

CHAPTER

11

Using the SQL Editor

<i>Overview</i>	205
<i>Opening the SQL Editor Window</i>	206
<i>Entering SQL Statements Directly</i>	206
<i>Entering an SQL Query</i>	206
<i>Entering Non-SELECT SQL Code</i>	207
<i>Creating Template SQL Programs</i>	208
<i>Using a Stored Query or Statement</i>	208
<i>Selecting Tables with the Select Window</i>	209
<i>Choosing Tables with Select Table</i>	210
<i>Using the Enhanced SQL Editor</i>	211
<i>Including and Writing SQL Code to External Files</i>	213
<i>Correcting Errors in SQL Code</i>	214
<i>Running an SQL Query and Customizing the Output</i>	215
<i>Saving SQL Code or Output</i>	217

Overview

This chapter describes how you use Structured Query Language (SQL) and the SQL Editor window to perform queries for both the SAS SQL procedure and DB2 SQL. This chapter is intended for users who are familiar with SQL statements and syntax, and who have worked with SQL programs.

This chapter shows examples based on SAS AIRLINE sample tables. See “Returning to SAS/ASSIST Windows from the Output Window” on page 9 for information on how to create these sample tables.

The SQL Editor window enables you to code and submit SQL SELECT statements (also called *queries*). You can also code and submit non-query SQL statements, such as CREATE or DROP. The SQL Editor window uses the Query Manager to get information about tables and views. Templates of SQL code are then generated based on these Query Manager selections.

You can also use the SQL Editor window to include queries that are generated from the Query window, queries that are stored in external files, or IBM QMF queries that have been migrated to Query and Reporting. Query results can be saved as SAS tables (that is, SAS data files or data views) or DB2 tables. For more information on migrating from QMF to Query and Reporting, see Appendix 4, “Migrating from QMF to SAS/ASSIST Software,” on page 255.

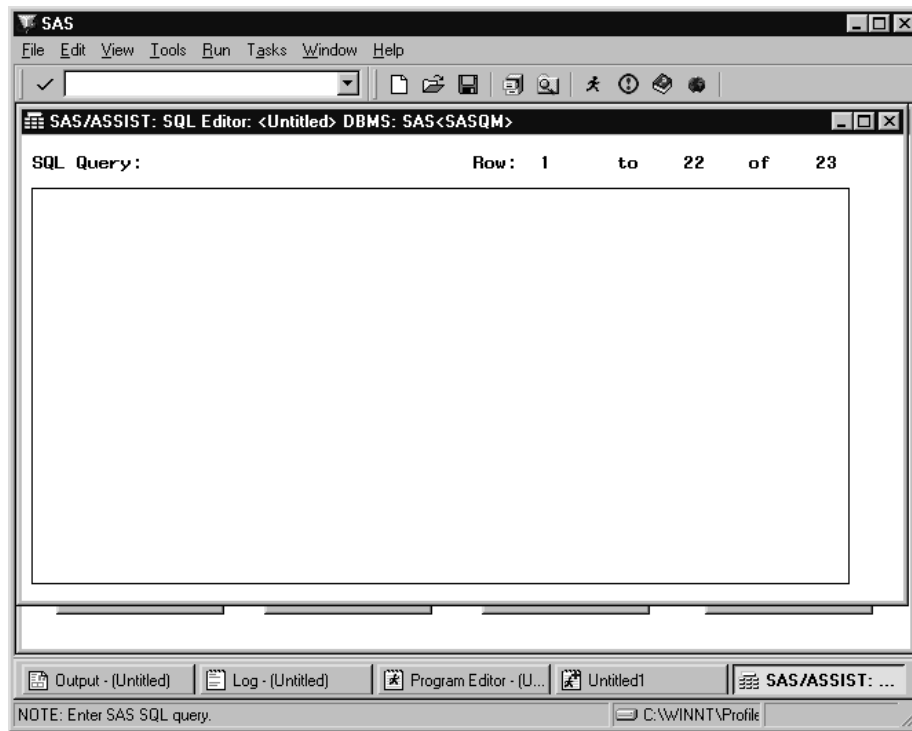
Opening the SQL Editor Window

To open the SQL Editor window, follow this path:



The SQL Editor window appears.

Display 11.1 SQL Editor Window



If the window is filled when you open it, it contains a last query that was loaded into this window. To clear the window, select **New** from the **File** menu.

Entering SQL Statements Directly

Entering an SQL Query

The SQL you use in the SQL Editor window depends on the kind of Query Manager that you are using. If you have a SAS Query Manager, you use the SAS System's SQL procedure in the SQL Editor window. If you have a DB2 Query Manager, you use IBM's DB2 SQL.

A message appears at the top of the SQL Editor window to indicate which SQL is being used. If you need to use a different set of tables, you can create a new Query Manager or talk to your SAS Support Consultant or database administrator (DBA).

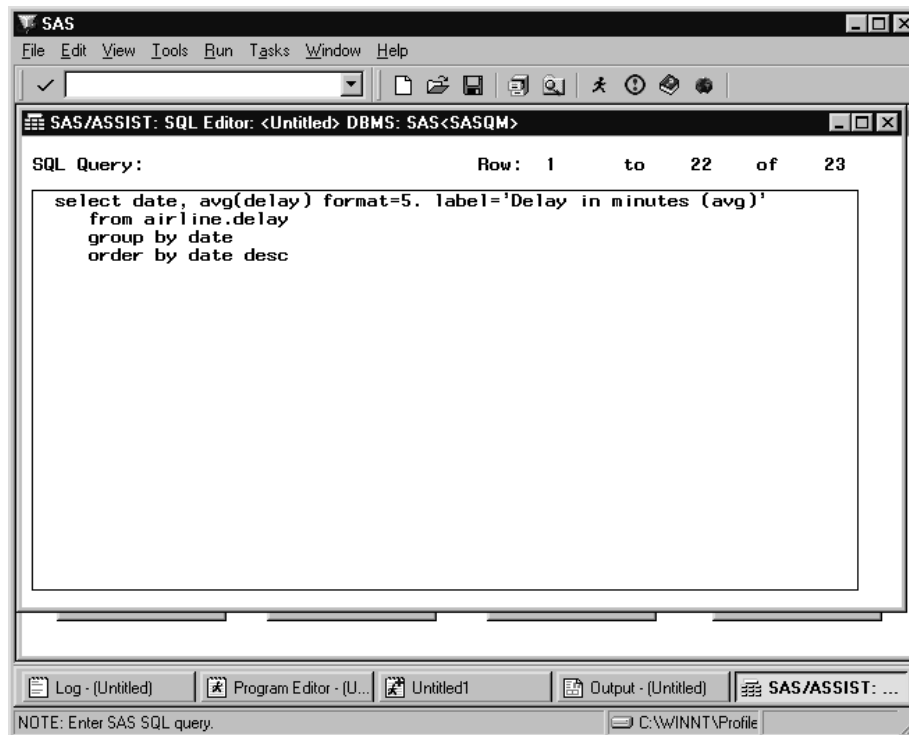
You can enter an SQL query in the SQL Editor window if you already know the libref or creator, table, and column names that you want to use.* The SQL Editor window is a complete editor with the following rules:

- Enter and submit only one SQL query at a time
- Enter one or more non-SELECT statements at a time, separating them with semicolons (;)
- Do not submit both SQL queries and non-SELECT statements at the same time.

You can insert or delete lines in the window by using items in the **Edit** menu.

The query in the following display retrieves the average number of delayed flights each day.

Display 11.2 SQL Editor Window With Query



You can now edit the query further or run it by selecting **Submit** from the **Run** menu. If you run the query, you can save your output to a SAS or DB2 table, as described in “Creating New Tables” on page 185.

Entering Non-SELECT SQL Code

Occasionally you might want to submit SQL statements that are not queries, such as statements that drop tables or create indexes. These non-query, non-SELECT statements do not produce output.

You can enter several non-SELECT SQL statements at one time, separated by semicolons. Do not end the final statement with a semicolon.

* For brevity, table means table or view unless otherwise noted. The “creator” refers to the SAS libname or your DB2 authorization identifier.

Note: Using more than one statement and semicolons applies only to non-SELECT SQL statements and not to SQL queries. Δ

Select **New** from the **File** menu to clear the SQL Editor window. Then enter your SQL code into the window. For example,

```
create index jobcode on airline.jobcde2|jobcode);
drop table airline.temp
```

Select **Submit** from the **Run** menu to submit the code to the SAS System or to DB2.

The following list includes some non-Select SQL statements that you might want to submit to your database.

CREATE	creates a table, view, index, or other objects, depending on how the statement is specified.
DELETE	deletes one or more rows from a table.
DROP	deletes a table, view, index, or other objects, depending on how the statement is specified.
INSERT	adds one or more rows to a table.
UPDATE	modifies the data in columns of a row in a table.
GRANT	gives users the authority to access or modify objects such as tables or views. (DB2 SQL only)
REVOKE	revokes the privileges that were given to users by the GRANT statement. (DB2 SQL only)

For more information and restrictions on these and other SQL statements, see your SQL documentation.

You can use the prompt facility to display a prompt window when non-SELECT SQL statements are submitted. See “Using the Prompt Facility” on page 177 for more information. Non-SELECT SQL statements can be saved to and executed from the Result Manager window. See “Saving SQL Code or Output” on page 217 and Chapter 9, “Doing More with Results,” on page 145 for information.

Creating Template SQL Programs

Many times you might want to begin a new program by using part of another program. Or, you might want to begin a new program by using the tools that Query and Reporting offers to create a template program. The following sections show you how to use these tools to create a template that you can edit and customize for your purposes.

Using a Stored Query or Statement

You can include a query that you have previously created in the Query window or include a query or non-SELECT statement that you previously created in the SQL Editor window. From the SQL Editor window, select **Open** from the **File** menu. The Open window appears with a list of your saved queries or non-SELECT statements (of types QUERY and SQL). Type an **S** (for Select) next to the query or statement that you want to include.

The Open window has the fields shown in the following table.

Table 11.1 Fields in the Open Window

Field	Description
Catalog	Specifies the current catalog name. To change it, type a two-level name or ? to list all possible catalogs. Then make a selection from the resulting Select a Member window.
Name	At the top of the display, the Name field is empty, by default. To search for and limit the listing to certain entries, type the complete name of an entry, or type a portion of a name with a pattern-matching character (such as % or _). For example, specifying JOB% retrieves all of the queries that begin with JOB . Type % for a list of all the saved entries.
Cmd	Type S or an asterisk (*) to select the query or non-SELECT statement.
Row	Displays the number of queries and non-SELECT statements. You can edit the value of the starting number to change the number of the rows displayed.
Name	In the window's listing, Name displays the names of your queries or non-SELECT statements.
Description	Describes your queries or non-SELECT statements.
Type	Lists whether the entry was saved from the SQL Editor window (SQL) or from the Query window (QUERY).
Date	Lists the date when the query or non-SELECT statement was created or last edited.

You can also sort the contents of the window by **Name**, **Description**, **Date**, or **Type** by selecting **Sort By** from the **View** menu.

After you type **S** in the **Cmd** field and press ENTER, the query or non-SELECT statement is loaded into the SQL Editor window and a message is displayed to indicate that the query has been loaded. Select **Goback** to return to the SQL Editor window.

If you load a query of type **QUERY**, changes that you make to your SQL code in the SQL Editor window are not reflected in the query that you display with the Query window. However, you can save your changed SQL code, as described in “Saving SQL Code or Output” on page 217.

You can now edit or run your SQL code. You can edit it in the SQL Editor window or select **Enhanced Editor** from the **Edit** menu. The Enhanced Editor enables you to make more significant changes to your code. See “Using the Enhanced SQL Editor” on page 211 for more information.

Selecting Tables with the Select Window

You can use the Query and Reporting Select window to include tables in the SQL Editor. Select **Query Manager** from the **Tools** menu to open the Query and Reporting Select window. Choose one or more tables and columns, as described in “Selecting Tables and Columns for Queries” on page 168.

Note: Selections made from the Query and Reporting Select window replace non-SELECT SQL statements in the SQL Editor window and append SQL queries in the SQL Query window. If you want to append table selections to non-SELECT SQL code, or want to replace SQL queries, do not use the Query and Reporting Select window; instead, use the SQL Select Table window as described in the next section. △

After you select your columns and select **C**lose from the **F**ile menu, you return automatically to the SQL Editor window. SQL code that is based on your selections is displayed in the SQL Editor window.

Under DB2 SQL, if a column name has special or national characters (such as FLIGHT#), the column automatically appears in quotation marks.

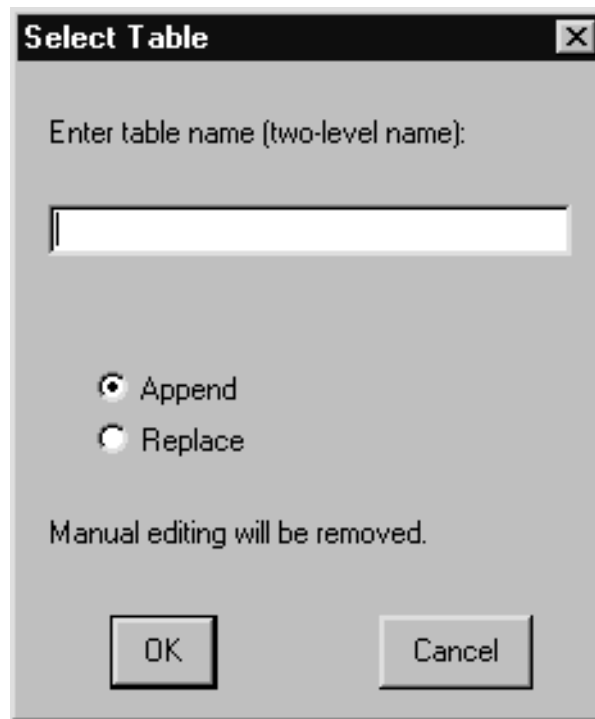
If relations are defined on the tables, the tables are joined automatically; a WHERE clause is automatically generated in the SQL code. If relations are not defined, edit the SQL code to add a WHERE clause to complete the join.

Choosing Tables with Select Table

Another way to begin a template program is to follow these instructions:

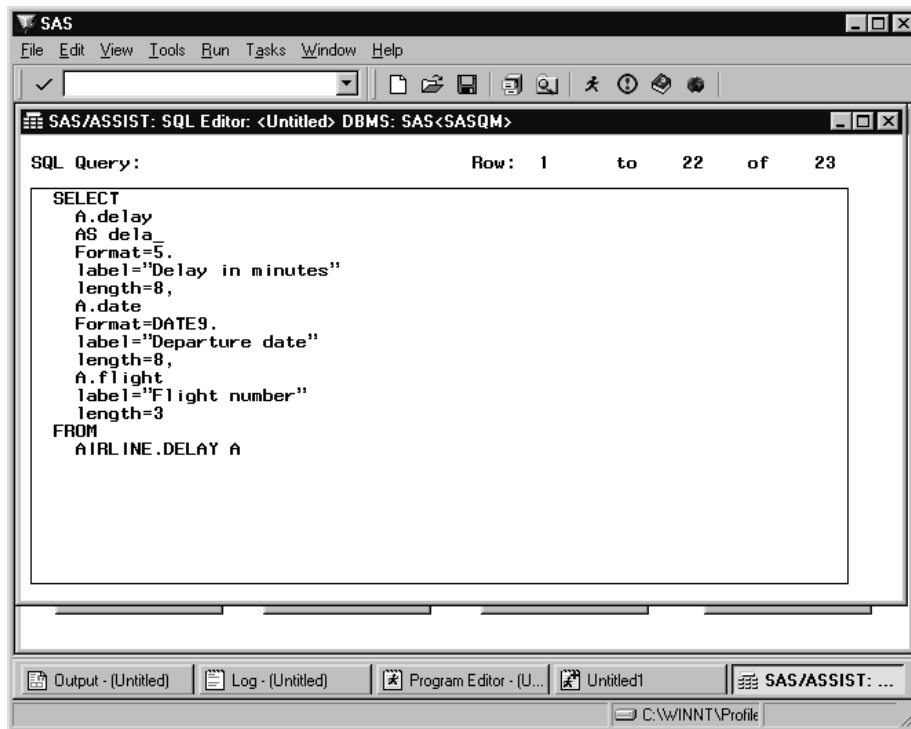
- 1 Choose **S**elect **T**able from the **E**dit menu. The SQL Select Table window appears.

Display 11.3 SQL Select Table Window



- 2 Type the two-level name of a table that you want as the basis of your query.
- 3 Select **A**ppend to append the table to the query currently in the SQL Editor window, or select **R**eplace to replace it.
- 4 Select **O**K. The SQL template program appears in the SQL Editor window. In this example, all the columns are selected from the AIRLINE.DELAY table, as shown in the following display. Table and column aliases, formats, labels, and lengths are automatically generated for the table(s) that you include.

Display 11.4 SQL Editor Window With AIRLINE.DELAY Table Selected



- 5 To join the AIRLINE.DELAY with the AIRLINE.FLINFO table, choose **Select Table** again and type the table name AIRLINE.FLINFO. Select **Append** to append the two tables. Select **OK**. If relations are defined on the tables, a WHERE clause is automatically generated in the SQL code and the join is complete.

In this example, no relations are defined on the AIRLINE.DELAY and AIRLINE.FLINFO tables. For a join to work efficiently, it must have a WHERE (or ON) clause, and so you must edit the SQL code.

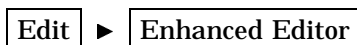
You can do minor editing in the SQL Editor window by using the **Edit** menu's **Insert Line** and **Delete Line** items. For more substantial editing, use the SQL Enhanced Editor, as described in the following section.

Note: If you decide that you do not want to join AIRLINE.DELAY with AIRLINE.FLINFO and instead want only AIRLINE.FLINFO, choose **Select Table** again. Type the name AIRLINE.DELAY and select **Replace**. The SQL template program for the join is then replaced by a template program for AIRLINE.FLINFO. △

Using the Enhanced SQL Editor

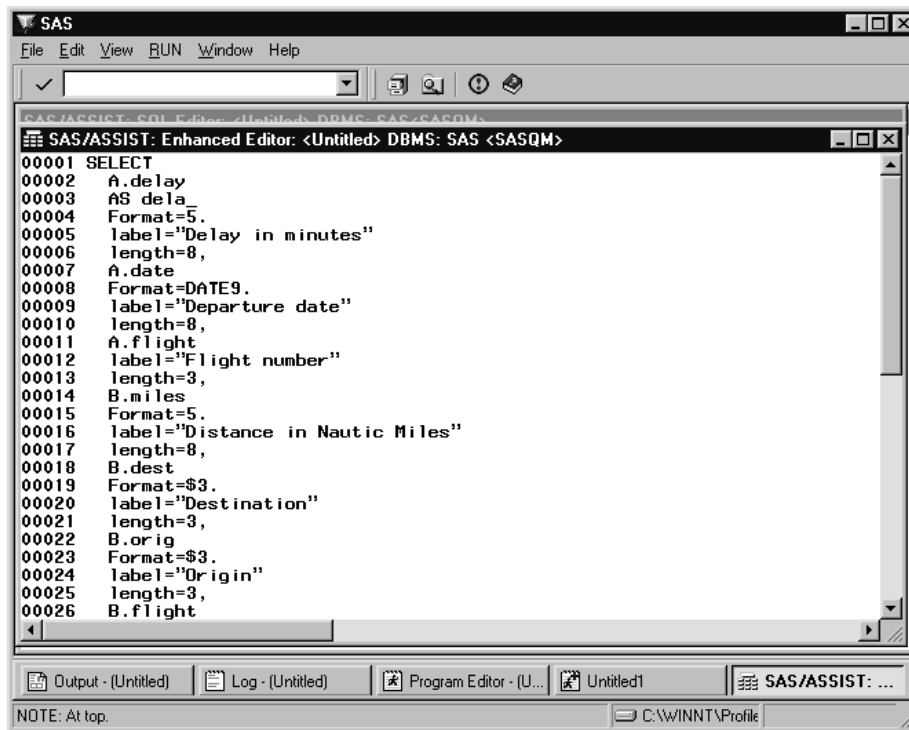
The Enhanced Editor window is useful when you want to make changes in the SQL code. You can open the Enhanced Editor window from the SQL Editor window at any time. Any code in the SQL Editor appears in the Enhanced Editor window. In this example, you edit the join between the AIRLINE.DELAY and AIRLINE.FLINFO tables that you selected in the previous section.

- 1 Follow this selection path:



The Enhanced Editor window appears with the SQL statements from the SQL Editor window.

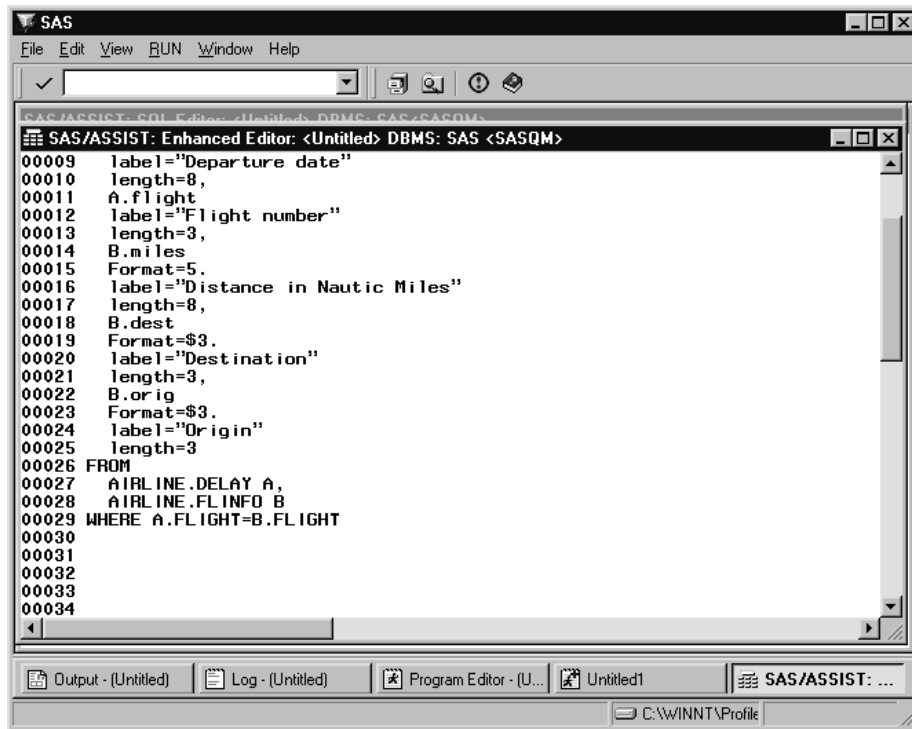
Display 11.5 Enhanced Editor Window



- 2 Edit the SQL code as follows. For a complete explanation of commands for using the Enhanced Editor, refer to the window's online help.
 - Add a WHERE clause (**WHERE A.FLIGHT=B.FLIGHT**) at the end of the SQL code to complete the join.
 - Edit the SELECT clause to remove the duplicate B.FLIGHT column along with its format, length, and label.
 - Delete the comma after **length=3** at the end of the B.ORIG column description.

The edited query is shown in the following display. The display is scrolled forward so that you can see the WHERE clause.

Display 11.6 Edited SQL Query



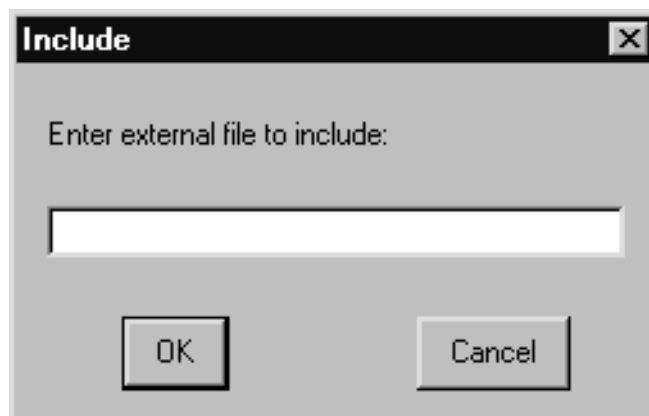
- 3 Select **C**lose from the **F**ile menu to return to the SQL Editor window. You can perform further modifications to your query, run your query, or save it.

Note: You can also run your query from the Enhanced Editor window by selecting **R**un from the **R**UN menu. △

Including and Writing SQL Code to External Files

In the Enhanced Editor window, you can include an external file that contains SQL code. Select **I**nclude from the **F**ile menu. The Include... window appears and prompts you to type a filename, as shown in the following display.

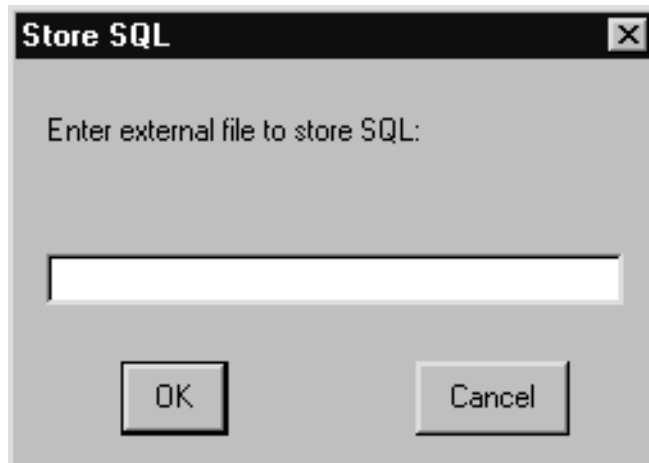
Display 11.7 Include Window



Type an external file name and select **OK** to copy the SQL code into the SQL Editor window.

To write your SQL code to an external file, select **Store SQL** from the **File** menu. The Store SQL...window prompts you for the external file name, as shown in the following display.

Display 11.8 Store SQL Window



Type an external file name and select **OK** to write the SQL code to your external file.

Note: You can include IBM QMF queries in the SQL Editor window if you convert them for use in Query and Reporting. See Appendix 4, "Migrating from QMF to SAS/ASSIST Software," on page 255 for more information. Δ

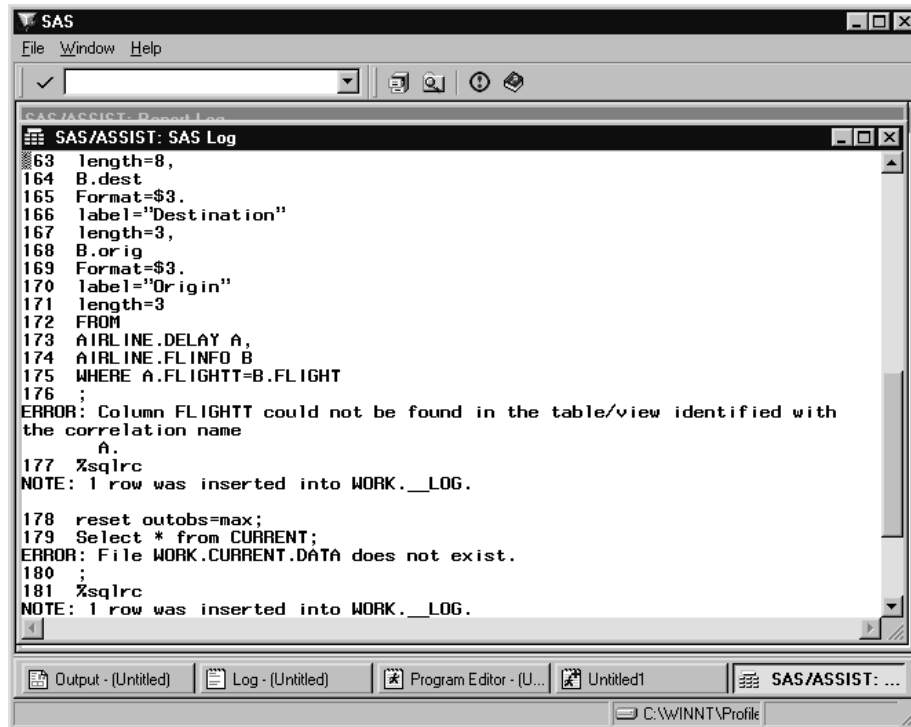
Correcting Errors in SQL Code

It is usually difficult to create perfectly correct SQL code the first time. If you run incorrect SQL code, you will get errors. The SQL Editor window has a SAS log for you to review so that you can correct your code.

Using the example from earlier in this chapter, if you incorrectly type the A.FLIGHT column name as A.FLIGHTT in the WHERE clause and you run the program, the Report Log window appears, informing you that an error has occurred.

Select **View SAS Log** from the **Run** menu to open a SAS log that is customized for SAS/ASSIST software. The following display indicates that the A.FLIGHTT column was not found in the AIRLINE.DELAY table.

Display 11.9 SAS Log Window



Select **Close** from the **File** menu twice to return to the SQL Editor window. Correct your code and run it again.

When you run your SQL code successfully, your output appears in the Report Output window (or in the output window that is defined for your system). From the SQL Editor window, you can select **Report Log** from the **View** menu to see the most recent SAS log for your output.

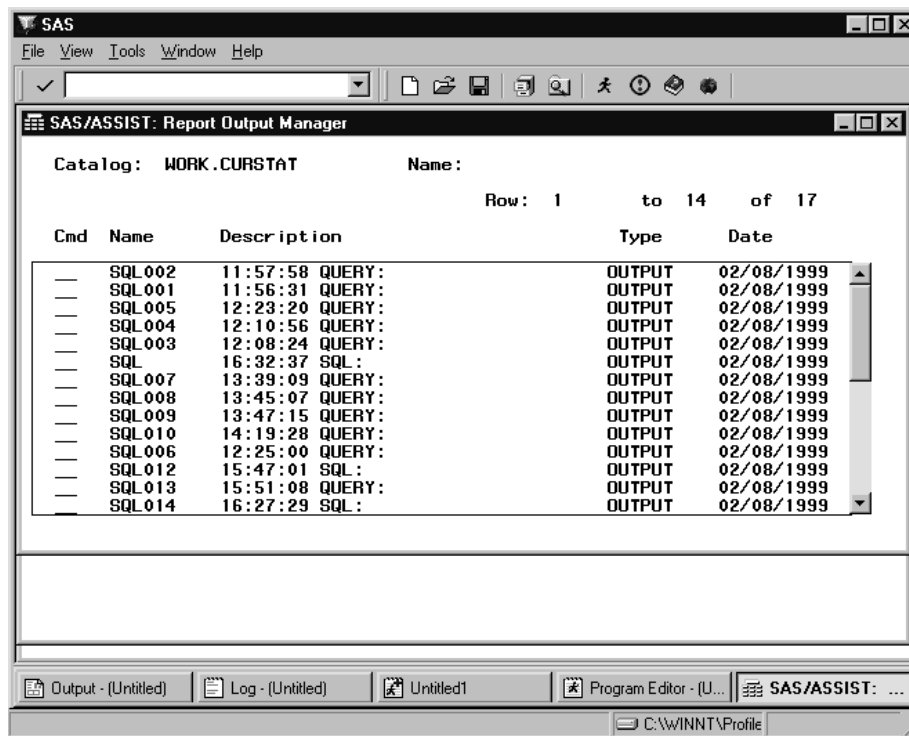
Running an SQL Query and Customizing the Output

When your SQL query is ready, you can run the query immediately or you can add a title, change its options, limit its output, and so on, to make the report look better. See “Viewing the SQL Code, Running the Query, and Refining the Output” on page 181 for more information.

To run your report immediately, select **Submit** from the SQL Editor window’s **Run** menu. Your output is displayed in the Report Output window (or in the output window that is defined for your system).

In some cases, you might want to edit your output to make your report look better or to correct minor errors, such as a mistyped name. It is more efficient to change the output for a report rather than to run the query again, especially if the tables used to create the report are very large.

To edit your output directly, select **Report Output Manager** from the **View** menu. The Report Output Manager window opens, as shown in the following display.

Display 11.10 Report Output Manager Window

The Report Output Manager lists every run of your queries in a temporary catalog, WORK.CURSTAT (current status). Type **E** or **S** in the **Cmd** field next to the output that you want to edit. This opens a window with the same text editing capabilities and commands as the Enhanced Editor window. You can edit any of the data or text in the output.

To save your changes, you must save the edited output either to an external file or to a SAS catalog entry.

- To save the edited output temporarily in the WORK.CURSTAT catalog, select **Save** from the **File** menu. When prompted, select **R** to overwrite the original OUTPUT entry in the WORK.CURSTAT catalog.
- To save your edited output permanently, select **Save As** from the **File** menu. Specify a name, description, and catalog and select **OK**.
- To save your output to an external file, select **Save As File** from the **File** menu. Specify an external file name in the Save As window and select **OK**.

From the Edit window, select **Close** from the **File** menu to return to the Report Output Manager window.

When you exit from the Report Output Manager and return to the Report Output window, you see that the changes you made are not shown in your output. The Report Output window does not automatically update to include your changes.

Note: If you do not save your changes to a permanent catalog or external file, the changes will be lost when you exit from the SAS System because WORK.CURSTAT is a temporary catalog. Δ

Select **Close** from **File** to return to the SQL Editor window.

Saving SQL Code or Output

The previous section describes how to run your query and to make minor editing changes to the output. After you have run your query, you have several options for saving your work.

- You can create a new SAS, DB2, or other database management system table from your output; see “Creating New Tables” on page 185 for details
- You can save the SQL query to a catalog entry of type SQL.
- You can save the output to a catalog entry or external file, as described in the previous section.
- save non-Select SQL statements to a catalog entry. From the SQL Editor window, select **save** from the **File** menu.

Your saved SQL queries (type SQL) can be used to make reports with the Report Engine. See Chapter 5, “Doing More with the Report Engine,” on page 61.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *Doing More With SAS/ASSIST Software, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 282.

Doing More With SAS/ASSIST Software, Version 8

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

ISBN 1-58025-520-5

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.

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.