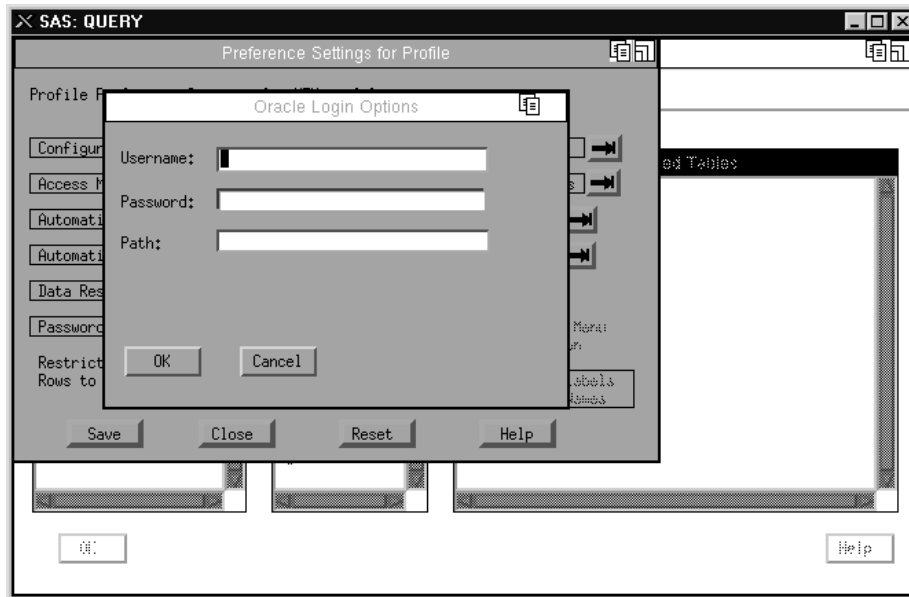
Access Mode

Access Mode specifies the source of the data that you will access. The source can be either SAS (for SAS data files and views), or most of the database management systems (DBMSs) for which the PROC SQL Pass-Through facility is available if you have SAS/ACCESS software installed. If you are using a SAS/ACCESS libname engine to query DBMS data, set the access mode to SAS. This will allow you to access the DBMS data via the libraries defined in your SAS session. The default access mode is SAS.

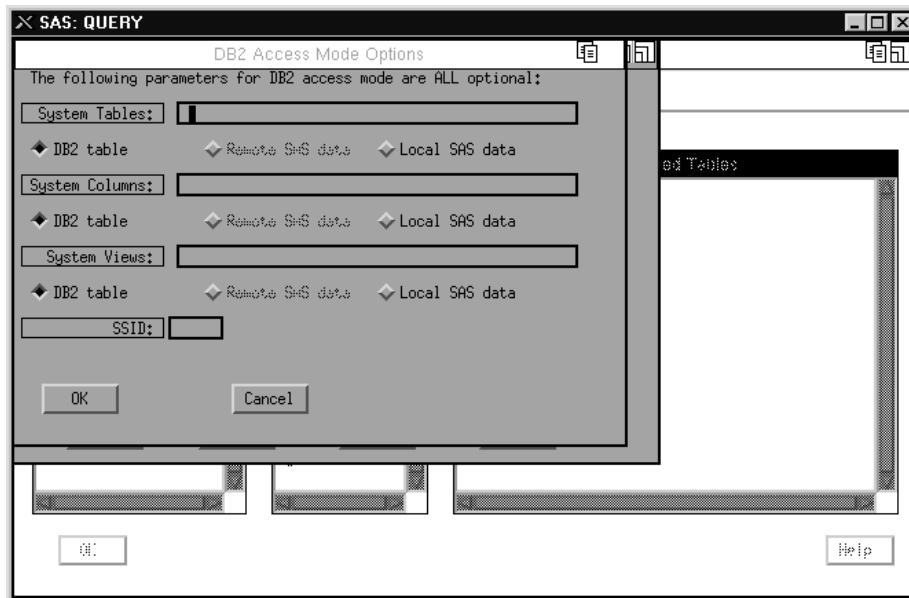


Access Mode Options

For some DBMSs such as SYBASE and ORACLE, you must specify access mode options such as the user name, password, and server. When you select one of these DBMSs that require options from the Access Mode window, an Access Mode Options window appears.



For access modes such as DB2 that do not require any options, you can select **Access Mode Options** to set additional options.



Automatic Join

An automatic join data set contains the table names and column links that are required to join tables automatically in a Query Window session. When tables that are defined in the automatic join data set are selected together, the corresponding column links are used to automatically start the query's WHERE expression.

Select

Automatic Join ► Set Name for Automatic Join Data Set...

to specify the automatic join data set.

Select

Automatic Join ► Create an Automatic Join Data Set...

to create an automatic join data set.

Available Table Sources

You can use the list of Available Table Sources to create an autojoin data set. An autojoin data set is a SAS data set that contains the column links for any table joins. This data set can be shared by many users.

Available Tables

You can add tables and views to the list of Available Tables by selecting one or more of the table sources. Choose two tables at a time from the list of Available Tables. Select **OK** to display the Column Links window.

A column link defines a column relationship between two tables in which the value of the column in the first table equals the value of the column in the second table.

The columns for each table appear in their own Columns list. Choose a column from each table and select **OK** to generate the column link.

When the column links are built, select **Show Links** to view all the links.

Saving Your Automatic Join Data Set

Select **Save** to create your automatic join data set. The name of the automatic join data set will be included in the profile that you are creating. You can also select a different library name to choose a SAS data library that is associated with your current SAS session.

To replace an existing SAS data set, enter a different data set name in the Table field or select the \equiv to choose a SAS data set name from the selected library. If the SAS data set is new, type in a new SAS data set name in the Table field.

The Set Automatic Join Data Set Name window enables you to specify your own automatic join data set. The default data set name is SASUSER.AUTOJOIN.

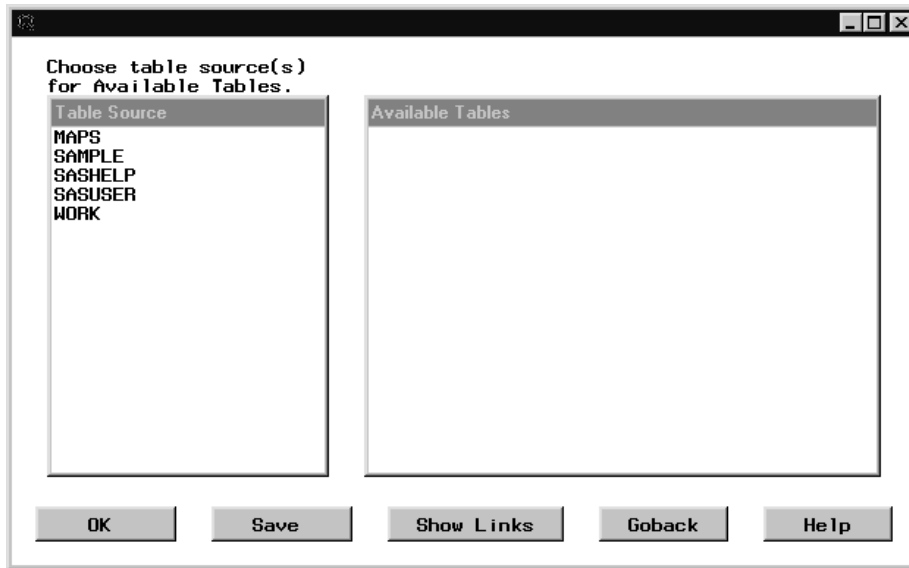
Updating Your Automatic Join Data Set

You can update an automatic join data set with PROC FSEDIT or PROC SQL. Automatic join data sets contain two columns, AUTOCOL1 and AUTOCOL2. Each column contains the library name, table name, and column name for one of the column links.

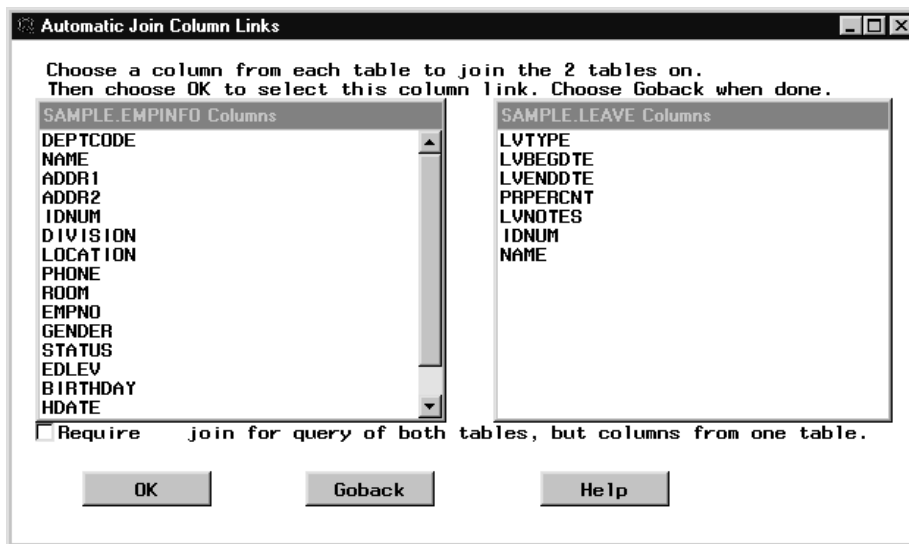
Creating an Automatic Join Data Set

The following example illustrates creating an automatic join data set. Select

Automatic Join ► Create an Automatic Join Data Set...



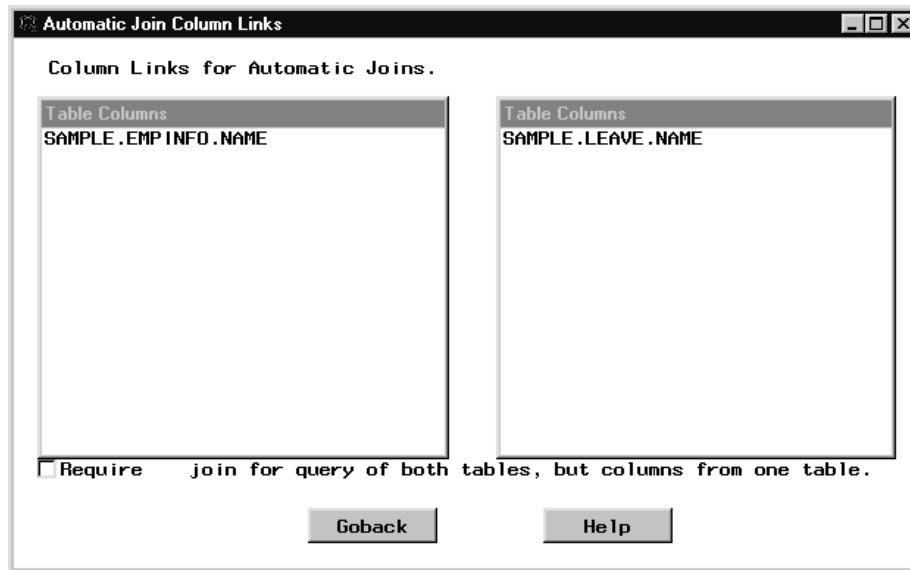
Select **SAMPLE** from the Table Source column to display a list of Available Tables. Select **SAMPLE.EMPINFO** and **SAMPLE.LEAVE**. Select **OK**. The Automatic Join Column Links window is displayed.



These two tables have the **NAME** column in common. Select **NAME** from the **SAMPLE.EMPINFO Columns** and **SAMPLE.LEAVE Columns**. Select **OK**. If the two tables had any other columns in common, you would be able to select these columns and store the column links in the automatic join data set.

Select **Goback** to return to the list of Available Tables.

Select **Show Links** to display the link that you have created between the two data sets.



Select **Goback**. You can select two tables again to define their column links. Select **Save** to save your automatic join.



Type **Employee Leave** as a label for your automatic join SAS data set. Select **OK**.

Select **Goback** to return to the Preference Settings for Profile window. Select **Goback** to return to the SQL QUERY TABLES window.

When you invoke the SQL Query window, the automatic join information in SASUSER.AUTOJOIN or in another SAS library that you have specified is included. If you invoke the SQL Query Window with a profile that contains an automatic join data set, the data set is used for automatic joins. If no automatic join data set has been defined in the profile, the default automatic join data set is SASUSER.AUTOJOIN.

Select



to end your SQL Query Window session. Select **OK** in the dialog to return to the PROGRAM EDITOR window. Issue the following command in the PROGRAM EDITOR window:

query

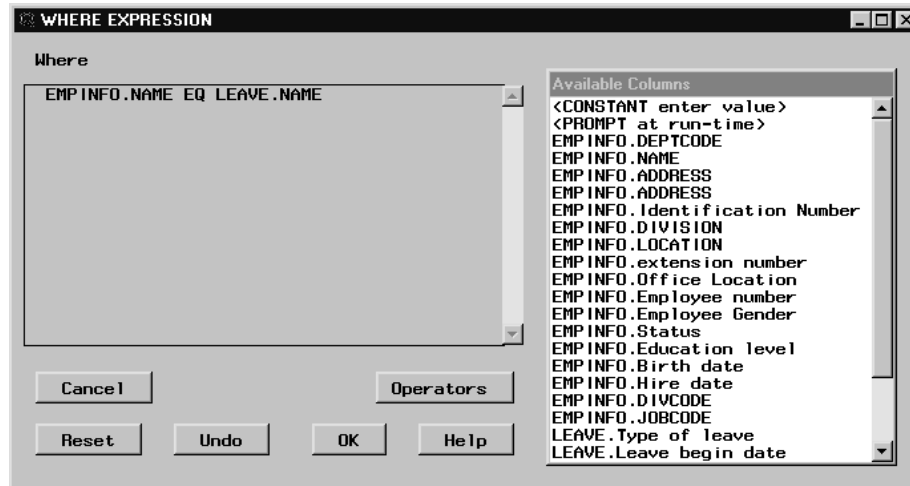
to invoke another SQL Query Window session. Select SAMPLE.EMPINFO and SAMPLE.LEAVE from the list of Available Tables and add them to the list of Selected Tables. Select **OK** to display the SQL QUERY COLUMNS window.

Select DIVISION from the list of Available Columns and add it to the list of Selected Columns.

Select

View ► Where Conditions for Subset...

to display the WHERE EXPRESSION window.



The WHERE expression begins with the column link that you specified in your autojoin table.

Automatic Lookup

Automatic Lookup specifies a lookup table. Select **Library** and **Table** to specify existing library and table names or to name new ones. If you do not specify a lookup table name, the table name defaults to SASUSER.LOOKUP when you save the profile. If you invoke the SQL Query window without specifying a profile, the Query window uses SASUSER.LOOKUP if it exists.

Data Restrictions

Data Restrictions specifies the table sources, tables, and columns that will be available in an SQL Query window session that is invoked with this profile. **Data Restrictions** also shows you which table sources, tables, and columns you have made available for the profile.

Password Protect

Password Protect enables you to specify a SAS System password for your profile. After you enter the password, you are prompted to re-enter it for verification. Thereafter, users can invoke the SQL Query window with this profile without knowing

the password; however, a user cannot update the profile without supplying the password.

Restrict Input Rows to Query

Restrict Input Rows to Query imposes a limit on the number of rows (observations) that the SQL Query window will process from any single table. This item is useful for debugging queries on large tables, or for preventing the excessive expenditure of computer resources that would result from running queries on huge tables.

Set SQL Options

Set SQL Options enables you to set SQL options for the execution of the query.

The screenshot shows a dialog box titled "SQL Option Settings for Profile". It contains the following elements:

- Input fields for: INOBS=, FLOW=, OUTOBS=, SORTSEQ=, and LOOPS=.
- Checkboxes for:
 - NUMBER, include a column for row number.
 - FEEDBACK, display expanded references.
 - PROMPT stop or continue when limits met.
 - STIMER log execution time for queries.
- Buttons: OK, Reset, and Cancel.

INOBS=

restricts the number of rows that are processed from any single source.

OUTOBS=

restricts the number of rows that are processed as the target.

LOOPS=

limits the number of iterations in the inner loop.

FLOW=

specifies the limit beyond which character columns are to be flowed to multiple lines.

SORTSEQ=

specifies the collating sequence to be used with an ORDER BY clause. Use this option to specify a collating sequence other than the default.

Keep Profile in Menu

Keep Profile in Menu enables you to remove or retain the **Profile** item in the PMENU and to turn on or off the ability to switch to a new profile from the PMENU.

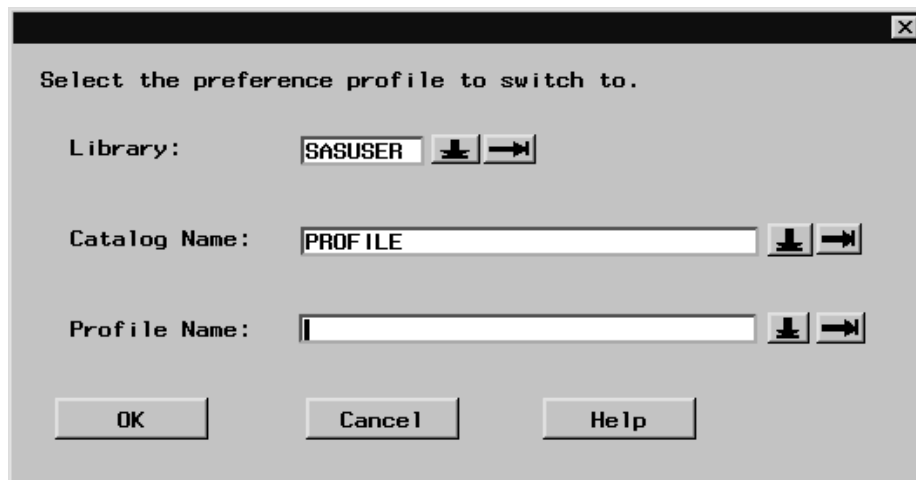
Exit Confirmation

Exit Confirmation enables you to turn off the dialog that asks you if you want to end the query session. The dialog is displayed when you choose End from the **File** PMENU.

Switching to Another Profile

To change to a different profile during an SQL Query window session, select

Tools ► Switch to New Profile...



If you have stored your profiles in a different library or catalog, select **Library** or **Catalog Name** and select a library or catalog that is associated with your current SQL Query window session. Then select **Profile Name** to display a list of the profiles that are stored in that library and catalog. Select the profile you want, then select **OK**.

Handling Missing Values

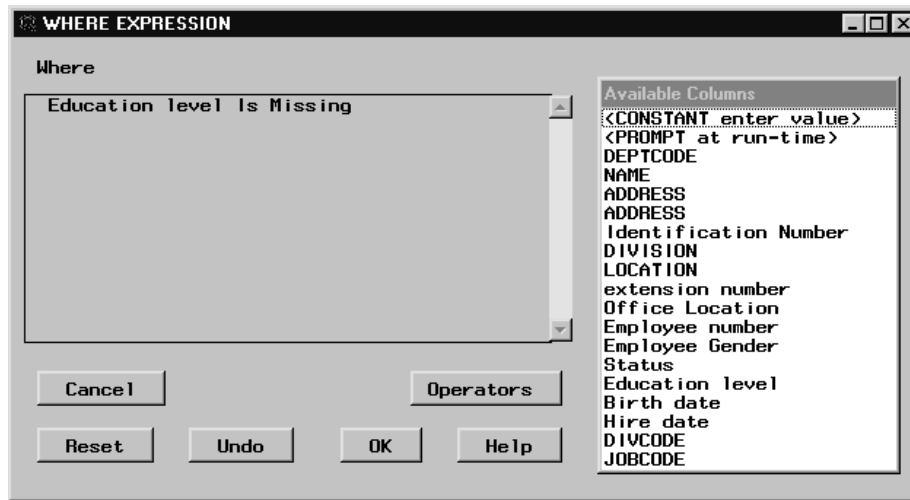
You can use the SQL Query Window to test for missing values in a data set. From the SQL QUERY TABLES window, select SAMPLE.EMPINFO from the list of Available Tables and add it to the list of Selected Tables. Select **OK**.

In the SQL QUERY COLUMNS window, select NAME and Education Level from the list of Available Columns and add them to the list of Selected Columns.

Select

View ► Where Conditions for Subset...

In the WHERE EXPRESSION window, select Education level from the list of Available Columns. Select **Is Missing** from the list of **OTHER Operators**.



Select **OK** to return to the SQL QUERY COLUMNS window.
Select



to display a list of employees whose education level is missing from the data set.

NAME	Education level
Anaya, John K.	.
Armstrong, Arnistead M.	.
Aschenbrenner, Edward T.	.
Blue, Kenneth N.	.
Brayboy, Kathleen	.
Loflin, Laura Anne	.
Sasson, Sammy E.	.
Smith, Jonathan D.	.
Thompson, Ann A.	.

Defining a Format Outside the SQL Query Window

You can use the FORMAT procedure to define additional output formats. In the PROGRAM EDITOR window, submit the following SAS code:

```
proc format; value edlevel 1-12 = 'No High School Diploma'
                        12 = 'High School Diploma'
                        13 = 'Completing Associate'
                        14 = 'Associate'
                        15 = 'Completing Bachelors'
                        16 = 'Bachelors'
                        17 = 'Completing Masters'
                        18 = 'Masters'
                        19 = 'Completing PhD'
                        20-99 = 'PhD'
                        . = 'No Education Data';
```

Invoke the SQL Query Window by typing

query

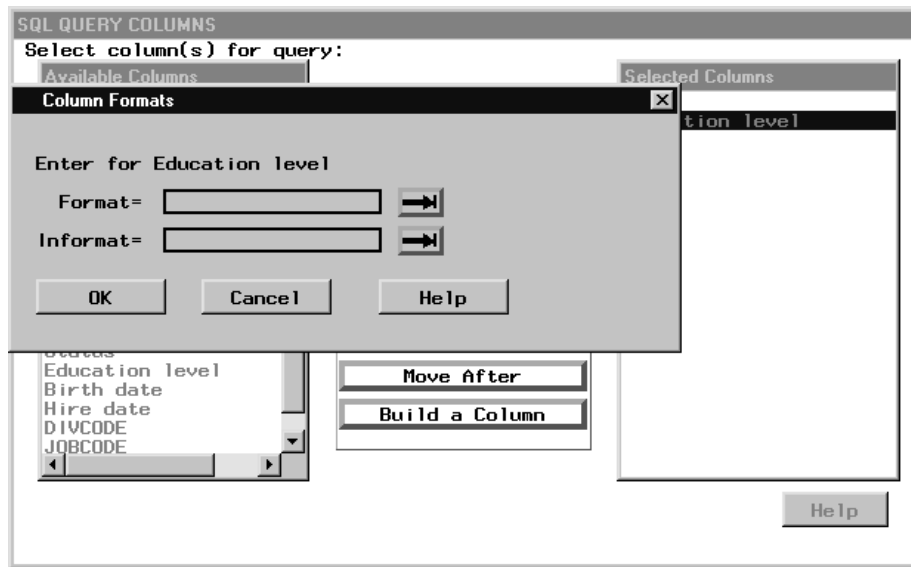
on the PROGRAM EDITOR window command line.


In the SQL QUERY TABLES window, select SAMPLE.EMPINFO from the list of Available Tables and add it to the list of Selected Tables. Select **OK**.

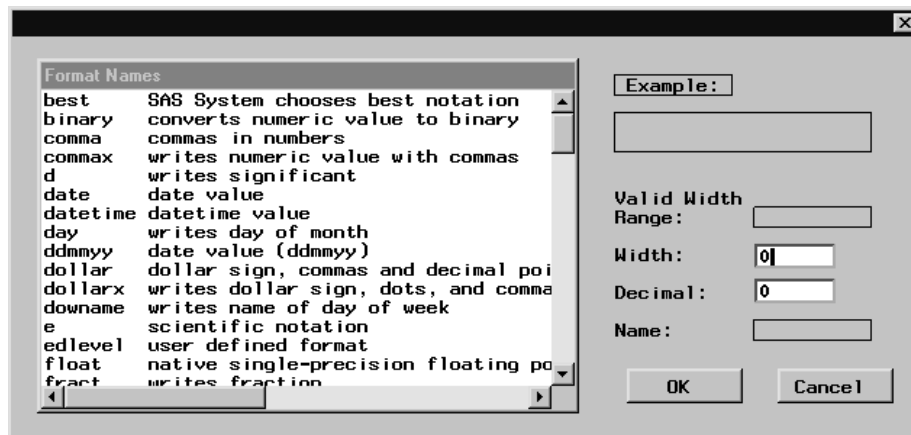
In the SQL QUERY COLUMNS window, select NAME and Education Level from the Available Columns List and add them to the list of Selected Columns.

Using the Column Formats Window to List Your Format

Select **Education Level** from the Selected Columns List. Select **Column Formats** to display the Column Formats window.

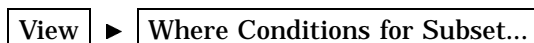


Select  next to **Format=** to display a list of Format Names.



Select **EDLEVEL** from the list of Format Names. Select **OK** to return to the Column Formats window. Select **OK** to return to the SQL QUERY COLUMNS window.

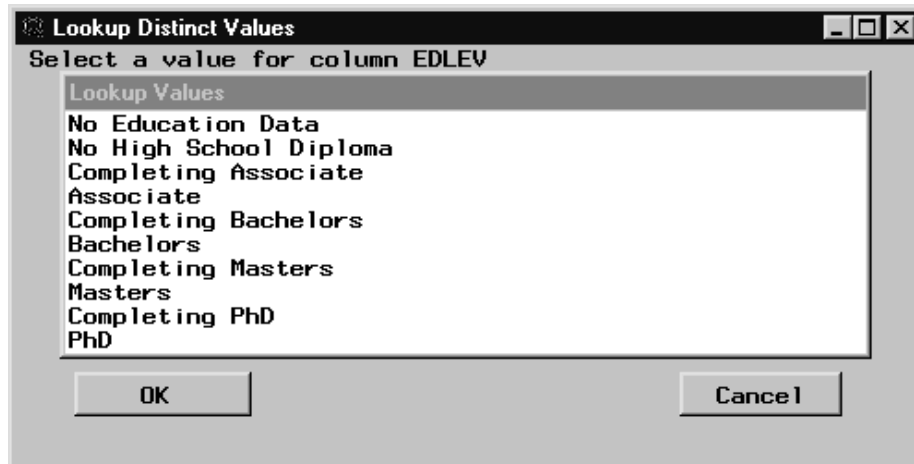
Select



to display the WHERE EXPRESSION window. Select **Education Level** from the list of Available Columns. Select **EQ** from the list of Column Operators.

Using the Format Procedure to Assign Lookup Values

Select **<LOOKUP distinct values>** from the list of Available Columns. The list of Lookup Values contains the distinct values for the Education Level column using the format you defined for EDLEVEL.



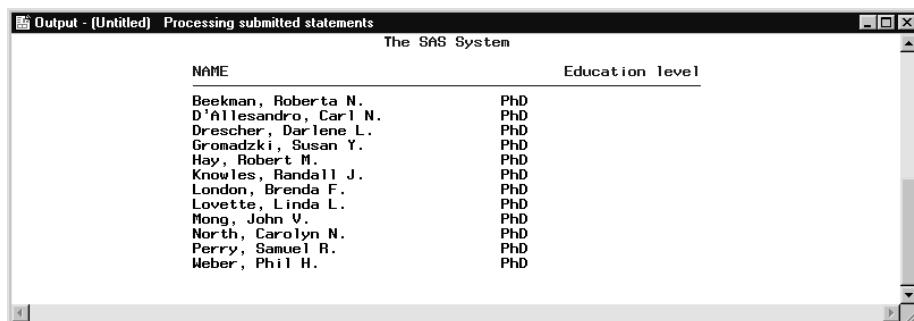
Select **PhD** from the list. Because the EQ operator can take only one value, you are automatically returned to the WHERE EXPRESSION window. Select **OK** to return to SQL QUERY COLUMNS window.

Viewing Your Output

Select



to display a list of the employees whose education level is PhD.



Select



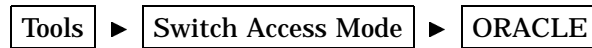
to reset the query.

Changing Access Modes

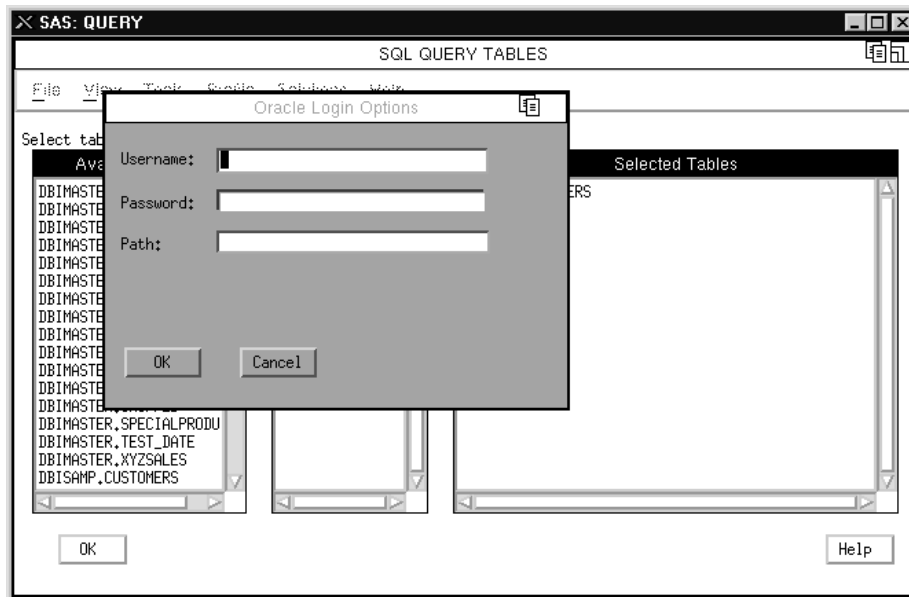
If you have SAS/ACCESS Interface to Oracle installed, you can switch access modes and use the SQL Pass-Through facility to query ORACLE tables.

ORACLE Access Mode Options

From the SQL QUERY TABLES window, select



to display the Oracle Access Mode Options window.

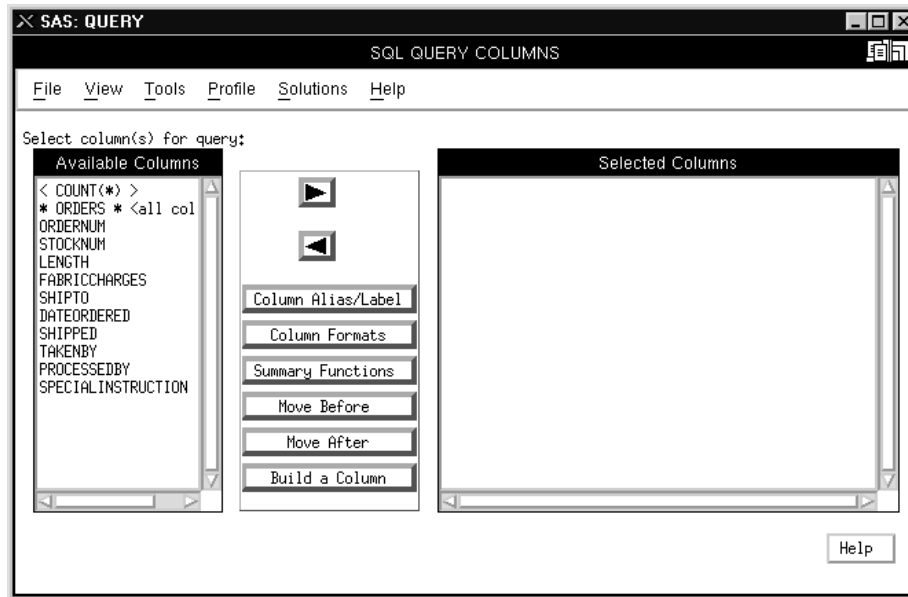


Fill in the fields with the information appropriate for your site. Contact your ORACLE administrator for more information.

Creating a WHERE Expression

Select **OK** to return to the SQL QUERY TABLES window. The sample tables that are available with your ORACLE DBMS are listed in the Available Tables column.

Select **ORDERS** from the list of Available Tables and move it to the list of Selected Tables. Select **OK** to display the SQL QUERY COLUMNS window



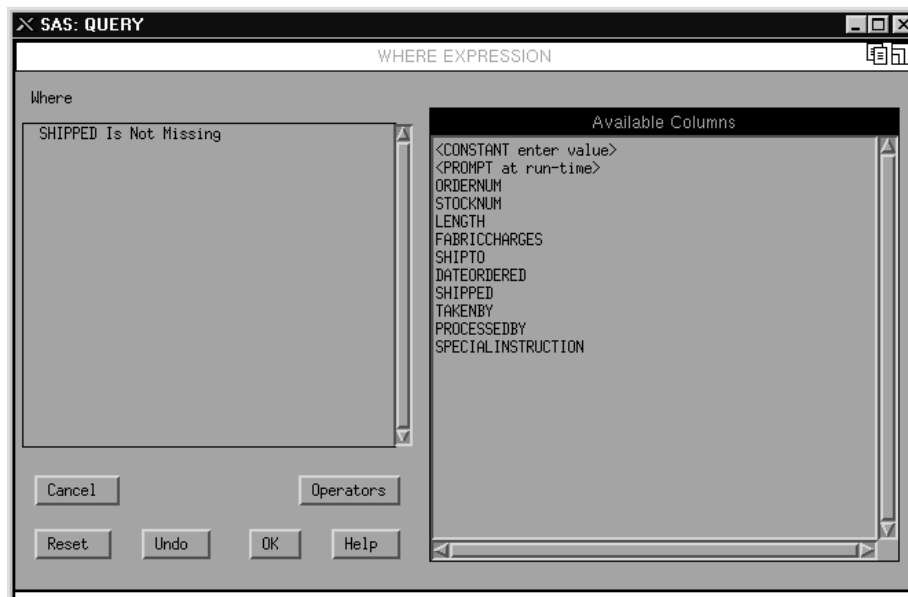
Select **FABRICCHARGES**, **SHIPTO**, **DATEORDERED**, **TAKENBY**, and **PROCESSEDBY** from the list of Available Columns and add them to the list of Selected Columns.

Select

View \blacktriangleright **Where Conditions for Subset...**

to display the WHERE EXPRESSION window.

Select **SHIPPED** from the list of Available Columns. Select **Is Not Missing** from the menu of **OTHER Operators**.



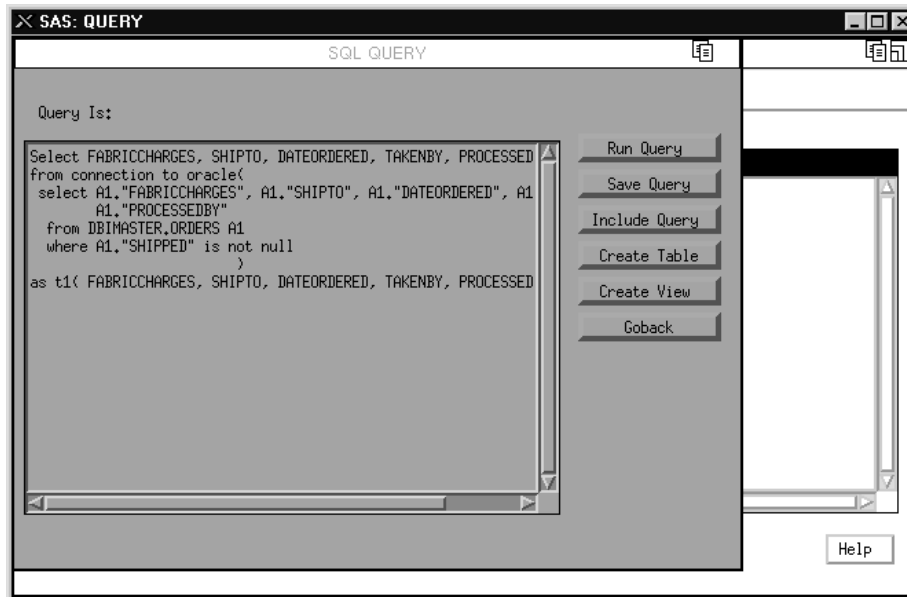
Select **OK** to close the WHERE EXPRESSION window.

Viewing Your Query

From the SQL QUERY COLUMNS window, select

Tools ► Show Query...

to view your query.



This query reports information for orders that have been shipped. ORACLE SQL is generated as the query is built.

The syntax enclosed by the parentheses following **ORACLE** is transported through the SQL Procedure Pass-Through facility to the ORACLE DBMS for processing. The **is Not Missing** operator from the WHERE expression is converted to the **is not null** ORACLE operator.

The syntax outside of the parentheses following **ORACLE** is processed by the SAS System.

Using SAS Data Sets to Store System Tables Information

For access modes other than SAS, the individual database management system (DBMS) tables are queried for the Available Tables and Available Columns lists. For DB2, DB2/2, DB2/6000, SYBASE, and ODBC, the system tables information that fills the Available Tables and Available Columns can now be stored in SAS data sets. One data set contains Tables information. The other data set contains Columns information. When you use a remote session for querying, these SAS data sets can be stored locally for enhanced performance.

If you have read authority to the system tables, you can assign the DB2, DB2/2, or DB2/6000 system tables to be read to SAS tables that are mirror images of the DB2 tables. If you are executing a remote session, you can specify whether these SAS DB2 system tables are to be read locally or remotely.

You can create these mirror image tables by querying the DB2 system tables in a Query window session and creating SAS tables of the queries built. They can also be

created with a PROC SQL program that queries the DB2 system tables. The PROC SQL statements can be saved and the SAS program can be run in batch whenever you need to update the mirror image tables.

SAS tables (data sets) created by automatic joins can also be created for any of the access modes that provide system tables/dictionaries with a PROC SQL program.

For ODBC, you can generate SAS data sets that contain system tables:

- 1 Select **Set Preferences...** from the Profile pmenu.
- 2 In the Preference Settings for Profile window, set **Access Mode** to ODBC.
- 3 In the Preference Settings for Profile window, select **Access Mode Options**.
- 4 In the Preference Settings for ODBC Access Mode window, enter the SAS data set names for system tables information.
- 5 If SAS data sets have not been created, select **Create Table** and **Create Column**.

For SYBASE, the system table information can be read from a SAS data set:

- 1 Select **Set Preferences...** from the Profile pmenu.
- 2 In the Preference Settings for Profile window, set **Access Mode** to SYBASE.
- 3 In the SYBASE Access Mode Options window, enter the parameters for the SYBASE access mode.
- 4 Select **SAS Data Sets**.
- 5 In the Available Tables and Columns for SYBASE window, enter the SAS library name and SAS data set name for SYBASE system tables. If the SYBASE system tables do not already exist select **Create Table** and **Create Column**.

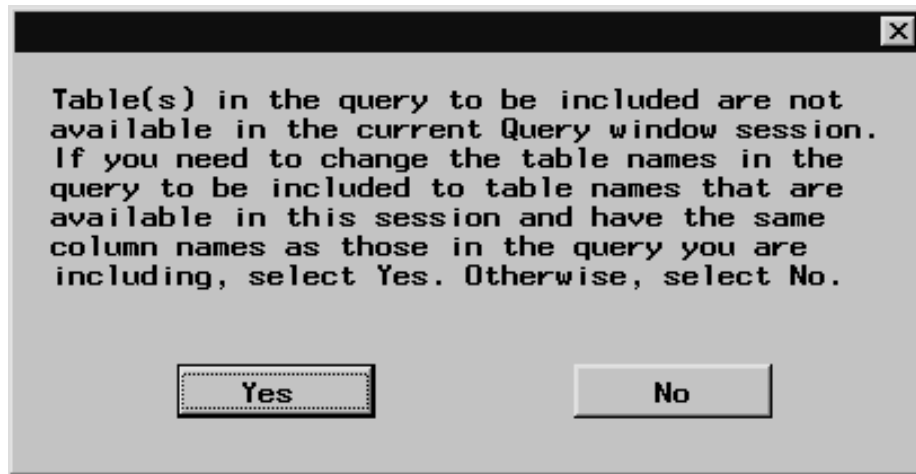
Embedded Blanks in Column Names

DB2, DB2/2, DB2/6000, ORACLE, and ODBC column names can contain blanks. The Query Window encloses column names in double-quotes for these access modes to support any blanks within the column name string. Some ODBC drivers may not support double-quoted column names. To avoid conflicts with ODBC drivers:

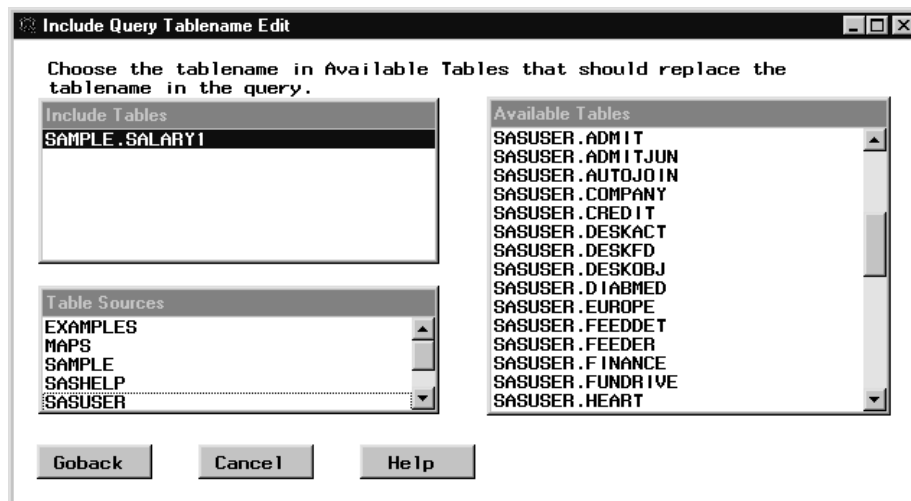
- 1 Select **Set Preferences...** from the Profile pmenu.
- 2 In the Preference Settings for Profile window, set **Access Mode** to ODBC.
- 3 In the Preference Settings for ODBC Access Mode window, select **Exclude double quote around column names**.

Including Saved Queries

When you save a query, references in the query to tables are saved as two-level names(*libref.filename*). If you try to include a saved query that specifies a libref that is not currently assigned, or tables that have been moved or deleted, the Query window will inform you that the tables cannot be found.



If the tables are available in another library, or you want to run the query against different tables, select **Yes**. The Include Query Tablename Edit window appears (probably with different names than in this example).



If the tables exist in another library, select that library from the list of Table Sources, then select the tables from the list of Available Tables.

If you want to run the query against different tables, select the library containing the tables from the list of Table Sources, then select the tables from the list of Available Tables. The tables must have identical structures to the tables on which the query was built.

This feature enables you to create a query that can be used on any identically structured table. For example, you could create a query on a table containing March sales data, and then use that query on a table containing April sales data. Remember that, to be prompted for a new table for the query, the original table on which the query was created must not be available in the Query window session. By moving the March table from, for example, the CUR_MON (Current Month) library, and moving the April table into the CUR_MON library, you would be prompted to supply the table for the query.

See “Saving Queries” on page 32 for more information on creating and saving queries.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS SQL Query Window User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 120.

SAS SQL Query Window User's Guide, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-554-X

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS[®] and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. [®] indicates USA registration.

DB2[®], DB/2[™], DB2/6000[™], 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.

The Institute is a private company devoted to the support and further development of its software and related services.