Paper 333-2010

Understanding SAS/ACCESS® Software's Role with Querying and Reporting in a Business Intelligence Environment

Varsha Desai, SAS Institute Inc., Cary, NC

ABSTRACT

Sometimes, SAS® software users encounter errors when they use business intelligence (BI) clients to query database tables. For example, a user who queries an Oracle table using SAS® Enterprise Guide® or the SAS® Add-In for Microsoft Office might receive an error that indicates that the data file cannot be opened when she tries to access a table. Errors such as this can mystify users or stop the progress of an entire project. To correct such problems, users need to understand how SAS/ACCESS® software is used to access database management system (DBMS) libraries.

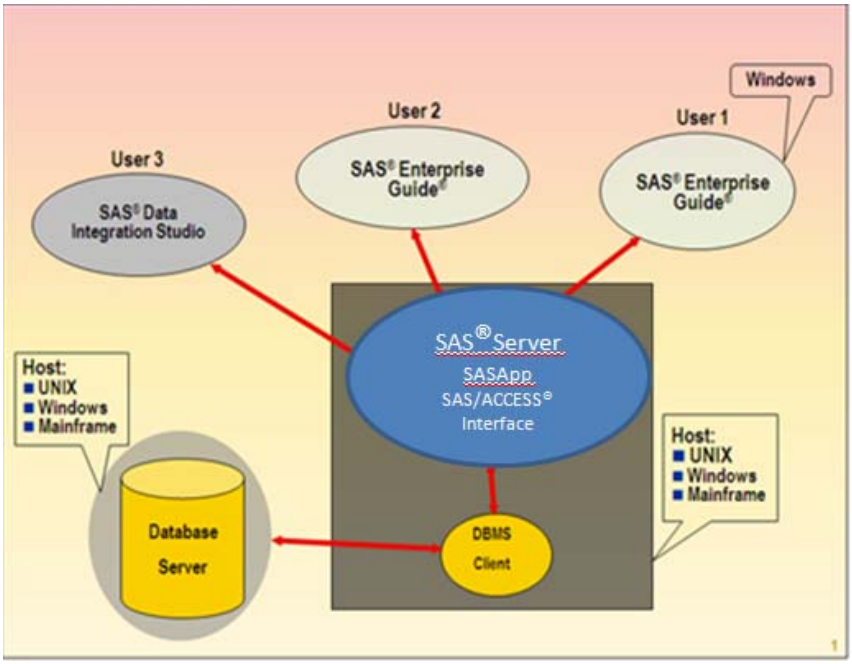
This paper is suitable for all levels of BI client users and explains the role that SAS/ACCESS software plays in such a process. It guides you through a case in which an error occurs when SAS Enterprise Guide is the BI client. Understanding how SAS/ACCESS software works in this case study will enable you to resolve issues related to similar errors that you might encounter. The paper also provides debugging tips that can help you be more productive in your daily query-and-reporting activities.

SAS/ACCESS® SOFTWARE: OVERVIEW

SAS/ACCESS software is a group of interfaces that provide data access and integration between SAS and third-party databases. SAS/ACCESS interfaces are available for the most current releases of many database management systems on the most popular platforms on the market. For a list of supported databases and operating systems, see the SAS/ACCESS Validation Matrix at support.sas.com/matrix.

SAS/ACCESS interfaces enable SAS to read, write, and update data regardless of the database type or platform. With SAS/ACCESS software, you can obtain and analyze data from third-party databases that is still stored in its native format. This can help you make critical business decisions efficiently using SAS BI client applications such as SAS Enterprise Guide. Most SAS/ACCESS interfaces call native client libraries (for example, OCI for Oracle, CLI for DB2, and CLIv2 for Teradata) that are supplied by database vendors. Database client software must be installed on the same machine as SAS, and SAS uses the client to communicate with the database server.

Note: SAS Enterprise Guide does not interact directly with the database server. SAS Enterprise Guide interacts with the SAS server. As shown in the following diagram, the SAS server interacts with the DBMS client, which subsequently interacts with the database server.



This paper examines a case in which SAS Enterprise Guide is the BI client and an error occurs during a query that is submitted through the client. When you submit a query from SAS Enterprise Guide against a table in the Oracle database, the query proceeds in the following sequence:

1. The query is submitted to the SAS server.
2. The SAS server sends a request to the Oracle client.
3. The Oracle client sends a request to the Oracle server.
4. The query runs on the Oracle server and returns the result to SAS Enterprise Guide via the SAS server.

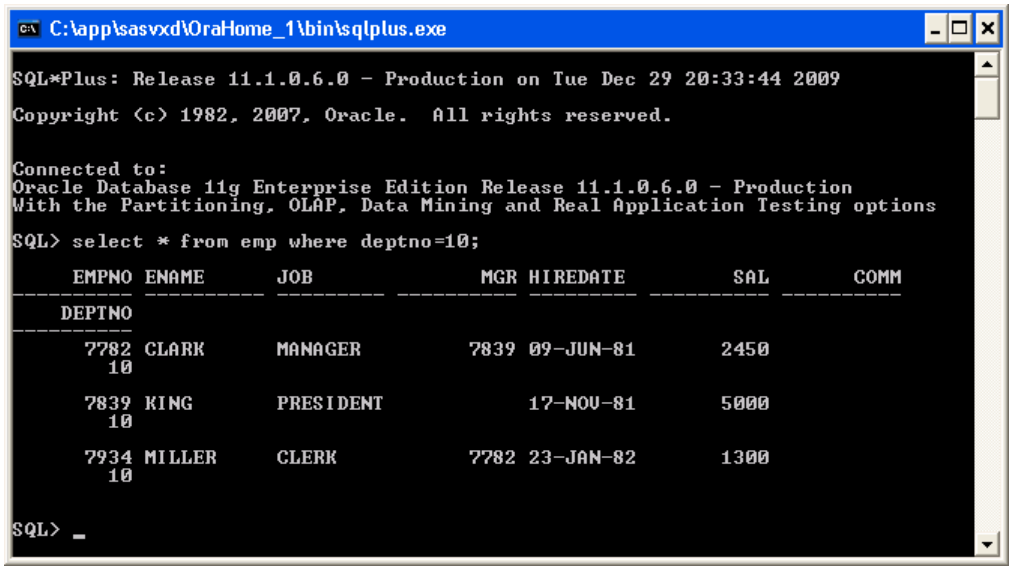
If you receive an error during a query, the problem is either with the Oracle client, the SAS server, or SAS Enterprise Guide. To locate the error, you must test each of those applications. The following sections explain how to test the connections from those applications to the database.

TESTING THE ORACLE CLIENT WITH ORACLE SQL*PLUS

In order to connect to an Oracle database using SAS/ACCESS software, you must be able to connect to the same database outside of SAS. One method for doing this is to test the connection with Oracle SQL*Plus, which is an interface to the Oracle database server. You test the connection using SQL*Plus as follows:

1. Select **Start ► Run** to open the Run dialog box.
2. In the **Open** text box, type **`sqlplus` *`oracle-user/oracle-password@oracle-path`***, where **`oracle-user`** is your Oracle user ID, **`oracle-password`** is your Oracle password, and **`oracle-path`** is the path to your Oracle server. If you do not know this information, contact your database administrator.

3. Click **OK** to invoke SQL*Plus, which displays in a command window, as shown here:



If SQL*Plus opens, proceed to the next section, "Connecting to the Oracle Database from the Foundation SAS® Server."

If SQL*Plus does not open, there is a problem with your connection to the database. If a problem exists, contact your database administrator for assistance.

Once you connect successfully to the database using SQL*Plus, you should be able to connect to the same database using SAS/ACCESS software. Verify the following information before you attempt to connect from SAS:

- If your SAS server is running in a Windows operating environment, ensure that the path to the Oracle client libraries is appended to the Windows PATH system variable. To locate this variable, follow the instructions for your specific Windows environment in the *Configuration Guide for SAS® 9.2 Foundation for Microsoft® Windows®* (support.sas.com/documentation/installcenter/92/sasinstall/9.2/win/win/config.pdf). The Oracle libraries are usually located in the directory **`ORACLE_HOME/bin`**.
- If your SAS server is running in a UNIX operating environment, you must set appropriate environment variables in the sasenv_local file that is located in the **`!SASROOT/bin`** directory. The following examples illustrate these settings for various UNIX operating systems:
  - **AIX environment:**

    ```
    $ LIBPATH=$ORACLE_HOME/lib:$LIBPATH
    ```

  - **Linux and Solaris environments:**

    ```
    $ LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
    ```

  - **HP-UX environment:**

    ```
    $ SHLIB_PATH=$ORACLE_HOME/lib:$SHLIB_PATH
    ```

  For more information, see the *Configuration Guide for SAS® 9.2 Foundation for UNIX® Environments* (support.sas.com/documentation/installcenter/92/sasinstall/9.2/unx/config.pdf).

CONNECTING TO THE ORACLE DATABASE FROM THE FOUNDATION SAS® SERVER

After you obtain a successful test connection from SQL*Plus and you have set your environment variables, you can access the Oracle tables with SAS/ACCESS software using either the Oracle LIBNAME engine or the Pass-Through Facility with the SQL procedure.

USING THE ORACLE LIBNAME ENGINE

A *LIBNAME engine* provides a seamless and transparent way to surface data. It requires minimal knowledge of the database and the Structured Query Language (SQL). As shown in the following example, you can use this engine in a LIBNAME statement to test your connection using SAS/ACCESS Interface to Oracle from Base SAS software:

```
libname oralib oracle user=oracle-user
                      password=oracle-password
                      path=oracle-path;
```

Again, you need to substitute your own Oracle connection information for **`oracle-user`**, **`oracle-password`**, and **`oracle-path`**.

When you use LIBNAME engine technology, SAS attempts to pass the query to the DBMS server. However, there are cases when this does *not* happen. If you suspect that a query is not being passed to the DBMS server when it is processing in SAS, you can use the SASTRACE= option to generate information about what SAS is passing. If the query is not being passed to the DBMS server, the following message appears in the SAS log when you use the SASTRACE= option:

```
SQL statement was not passed to the DBMS, SAS will do the processing
```

For more information, see “Passing SAS Functions to Oracle" in *SAS/ACCESS® 9.2 for Relational Databases: Reference, Second Edition* (support.sas.com/documentation/cdl/en/acreldb/63023/HTML/default/a003113612.htm.)

USING THE PASS-THROUGH FACILITY

The *Pass-Through Facility* (explicit pass-through processing) requires that you have knowledge of the SQL that is passed to the database via PROC SQL. With this method, you invoke PROC SQL and specify a statement that requests connection to the database server, followed by SQL statements. PROC SQL submits your SQL statements, to the database server. The database then performs the processing request and returns the result. In the following example, the second SELECT statement is passed **as is** to Oracle:

```
proc sql;
   connect to oracle(user=oracle-user password=oracle-password
                     path=oracle-path);
   create table test as
          select * from connection to Oracle
          (select * from EMP where empid=10 rownum < 100);
quit;
```

DISPLAYING THE ORACLE TABLES

If you use the LIBNAME engine, you can view your Oracle tables in your library in the SAS Explorer window or, if you prefer, you can submit a DATASETS procedure that is similar to the following to display the tables in the SAS log:

```
proc datasets lib=oralib;
quit;
```

If the Oracle tables are not listed in the library, you might need to add the SCHEMA= option in the LIBNAME statement. If you are not sure which schema to use, submit the following SQL procedure to find the appropriate schema:

```
proc sql;
   connect to oracle(user=oracle-user
                     password=oracle-password
                     path=oracle-path);
   create table list_tab as
          select * from connection to oracle
          (select owner, table_name from all_tables);
quit;
```

The SAS data set List_Tab displays a list of tables and their associated schema names. Once you know the schema name, you can add it to the LIBNAME statement, as shown in this example:

```
libname oralib oracle user=oracle-user password=oracle-password
                      path=oracle-path schema=schema-name;
```

Note: When a schema is not specified, the Oracle user ID that you submit in the LIBNAME statement becomes the schema name.

ACCESSING THE ORACLE DATABASE FROM A BUSINESS INTELLIGENCE CLIENT

There are several ways to access an Oracle database from a BI client application. However, the preferred method is to define the library in SAS® Management Console, which provides a point of control for security management for a SAS library. For step-by-step instructions about how to define an Oracle library in SAS Management Console, see "TS771: Define an Oracle Library in SAS® Management Console" (support.sas.com/techsup/technote/ts771.pdf).
Note: Displays that appear in TS771 and this paper might have some slight differences because the displays in TS771 were created using SAS 9.1.3.

After you define the library in SAS Management Console, you can make BI Client aware of the library definition in one of two ways:

- Pre-assign the library so that the library is always assigned when the work space server starts. When a library is pre-assigned, you do not need to register the tables. In this case, all of the tables are available from the client application in that library, and you cannot set permissions at the table level. Pre-assigning a large number of libraries can slow the execution of SAS programs for all users. For additional details about pre-assigning libraries, see "Chapter 3: Assigning Libraries" in the SAS® *9.2 Intelligence Platform: Data Administration Guide* (support.sas.com/documentation/cdl/en/bidsag/61236/PDF/default/bidsag.pdf).

- Enable the client application to assign the library. When libraries are assigned by the client applications, each application can assign the library in a way that is most suitable for its intended user base. Library connections to the database are established only as necessary.

Once you define an authentication domain in metadata and the associated credentials in a metadata repository, then you can access your data from your BI client (in this case, SAS Enterprise Guide) by submitting the following SQL procedure (including the AUTHDOMAIN= option) in the SAS Enterprise Guide program window:

```
proc sql;
   connect to oracle(authdomain=oracle-authentication-domain path=oracle-
   path);
   create table foo as
          select * from connection to oracle
          (select * from emp where deptno=10);
quit;.
```

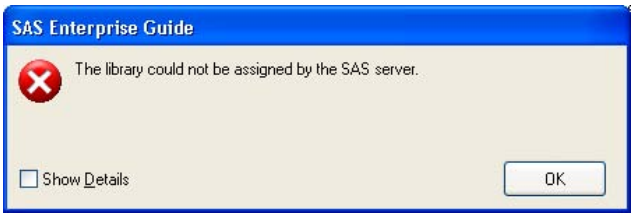
To the ORACLE engine, this procedure is the equivalent of specifying the USER= and PASSWORD= options, thereby granting you access to the database.

TROUBLESHOOTING PROBLEMS

This section presents common errors that you might encounter during a query and suggests solutions for solving the errors.

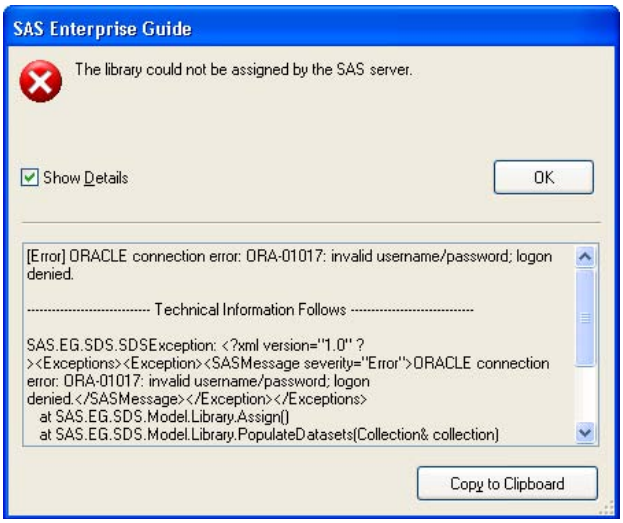
Problem 1

You want to run a query against an Oracle table, but you receive the following error when you try to expand the Oracle library from the **Server List** section in SAS Enterprise Guide:



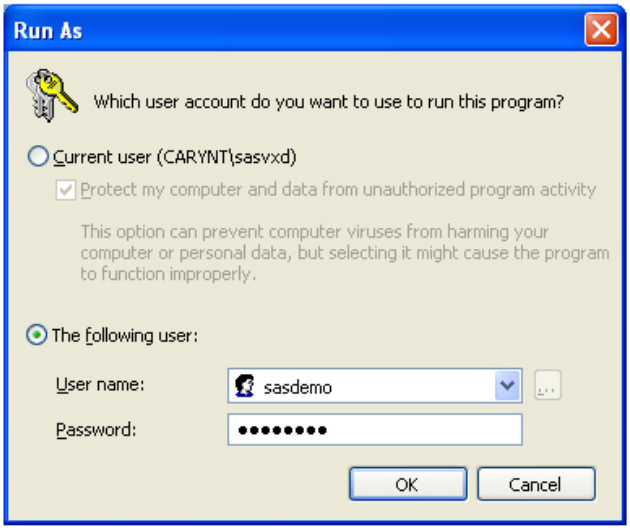
Solution

Many times, you can obtain helpful information by selecting the **Show Details** check box that appears in the error dialog box. For example, the following display is similar to what you see if you select **Show Details**. For this example, it indicates that the Oracle logon information is either incorrect or it is not present for this SAS Enterprise Guide user.



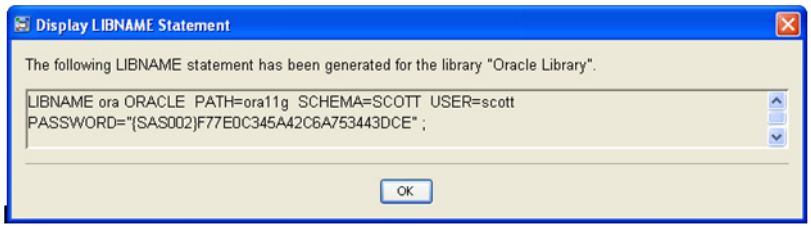
To solve this problem, make sure that the libref is assigned successfully when you manually submit the LIBNAME statement to the SAS® Workspace Server. To verify that the libref was properly assigned:

1. In Base SAS software, start a workspace-server session under the same account that you are using in SAS Enterprise Guide. To start a workspace-server session:
   a. Select **Start ► Programs ► SAS**.
   b. Right-click **SAS 9.2 (English)**, and select **Run As** from the menu that appears.
   c. In the Run As dialog box, enter the user ID and password that you are using in SAS Enterprise Guide, as shown in the following display:



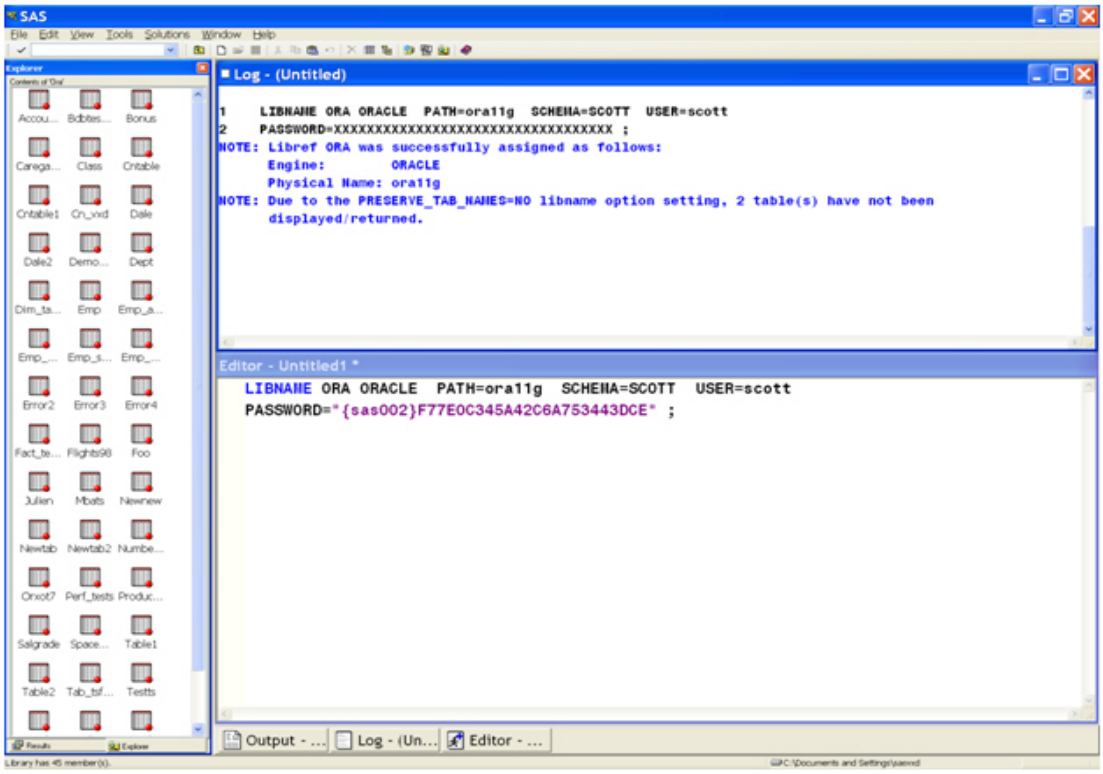
   d. Click **OK** to start the workspace-server session.

2. In SAS Management Console, right–click the library name under the **Data Library Manager** plug–in and select **Display LIBNAME Statement.** The Display LIBNAME Statement dialog box appears as follows with the LIBNAME statement that you submitted for the Oracle library:



   Note: If the library is pre-assigned, you cannot view it by right-clicking the library. You can temporarily uncheck the pre–assigned flag in order to display the LIBNAME statement.

3. Copy and paste this LIBNAME statement into to your SAS Workspace Server session and submit it, as shown in the following display:



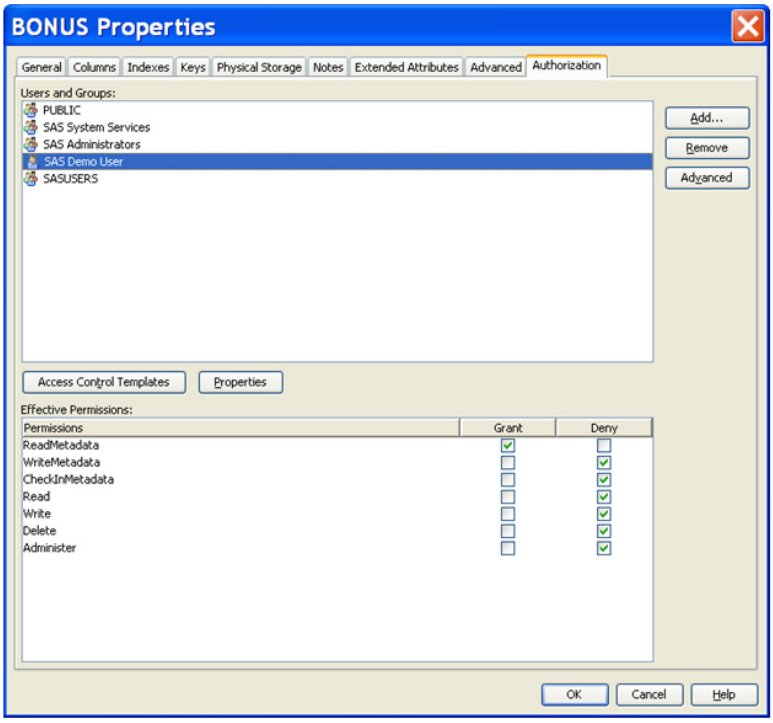
After the LIBNAME statement is submitted, you can then check the SAS log for messages, as shown in the preceding display.

Problem 2

When you use SAS Enterprise Guide to run a report on the Oracle table BONUS, you receive the following error:

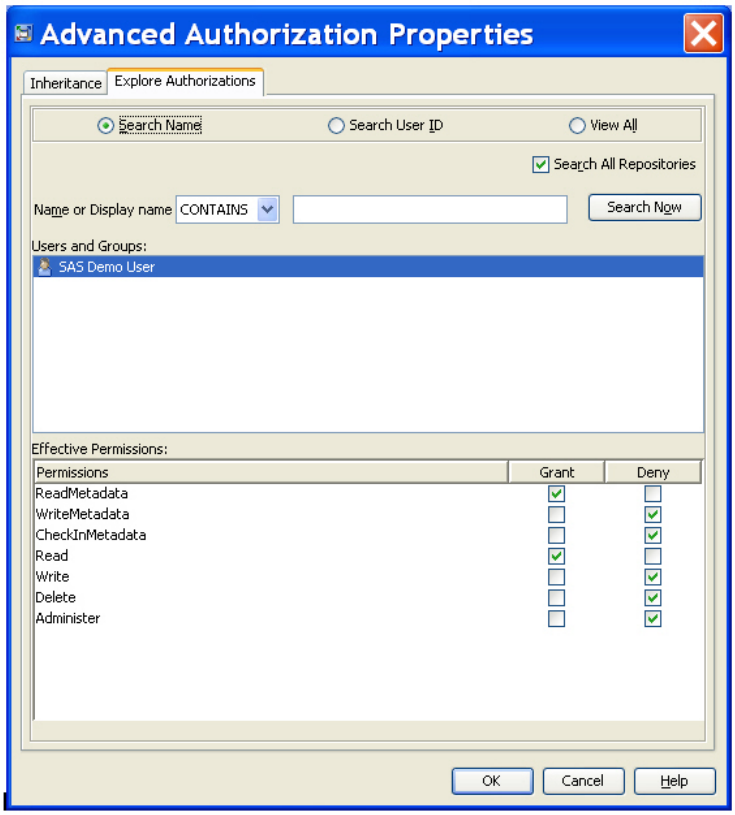
To solve this problem:

1. Invoke SAS Management Console.
2. Right-click the library that contains the BONUS table and select **`Properties`**. Then click the **`Authorizations`** tab, as shown in the following display:



3. Click the **Advanced** button to open the Advanced Authorization Tab dialog box. Then click the **Explorer Authorizations** tab.

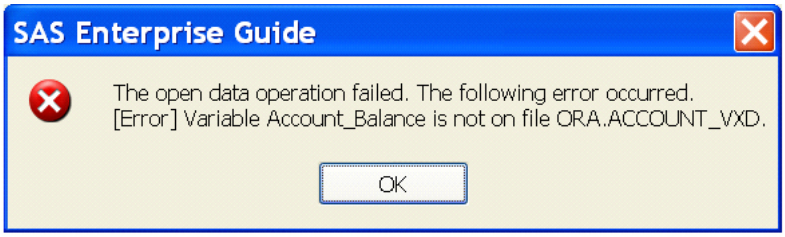
4. Type your SAS Enterprise Guide user ID in the **Name or Display Name** text box and click **Search Now**. After the application finds your user ID, you should see the effective permissions for that ID, as shown in the following display. **Note:** You must have set at least the Readmetadata and the Read permissions.



Once these permissions are set, you should be able to run a report on the BONUS table.

Problem 3

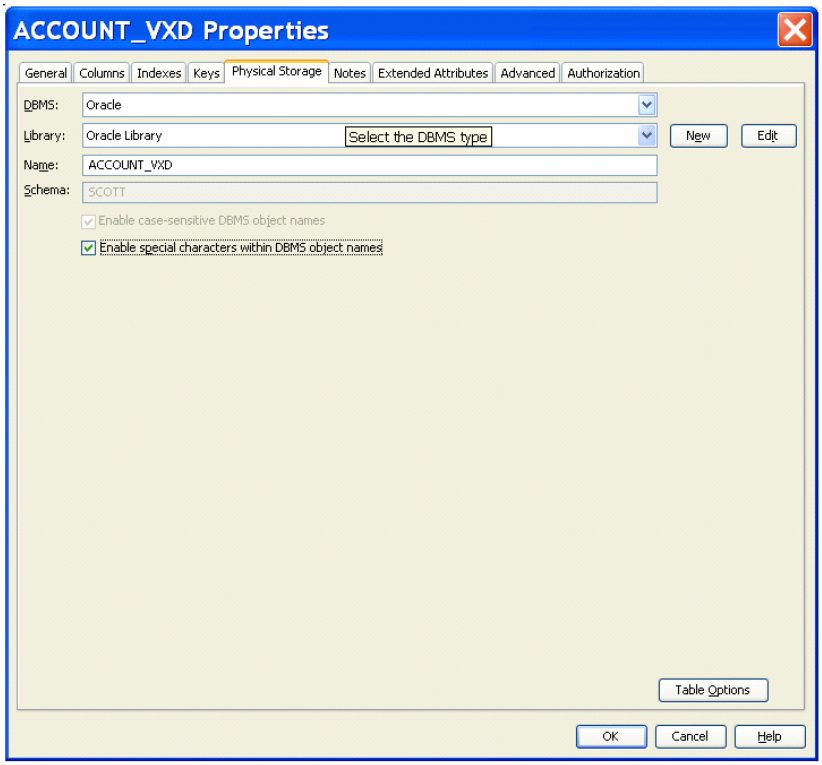
You are able to run a query from SQL*Plus and also from Base SAS software, but when you try to run the query from SAS Enterprise Guide, you receive the following error:



The error occurs because the Account Balance column name in the table contains a space.

To solve this problem:

1. In SAS Management Console, right-click the table and select **Properties** from the menu that appears.
2. In the Properties dialog box that displays, click the **Physical Storage** tab. As shown in the following display, select the **Enable Case sensitive DBMS object names** and **Enable special characters within DBMS object names** check boxes:



Note: If you select **Enable Case sensitive DBMS object names**, the **Enable special characters within DBMS object names** check box remains active and you can select it. However, if you select the second check box first, the application automatically selects both check boxes for you, and the first check box is no longer selectable.

3. Click **OK**.
4. In SAS Management Console, right-click the table again and select **Update Metadata** from the menu.

After the table metadata is updated, you should be able to open the table from SAS Enterprise Guide.

CONCLUSION

Knowing how a database library works in BI environment is important when you are querying and reporting from a BI client such as SAS Enterprise Guide. As explained in this paper, understanding the configuration of SAS/ACCESS software also plays a major role when you try to debug errors. In addition, knowledge about different methods for accessing a database from SAS can also help you to reduce the response time for a query to the external database.

RESOURCES

Plemmons, Howard. 2009. "Top Ten SAS® DBMS Performance Boosters for 2009." *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. Available at support.sas.com/resources/papers/sgf09/309-2009.pdf.

SAS Institute Inc. 2009. SAS/*ACCESS*® *9.2 for Relational Databases: Reference, Second Edition*. Cary, NC: SAS Institute Inc. Available at support.sas.com/documentation/cdl/en/acreldb/63023/PDF/default/acreldb.pdf.

SAS Institute Inc. 2009. *SAS*® *9.2 Intelligence Platform: Data Administration Guide*. Cary, NC: SAS Institute Inc. Available at support.sas.com/documentation/cdl/en/bidsag/61236/PDF/default/bidsag.pdf.

Whitcher, Mike. 2008. "New SQL Performance Optimizations to Enhance Your SAS® Client and Solution Access to the Database." *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. Available at www2.sas.com/proceedings/forum2008/296-2008.pdf.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Varsha Desai
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Phone: (919) 677-8008
Fax: (919) 531-9449
E-mail: support@sas.com
Web: support.sas.com