

## CHAPTER

**10****Query and Reporting**

<i>Introduction to Query and Reporting</i>	158
<i>Additional Information</i>	158
<i>Querying Data</i>	158
<i>Opening the Query Window</i>	159
<i>Selecting a Query Manager</i>	160
<i>Opening and Using the Select Window</i>	161
<i>Typing Names into the Fields</i>	162
<i>Using Commands to Manipulate the Listings</i>	164
<i>Expanding a List to Get More Information</i>	165
<i>Selecting Tables and Columns for Queries</i>	168
<i>Using the View Menu to Manipulate Columns</i>	168
<i>Browsing or Editing the Selected Table</i>	169
<i>Updating a Query Manager</i>	170
<i>Building a Query</i>	171
<i>Customizing Your Query</i>	174
<i>Using the Cmd Field</i>	174
<i>Using the No Field</i>	174
<i>Using the Order Field</i>	175
<i>Using the Usage Field</i>	175
<i>Retrieving a Subset of Data</i>	176
<i>Using the Prompt Facility</i>	177
<i>Defining an Expression</i>	179
<i>Viewing the SQL Code, Running the Query, and Refining the Output</i>	181
<i>Viewing the SQL Code</i>	181
<i>Running the Query</i>	181
<i>Refining the Output</i>	183
<i>Limiting the Report's Output</i>	183
<i>Saving a Query</i>	184
<i>Resetting the Query Window</i>	185
<i>Performing a COUNT(*) on a Table</i>	185
<i>Creating New Tables</i>	185
<i>Creating a New SAS Table or View</i>	186
<i>Creating a New DB2 Table (OS/390 Only)</i>	187
<i>Joining Tables</i>	187
<i>Introducing Table Joins</i>	187
<i>Joining Tables Manually</i>	190
<i>Joining Tables Without Defined Relations</i>	190
<i>Joining Three Tables</i>	192
<i>Joining Tables with Outer Joins</i>	194
<i>Defining Relations on Tables</i>	196
<i>Modifying and Deleting Relations</i>	198

<i>Joining Tables Automatically</i>	198
<i>Joining Tables with Multiple Relations</i>	199
<i>Additional Considerations for Joining DB2 Tables</i>	201
<i>Referential Integrity</i>	202
<i>Query and Reporting Setup Options</i>	202
<i>Query migration</i>	203
<i>Forms migration</i>	203
<i>Administration</i>	203

---

## Introduction to Query and Reporting

This chapter introduces Query and Reporting, a menu-driven interface that is part of SAS/ASSIST software. You use this interface to query and report on both SAS data and DB2 data from the SAS System. Query and Reporting (Q & R) consists of these windows:

- Query window, which includes an intelligent Query Manager
- SQL Editor window
- Report Engine window
- Result Manager window
- Setup window.

*Note:* The Setup window is titled Query and Reporting: Setup when you access it through Query and Reporting. It is titled Setup when you access it from the WorkPlace menu or a task window. The two Setup windows contain different options.  $\Delta$

The Report Engine and the Result Manager are discussed elsewhere in this book and in *Getting Started with the SAS System Using SAS/ASSIST Software*. This chapter and the next chapter describe the capabilities of the other windows.

---

## Additional Information

For more information on performing queries with SQL, refer to “The SQL Procedure” in *SAS Procedures Guide*.

Query and Reporting provides access to DB2, IBM’s relational database management system. You must license SAS/ACCESS software and the SAS/ACCESS interface to DB2 to use this feature. For more information on DB2, refer to either *SAS/ACCESS Software for Relational Databases: Reference (DB2 under OS/390 Chapter)* or *SAS/ACCESS Software for Relational Databases: Reference (DB2 under UNIX and PC Hosts Chapter)*, depending on your operating environment. For general information on relational databases, see *SAS/ACCESS Software for Relational Databases: Reference*.

---

## Querying Data

The Query and Reporting Query window is a menu-driven interface that enables you to query SAS or DB2 data. You can create queries that access one or more tables or views. It is not necessary to know your system’s SQL syntax. In the Query window you can

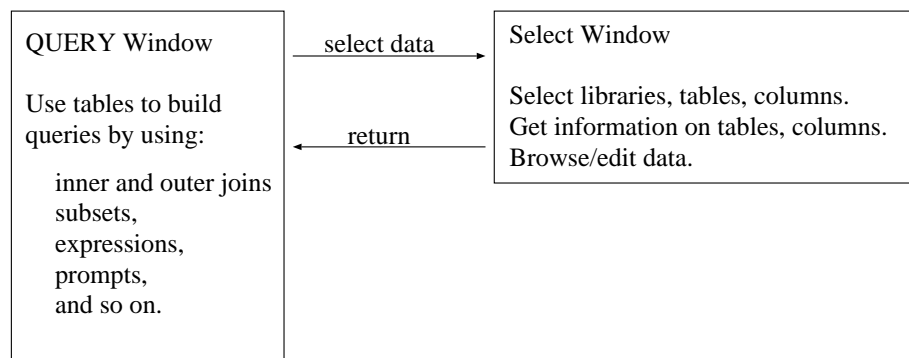
- create queries on one or more tables or views that you select
- view, browse, and save the SQL statements that are automatically generated by the queries
- save and run the queries
- create SAS tables, DB2 tables, tables from other DBMSs, or flat files from the queries' output
- perform inner and outer (left, right, and full) joins that retrieve matching and non-matching rows from the contributing tables.

The Query window supports subsetting data, arithmetic expressions, and column functions, as well as group processing and ordering. Queries can contain an unlimited number of prompts. A prompt displays a window that asks users for values when the query is executed.

The output from queries can be saved as SAS tables (that is, SAS data files or data views), as DB2 tables, as tables in any other database for which you have SAS/ACCESS software licensed, or as external files. The generated SQL code can be browsed, and queries can be saved for editing in the SQL Editor window.

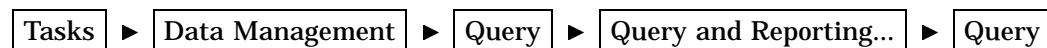
To select data for your queries, you use the Select window, which is accessed from the Query window. You select the tables, views, and columns that you use to create your queries. Then in the Query window, you can customize your queries further. The following figure illustrates this process.

**Figure 10.1** The Query Process



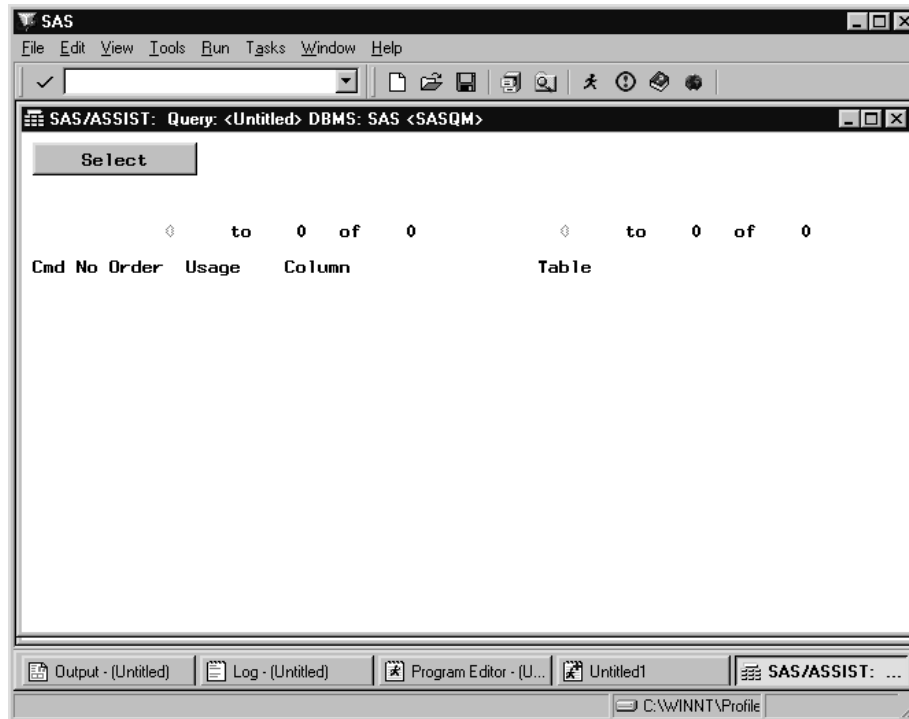
## Opening the Query Window

To open the Query window, follow this selection path:



The following display shows the Query window.

Display 10.1 Query Window



The Query window may contain columns and tables that you selected the previous time that you used the Select window. To clear the window for selecting different tables or columns, select **New** from the **File** menu.

The data that is available to you from the Query window and that is presented in the Select window is based on which Query Manager has been chosen and how that Query Manager is set up, as described in the next section.

---

## Selecting a Query Manager

A Query Manager controls access to your databases. Different Query Managers can be used to access different databases or different parts of the databases. For example, different Query Managers can be used to access different SAS libraries or different DB2 subsystems. A Query Manager defines a group of tables or views (SAS or DB2). The Query Manager maintains information such as the table and column names, indexes, column types, and relationships. The Query Manager also determines which tables, views, and columns you can use to build a query in the Select window.

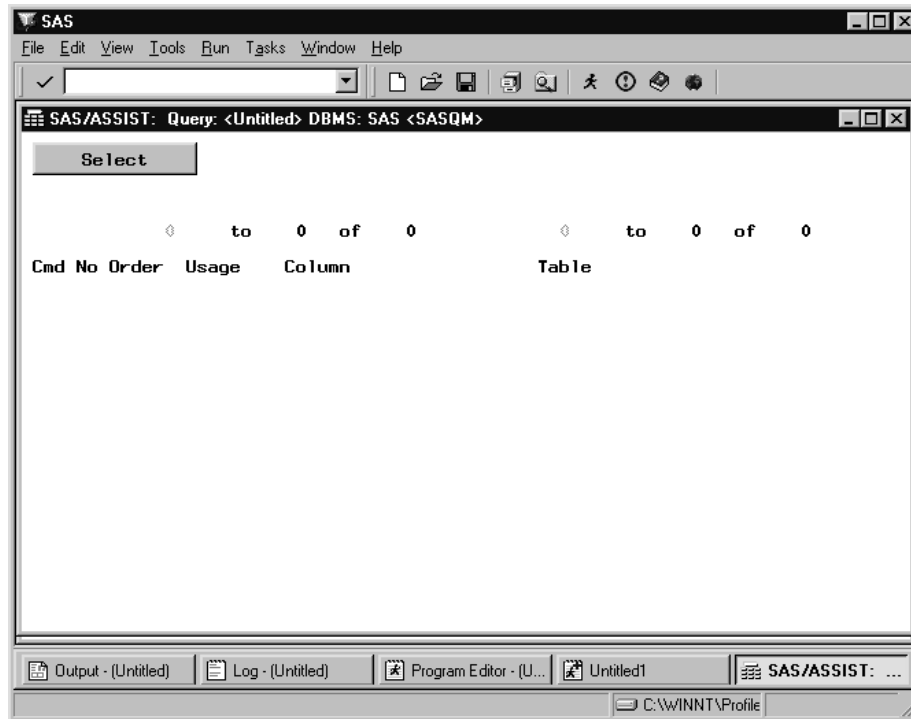
By default, a personal SAS Query Manager, SASQM, is created the first time you enter the Query window. The default personal Query Manager uses all available SAS data libraries in the current SAS session except SASHELP and MAPS.

To choose a Query Manager, follow this path from the Query window to display a list of available Query Managers.

Tools ► Select Query Manager

The following display shows a list of Query Managers.

Display 10.2 Query Manager List



Select the Query Manager you want to choose. For this chapter, the SASQM Default SAS Query Manager is used. The list of Query Managers shows the Query Manager name, description, and type. The type of Query Manager can be **M** for Master, **G** for Group, or **U** for User. In the title of the Query window, the **DBMS** field shows the type of Query Manager in use, SAS or DB2, and the name of the Query Manager, for example, SASQM.

---

## Opening and Using the Select Window

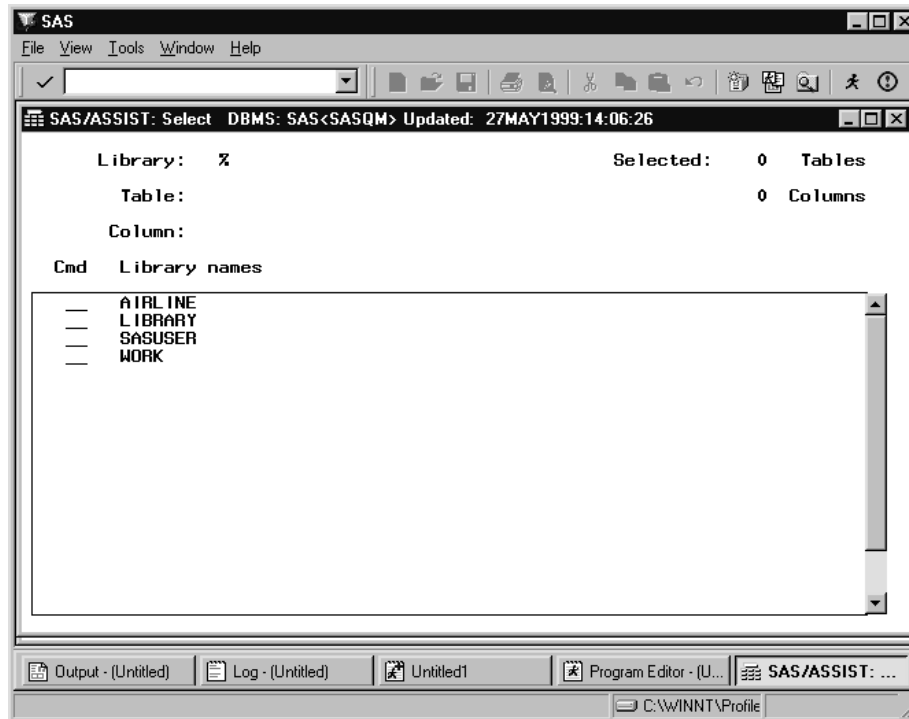
The Select window is part of the Query window. In this window, you can select the tables and columns that you want to use to build queries. You can get more information about the tables and columns in this window, including

- table and column attributes (column labels, lengths, and so on)
- indexes and key columns
- relations between database tables.

This information is gathered from database systems and additional sources (for example, third-party vendor data dictionaries) and stored in SAS files for fast access. This information is used to generate selection lists and to provide additional information about table and column descriptions, indexes, relations, and so on, between database tables. These relations are used to join tables in the Query window. Furthermore, you can browse or edit data interactively.

To open the Select window from the Query window, choose **select**. The following display shows the Select window. In this window, you specify the SAS data libraries or DB2 authorization identifiers (called *creators*), tables, and columns that you use to create your queries.

Display 10.3 Select Window



The data presented in the Select window is based on which Query Manager has been chosen and how that Query Manager is set up, as described in the previous section.

You can list your libraries (in DB2, creators), tables, and columns in this window using one of the following methods:

- 1 Type names or use pattern-matching characters in fields at the top of the window.
- 2 Manipulate a vertical listing of libraries, tables, and columns with commands.

---

## Typing Names into the Fields

The Select window has three fields where you can type names or pattern-matching characters: **Library**, **Table**, and **Column**.

The pattern-matching character (%) appears in **Library** by default and causes the library names to be listed below the fields. You can use the following pattern-matching characters as shown in the following display:

- \_ (underscore)  
 matches any single character
- % (percent)  
 matches any sequence of zero or more characters.

You can use pattern-matching characters before or after the characters that you want to match. You can use one or more pattern-matching characters in a field.

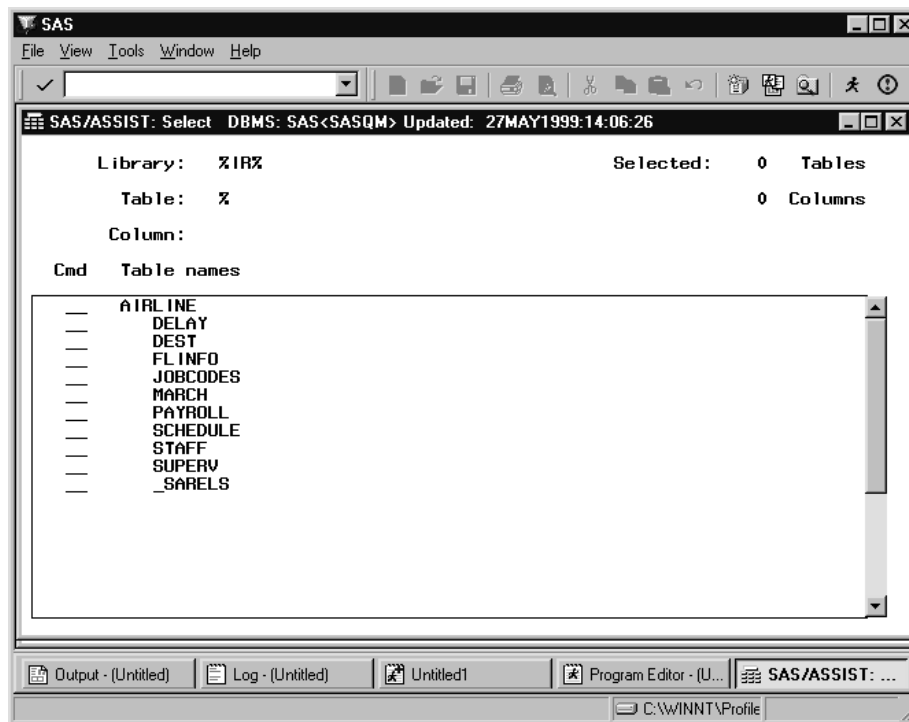
Table 10.1 Pattern-Matching Examples

Field Entry	Displays
%	all the libraries, tables, or columns, depending on the field where it is typed
<b>AIRLINE</b>	one specific library
<b>AI%</b>	one or more libraries that start with AI
<b>%NE</b>	one or more libraries that end with NE
<b>%IR%</b>	one or more libraries that contain IR
<b>AIR_INE</b>	a seven-letter library name that starts with AIR and ends with INE and can have any value in the fourth position.

Searches for the names of the libraries, tables, and columns are fast because this information is stored in the Select window.

In the following display, **AIRLINE** is typed in the **Library** field and a % is typed in the **Table** field, so that all the tables for the AIRLINE library are shown.

Display 10.4 Select Window With Pattern-Matching Character



The following table shows other ways to combine the pattern-matching characters and object names.

**Table 10.2** Examples of Ways to Use the Fields

<b>Action</b>	<b>Results</b>
Type the name of a library	The library name is listed.
Type the names of a library and table	Both the library and table names are listed.
Type the name of a library and % in <b>Table</b>	The library name and all the tables in that library are listed.
Type the name of a library and % in <b>Table</b> and <b>Column</b>	The library name, all the tables, and all of the columns in those tables are listed.
Type % in all three fields	All the library, table, and column names are listed.
Type % in <b>Library</b> and a table name in <b>Table</b>	The library name(s) for the table are listed, and duplicates of the tables are listed.
Type % in <b>Library</b> and a column name in <b>Column</b>	All library and table names for that column are listed.

## Using Commands to Manipulate the Listings

Specifying commands in the **Cmd** field next to the listed libraries, tables, and columns enables you to

- expand or collapse the list of libraries, tables, and columns
- display more information about the tables and columns
- browse or edit the data
- select or deselect tables and columns for your queries.

The following table summarizes the Select window commands that give you more information about your objects. Examples and more information follow this table.

**Table 10.3** Select Window Commands

<b>Cmd</b>	<b>Action</b>	<b>Description</b>
?	List commands	Lists all possible commands.
+	Expand	Shows the tables of a specific library or the columns of a specific table. When columns are listed, any indexes defined on those columns are indicated in the <b>Index</b> field.
-	Collapse	Returns to the previous level; for example, returns from listing the columns to listing just the table names.
<b>S</b>	Select	For a table or column, selects it for inclusion in a query. Specify <b>S</b> to select individual columns in a table (rather than selecting the entire table and then deselecting columns that you do not want).
<b>D</b>	Deselect	Removes a table or column from the query; it has no effect on the table itself.
<b>I</b>	Info	Provides detailed information about a table or column, including its indexes, relations, and so on.
<b>EL</b>	Edit List	Edits multiple rows in a table by using the SAS FSVIEW procedure in edit mode. <b>EL</b> requires UPDATE privileges and SAS/FSP software.



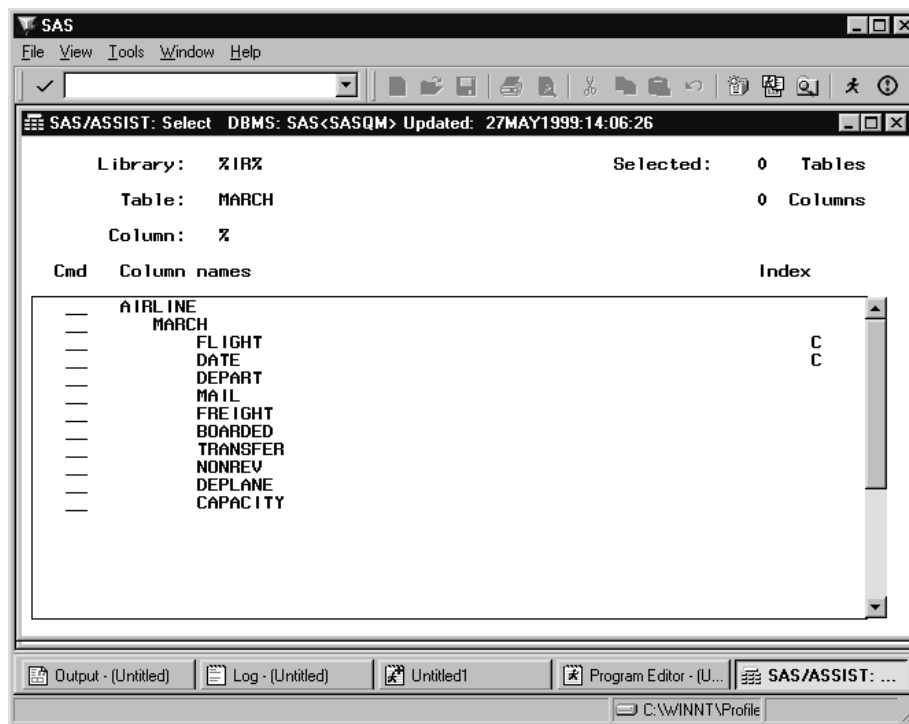
Cmd	Action	Description
<b>E</b>	Edit Rows	Edits a table one row at a time by using the SAS FSEDIT procedure. <b>E</b> requires UPDATE privileges and SAS/FSP software.
<b>BL</b>	Browse List	Browses multiple rows of data in a table by using the SAS FSVIEW procedure. <b>BL</b> requires SAS/FSP software.
<b>B</b>	Browse Rows	Browses the data in a table one row at a time by using the SAS FSBROWSE procedure. <b>B</b> requires SAS/FSP software.

*Note:* You can type as many commands as you want before you press ENTER. For example, you can type ten **S** commands to select ten columns and then submit all of them by pressing ENTER once.  $\Delta$

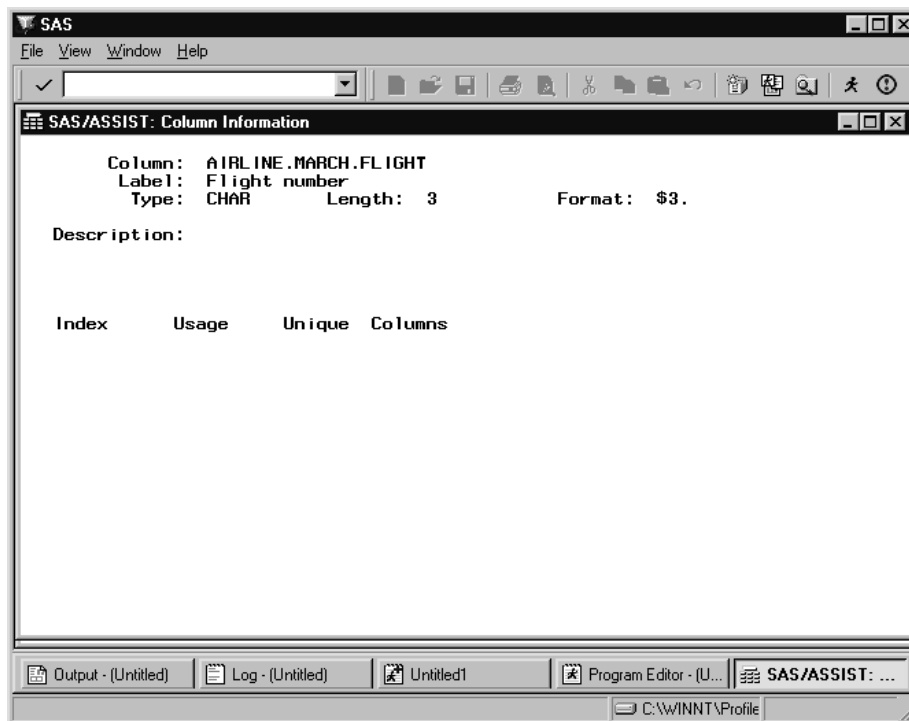
## Expanding a List to Get More Information

To expand the AIRLINE.MARCH table shown in Display 10.4 on page 163 to list the columns, type **+** in the **Cmd** field next to **MARCH** and press ENTER. The following display shows the expanded table. Notice that the fields at the top of the display change, and that columns with composite indexes are indicated.

Display 10.5 Select Window With Expanded List



Type **I** next to the FLIGHT column to open the Column information window, as shown in the following display. **I** can also be used with tables or columns listed in the Query window and its subwindows.

**Display 10.6** Column Information Window

In this example, the Column Information window displays the full column name, its label, and other column attributes.

The following table gives more information about each field in the Column information window:

**Table 10.4** Column Information Fields

Field	Description
<b>Column</b>	Fully qualified name for the column.
<b>Label</b>	Label for the column.
<b>Type</b>	Data type for the column.
<b>Length</b>	Width of the column.
<b>Format</b>	(SAS only.) Column format.
<b>Nulls</b>	(DB2 only.) <b>N</b> means that the column cannot contain NULL values. <b>Y</b> means that the column can contain NULL values.
<b>Description</b>	Additional information about the column.
<b>Index</b>	Name(s) of the index(es).
<b>Usage</b>	(SAS only.) <b>Simple</b> means an index based on a single column. <b>Composite</b> means an index based on two or more columns.
<b>Cluster</b>	(DB2 only.) <b>Y</b> means that the index is a cluster index. <b>N</b> means that the index is not a cluster index.

Field	Description
<b>Unique</b>	Whether the index is unique. <b>Yes</b> (SAS only) means that the index is unique. <b>D</b> (DB2 only) means that duplicates are allowed. <b>U</b> (DB2 only) means that the index is unique. <b>P</b> means that the index is primary key and unique.
<b>Order</b>	(DB2 only.) <b>A</b> means ascending, <b>D</b> means descending.
<b>Columns</b>	Columns that contribute to the index.

If information does not apply to a field, the field is left blank. Some fields do not apply to views. If a > sign appears in a field, select it to see additional information. Select **Close** from the **File** menu to return to the Select window.

From the Select window, type **I** next to a table name to open the Table information window. This opens a window that is similar to Display 10.6 on page 166. The following table lists information about each field in the Table information window.

**Table 10.5** Table Information Fields

Field	Description
<b>Table</b>	Name of the table or view.
<b>Label</b>	Description of the contents of the table.
<b>Created</b>	(SAS only.) Date when the table was created.
<b>Modified</b>	(SAS only.) Date when the table was last modified.
<b>Primary Key</b>	(DB2 only.) Column(s) that make up the table's primary key, if the key is defined.
<b>Description</b>	Additional information about the table.
<b>Database</b>	(DB2 only.) Name of the DB2 database where the table resides.
<b>Tablespace</b>	(DB2 only.) Name of the DB2 table space within the database where the table resides.
<b>Columns</b>	(SAS only.) Number of columns in the table.
<b>Pages</b>	(DB2 only.) Number of pages (typically of 4Kb) that the table uses. This value is -1 if no statistics have been gathered or if it is a DB2 view.
<b>Rows</b>	Number of rows in the table. For DB2 only, <b>Rows</b> is -1 if no statistics have been gathered using the DB2 RUNSTATS utility.
<b>Type</b>	For SAS, <b>DATA</b> or <b>VIEW</b> . For DB2, <b>T</b> means table, <b>V</b> means view.
<b>Relation</b>	Names of referential constraint(s) in the table.
<b>References</b>	Column(s) in the current table that reference the columns of the referenced table.
<b>Foreign key(s)</b>	(DB2 only.) Column(s) in the current table that reference the primary key of the referenced table.

If information does not apply to a field, the field is left blank. Some fields do not apply to views. If a > sign appears in a field, select it to see additional information.

*Note:* The Information windows are data specific and may vary depending on the data being queried.  $\Delta$

Depending on the setup at your site, you may be able to access additional information about a column or table by selecting **Additional Information** from the **View** menu. Select **Goback** to return to the Column Information or Table Information window from the Additional Information window. The ability to retrieve additional information is controlled by the **Additional Information** User Profile option. See Appendix 2, “Customizing SAS/ASSIST Software,” on page 241 for details.

Select **Close** from the **File** menu to return to the Select window.

## Selecting Tables and Columns for Queries

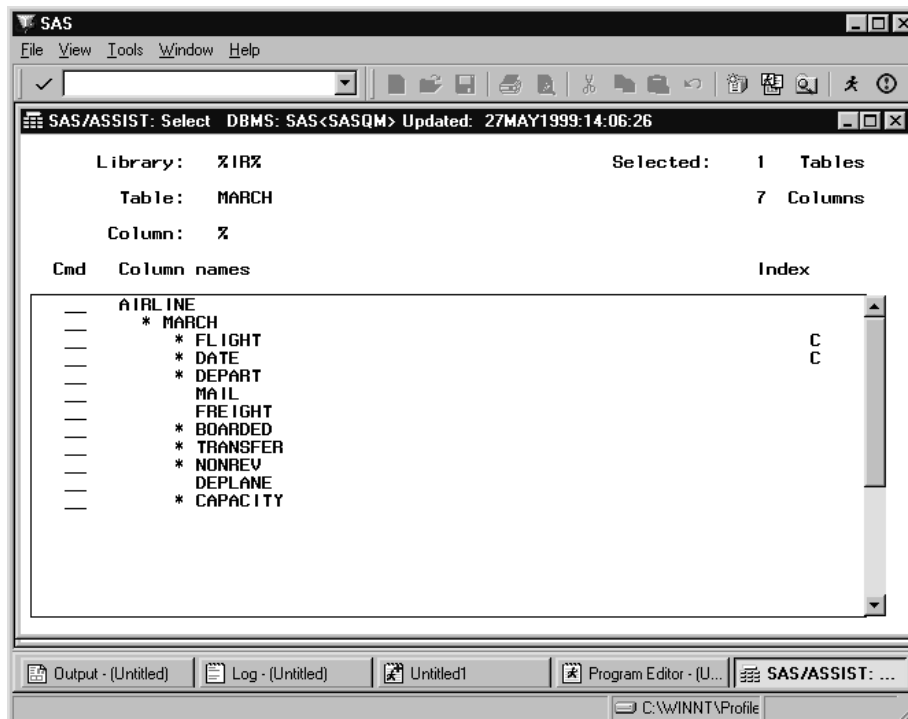
To select tables and columns for inclusion in your queries, type **S** next to the table or column name or names in the Select window. You can type as many commands as you want before you press ENTER.

When you select a table name, you automatically select all of the columns in that table. Asterisks (\*) appear next to the table name and all of the columns.

If you want only a subset of the columns, type **D** (for deselect) next to the previously selected columns that you want to omit. The asterisk next to the name is removed. You can also use **D** to deselect a table. Or you can specify **S** to select individual columns in a table (rather than selecting the entire table and then deselecting columns that you do not want).

The following display shows the AIRLINE.MARCH table after it has been selected and some of its columns have been deselected.

**Display 10.7** Columns Selected for the Query



## Using the View Menu to Manipulate Columns

You can use the items on the **view** menu in the Select window to display all the columns or just selected columns, and to deselect columns. You can also use this menu

to display the column names or column labels. You can use these menu items instead of the commands.

Show ► All

Displays all the columns in the table(s).

Show ► Selected

Displays only selected columns in the table(s).

Deselect All Columns

deselects the table(s) and all the columns.

Show Column ► Labels

Displays the columns' labels instead of the names.

Show Column ► Names

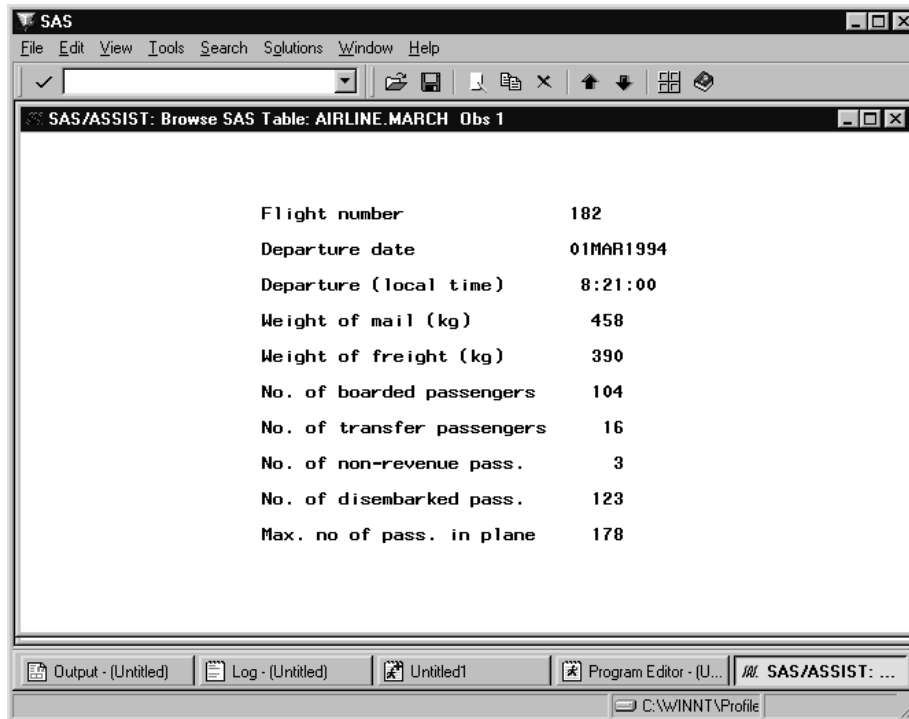
Displays the columns' names instead of the labels.

The **Show** item has two choices that act as a toggle. If you choose **Selected**, only the columns you have selected are displayed. The search fields are shown in another color or are "grayed out." Choose **All** to show all columns and to be able to type names in the **Library** and other fields at the top of the window.

## Browsing or Editing the Selected Table

If you have SAS/FSP software installed at your site, the Select window also enables you to display the data that you have selected. To browse the data, in the **cmd** field next to a table name, type **B** (to browse one row at a time) or **BL** (to browse multiple rows). The following table shows the AIRLINE.MARCH table that is displayed by using **B**. All of the table's columns are displayed, not just the columns selected for the query.

Display 10.8 Browsing the AIRLINE.MARCH Table



Variable	Value
Flight number	182
Departure date	01MAR1994
Departure (local time)	8:21:00
Weight of mail (kg)	458
Weight of freight (kg)	390
No. of boarded passengers	104
No. of transfer passengers	16
No. of non-revenue pass.	3
No. of disembarked pass.	123
Max. no of pass. in plane	178

*Note:* For OS/390 only, settings of the Table Browse and Table Edit options in your User Profile can affect your ability to browse or edit DB2 tables in the Select window (see Appendix 2, “Customizing SAS/ASSIST Software,” on page 241) .  $\Delta$

To display a subset of the data, from the Browse window, select **where** from the **Search** menu, specify a WHERE clause, and select **OK**. Select **where** again and **OK** without a where clause in the Where window to clear the clause. To edit or update the data in a table that you have selected, type **E** (to edit one row) or **EL** (to edit multiple rows) beside the table name in the Select window. You must have UPDATE access to the table to open it with these commands. Select **Close** from the **File** menu to exit the EDIT or BROWSE window.

After you have selected the tables and columns for queries, you can build your queries, customize them, and run the queries.

---

## Updating a Query Manager

A Query Manager can be static or dynamic. A static Query Manager gives you quick access to information about the structure of your SAS or DB2 tables. However, a static Query Manager requires that you update the Query Manager tables when the structure of the tables changes, for example, when new tables are added to your SAS data libraries or when new columns are added to the tables.

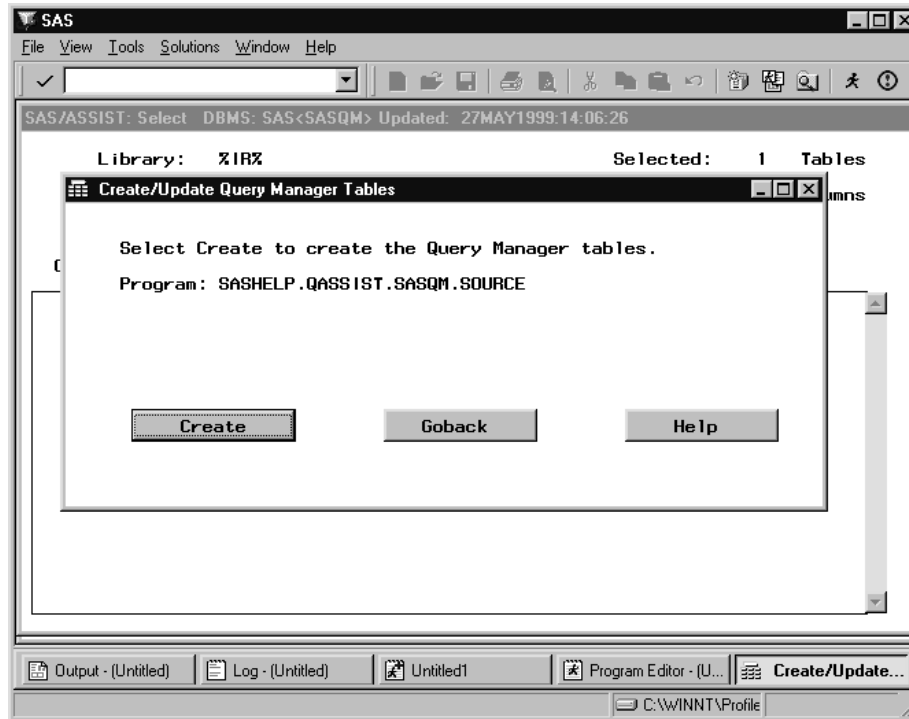
A dynamic Query Manager always reflects the most current structure of the SAS or DB2 tables to which you have access in your SAS session. For example, if you add new tables (SAS data sets) to your SAS data libraries or add new columns to your tables, these changes are automatically reflected by the dynamic Query Manager. However, information about the structure of the tables is displayed more slowly when you use a dynamic Query Manager.

If you use a static Query Manager, you should update your Query Manager periodically to reflect the most current structure of your tables. The time stamp in the

title line of the Select window shows when the static Query Manager tables were last updated.

To update the Query Manager, select **Update Query Manager** from the **Tools** menu of the Select window. The Create/Update Query Manager Tables window appears.

**Display 10.9** Create/Update Query Manager Tables Window



If a Query Manager creation program has been defined, you can select **Create** to execute this program. The program updates your Query Manager to reflect the current structure of your tables. The name of the program that is used to update the Query Manager tables is shown in the Create/Update Query Manager Tables window.

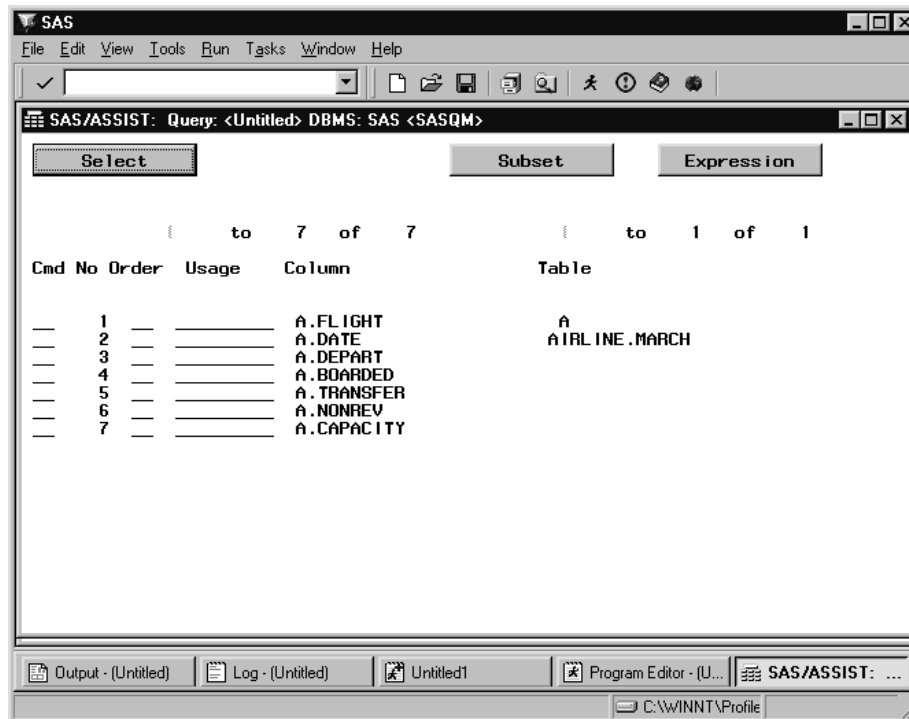
The decision to make a Query Manager static or dynamic is normally made by the designer of the Query Manager, usually a system administrator. Contact your SAS Support Consultant for details.

---

## Building a Query

To build a query on the table that you have selected, you must first return to the Query window; from the Select window, choose **Close** from the **File** menu. Any tables that you have selected in the Select window are displayed in the Query window, as shown in the following display. **<UNTITLED>** appears in the Query window name because you have not given your query a name yet (you do so later in “Saving a Query” on page 184).

Display 10.10 Query Window With AIRLINE.MARCH Table



The Query window assigns aliases to the tables and columns. An *alias* is a temporary, alternate name for a table. If you select two tables that have columns with the same name, the aliases are used so that you can distinguish them. In the previous display, the AIRLINE.MARCH table is assigned an alias of **A**.

If you chose one table in the Select window, **Subset** and **Expression** are displayed. If you select multiple tables, **Join** is also displayed in the window.

If your query has many columns or several tables so that the list of columns extends beyond one display of this window, you can see the other columns in one of two ways:

- Scroll forward (down) or backward (up) by using the **View** menu items.
- Change the numbers in the ranges above the listed columns and tables. For example, the AIRLINE.MARCH table in the previous display shows this range:

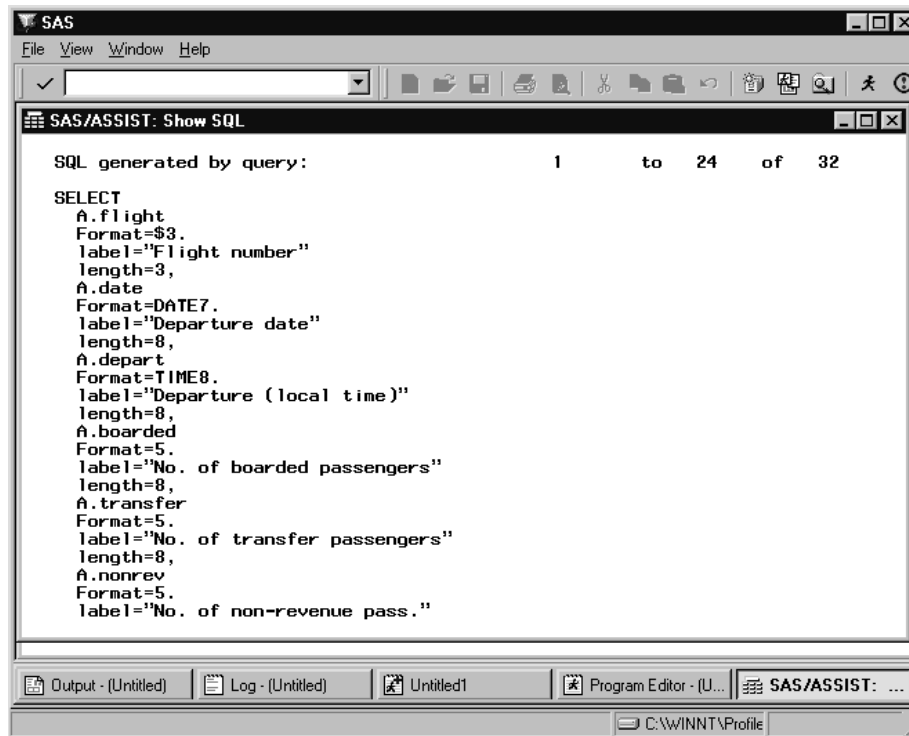
1 to 7 of 7      1 to 1 of 1

The range on the left refers to the columns in the table. This example shows selected columns for the query on AIRLINE.MARCH. However, if you had many columns in your table, you could specify the first number of the range in the first field. For example, if you have 27 columns in the table, and you change the first number in the left range to 5 and press ENTER, the range changes to **5 to 20 of 27**. Columns 5 to 20 are displayed in the window. The total number of columns in the table does not change.

The range on the right refers to the number of tables in the query. Again, you can specify the first number of the range that you want to see. The total number of tables does not change.

To see the SQL that has been created by selecting columns from the AIRLINE.MARCH table, select **Show SQL** from the **View** menu. The first page of the SQL code is shown in the following display.



**Display 10.11** Show SQL Window

```
SAS
File View Window Help
SAS/ASSIST: Show SQL
SQL generated by query: 1 to 24 of 32
SELECT
  A.flight
  Format=$3.
  label="Flight number"
  length=3,
  A.date
  Format=DATE7.
  label="Departure date"
  length=8,
  A.depart
  Format=TIME8.
  label="Departure (local time)"
  length=8,
  A.boarded
  Format=5.
  label="No. of boarded passengers"
  length=8,
  A.transfer
  Format=5.
  label="No. of transfer passengers"
  length=8,
  A.nonrev
  Format=5.
  label="No. of non-revenue pass."
```

The screenshot shows the SAS/ASSIST interface with the 'Show SQL' window open. The window title is 'SAS/ASSIST: Show SQL'. The main content area displays the SQL code generated by the query, starting with 'SELECT' and listing various columns with their respective formats and labels. The columns include flight numbers, departure dates, departure times, boarded passengers, transfer passengers, and non-revenue passengers. The window also shows a range indicator '1 to 24 of 32' and a taskbar at the bottom with several open windows.

This window also has a range so that you can scroll to a particular line of code. Select **C**lose from the **F**ile menu to return to the Query window.

To run this query, within the Query window, select **S**ubmit from the **R**un menu. The following display shows the results.

**Display 10.12** Query Results

Flight number	Departure date	Departure (local time)	No. of boarded passengers	No. of transfer passengers	No. of non-revenue pass.
182	01MAR1994	8:21:00	104	16	3
114	01MAR1994	7:10:00	172	18	6
202	01MAR1994	10:43:00	151	11	5
219	01MAR1994	9:31:00	198	17	7
439	01MAR1994	12:16:00	167	13	5
387	01MAR1994	11:40:00	152	8	3
290	01MAR1994	6:56:00	96	16	7
523	01MAR1994	15:19:00	177	20	3
982	01MAR1994	10:28:00	49	19	2
622	01MAR1994	12:19:00	207	15	5
821	01MAR1994	14:56:00	205	13	4
872	01MAR1994	13:02:00	145	13	5
416	01MAR1994	9:09:00	71	18	4
132	01MAR1994	15:35:00	115	24	5
829	01MAR1994	13:38:00	75	16	5
183	01MAR1994	17:46:00	80	19	3
271	01MAR1994	13:17:00	138	14	6
921	01MAR1994	17:11:00	122	8	4

When you return to the Query window, a message indicates how many rows were retrieved.

*Note:* You can use the Report Engine window to enhance the report, as described in Chapter 5, “Doing More with the Report Engine,” on page 61. Select **Report Engine** from the **view** menu to go directly to the Report Engine window.  $\Delta$

---

## Customizing Your Query

You can customize your query by re-ordering the columns, applying aggregate (summary) functions to the columns, retrieving a subset of the data, setting up a prompt for queries, or adding fields that include arithmetic expressions to the selection list.

### Using the Cmd Field

The **cmd** field accepts these commands:

- I** Displays information about the column, as described in “Expanding a List to Get More Information” on page 165.
- D** Permanently deselects (excludes) one or more columns from your query.
- R** Repeats the row, adding the duplicate after the row to be repeated.
- ?** Displays a list of available commands.

### Using the No Field

Although the **no** field (abbreviated from “Number”) does not accept commands, you can use it to perform these actions:

- 1 Blanking out or “erasing” a number in this field omits the column temporarily from the query. The omitted column moves to the bottom of the column list. You can retype the number to select the column again.
- 2 Typing a new number over a current number reorganizes the column order accordingly. For example, if you want **A.NONREV** to be the first column in Display 10.10 on page 172, type **1** in the **NO** field next to **A.NONREV** and a different number for **A.FLIGHT** (the previous Column 1). When you press ENTER, the columns are reordered.

## Using the Order Field

The **Order** field accepts these commands:

- A** Organizes the data within each column in ascending order, that is, in numeric order or alphabetic order from A to Z.
- D** Organizes the data within each column in descending (reverse) order, such as the highest miles traveled to the lowest miles traveled.
- ?** Displays a list of available commands.

You can designate one column as **A** and another as **D** within the same query. If you want the sorting order (ascending or descending) to be different from the column order, you can combine a letter (**A** or **D**) with a number in the **Order** field. This affects only the query’s output and not the ordering of any columns in the Query window.

For example, to list the dates and departure times in the AIRLINE.MARCH table in ascending order, you specify **A1** next to the **A.DATE** and **A2** next to **A.DEPART**. This would have the same effect as if you wrote the following SQL query:

```
select a.depart, a.date
      from airline.march a
      order by a.date asc, a.depart asc;
```

## Using the Usage Field

The **Usage** field accepts several commands that apply summary (aggregate) functions on the values in the designated column as shown in the following table.

Table 10.6

Command	Function
<b>GROUP</b>	Groups data by the values in the column
<b>SUM</b>	Computes the sum of values in the column
<b>MIN</b>	Lists the lowest value (minimum)
<b>MAX</b>	Lists the highest value (maximum)
<b>MEAN</b>	Averages or finds the mean of values in the column; equivalent to AVG
<b>AVG</b>	Averages or finds the mean of values in the column
<b>COUNT</b>	Counts the number of distinct, nonmissing values
<b>N</b>	Counts the number of distinct, nonmissing values
<b>CSS</b>	Reports the corrected sum of squares
<b>CV</b>	Returns the coefficient of variation (percent)

Command	Function
<b>NMISS</b>	Reports the number of missing values
<b>PRT</b>	Reports the probability of a greater absolute value of Student's $t$
<b>RANGE</b>	Reports a range of values
<b>STD</b>	Shows the standard deviation
<b>STDERR</b>	Returns the standard error of the mean
<b>SUMWGT</b>	Returns the weighted sum
<b>T</b>	Returns the Student's $t$
<b>USS</b>	Returns the uncorrected sum of squares
<b>VAR</b>	Returns the variance

When you use summary functions in queries, columns with no usage are treated as if they have GROUP usage. With SAS data, use the **Grouping in summary** User Profile option to specify which action to take when using summary functions in queries. If you set this option to **Pop-up**, you are prompted whether or not to apply the GROUP usage to all columns. If you set this option to **Yes**, the GROUP usage is automatically applied to all columns. If you set this option to **No**, the GROUP usage is not applied to any columns. See Chapter 9, "Doing More with Results," on page 145 and Appendix 2, "Customizing SAS/ASSIST Software," on page 241 for additional information. Note that in DB2 queries, the GROUP usage is always applied to all columns.

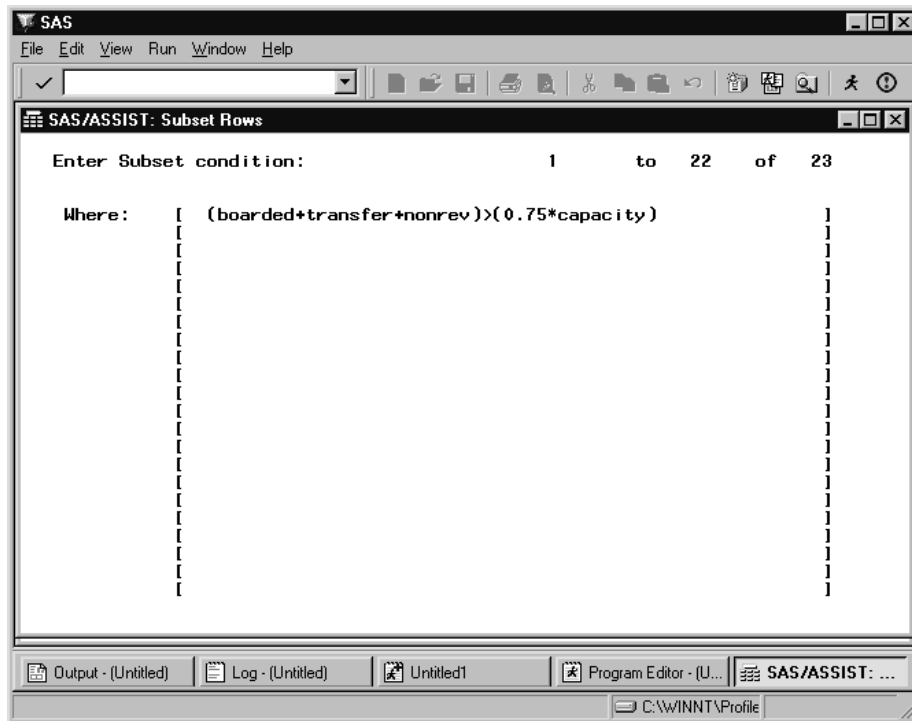
## Retrieving a Subset of Data

To retrieve some but not all rows of data in a query, apply subsetting criteria to the data. In this example, you apply a subsetting criterion that retrieves information about flights that were filled to at least three-quarters capacity.

- 1 Select **Subset** from the Query window to open the Subset Rows window. In this window, you specify your subsetting criteria (also called WHERE predicates). Do not include the word WHERE.
- 2 On the first line, type the following:

```
(boarded+transfer+nonrev)>=(0.75*capacity)
```

The following display shows the edited Subset Rows window.

**Display 10.13** Subset Rows Window

Note that if only one table is selected for the query, the alias is not required for identifying columns in the Subset Rows window.

- 3 Follow this selection path to save your subset criteria and return to the Query window:

File

▶
Close

The word **Specified** appears under **Subset**, so you know that selection criteria have been applied to this query.

If you do not remember the name of a column, you can choose it from a selection list. Position your cursor in your subsetting criteria where you want the column name. Select **Add Column Name** from the **Edit** menu. A list of columns in the current tables appears. First select a table from the list, and then select a column name. The column name is added to your subsetting criteria.

## Using the Prompt Facility

You can use prompts to run the same query several times on different subsets of data. When you run a query with a prompt, a window opens where you type a subset value. The rest of the query is then run with this value. Prompts give you more flexibility in customizing your program.

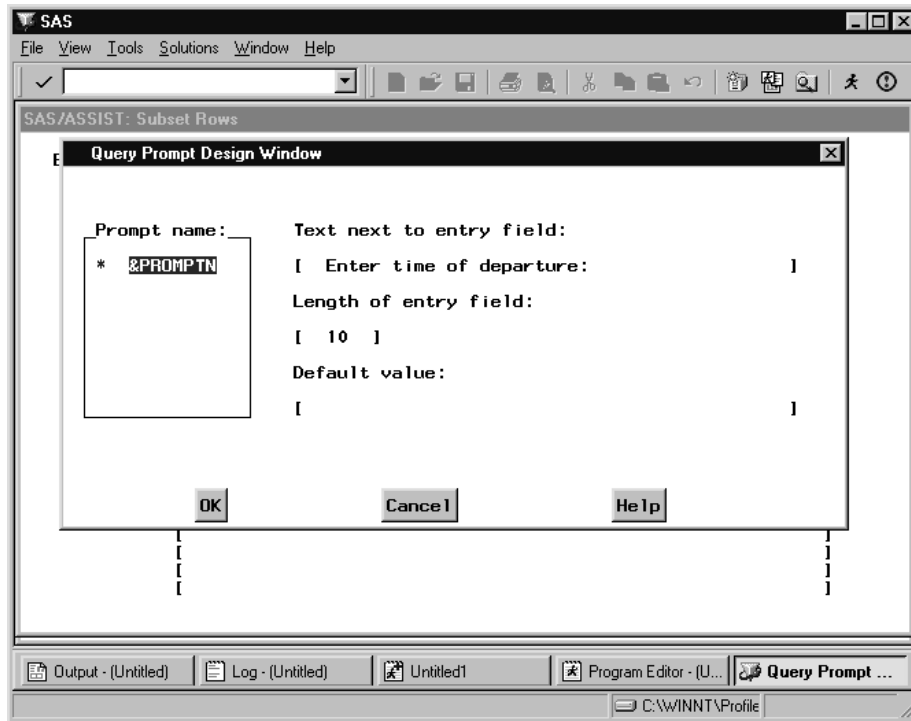
The following steps show you how to create a prompt window. In this example, you create a prompt that asks for a time value. The prompt is used to return only rows with a departure time of or earlier than the user-supplied time value.

- 1 Re-open the Subset Rows window.
- 2 On the second line, type the following:

```
and depart<=&promptn
```



Display 10.15 Query Prompt Design Window



The **Default value** field is optional; if used, this value appears in the prompt window for the user to either accept or overwrite.

- 5 Select **OK** to return to the Subset Rows window.
- 6 Select **Close** from the **File** menu to return to the Query window.

You can create an unlimited number of prompts for a query. You can use a prompt to substitute any part of the subsetting criteria, such as a logical operator (AND, OR, NOT), and you can use SAS macro functions.

*Note:* The prompt facility is also available from the SQL Editor window and from the Query Expression window. △

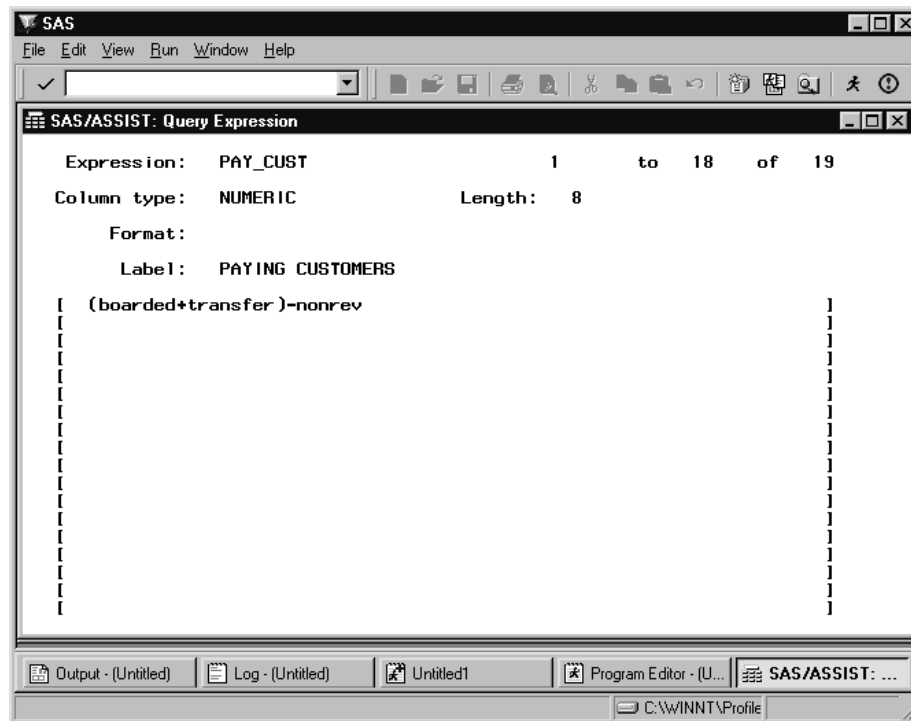
## Defining an Expression

An expression uses column names, functions, or constants to compute a result. The expression creates a new column in the query. In this example, you create a column that contains the number of paying customers on each flight.

- 1 Select **Expression** from the Query window. The Query Expression window appears.
- 2 The expression is called **EXP\_nnn** by default. If desired, change it by typing over the name in the **Expression** field. You can change the data type in the **Column type** field by typing over the default value or by typing a ? in the field to display a selection list of data types. You can also supply values for the **Format** and **Label** fields.

Add your expression below the **Label** field as shown in the following display.

Display 10.16 Query Expression Window



Note that if only one table is selected for the query, the alias is not required when identifying columns in the Query Expression window.

The previous display shows the Query Expression window filled in, with the name changed to PAY\_CUST, the column type as NUMERIC, a LENGTH of 8, and the label "PAYING CUSTOMERS" assigned. An expression to calculate the number of paying customers is added after the label.

If you do not remember the name of a column, you can choose it from a selection list. Position your cursor in your expression where you want the column name. Select **Add Column Name** from the **Edit** menu. A list of columns in the current tables appears. First select a table from the list and then select a column name. The column name is added to the expression.

- 3 Select **Close** from the **File** menu to return to the Query window. The word **Specified** appears under **Expression** to indicate that an expression has been applied to this query, and the new column, PAY\_CUST, appears in the list of columns.

The next time you select **Expression** from the Query window, a list of expressions appears; select an expression to modify or select **<NEW>** to create a new expression. Alternatively, you can edit an expression by typing **I** in the **Cmd** field next to the expression name.

You can add unlimited expressions to a query. To delete an expression, select **Expression** from the Query window and select the expression to be selected. From the Query Expression window, select **Clear** from the **Edit** menu. Alternatively, delete the column in the Query window by typing **D** in the **Cmd** field next to the expression name.



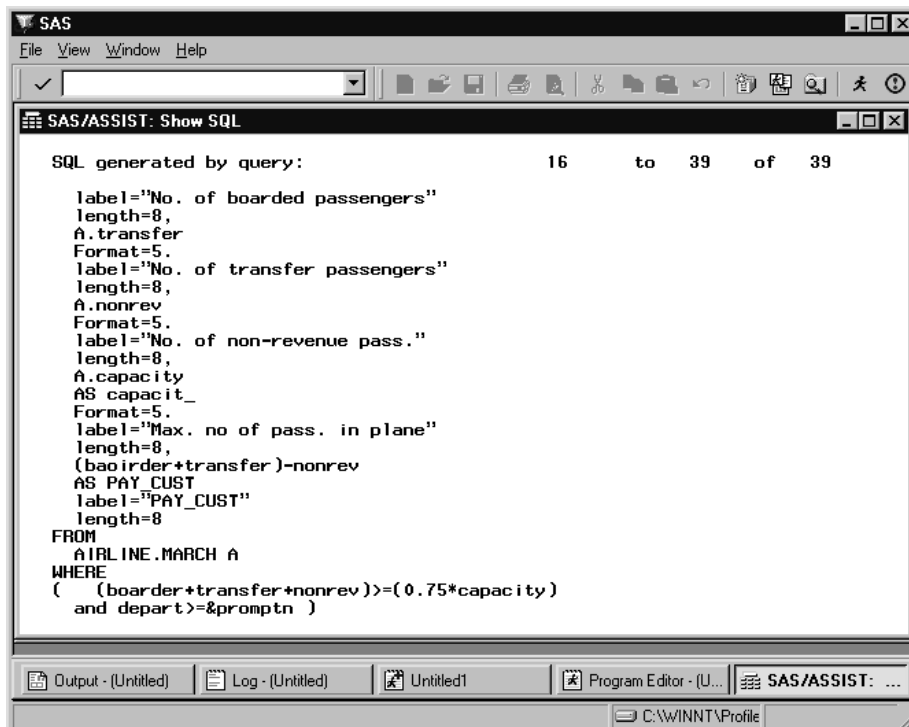
## Viewing the SQL Code, Running the Query, and Refining the Output

### Viewing the SQL Code

As you make changes to a query, the SQL code changes. You can browse the code by selecting **Show SQL** from the **View** menu in the Query window, the Subset Rows window, the Query Expression window, or the Join Manager window.

The SQL code in the following display is based on the query on the AIRLINE.MARCH table constructed in the previous sections. Scroll down to see the changes.

Display 10.17 Show SQL Window



```

SAS
File View Window Help
SAS/ASSIST: Show SQL
SQL generated by query: 16 to 39 of 39
  label="No. of boarded passengers"
  length=8,
  A.transfer
  Format=5.
  label="No. of transfer passengers"
  length=8,
  A.nonrev
  Format=5.
  label="No. of non-revenue pass."
  length=8,
  A.capacity
  AS capacity_
  Format=5.
  label="Max. no of pass. in plane"
  length=8,
  (boarder+transfer)-nonrev
  AS PAY_CUST
  label="PAY_CUST"
  length=8
FROM
  AIRLINE.MARCH A
WHERE
  ( (boarder+transfer+nonrev)>=(0.75*capacity)
  and depart>=&promptn )
  
```

Select **Close** from the **File** menu to return to the Query window.

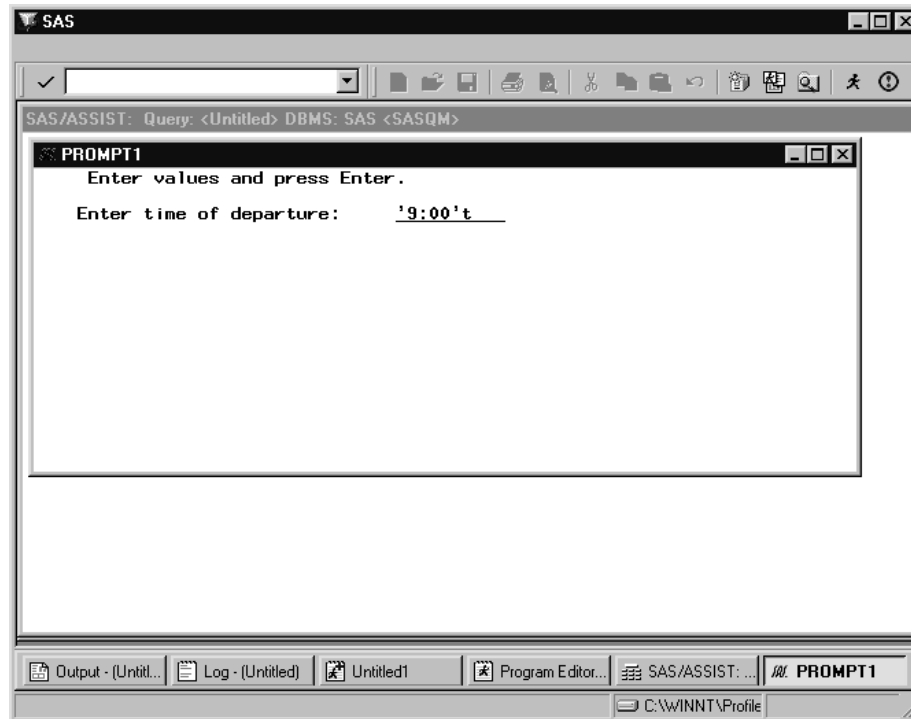
### Running the Query

- 1 To run the query, follow this selection path from the Query window:

►

The Prompt window appears.

Display 10.18 Prompt Window



For this report, specify `'9:00't`. Make sure to use single quotation marks (') around the date value, and include the `t` after the closing quote to convert the time to a SAS time value.

- 2 Press ENTER. The query returns your output.

Display 10.19 Query Results

Flight number	Departure date	Departure (local time)	No. of boarded passengers	No. of transfer passengers	No. of non-revenue pass.	Max. no of pass. in plane	PAYING CUSTOMERS
114	01MAR1994	7:10:00	172	18	6	210	184
182	02MAR1994	8:21:00	116	19	6	178	129
182	03MAR1994	8:21:00	137	17	4	178	150
114	03MAR1994	7:10:00	197	9	2	210	204
290	03MAR1994	6:56:00	114	22	3	180	133
182	04MAR1994	8:21:00	160	18	0	178	178
114	04MAR1994	7:10:00	178	23	5	210	196
290	04MAR1994	6:56:00	131	9	4	180	136
182	05MAR1994	8:21:00	125	24	5	178	144
182	06MAR1994	8:21:00	122	12	7	178	127
182	07MAR1994	8:21:00	155	23	0	178	178
114	07MAR1994	7:10:00	160	23	5	210	178
290	07MAR1994	6:56:00	168	12	0	180	180
182	08MAR1994	8:21:00	164	14	0	178	178
182	09MAR1994	8:21:00	140	26	3	178	163
114	09MAR1994	7:10:00	156	16	6	210	166
182	10MAR1994	8:21:00	146	8	6	178	148
114	10MAR1994	7:10:00	183	10	5	210	188

## Refining the Output

To enhance the appearance of your report, make the following changes:

- 1 From the Report Output window, select **Titles** from the **Edit** menu. Add the title **Information on Early Morning Flights**. Select **Close** from the **File** menu; when prompted, select **Yes** to save the changes to the Titles window.
- 2 Select **Page Layout** from the **Edit** menu. If the **Date** value is **YES**, set it to **NO** to remove the date and time listing in your report. Set the **Center** value to **YES**, which centers the title. Select **OK**.

Select **Close** from the **File** menu to return to the Query window, then re-run the query by selecting **Submit** from the **Run** menu. The following display shows the modified report output.

Display 10.20 Query Results With Refinements

The screenshot shows the SAS Report Output window titled "SAS/ASSIST: Report Output - SQL <Untitled>". The report title is "Information on Early Morning Flights". The table contains the following data:

Flight number	Departure date	Departure (local time)	No. of boarded passengers	No. of transfer passengers	No. of non-revenue pass.	Max. no of pass. in plane	PAYING CUSTOMERS
114	01MAR1994	7:10:00	172	18	6	210	184
182	02MAR1994	8:21:00	116	19	6	178	129
182	03MAR1994	8:21:00	137	17	4	178	150
114	03MAR1994	7:10:00	197	9	2	210	204
290	03MAR1994	6:56:00	114	22	3	180	133
182	04MAR1994	8:21:00	160	18	0	178	178
114	04MAR1994	7:10:00	178	23	5	210	196
290	04MAR1994	6:56:00	131	9	4	180	136
182	05MAR1994	8:21:00	125	24	5	178	144
182	06MAR1994	8:21:00	122	12	7	178	127
182	07MAR1994	8:21:00	155	23	0	178	178
114	07MAR1994	7:10:00	160	23	5	210	178
290	07MAR1994	6:56:00	168	12	0	180	180
182	08MAR1994	8:21:00	164	14	0	178	178
182	09MAR1994	8:21:00	140	26	3	178	163
114	09MAR1994	7:10:00	156	16	6	210	166
182	10MAR1994	8:21:00	146	8	6	178	148
114	10MAR1994	7:10:00	183	10	5	210	188

Select **Close** from the **File** menu to return to the Query window.

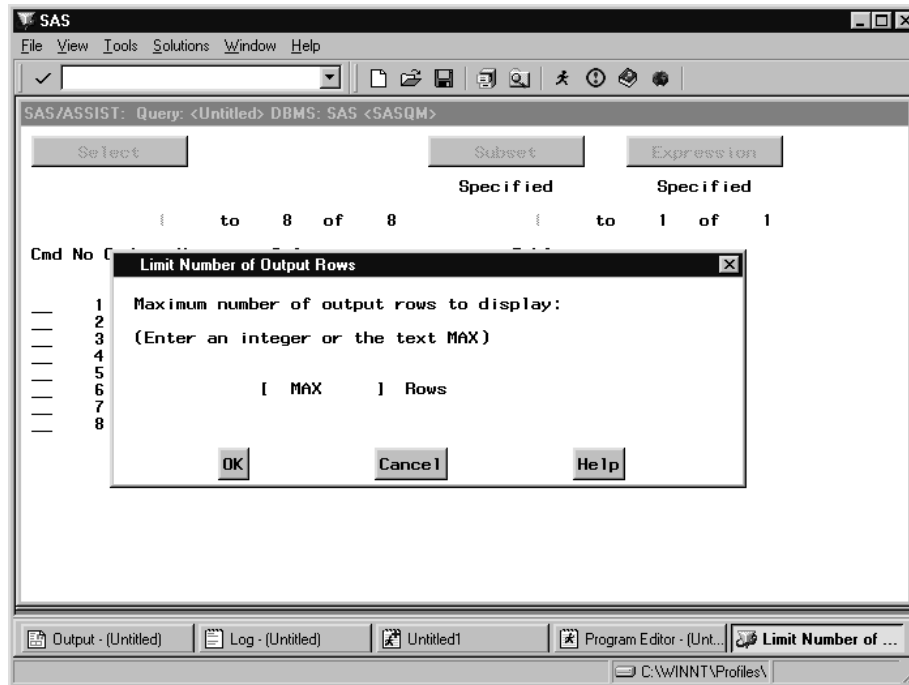
To save your output as a SAS or DB2 table, see "Creating New Tables" on page 185

## Limiting the Report's Output

If you anticipate that your report may have many rows of output and you need to see only some of them, you can limit the number of rows processed. Doing so might also shorten the time the query runs, unless you have used summary (aggregate) functions that must be computed on every row in the query.

To limit the number of rows in your output before you run the query, select **Limit Output Rows** from the **Edit** menu of the Query window. The Limit Number of Output Rows window appears.

Display 10.21 Limit Number of Output Rows Window



The default value is **MAX**. Type the number of rows that you want to display and select **OK**.

When you exit the window, be aware that the specified limit always limits the number of rows displayed in the Report Output window. It also limits SAS or DB2 processing unless the query contains summary functions that force processing of all rows of the table(s) before returning the output to SAS.

If a limit has been placed on the number of output rows in your User Profile, you might be prompted when a query is run. A prompt window appears, verifying that you want to limit the number of rows to be output when this option has been selected in your profile. Type **S** to stop at the specified number of rows, or **C** to ignore the limit, and press ENTER. This feature is controlled by the User Profile options **Limit output rows** and **Limit Rows Pop-up Window** (see Appendix 2, "Customizing SAS/ASSIST Software," on page 241).

---

## Saving a Query

The Query window remains untitled until you save a query by using **Save** or **Save As** from the **File** menu. When you save a query, you give it a name and, optionally, a description. After the query is named, its name appears as part of the Query window title. To save and name your query, select **Save As** from the **File** menu in the Query window. The Save As window opens. Type the name of the query and a short description. Change the default catalog if you want and select **OK**. When you return to the Query window, the name of your query is displayed after the window title.

Saved queries can be used to create reports with the Report Engine. See Chapter 5, "Doing More with the Report Engine," on page 61 for more information.

---

## Resetting the Query Window

To reset the Query window in order to build another query, select **New** from the **File** menu. Doing so clears all existing data and specifications from the Query window. To start another query, choose **Select** to open the Select window.

---

## Performing a COUNT(\*) on a Table

In order to display the number of rows returned by a query (that is, to perform a COUNT(\*) on the table), follow these instructions:

- 1 Use the Select window to select tables and columns for the query.
- 2 Modify the query with subsets, expressions, repeated and deleted columns, and usages as desired.
- 3 Follow this selection path:



Query and Reporting creates an expression called **COUNT(\*)** with a usage of **\***. This expression appears in the Query window.

- 4 Select **Submit** from the **Run** menu. If you have not applied a usage to all columns, a prompt appears asking you if you want to apply a **GROUP** usage to all columns. Select **Yes** or **No**. The output includes a count of the returned rows.
- 5 Select **Close** from the **File** menu to return to the Query window.

To perform a COUNT DISTINCT(\*) on a table:

- 1 Use the Select window to select table for the query.
- 2 From the Query window, select **Expression** and specify **\*** as your expression. Give the expression a name and select **Close** from the **File** menu to return to the Query window.
- 3 Delete the numbers in the **No** column for all columns, except the new expression, to remove them from the output.
- 4 Select **Submit** from the **Run** menu. The output shows the number of distinct nonmissing rows for the entire table.
- 5 Select **Close** from the **File** menu to return to the Query window.

To perform a COUNT(\*) within a group:

- 1 Use the Select window to select tables and columns for the query.
- 2 From the Query window, select **Expression** and specify **\*** as your expression. Give the expression a name and select **Close** from the **File** menu to return to the Query window.
- 3 Specify the **GROUP** usage for the column of interest. Delete the numbers in the **No** column for all columns except the new expression and the column of interest.
- 4 Select **Submit** from the **Run** menu to run your query. The output shows the number of distinct values (groups) in the column of interest.
- 5 Select **Close** from the **File** menu to return to the Query window.

See Chapter 5, “Doing More with the Report Engine,” on page 61 for more information on the **Usage** field.

---

## Creating New Tables

You can create a table from a query using the SQL Editor window or the Query window.

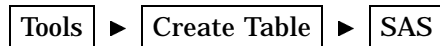
In all operating environments, you can create SAS tables and views. In OS/390, you can also create DB2 tables. In some cases you must be granted certain access or privileges to create new tables. If you do not have these privileges, contact your SAS Support Consultant or database administrator.

To create a table from your query results, run your query, then return to the Query window by selecting **Close** from the **File** menu. Follow the instructions in one of the following two sections, depending on which type of table you want to create.

## Creating a New SAS Table or View

Follow these steps to create a new SAS table or view from your query results:

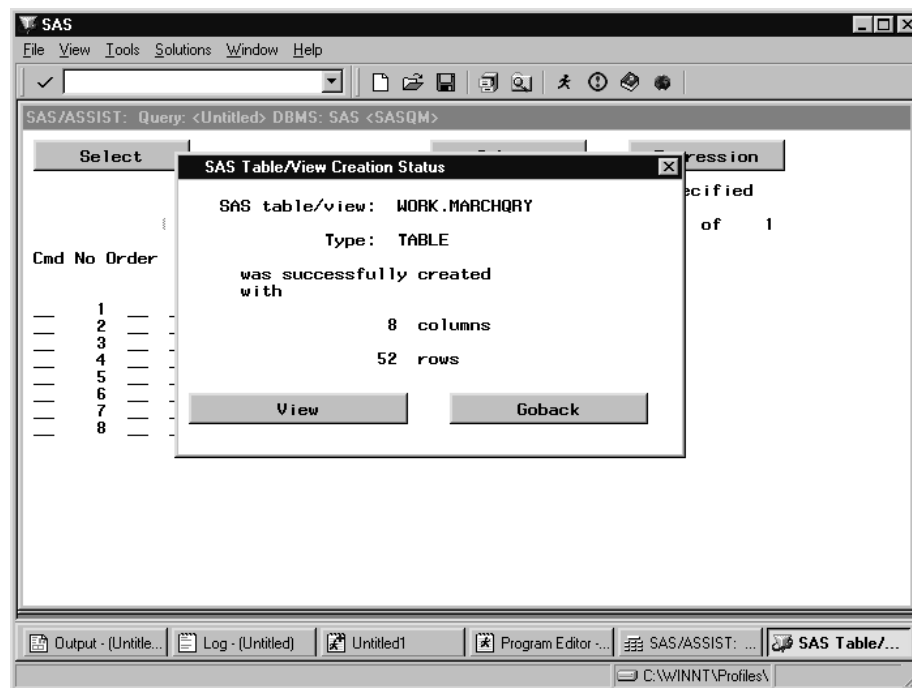
- 1 Follow this selection path from the Query window:



The Create SAS Table/View window appears.

- 2 In the **Name** field, type the name of the table or view. Use a two-level name in the form *libref.table-name* if you want to store the table or view permanently.
- 3 Select **Table** or **View**. **Table** creates a SAS data file; **view** creates a PROC SQL view.
- 4 When you select **OK**, the query is processed and a window verifies the SAS table's creation, as shown in the following display.

Display 10.22 SAS Table/View Creation Status Window



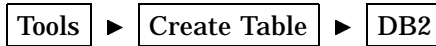
- 5 Select **view** to display the new SAS table, or select **Goback** to return to the Query window.

---

## Creating a New DB2 Table (OS/390 Only)

In OS/390, to save the results of your query to a DB2 table, do the following:

- 1 Follow this selection path from the Query window:



The Create DB2 Table window appears.

- 2 In the **Name of DB2 table** field, type the name of the table.
- 3 Select **OK** SAS/ASSIST software verifies whether or not the table exists; if it does, you are prompted to append or replace it. The new table is created, and its authorization ID is automatically set to your userid.

*Note:* If you plan to use DB2 data in several SAS programs, you usually get better SAS performance if you save the DB2 data to a SAS table or SAS view. △

---

## Joining Tables

You often need data that are stored in separate tables. For example, you may want to produce a report that gets information about flight delays from one table and boarding capacity from another. Information from separate tables can be combined by matching values in columns that relate the tables. This is called *joining* tables. You can join a maximum of 16 tables in one query. Note that some limitations apply when you join DB2 tables. For more information, see “Additional Considerations for Joining DB2 Tables” on page 201.

---

## Introducing Table Joins

A join combines two or more tables side by side. If you do not specify how to join the tables, you get a *Cartesian product*. This means that SQL combines each row from the first table with every row from the second table, the second row of the first table is then matched with every row of the second table, and so on. Therefore, two tables with 100 rows each produce  $100 * 100$ , or 10,000, rows. You seldom want all this information, and it often causes you to exceed the available disk space.

To avoid a Cartesian product, you must specify how the tables should be combined. Typically, you want to pair rows based on matching values in one or more key columns of each table. The following example uses the tables in the following figure to illustrate a join.

Figure 10.2 Joining Two Tables

Table A		Table B	
ID	NAME	ID	SALARY
1	Elisabeth	1	2000
2	Jens	2	900
3	Michael	4	3000
4	Marie	5	500
		6	750

Pair the SALARY information in Table B with the NAME of the person in Table A by matching the values in the ID columns. Therefore, your join criterion is **A.ID=B.ID**. In SQL, the code is:

```
select name, salary
  from a, b
 where a.id = b.id;
```

Only rows with matching values of ID are output. This is known as an *inner join*. The following figure shows the results of this query.

Figure 10.3 Output from Joined Tables

OUTPUT	
NAME	SALARY
Elisabeth	2000
Jens	900
Marie	3000

You can also join tables that have missing values. For example, ID 3 is missing from Table B. A *left join* outputs all the names in Table A (the left table) and matches them to the available salary information in Table B. If a row in Table B is missing, missing values are supplied in the joined output. In this case, **Michael** from Table A is output with a missing value for SALARY next to his name. The following figure shows the results of this query.



**Figure 10.4** Output from Left Joined Tables

## OUTPUT

NAME	SALARY
Elisabeth	2000
Jens	900
Michael	.
Marie	3000

A *right join* works similarly. All the rows from Table B (the right table) are output, and any missing rows from table A are output as missing values (or blanks, if the values are character data). The following figure shows the results of this query.

**Figure 10.5** Output from Right Joined Tables

## OUTPUT

NAME	SALARY
Elisabeth	2000
Jens	900
Marie	3000
	500
	750

If you join two tables that both have missing rows, you can combine the effect of a left and a right join by using a *full join*. The following figure shows the results of this query.

**Figure 10.6** Output from Fully Joined Tables

## OUTPUT

NAME	SALARY
Elisabeth	2000
Jens	900
Michael	.
Marie	3000
	500
	750

Inner joins can be performed on up to 16 tables at a time, whereas outer joins (left, right, full) work on only two tables at a time.

## Joining Tables Manually

If tables selected for joining have a relation defined between them, Query and Reporting joins them automatically; see “Joining Tables Automatically” on page 198 for more information. If the tables do not have a relation defined between them, or if they have more than one such relation defined, you have to join the tables manually. The following sections show examples of how to join tables manually in both cases, as well as an example of performing outer joins, which must be done manually.

### Joining Tables Without Defined Relations

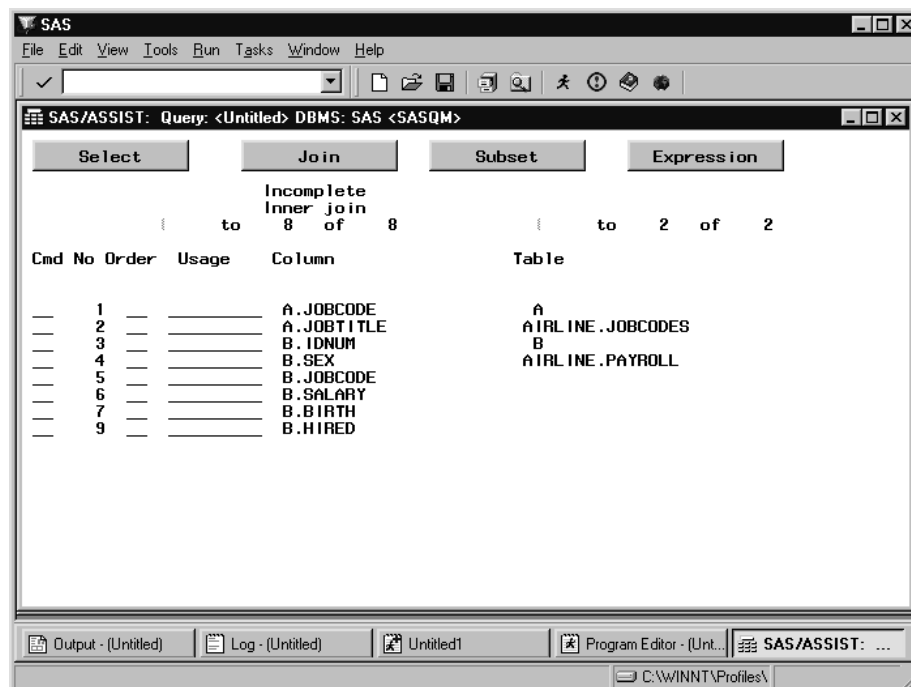
Follow these instructions to join two tables manually:

- 1 Choose **Select** to open the Select window.
- 2 Select the AIRLINE.JOBCODES and AIRLINE.PAYROLL tables.
- 3 Select **Close** from the **File** menu to return to the Query window.

The **Incomplete Inner join** message appears under **Join**. The warning **Not all tables are joined** appears because the tables do not contain a relation defined on them. Therefore, Query and Reporting has not automatically joined the tables. The following display shows the Query window with the two tables to be joined.

*Note:* To clear previously-defined subsets or expressions, select **subset** or **Expression**. Select **Clear** from the **Edit** menu, then select **Close** from the **File** menu.  $\Delta$

**Display 10.23** Query Window with Tables for Manual Join



#### **CAUTION:**

**Interrupting Processing** If you run a query with an incomplete join, you create a Cartesian product, as described in “Introducing Table Joins” on page 187.

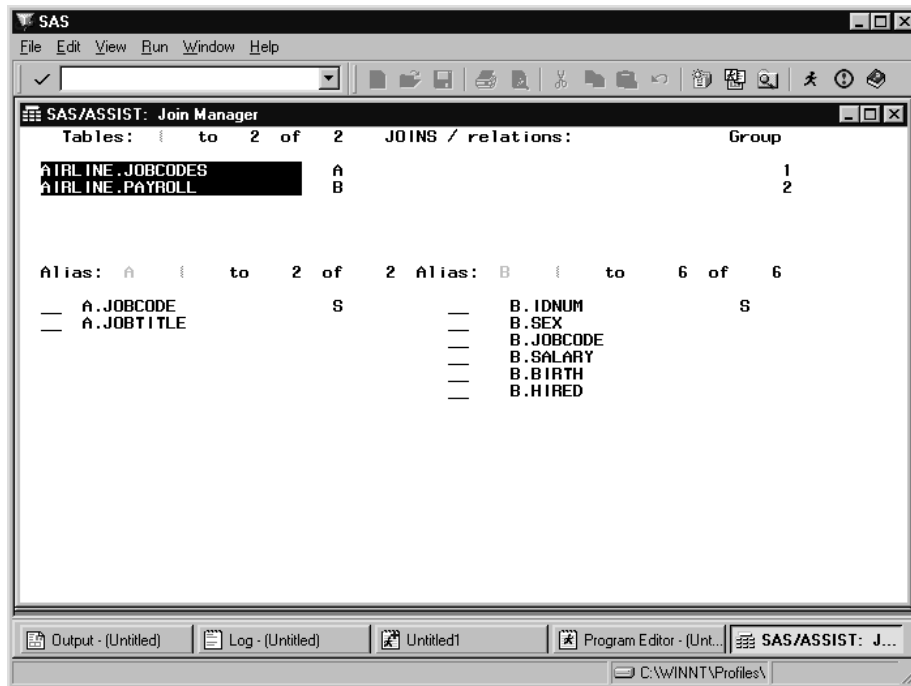
Depending on the setup at your site, either the query is rejected or a WARNING window appears, verifying whether or not you really want to run this query. If you select **Yes** to run the query, your query may take a long time to process, depending on the size of the tables.  $\Delta$

- 4 Select **Join** to open the Join Manager window. The Join Manager window shows the selected tables and all of their columns. Before the tables are joined, their names are highlighted.

*Note:* If you are joining more than five tables, not all of the table names appear in the Join window; overwrite the first number in the **Tables** field to see more tables.  $\Delta$

The following display shows the Join Manager window with AIRLINE.JOBCODES and AIRLINE.PAYROLL before they are joined. The **Group** field indicates whether groups of tables are joined. **1,2** indicates that this group is not joined.

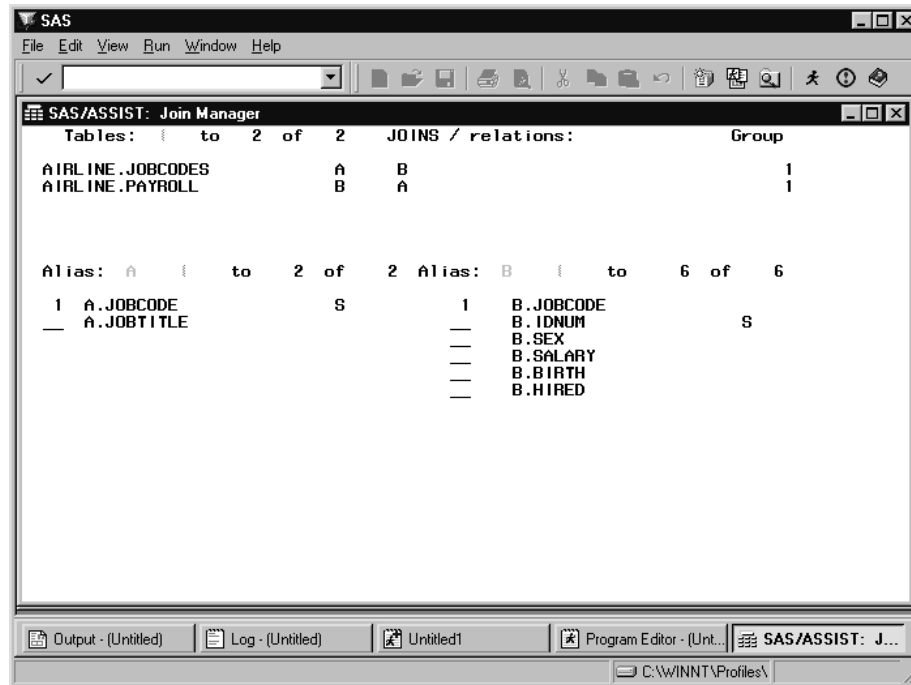
**Display 10.24** Join Manager Window With Two Tables To Be Joined



- 5 Type **1** next to **A.JOBCODE** and **1** next to **B.JOBCODE** and press ENTER. Note that the column names for the two tables are the same in this example. The column names do not have to be the same, but their data types and values must match. You can select **Sort by** from the **Edit** menu to reorder columns in the tables by their names, join number, index, or matching names.

After you press ENTER and the tables are joined, the highlighting of the table names is removed, the **Group** field changes to **1,1** (indicating one group), and a message appears saying **All tables are joined**. The following display shows the Join Manager window after the tables have been joined.

Display 10.25 Join Manager Window With Two Joined Tables



- 6 Select **Close** from the **File** menu to return to the Query window. Note that the message under the **Join** button says **Complete inner join**.

### Joining Three Tables

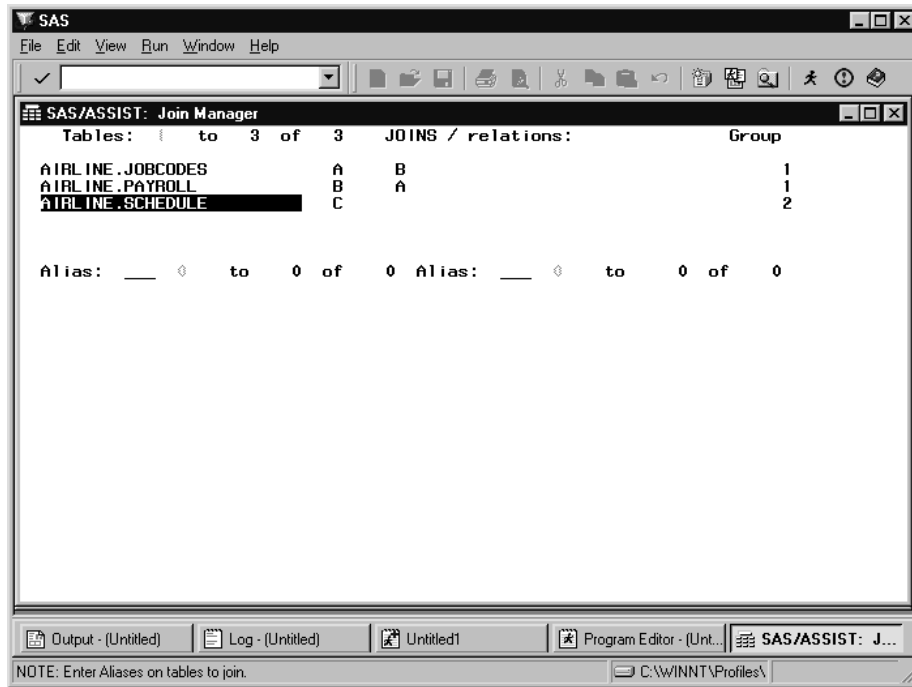
To add a third table to the joined AIRLINE.JOBCODES and AIRLINE.PAYROLL tables, follow these instructions:

- 1 Return to the Select window by choosing **Select** from the Query window.
- 2 Select one more table, AIRLINE.SCHEDULE
- 3 Select **Close** from the **File** menu to return to the Query window. The message **Incomplete Inner join** appears again.
- 4 Select **Join** to open the Join Manager window.

The table names are displayed at the top of the Join Manager window. AIRLINE.SCHEDULE is highlighted because it is not yet joined with another table. AIRLINE.SCHEDULE has no table alias under the **JOINS/relations** field for the same reason. This table is also listed as a separate group in the **Group** field.

The following display shows the Join Manager window before the third table AIRLINE.SCHEDULE is joined with AIRLINE.JOBCODES and AIRLINE.PAYROLL.

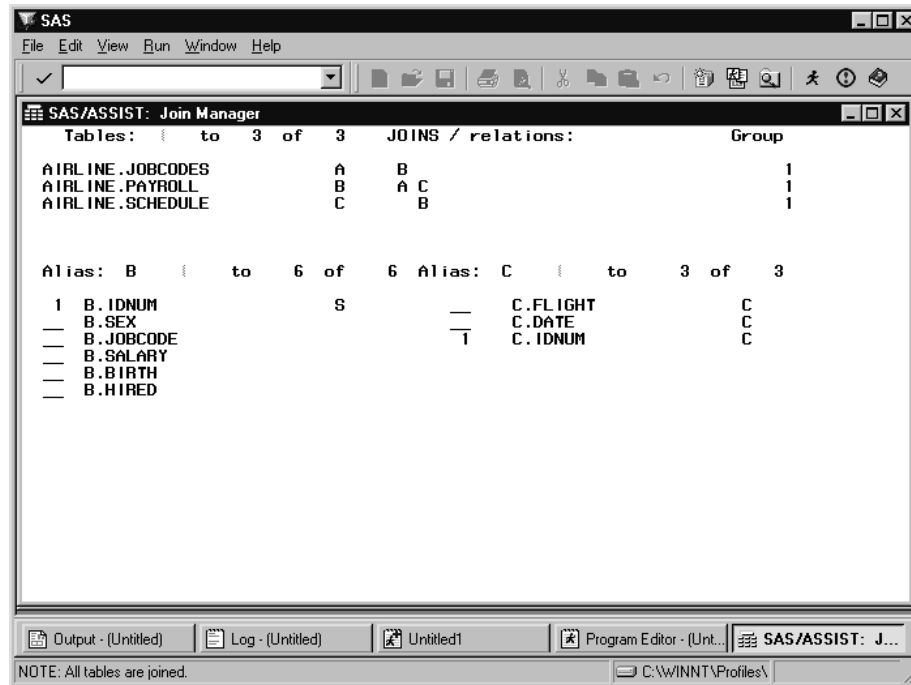
Display 10.26 Join Manager Window With Third Table Before Joining



- Because the columns from only two tables can be listed at the same time in this window, the **Alias** fields are blank and no columns are listed. AIRLINE.PAYROLL and AIRLINE.SCHEDULE have a common column, IDNUM. Type the alias **B** in the first **Alias** field and the alias **C** in the second **Alias** field. When you press ENTER, the tables' columns are listed. Type **1** next to **B.IDNUM** and **C.IDNUM**, respectively, and press ENTER.

When the three tables are joined, the highlighting is removed, the **Group** field changes to **1,1,1** (indicating one group), and a message appears saying **All tables are joined**, as shown in the following display.

Display 10.27 Join Manager Table With Three Joined Tables



- 6 Select **Close** from the **File** menu to return to the Query window. Save or run your query. After you run your query, you can save your output to a table. See “Creating a New SAS Table or View” on page 186 for more information.

## Joining Tables with Outer Joins

When you join tables, Query and Reporting uses an inner join by default. However, you can choose another type of join if you are joining only two tables. In DB2, outer joins are allowed on only two tables. Note that there may be other restrictions under DB2; for further information, see “Additional Considerations for Joining DB2 Tables” on page 201. Outer joins must be performed manually.

- 1 If desired, save the previous query using **Save** from the **File** menu.
- 2 Reset the Query window by selecting **New** from the **File** menu
- 3 Open the Select window and choose the AIRLINE.SCHEDULE and AIRLINE.SUPERV tables. These tables do not have a relation defined on them.
- 4 Return to the Query window, and note that the **Incomplete Inner Join** message appears under **Join**.
- 5 To change this query to a right join, select **Join Type** from the **Edit** menu.
- 6 Select **Right join** from the list of join types and **Goback**. When you return to the Query window, the **Incomplete Right join** message appears under the **Join** button.
- 7 Select **Join** to open the Join Manager window.
- 8 Type **1** in the **A.IDNUM** column and **1** in the **B.SUPERV** column and press ENTER. (These steps are described earlier in “Joining Tables Without Defined Relations” on page 190.)
- 9 Return to the Query window.
- 10 If desired, reorder the columns of the query and then select **Show SQL** from the **view** menu. The following display shows the second display of the SQL for the right join query.

**Display 10.28** Show SQL Window for Right Join

```

SAS
File View Window Help
SAS/ASSIST: Show SQL
SQL generated by query:          5      to 29  of 29

length=3,
A.date
Format=DATE7.
label="Date"
length=8,
A.idnum
Format=$4.
label="Id of crew member"
length=4,
B.supid
Format=$4.
label="Id of supervisor"
length=4,
B.state
Format=$2.
label="State"
length=2,
B.jobcat
Format=$3.
label="Job Category"
length=3
FROM
    AIRLINE.SCHEDULE A
RIGHT JOIN AIRLINE.SUPERV B
ON B.supid=A.idnum

```

Output - (Untitled) | Log - (Untitled) | Untitled1 | Program Editor - (U... | SAS/ASSIST: ...

C:\WINNT\Profile

Exit the Show SQL window.

- 11 Select **Submit** from the **Run** menu to run the query. The following display shows the results. This query creates a list of supervisors who have been assigned to flights as well as supervisors who have not been assigned to flights.

Display 10.29 Results of Right Join

Flight number	Date	Id of crew member	Id of supervisor	State	Job Category
114	05MAR1994	1106	1106	CT	PT
132	06MAR1994	1106	1106	CT	PT
132	01MAR1994	1118	1118	NY	PT
114	04MAR1994	1118	1118	NY	PT
			1126	NY	TA
114	04MAR1994	1352	1352	NY	NA
			1385	CT	ME
			1401	NJ	TA
			1405	NJ	SC
132	01MAR1994	1417	1417	NJ	NA
132	05MAR1994	1417	1417	NJ	NA
			1420	NJ	ME
114	06MAR1994	1431	1431	CT	FA
132	02MAR1994	1433	1433	NJ	FA
132	06MAR1994	1442	1442	NJ	PT
			1564	NY	SC
			1639	CT	TA
			1677	CT	BC
			1834	NY	BC
			1882	NY	ME

## Defining Relations on Tables

When you query SAS tables, *relations* provide a way to identify rows and to relate (or join) rows in one table to rows in another table. You define a relation on one or more columns in a table. Relations are based on the database concept of primary and foreign keys; see *SAS/ACCESS Software for Relational Databases: Reference* for more information.

You define a relation on tables based on one or more columns that tables contain. For example, the SAS AIRLINE.PAYROLL table has a column named IDNUM that has the same attributes as the IDNUM column in AIRLINE.STAFF. Therefore, you would define and name a relation for IDNUM on the two tables.

Defining a relation between or among tables enables you to build intelligence into your query creation process. When one relation is defined on tables that you join in a query, Query and Reporting automatically generates a WHERE or ON clause for that query. This saves you from defining a WHERE predicate for each join query. (See “Joining Tables Automatically” on page 198 for more information.) You can, of course, still select **subset** from the Query window to add other predicates to your WHERE clause; see “Retrieving a Subset of Data” on page 176 for more information.

If no relations are defined or if more than one relation is defined for the tables in the join, you must join the tables manually, as described in “Joining Tables Manually” on page 190 and “Joining Tables with Multiple Relations” on page 199.

The examples in this section are used to demonstrate joining tables automatically and joining tables with multiple relations later in this chapter.

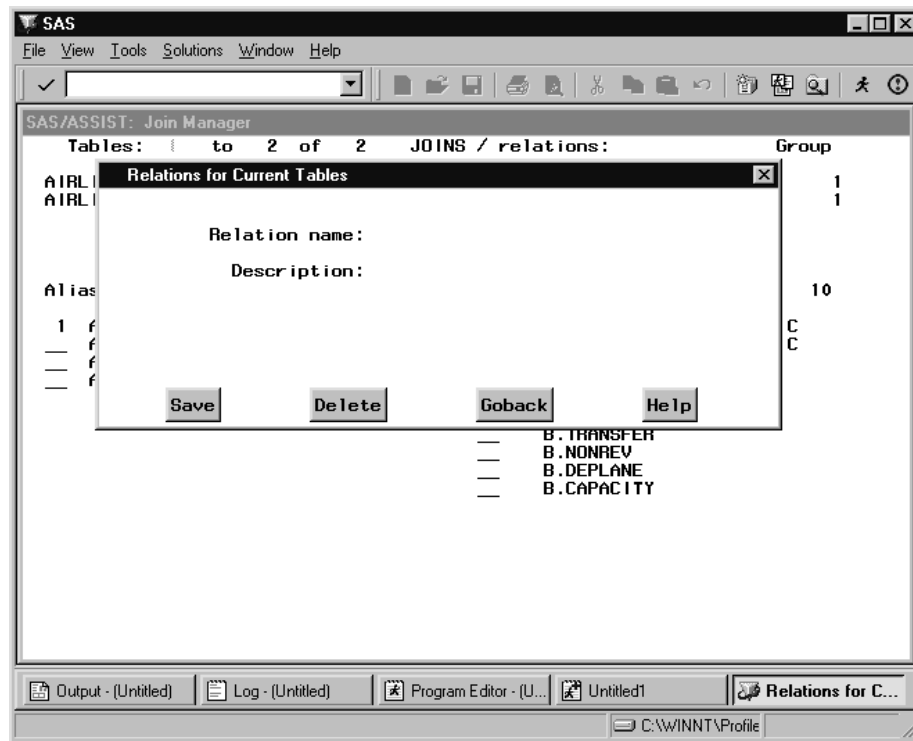
- 1 If desired, save the previous query by selecting **Save** from the **File** menu in the Query window.
- 2 Reset the Query window by selecting **New** from the **File** menu.



- 3 Go to the Select window and select the AIRLINE.MARCH and AIRLINE.FLINFO tables. Return to the Query window.
- 4 Select **Join** to open the Join Manager window.
- 5 Type a **1** next to the **A.FLIGHT** and **B.FLIGHT** columns and press ENTER. A message indicates that the tables are joined.
- 6 Follow this selection path to open the Relations for Current Tables window:

View ► Relations ► Current Tables

Display 10.30 Relations for Current Tables Window



- 7 In the **Relation name** field, type **MARFLT**. A relation name can be up to eight characters long.
- 8 In the **Description** field, type **March - Flight Info Relation**. A description can be up to 32 characters long.
- 9 Select **Save** to save the relation, then **Goback** to return to the Join Manager window.
- 10 Return to the Query window.

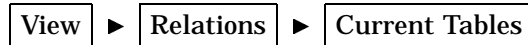
Repeat steps 2 through 9 above to define a relation on the AIRLINE.DEST and AIRLINE.FLINFO tables. Define your relation on the A.DEST and B.DEST columns. Name this relation **FLDEST** and give it a description of **Destination Name Join**. Then, repeat steps 5 through 9 to define a second relation on the AIRLINE.DEST and AIRLINE.FLINFO tables. By typing **2** (instead of **1**) next to the columns, define this relation on the A.DEST and B.ORIG columns. Name this relation **FLORIG** and give it a description of **Origin Name Join**.

## Modifying and Deleting Relations

In order to add, edit, or delete a relation, you must have the correct permissions to do so. Permission to modify relations is determined by the Query Manager in effect. Contact your SAS Support Consultant for details.

To modify a relation, follow these instructions:

- 1 Select the tables associated with that relation in the Select window
- 2 Select **Join** to open the Join Manager window.
- 3 If you want only to change the name or description of the relation, proceed to the next step. Otherwise, modify the join as desired.
- 4 Follow this selection path:

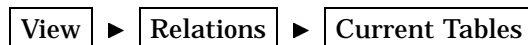


The Relations for Current Tables window appears, as shown in the previous display.

- 5 Type the name of the relation in the **Relation name** field. You can also enter a ? in the **Relation name** field and select the relation from the resulting list.
- 6 Select **Save**. When prompted, select **Yes** to overwrite the existing relation.

To delete a relation, follow these instructions:

- 1 Select the tables associated with that relation in the Select window.
- 2 Select **Join** to open the Join Manager window.
- 3 Follow this selection path:



The Relations for Current Tables window appears, as shown in the previous display.

- 4 Type the name of the relation in the **Relation name** field. You can also enter a ? in the **Relation name** field and select the relation from the resulting list.
- 5 Select **Delete**, then **Goback** to return to the Join Manager window.

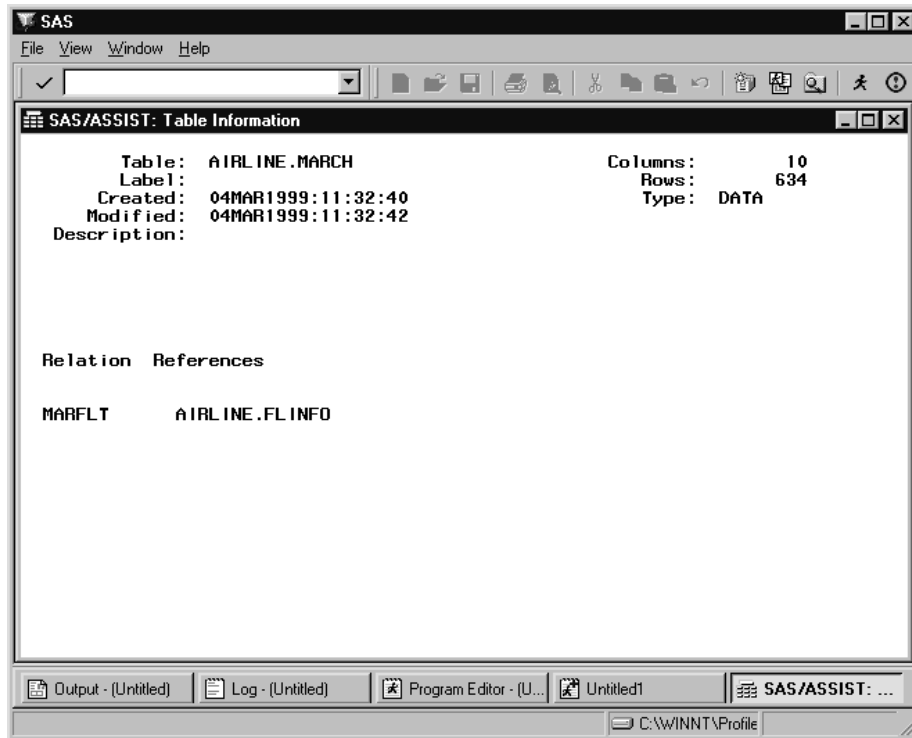
## Joining Tables Automatically

The Query window can join tables automatically if a relation is defined on the tables. In addition, as already discussed, relations can be defined either by a DBA or by a user. (Note that for DB2, if any referential constraints are defined, these constraints are used by the Query Manager to join tables automatically.)

Query and Reporting automatically joins tables that have exactly one relation defined on them. The default action is to create an inner join. (See “Joining Tables Manually” on page 190 for information on joining tables without a defined relation and for performing outer joins.)

In this section you join the AIRLINE.MARCH and AIRLINE.FLINFO tables on which you defined a relation in the previous section.

- 1 Reset the Query window by selecting **New** from the **File** window.
- 2 Open the Select window and select the AIRLINE.MARCH and AIRLINE.FLINFO tables. Return to the Query window.
- 3 By convention, relations are a property of the second of the two tables joined, in this case, AIRLINE.MARCH. To see this property, type **I** next to the table name in the Select window. The Table Information window appears, as shown in the following display.

**Display 10.31** Table Information Window

- 4 Select **Close** from the **File** menu twice to return to the Query window. The message **All tables are joined** appears in the Query window, and the message **Complete Inner join** appears under the **Join** button.

At this point, you can customize your query, run it, and save it as described in “Saving a Query” on page 184. Alternatively, you can save the query’s output to a table, as described in “Creating New Tables” on page 185. If you intend to access the joined data frequently, it is often more efficient to save the output to a SAS table and then specify the new SAS table in your SAS code.

---

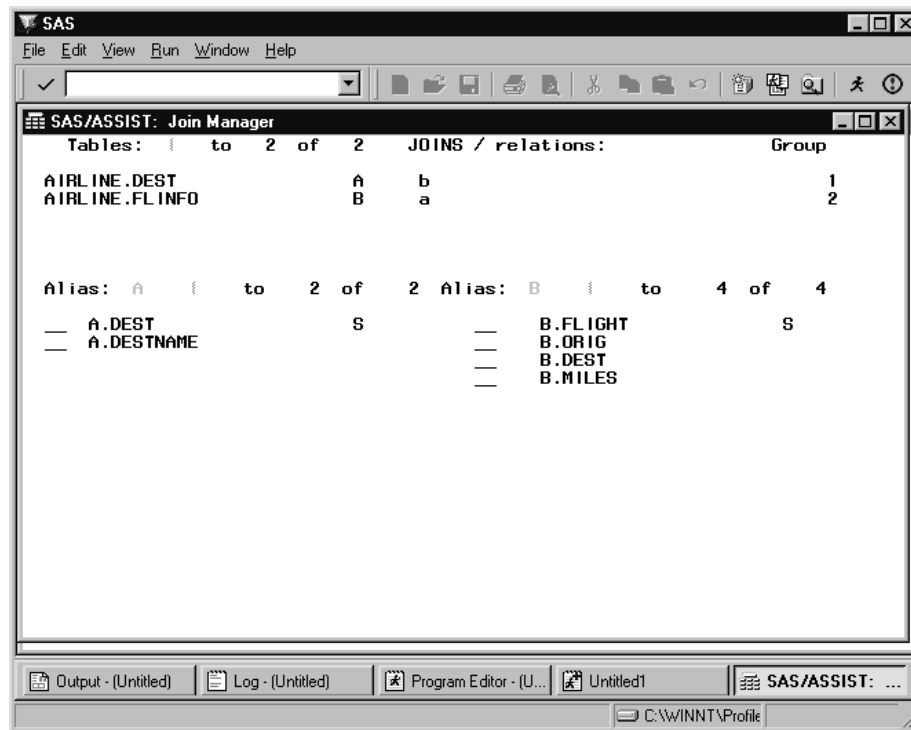
## Joining Tables with Multiple Relations

Tables with multiple relations must be joined manually because you have to decide which columns to use to join the tables. This example uses the relations you defined on the AIRLINE.DEST and AIRLINE.FLINFO tables in “Defining Relations on Tables” on page 196.

- 1 Reset the Query window by selecting **New** from the **File** menu.
- 2 Open the Select window and select AIRLINE.DEST and AIRLINE.FLINFO.
- 3 Open the Table Information window for the AIRLINE.FLINFO table to see the two relations defined on it.
- 4 Exit the Table information window and the Select window to return to the Query window. The **Not all tables are joined** message is displayed because you have not yet selected a relation.
- 5 Select **Join** to open the Join Manager window. The Join Manager window appears much as it does in Display 10.24 on page 191. The table aliases under the **JOINS/relations** field are in lowercase letters. The aliases are lowercase to indicate that

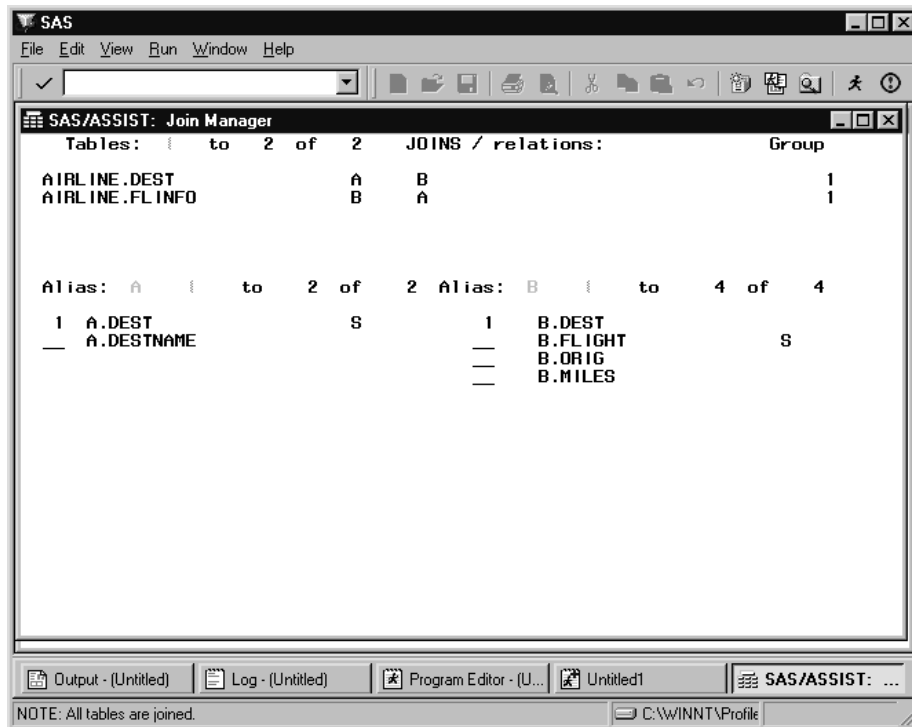
a relation exists between the tables but that it is not used yet. The following display shows the tables before they are joined.

**Display 10.32** Join Manager Window With Multiple Relations



- Choose **Select Relation** from the **Edit** menu, or select one of the lowercase aliases under the **JOINS/rerelations** field to select a relation with which to join. The relations for your tables are listed. Select **Destination Name Join**. The tables are joined and the lowercase aliases in **JOINS/rerelations** change to uppercase. Notice in the following display that a 1 automatically appeared in the **A.DEST** and **B.DEST** fields and the **Group** field changes to 1,1.

Display 10.33 Join Manager Window With Multiple Relations After Joining



Return to the Query window to customize, save, or run your query.

## Additional Considerations for Joining DB2 Tables

When you use the Query window, you can perform inner joins or outer joins (left, right, and full) on DB2 tables. If one primary key/foreign key relation is defined between the tables, Query and Reporting joins the related tables automatically. It generates a WHERE or ON clause that is based on the related column(s) in the tables, such as `where payroll.idnum=staff.idnum`.

If no primary key/foreign key relation is defined or if more than one relation exists between the contributing tables, you must specify manually which columns are used to join the tables (see “Joining Tables Manually” on page 190).

Versions 4 and later of DB2 includes outer join operators. Therefore, when you use the Query window and specify a left, right, or full join, Query and Reporting generates the WHERE or ON clause automatically.

DB2 versions prior to Version 4 lack outer join operators. You can still specify an outer join type in Query and Reporting. If relations are defined on the tables, a WHERE or ON clause is automatically generated. If no relations are defined, you have to join the tables manually. When you review your SQL code (by selecting **Show SQL** from the **View** menu), you see that DB2 SQL has used set operators (such as UNION) to produce an outer join result, instead of using outer join operators.

Because earlier versions of DB2 SQL do not have outer join operators, certain limitations apply to outer joins in Query and Reporting:

- Because you generate a WHERE or ON clause when you join the tables, you cannot specify *additional* WHERE predicates or expressions in the SELECT clause. For example, if you select **Subset** in the Query window, enter a WHERE predicate, and exit the window, the Messages window appears with **Subset is**

**disabled due to outer join.** Then **Disabled** appears under **Subset**. It works the same way with **Expression**.

- Arithmetic expressions (such as those in a SELECT clause) are ignored.
- Results from outer joins cannot be stored in a new DB2 table. (However, you can save the results to a SAS table.)

The following table shows the default values in outer joins.

**Table 10.7** Default Values for Data Types in Outer Joins

Data Type	Default Value
Character data	blank ( ' )
Numeric Data	0
Date	January 1, 0001
Time	Midnight
Timestamp	January 1, 0001, midnight

## Referential Integrity

Referential integrity is implemented in the DB2 database management systems by specifying conditions where all references from one database column to another column must be validated.

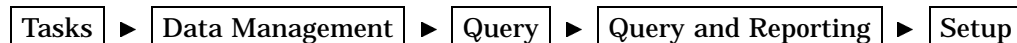
These referential constraints are established by defining *primary keys* and *foreign keys* for the columns in the relationship. A primary key defines a unique identifier for the rows of a table. A foreign key is a column where each value contains a value in the primary key of another defined table. After the relationship between a primary key and a foreign key has been established, DB2 enforces the referential integrity for you according to both the rules that DB2 has established and the referential constraints that you have specified.

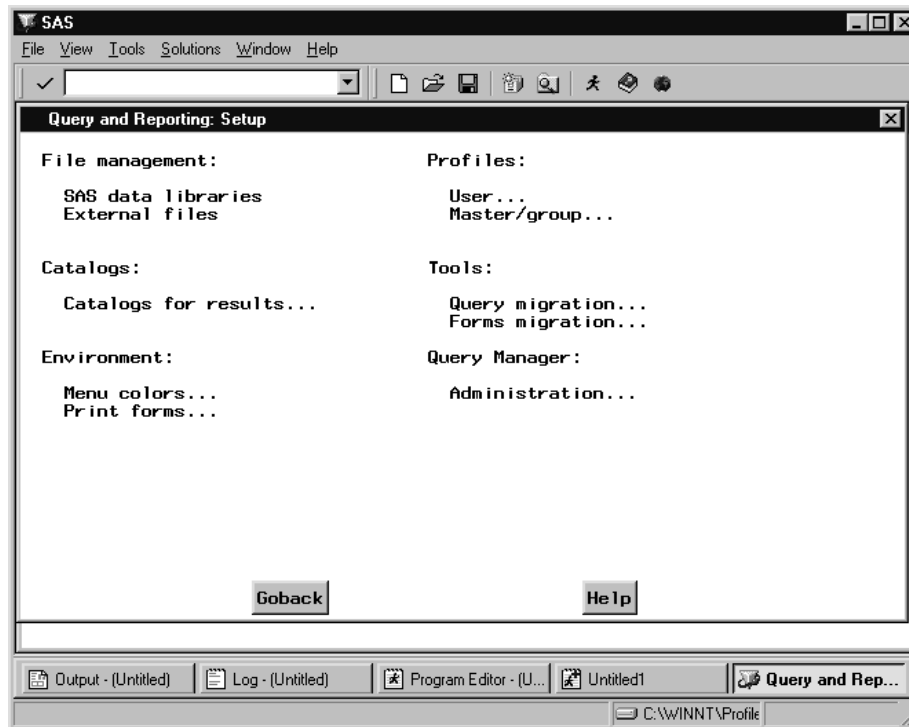
If referential constraints have been established, the Select window uses them to generate automatic joins between tables. In the Query, Select, or Report Engine windows, you can enter the **I** command next to a table or column name to open the Select Table information or Column information window. These windows display information on referential constraints for your convenience.

---

## Query and Reporting Setup Options

This section describes the options available in the Query and Reporting: Setup window. To access this window, follow this selection path:



**Display 10.34** Query and Reporting Setup Window

The items under **File management**, **Profiles**, **Catalogs**, and **Environment** have the same functions in the Query and Reporting: Setup window that they do in the main SAS/ASSIST Setup window. For details on these items, see Chapter 12, “Doing More with Setup,” on page 219.

---

## Query migration

The **Query migration** item opens the Query Migration window, which enables you to import QMF queries to Query and Reporting. For more information, see Appendix 4, “Migrating from QMF to SAS/ASSIST Software,” on page 255.

---

## Forms migration

The **Forms migration** item enables you to migrate QMF forms to the SAS System under the OS/390 operating environment. You must have the forms migration utility installed to use this item. For more information, contact SAS Institute Technical Support.

---

## Administration

The **Administration** item opens the Query Manager Administration window, which enables you to define and edit static and dynamic Query Managers.





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.