CHAPTER 2

Examples

The First Steps

To practice with the examples in this chapter, you will need to use the sample data library provided with the SQL Query Window.

Submit the following statement in the PROGRAM EDITOR window to assign the SAMPLE libname to the sample library:

```
libname sample 'sample library';
```

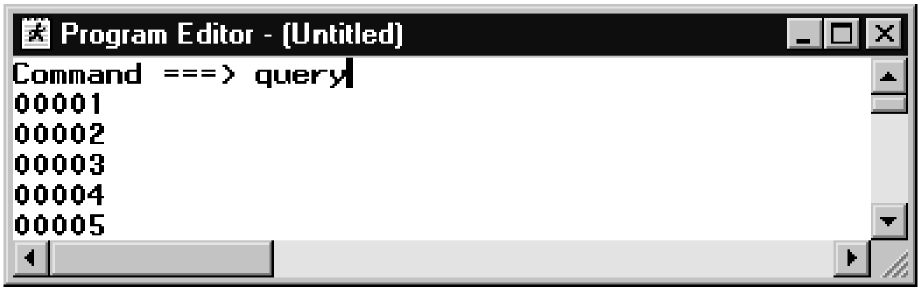
Consult your SAS Administrator for the location of the sample library. Some of the examples require that you save files to the sample library. If you do not have write-access to the sample library, you can save the files to another library of your choice.

Invoking the Query Window

See "Invoking the SQL Query Window" on page 2 for instructions on the different ways in which you can invoke the SQL Query Window. In this example, invoke the SQL Query Window by typing

```
query
```

in the Program Editor window.



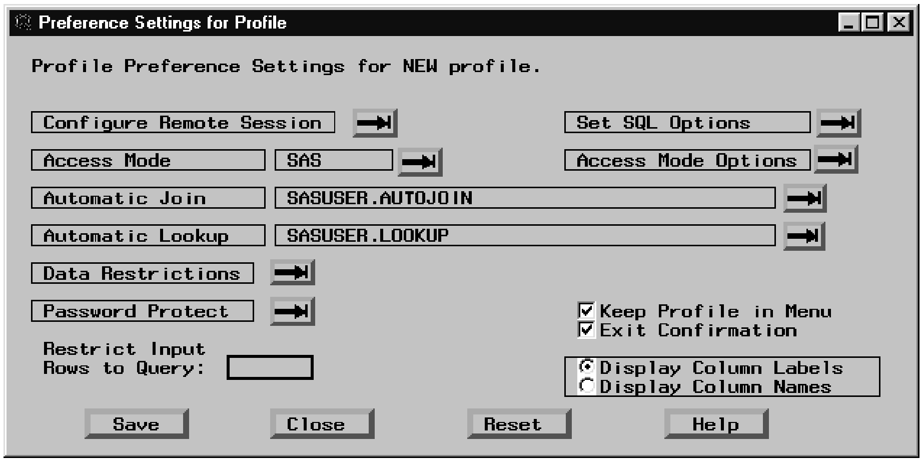
```
Command ===> query
00001
00002
00003
00004
00005
```

The SQL QUERY TABLES window is displayed. By default, the SASUSER libref is selected and the tables from that libref appear in the list of Available Tables.

Changing Your Profile

Set your SQL Query Window profile to include the tables in the sample data library. From the PMENU, select

Profile ▶ Set Preferences...



Profile Preference Settings for NEW profile.

Access Mode: SAS

Automatic Join: SASUSER.AUTOJOIN

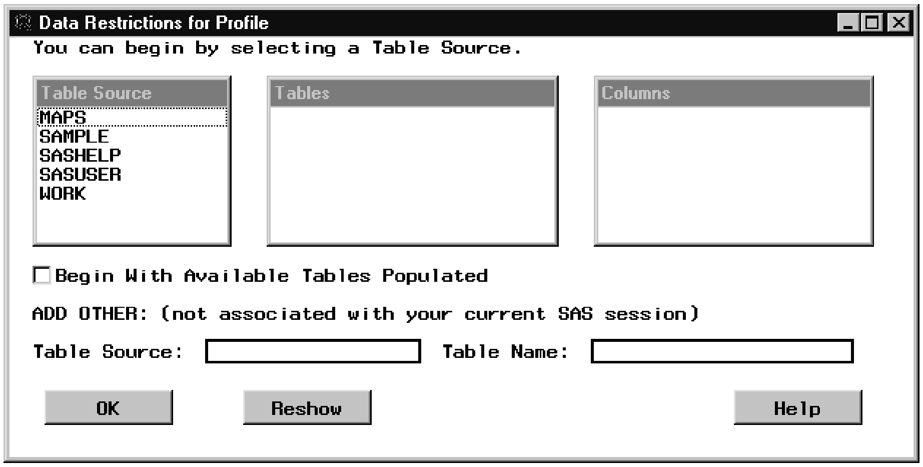
Automatic Lookup: SASUSER.LOOKUP

Restrict Input Rows to Query:

☑ Keep Profile in Menu
☑ Exit Confirmation

● Display Column Labels
○ Display Column Names

Select the ⇒ next to **Data Restrictions** to display the Data Restrictions for Profile window.



Data Restrictions for Profile

You can begin by selecting a Table Source.

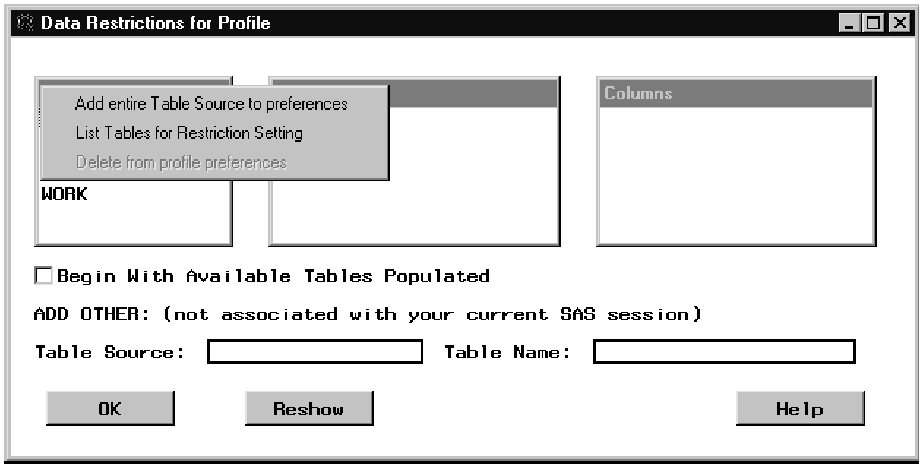
Table Source: MAPS, SAMPLE, SASHELP, SASUSER, WORK

□ Begin With Available Tables Populated

ADD OTHER: (not associated with your current SAS session)

Table Source: Table Name:

Select **SAMPLE** from the list of Table Sources. Select **Add entire Table Source to preferences** from the pop-up menu that appears.



Data Restrictions for Profile

□ Begin With Available Tables Populated

ADD OTHER: (not associated with your current SAS session)

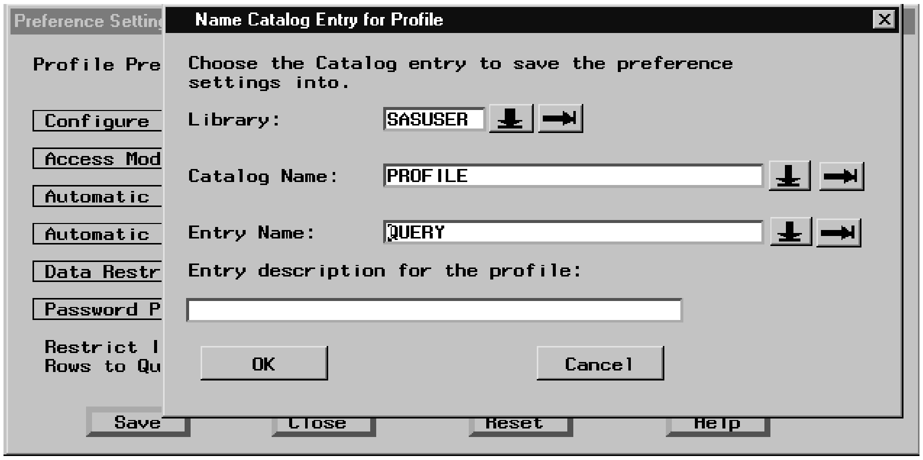
Table Source: Table Name:

Add entire Table Source to preferences
List Tables for Restriction Setting
Delete from profile preferences
WORK
Columns
OK
Reshow
Help

Select **WORK** from the list of Table Sources. Select **Add entire Table Source to preferences** from the pop-up menu.

Select **OK** to return to the Preference Settings for Profile window.

Select **Save** to save your new profile setting.



Name Catalog Entry for Profile

Choose the Catalog entry to save the preference settings into.

Library: SASUSER

Catalog Name: PROFILE

Entry Name: QUERY

Entry description for the profile:

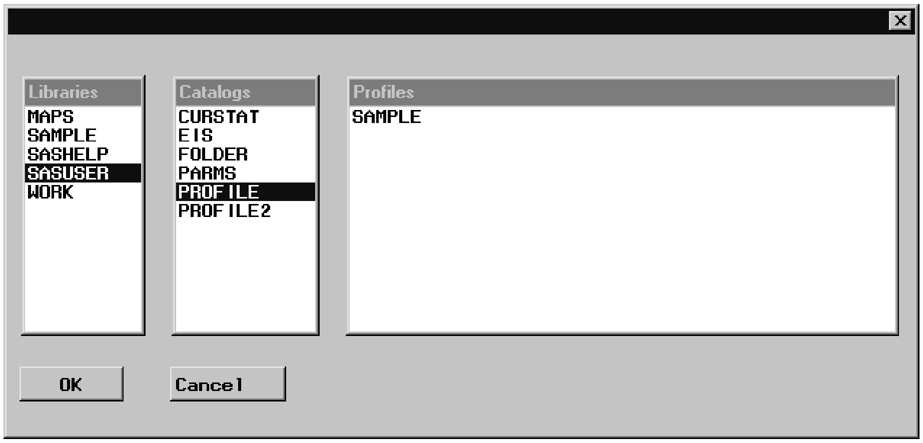
Type SAMPLE in the **Entry Name:** field of the Name Catalog Entry for Profile window. Select **OK**.

Select **Close** in the Preference Settings for Profile window.

From the SQL QUERY TABLES window PMENU, select

Tools ► Switch to New Profile

Select the → next to **Profile Name:** to display a list of profiles.



Select **SASUSER.PROFILE.SAMPLE** from the Preference Profiles in Catalog window.

Select **OK** to return to the SQL QUERY TABLES window and to complete the switch to the new profile. The new profile displays only the tables that are in the sample library.

To practice with the SQL Query Window examples, you will need the following data sets:

- SAMPLE.EMPINFO
- SAMPLE.JOBCODES
- SAMPLE.LEAVE
- SAMPLE.SALARY

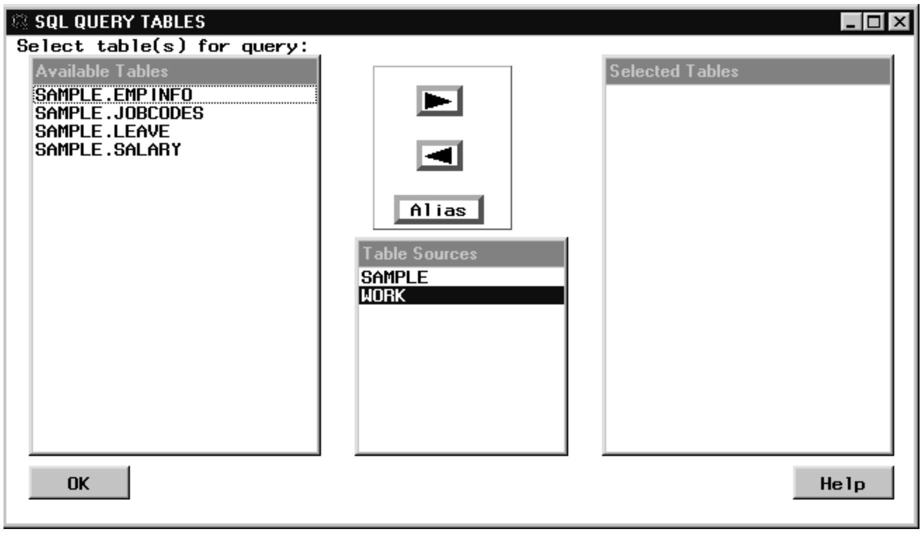
You will also need the following catalog:

- SAMPLE.PROGRAM

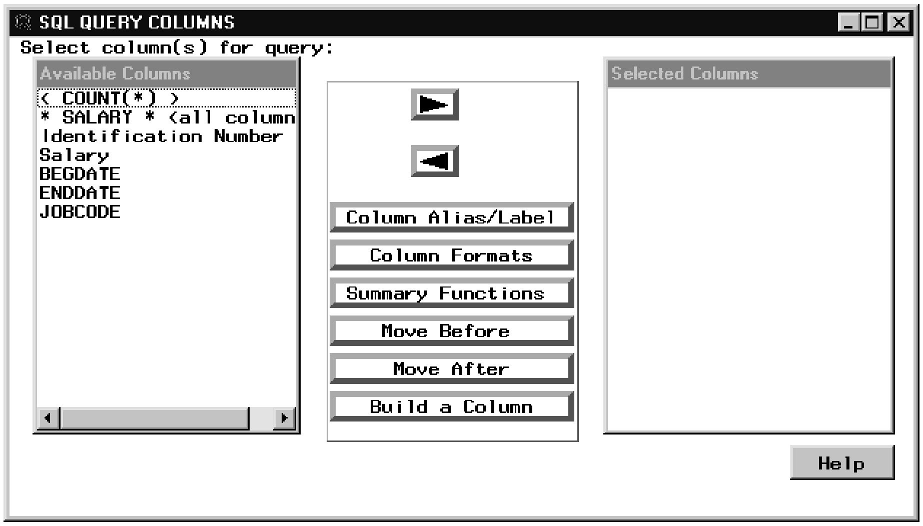
See “Setting Your Profile” on page 73 for more information on the SQL Query Window user profile.

Selecting a Table

First, you will analyze the relation between salary level, position, and hire date. Select **SAMPLE.SALARY** from the list of Available Tables.



Select ⇒ to add your selection to the Selected Tables list. You can also double-click on SAMPLE.SALARY to select it. Select **OK** to display the SQL QUERY COLUMNS window.

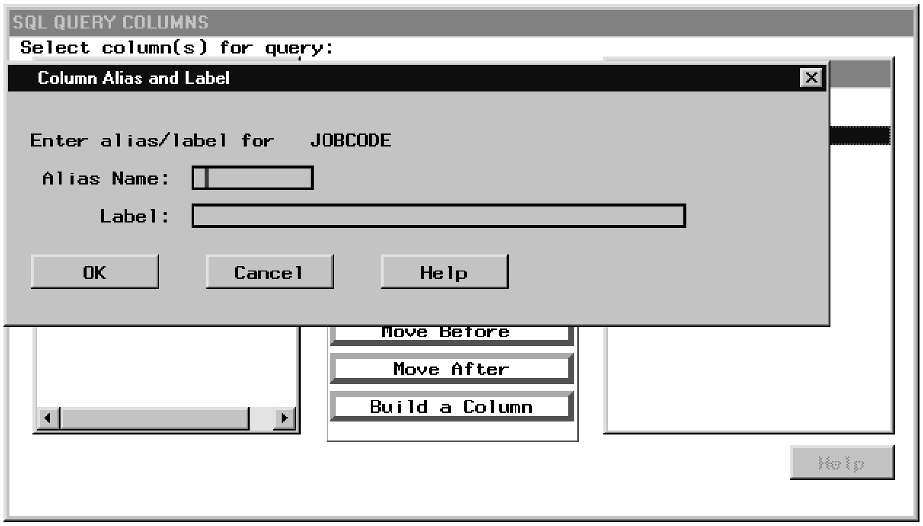


Selecting Columns

Select **Salary**, **BEGDATE**, and **JOBCODE** from the list of Available Columns. Select ⇒ to add your selections to the Selected Columns list.

Alias Names and Labels

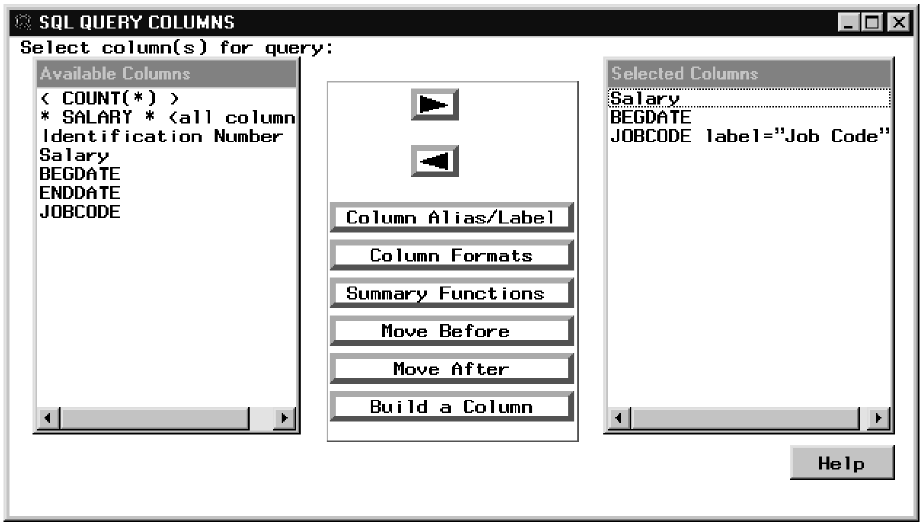
To create more descriptive labels for JOBCODE and BEGDATE, select **JOBCODE** from the list of Selected Columns. Select **Column Alias/Label** to assign a new label to the JOBCODE column.



Alias Name
: specifies an alias for the column. The alias is used in place of the column name both in the query and in any table or view that is created from the query. Aliases make a result table clearer or easier to read; they can also name a column expression.

Label
: associates a label with a column heading.

Type **Job Code** in the **Label:** field. Select **OK** to return to the SQL QUERY COLUMNS window. The assigned label is displayed next to JOBCODE in the Selected Columns List.


SQL QUERY COLUMNS

Select column(s) for query:

Available Columns

```
< COUNT(*) >
* SALARY * <all column
Identification Number
Salary
BEGDATE
ENDDATE
JOBCODE
```

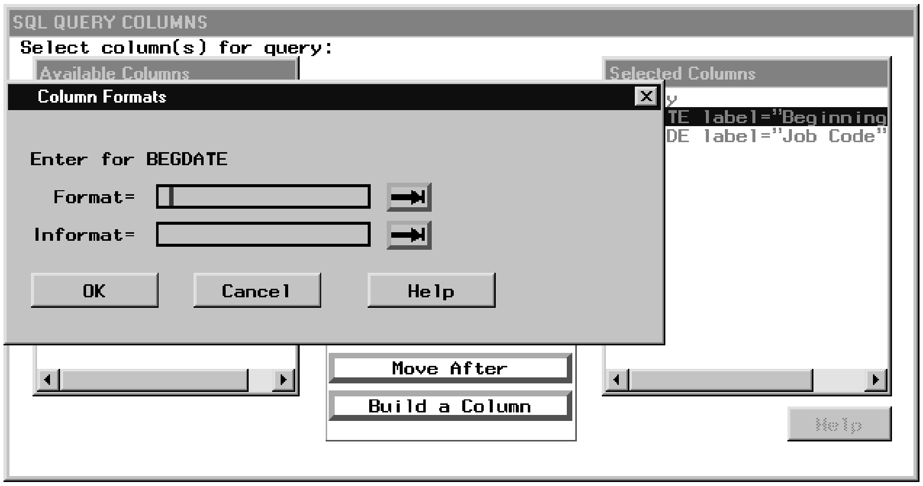
Selected Columns

```
Salary
BEGDATE
JOBCODE  label="Job Code"
```

Select **BEGDATE** from the Selected Columns list. Select **Column Alias/Label**. Type **Beginning Date** in the **Label:** field. Select **OK**.

Column Format

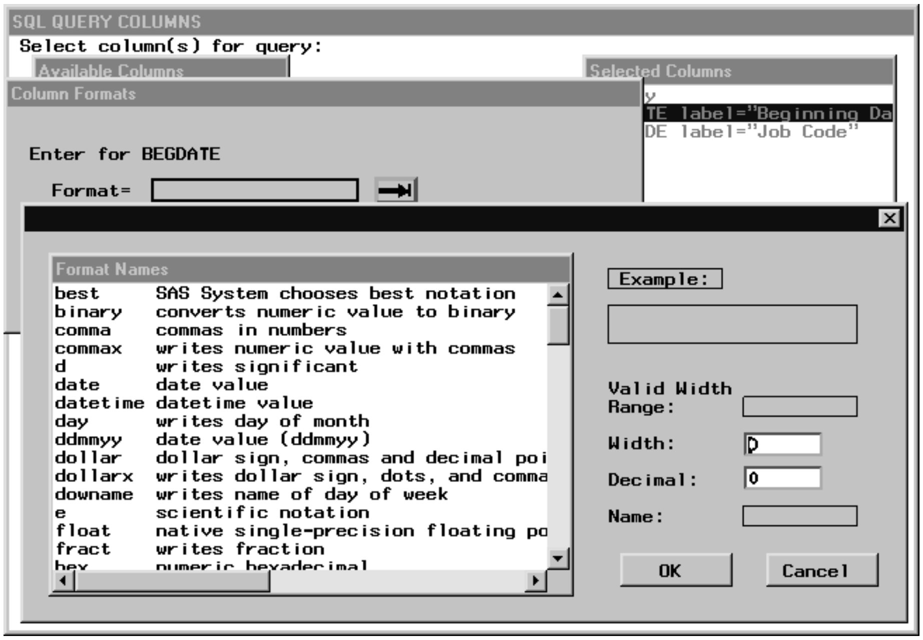
To modify the format of a column, select **BEGDATE** from the Selected Columns list. Select **Column Formats** to specify the format in which the beginning dates are presented.



Format
: specifies the form in which the column data are displayed. You can enter a format, or select → to see a list of valid formats. When you select a format, a formatted example appears, along with its width range, default width, default decimal, and name. You can either accept the default width and decimal values, or specify your own values in the **Width:** field.

Informat
: specifies the form in which the column data are read by other SAS procedures if you create a table or view from the query. You can enter an informat, or select → to see a list of valid informats. When you select an informat, a formatted example appears, along with its width range, default width, default decimal, and name. You can either accept the default width and decimal values, or specify your own values.

Select the → next to **Format** to display a list of formats.



```
y
TE label="Beginning Da
DE label="Job Code"
```

Enter for BEGDATE

Format=

```
best      SAS System chooses best notation
binary    converts numeric value to binary
comma     commas in numbers
commax    writes numeric value with commas
d         writes significant
date      date value
datetime  datetime value
day       writes day of month
ddmmyy    date value (ddmmyy)
dollar    dollar sign, commas and decimal poi
dollarx   writes dollar sign, dots, and comma
downame   writes name of day of week
e         scientific notation
float     native single-precision floating po
fract     writes fraction
hex       numeric hexadecimal
```

Example:

Valid Width Range:

Width: 0

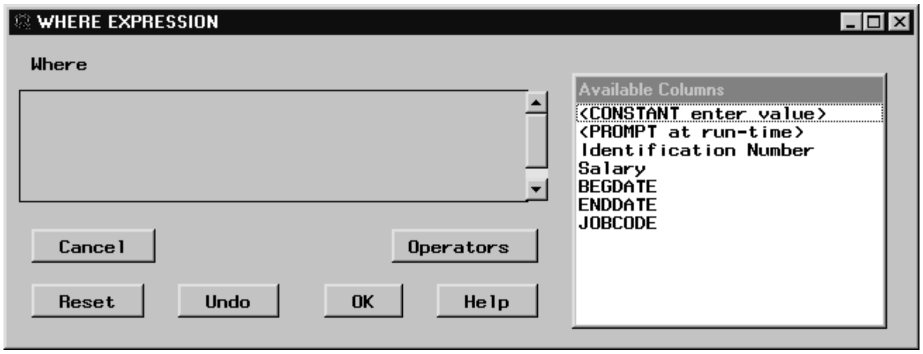
Decimal: 0

Name:

Select **date** from the list of Format Names. Type **9** in the **Width:** field. Select **OK**. Select **OK** to return to the SQL QUERY COLUMNS window.

Creating a WHERE Expression

A WHERE expression returns a subset of data that meet conditions you specify. Create a WHERE expression that displays the range of job codes whose employees were hired after October 1991 and whose salaries are less than $18,000.00. Select **Where Conditions for Subset...** from the **View** PMENU. The WHERE EXPRESSION window appears.



```
<CONSTANT enter value>
<PROMPT at run-time>
Identification Number
Salary
BEGDATE
ENDDATE
JOBCODE
```

Available Columns

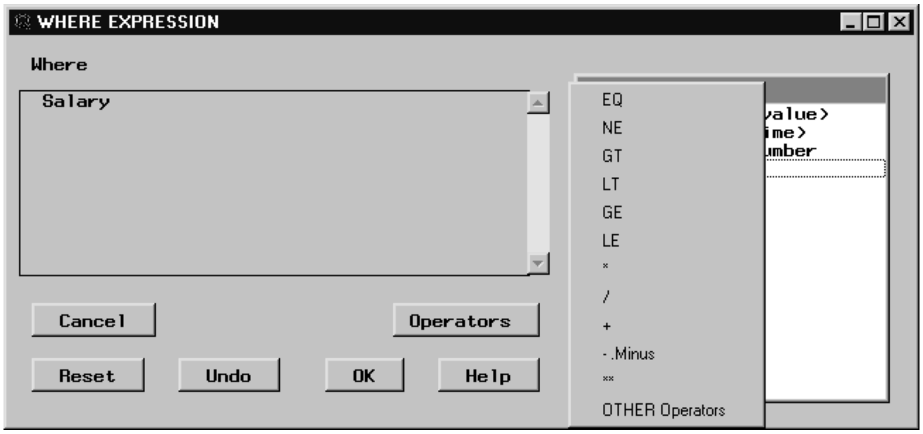
The Available Columns list contains all the columns from the selected tables, in addition to the following choices:

<CONSTANT enter value> enables you to enter a constant value for the Where expression

<PROMPT at run-time> enables you to enter a value for the Where expression when you run the query or create a table or view.

Comparison Operators

Select **Salary** from the Available Columns list. A list of numeric comparison operators appears.



The list of operators is specific to the data type.

EQ	is equal to
NE	is not equal to
GT	is greater than
LT	is less than
GE	is greater than or equal to
LE	is less than or equal to
*	multiplies by
/	divides by
+	adds
-	subtracts
**	raises to a power

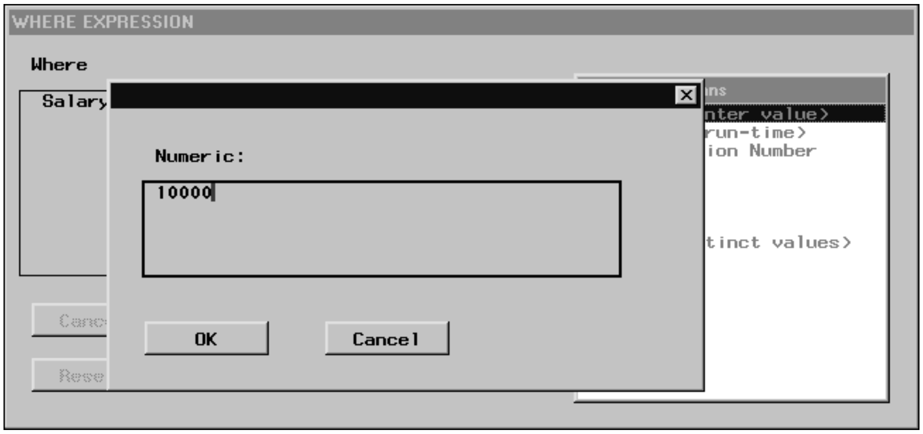
The **OTHER Operators** are:

Is Missing	selects rows in which a column value is missing or null.
Is Not Missing	selects rows in which a column value is not missing or is not null.
Between	Searches for values that lie within the specified parameters.
Not Between	Searches for values that lie outside the specified parameters.
In	Tests if the column value is a member of a set.
Not In	Tests if the column value is not a member of a set.

Select **LT** from the list of comparison operators.

Constant Values

Select **<CONSTANT enter value>**. Enter **10000** in the **Numeric:** field.



Salary

Numeric:

10000

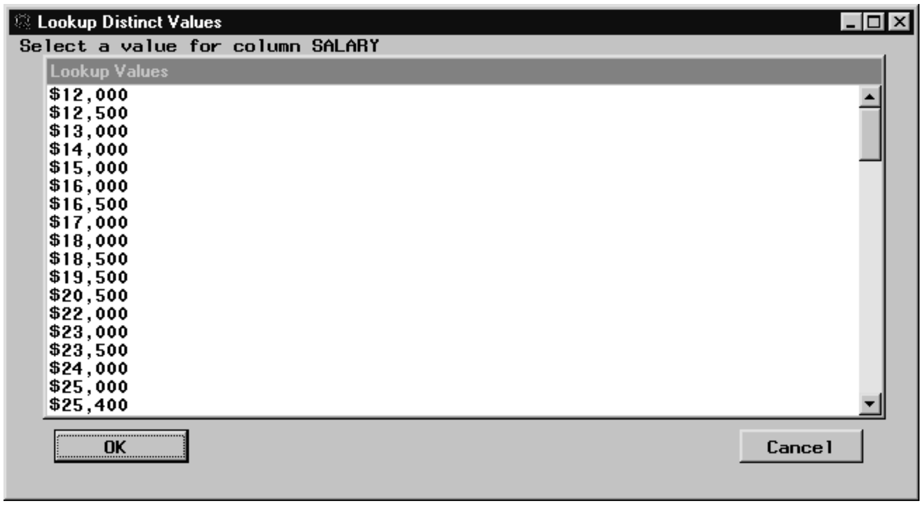
Select **OK**. The WHERE expression is built for you as you select new operators and values.

Undo

Select **Undo** to remove **10000** from the WHERE statement. You can delete your last addition to the WHERE statement by selecting **Undo**.

Lookup Distinct Values

Select **LOOKUP distinct values** to view all of the values for the SALARY column. Distinct values remove duplicate rows from your output table.



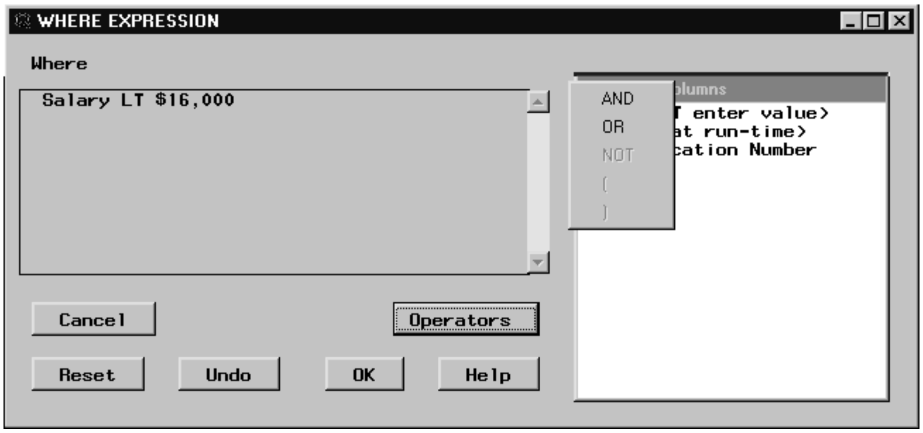
Select a value for column SALARY

```
$12,000
$12,500
$13,000
$14,000
$15,000
$16,000
$16,500
$17,000
$18,000
$18,500
$19,500
$20,500
$22,000
$23,000
$23,500
$24,000
$25,000
$25,400
```

Select **$18,000** from the list of values. Because the LT comparison operator requires only one value, you are automatically returned to the WHERE EXPRESSION window.

Logical Operators

Select **Operators** to display the list of operators. Note that the list of comparison operators has changed to a list of logical operators. Select **AND** from the list of operators.



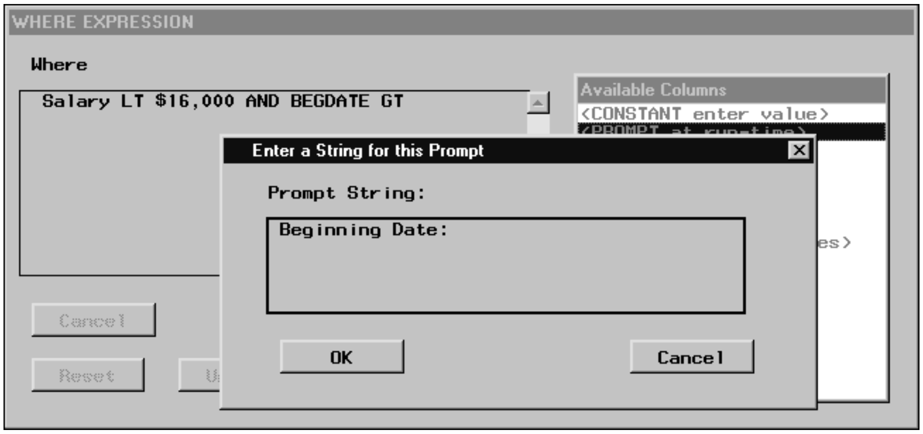
Where

Salary LT $16,000

Select **BEGDATE** from the list of Available Columns. Select **GT** from the list of comparison operators.

Run-Time Prompt

Select **PROMPT at run-time** to display the Prompt String window. Type **Beginning Date:** in the **Prompt String** field.



Where

Salary LT $16,000 AND BEGDATE GT

Prompt String:

Beginning Date:

Select **OK**. **&PROMPT1** in the Where expression indicates that you will supply a value for this variable when you run the query.

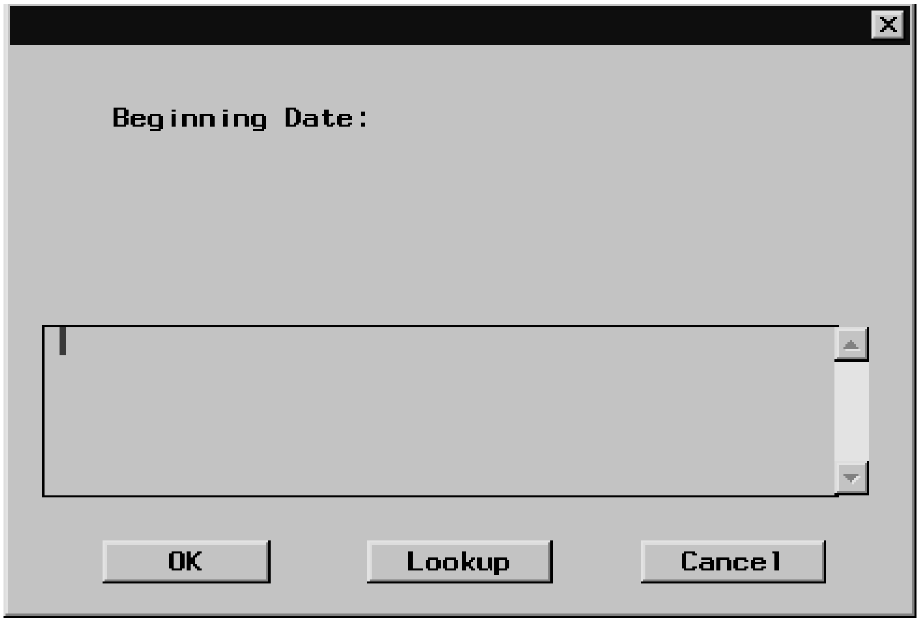
Select **OK** from the WHERE EXPRESSION window to return to the SQL QUERY COLUMNS window.

Running Your Query

To run your query, select

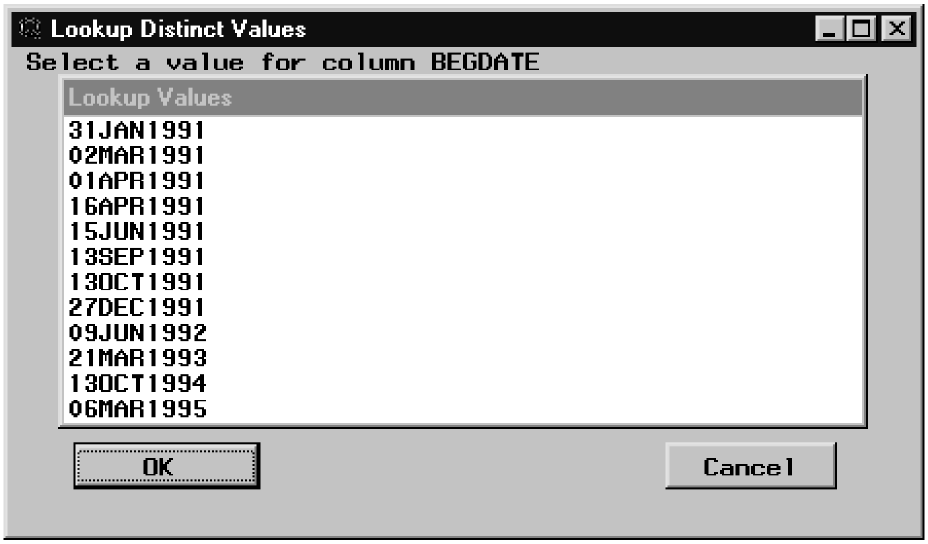
Tools ► Run Query ► Run Immediate

The Prompt at Run Time window appears, with the **Beginning Date:** prompt that you specified in the WHERE expression.



Beginning Date:

Select **Lookup** to display a list of values for **Beginning Date:**



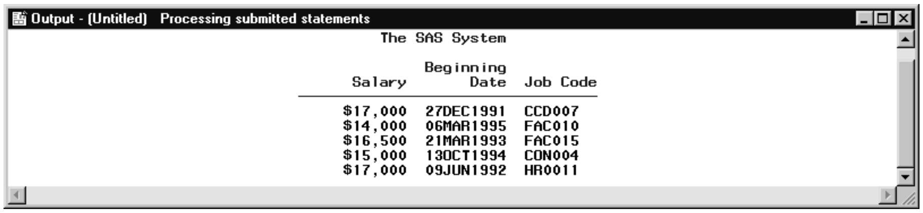
Lookup Distinct Values

Select a value for column BEGDATE

Lookup Values

```
31JAN1991
02MAR1991
01APR1991
16APR1991
15JUN1991
13SEP1991
13OCT1991
27DEC1991
09JUN1992
21MAR1993
13OCT1994
06MAR1995
```

Select **13OCT1991** from the list of values; the Prompt at Runtime Window is displayed with the value that you selected. Select **OK** to continue to run the query and to view your output in the OUTPUT window.



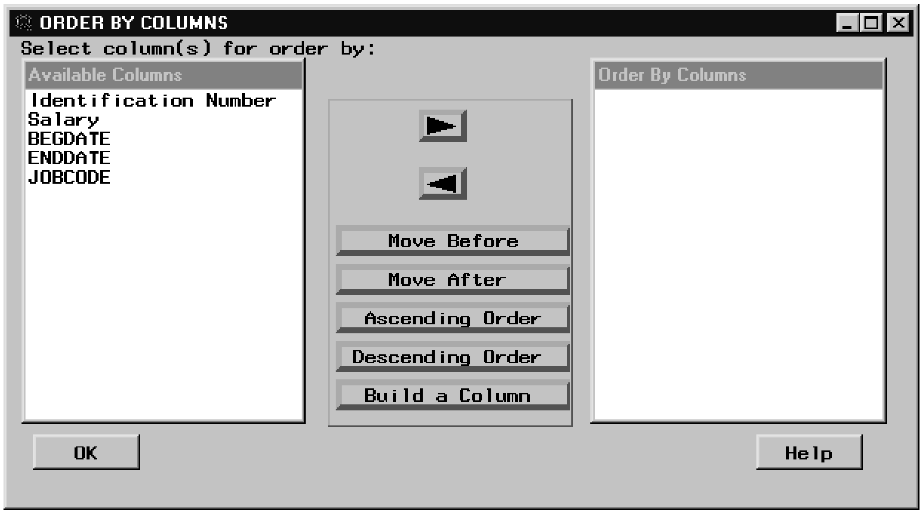
Output - (Untitled) Processing submitted statements

```
                     The SAS System

               Beginning
      Salary        Date  Job Code
  --------------------------------
     $17,000   27DEC1991  CCD007
     $14,000   06MAR1995  FAC010
     $16,500   21MAR1993  FAC015
     $15,000   13OCT1994  CON004
     $17,000   09JUN1992  HR0011
```

Sorting Your Output

Order By enables you to specify the order in which you want the output sorted. Using the query from the last example, you can change the ordering sequence of the columns in your OUTPUT window. From the SQL QUERY COLUMNS window, select

View ► Order By...



Move Before displays all columns in the Order By Columns list except the currently chosen one. The currently chosen column(s) will be inserted before the column(s) that you select.

Move After displays all the columns in the Order By Columns list except the currently chosen one. The currently chosen column(s) will be inserted following the column(s) that you select.

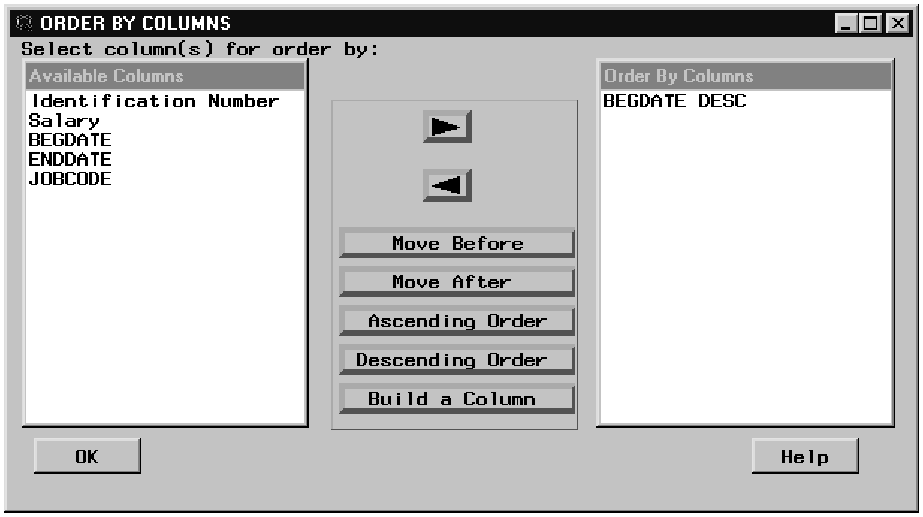
Ascending Order changes the ordering sequence of the selected column's values to ascending (lowest value to highest value).

Descending Order changes the ordering sequence of the selected column's values to descending.

Build a Column displays the Build a Column Expression window, which enables you to create your calculated column for use in sorting your output. Use the Build a Column Expression window to create new columns by performing calculations on existing (numeric) columns.

Order By Columns

Select **BEGDATE** from the list of Available Columns. Select → to move it to the list of Order By Columns. By default, columns are sorted in ascending order, so the abbreviation ASC appears next to the column name in the Order By Columns list. Select **BEGDATE ASC** and **Descending Order** to change the ordering sequence.



Select column(s) for order by:

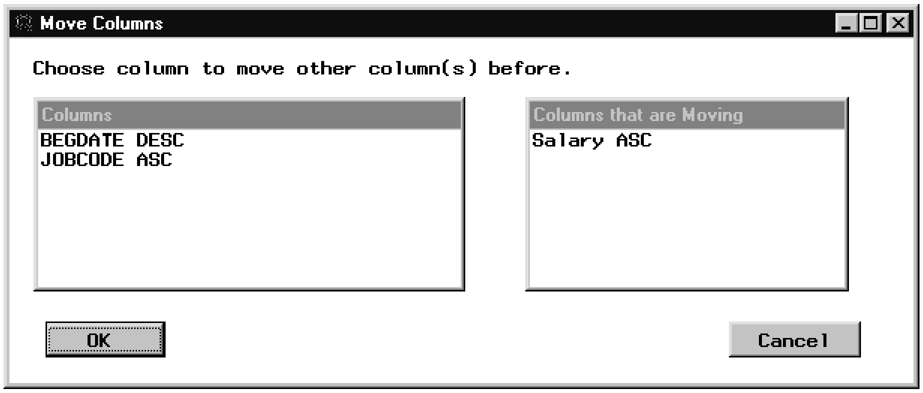
Available Columns: Identification Number, Salary, BEGDATE, ENDDATE, JOBCODE

Order By Columns: BEGDATE DESC

Select **Salary** and **JOBCODE**, and move them to the list of Order By Columns.

Move Columns

Select **Salary** and **Move Before**. The Move Columns window appears.

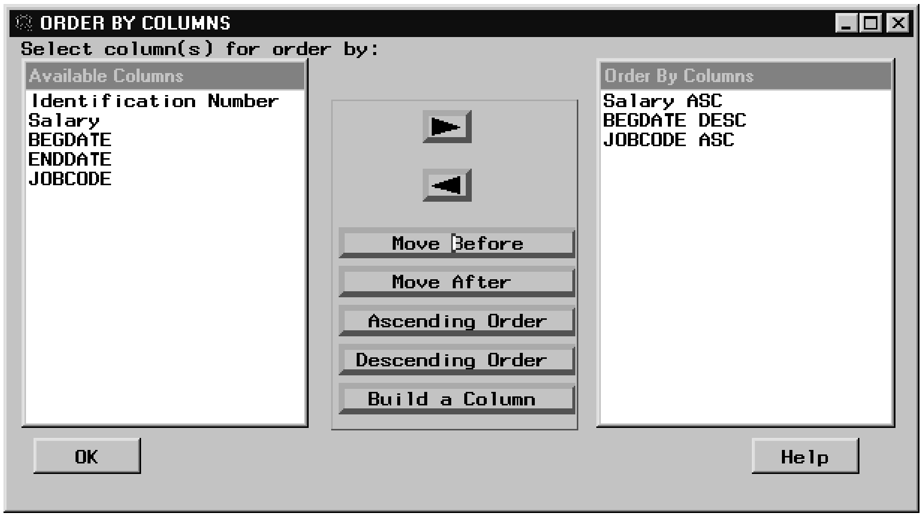


Choose column to move other column(s) before.

Columns: BEGDATE DESC, JOBCODE ASC

Columns that are Moving: Salary ASC

Select **BEGDATE** and **OK**. The ORDER BY COLUMNS window is displayed with Salary first in the list of Order By Columns.



ORDER BY COLUMNS

Select column(s) for order by:

Available Columns
Identification Number
Salary
BEGDATE
ENDDATE
JOBCODE

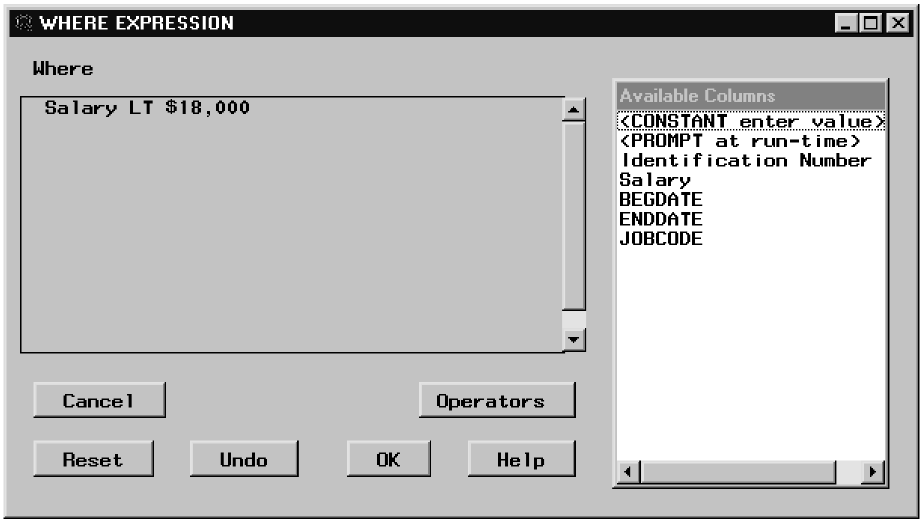
Order By Columns
Salary ASC
BEGDATE DESC
JOBCODE ASC

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

Select

View ► Where Conditions for Subset...

The WHERE EXPRESSION window appears. Select **Undo** four times, until only **SALARY LT $18,000** is displayed.



WHERE EXPRESSION

Where

Salary LT $18,000

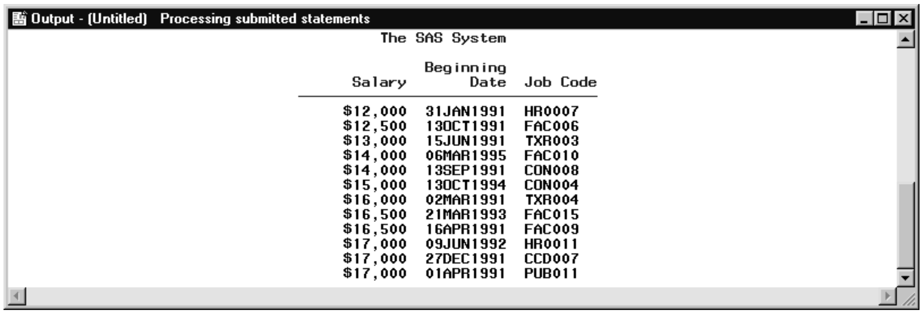
Available Columns
<CONSTANT enter value>
<PROMPT at run-time>
Identification Number
Salary
BEGDATE
ENDDATE
JOBCODE

Select **OK**.

Viewing Your Output

To run your query and view the output in the OUTPUT window, select

Tools ► Run Query ► Run Immediate



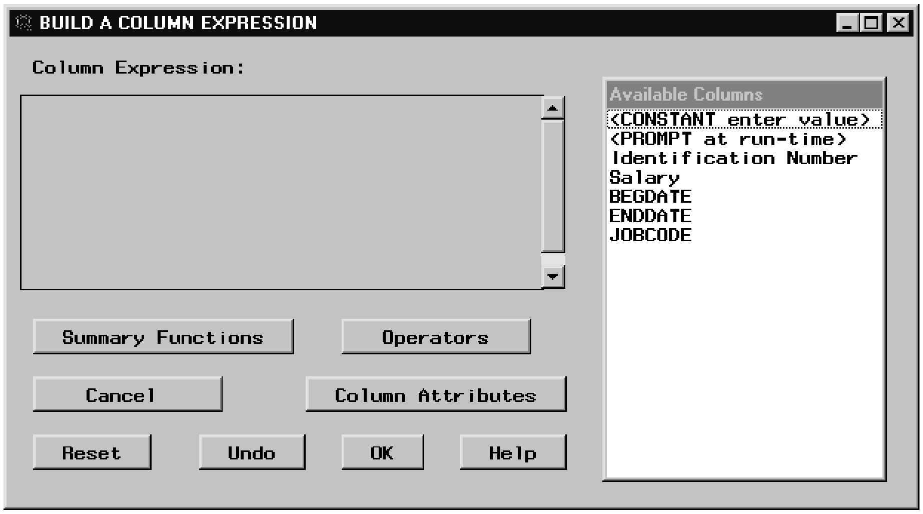
```
                                The SAS System

                                  Beginning
                       Salary          Date   Job Code
                  ------------------------------------
                      $12,000   31JAN1991   HR0007
                      $12,500   13OCT1991   FAC006
                      $13,000   15JUN1991   TXR003
                      $14,000   06MAR1995   FAC010
                      $14,000   13SEP1991   CON008
                      $15,000   13OCT1994   CON004
                      $16,000   02MAR1991   TXR004
                      $16,500   21MAR1993   FAC015
                      $16,500   16APR1991   FAC009
                      $17,000   09JUN1992   HR0011
                      $17,000   27DEC1991   CCD007
                      $17,000   01APR1991   PUB011
```

Building Calculated Columns

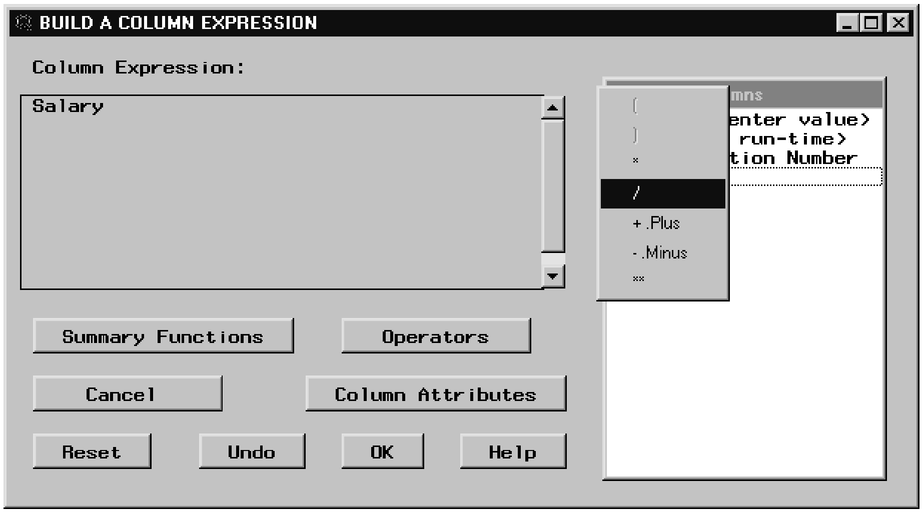
Using the query from the last example, you can create a new column that computes the hourly wage for each salary.

Build a Column Expression

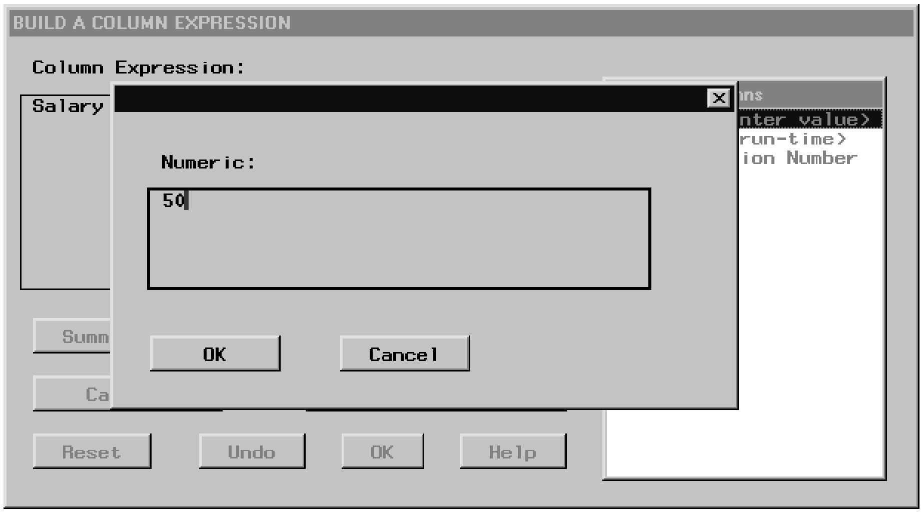
Select **Build a Column** from the SQL QUERY COLUMNS window to display the BUILD A COLUMN EXPRESSION window.



Select **Salary** from the Available Columns window. Select the division operator **(/)** from the list of operators.



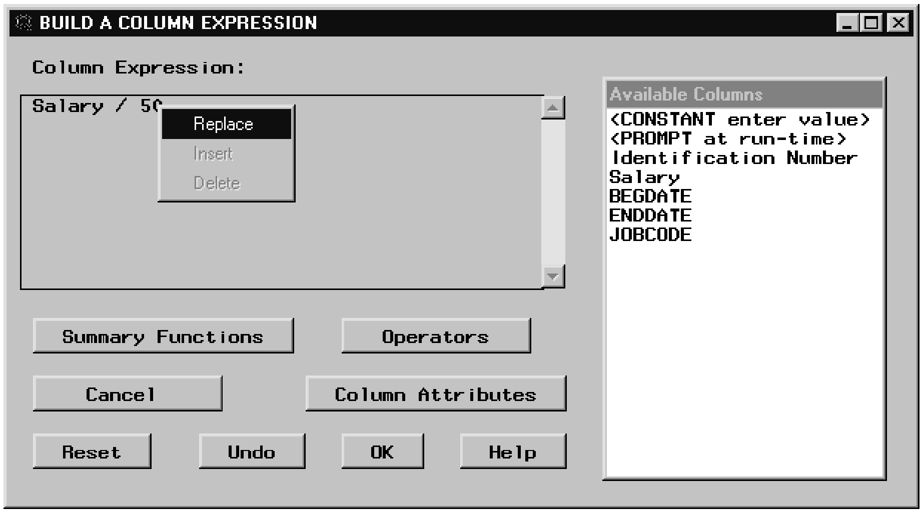
Select **<CONSTANT enter value>** from the Available Columns window. Enter **50** in the Numeric Constant dialog. Select **OK** to return to the BUILD A COLUMN EXPRESSION window.



Select the division operator again from the list of operators, and enter 40 to divide the number of weeks by the number of hours in each week. Click **OK**. Click outside the operator box to dismiss the list of operators.

Correcting Your Mistakes

You realize that you have made a mistake and that you want to divide Salary by 52, the number of weeks in a year. Select **50** in the WHERE expression. A pop-up menu displays a list of choices.



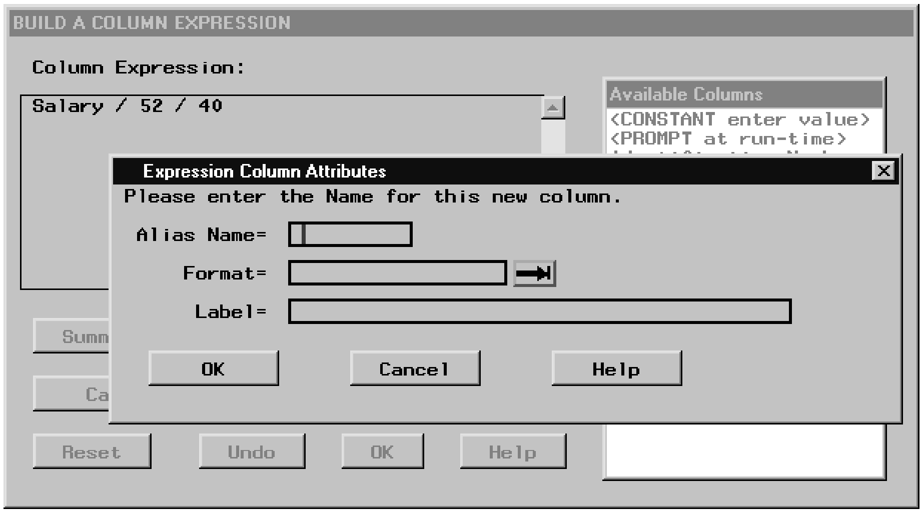
Column Expression: Salary / 50

Select **Replace** from the pop-up menu. The BUILD A COLUMN EXPRESSION window displays "Select from Available Columns to replace this value."

Select **<CONSTANT Enter Values>** from the list of Available Columns. Enter **52** as the new constant.

Defining the Column Format and Label

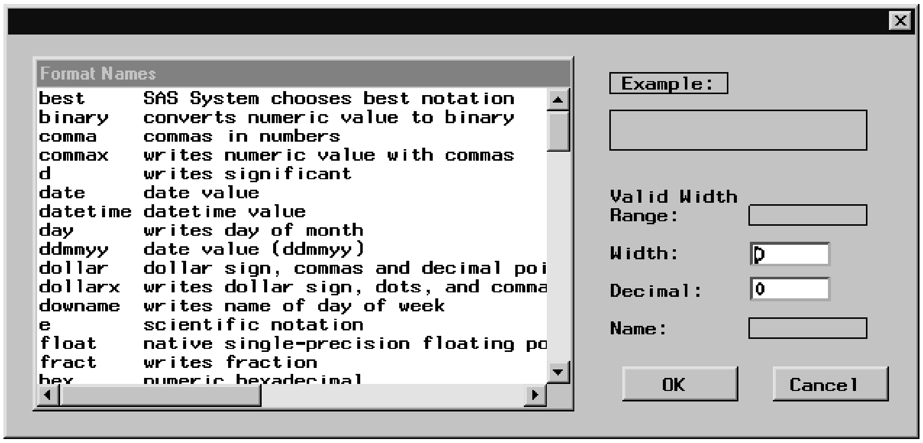
Select **Column Attributes** to define the format and label for your new column.



Column Expression: Salary / 52 / 40

Please enter the Name for this new column.

Enter **hourly** as the alias name. Select the ⇥ next to the **Format** field to choose the format in which the new column will appear.



Format Names

```
best     SAS System chooses best notation
binary   converts numeric value to binary
comma    commas in numbers
commax   writes numeric value with commas
d        writes significant
date     date value
datetime datetime value
day      writes day of month
ddmmyy   date value (ddmmyy)
dollar   dollar sign, commas and decimal poi
dollarx  writes dollar sign, dots, and comma
downame  writes name of day of week
e        scientific notation
float    native single-precision floating po
fract    writes fraction
hex      numeric hexadecimal
```

Example:

Valid Width Range:

Width: 0

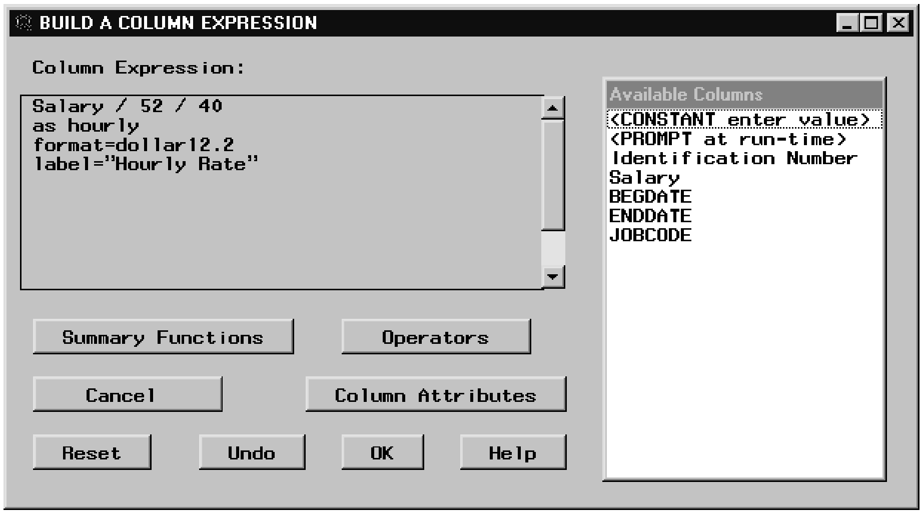
Decimal: 0

Name:

Select **dollar** from the list of Format Names. Enter **2** in the **Decimal:** field so that the hourly wage will be displayed to two decimal places. Select **OK**.

Enter **Hourly Rate** in the **Label** field for the column. Select **OK**.

The complete calculated column is displayed in the Column Expression window.



BUILD A COLUMN EXPRESSION

Column Expression:

```
Salary / 52 / 40
as hourly
format=dollar12.2
label="Hourly Rate"
```

Available Columns

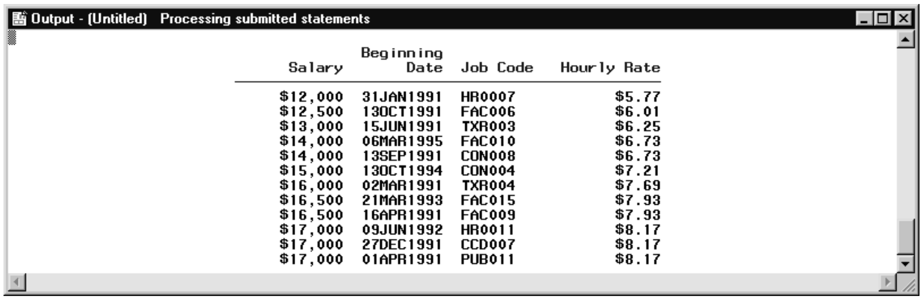
<CONSTANT enter value>
<PROMPT at run-time>
Identification Number
Salary
BEGDATE
ENDDATE
JOBCODE

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

Viewing Your Output

To run your query and view the output in the OUTPUT window, select

Tools ► Run Query ► Run Immediate



Output - (Untitled) Processing submitted statements

```
                Beginning
       Salary        Date  Job Code  Hourly Rate
       ------------------------------------------
      $12,000   31JAN1991  HR0007          $5.77
      $12,500   13OCT1991  FAC006          $6.01
      $13,000   15JUN1991  TXR003          $6.25
      $14,000   06MAR1995  FAC010          $6.73
      $14,000   13SEP1991  CON008          $6.73
      $15,000   13OCT1994  CON004          $7.21
      $16,000   02MAR1991  TXR004          $7.69
      $16,500   21MAR1993  FAC015          $7.93
      $16,500   16APR1991  FAC009          $7.93
      $17,000   09JUN1992  HR0011          $8.17
      $17,000   27DEC1991  CCD007          $8.17
      $17,000   01APR1991  PUB011          $8.17
```

Building and Adding Tables

Using the query from the last example, you can build a new table from the results of your latest query.

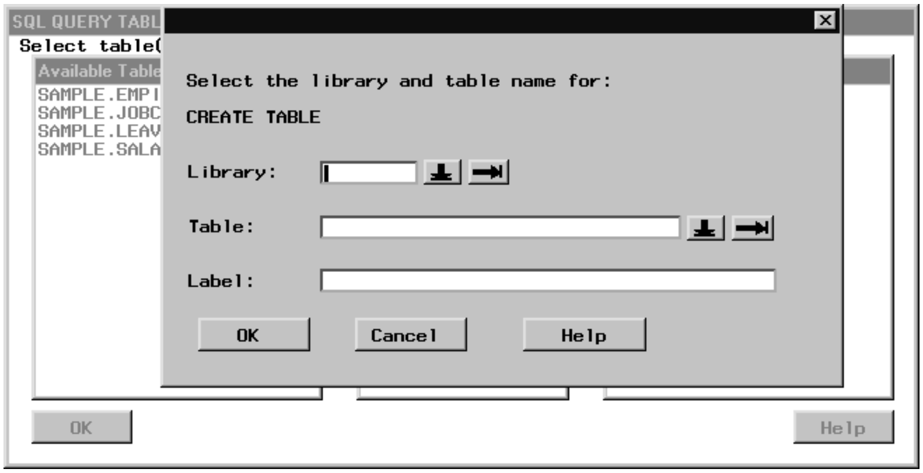
Creating a Table from Query Results

In the SQL QUERY COLUMNS window, select

View ▶ Tables...

to return to the SQL QUERY TABLES window.
From the SQL QUERY TABLES window, select

File ▶ Create Table From Query Results...



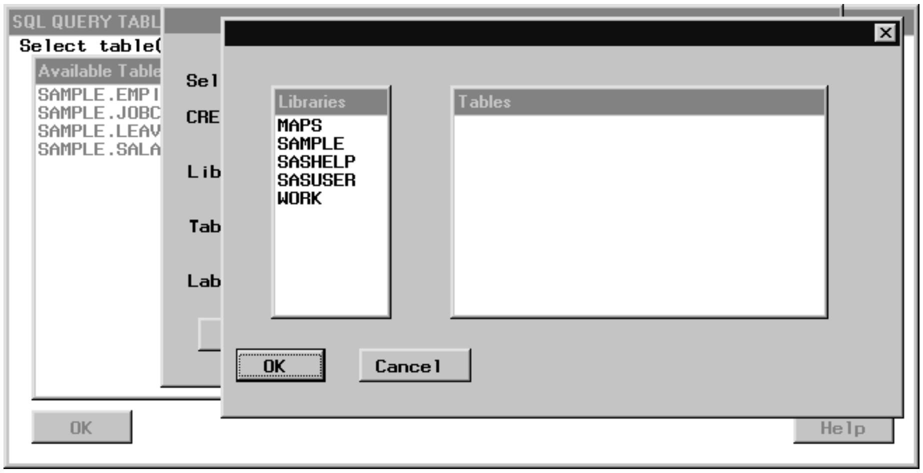
Select the library and table name for:
CREATE TABLE

Library:

Table:

Label:

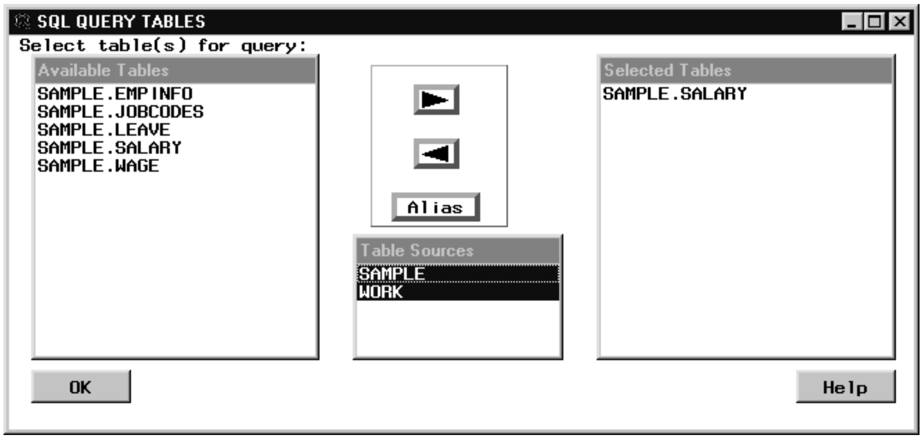
Select the ⇒ next to the **Library:** field to display a list of available libraries.



You can also enter the library name in the **Library:** field.

Select **SAMPLE** to include your new table in the SAMPLE library. Select **OK**. Type WAGE in the **Table:** field. Type **Hourly Wages** in the **Label:** field to specify the permanent label to be used when displaying that column in a table.

Select **OK** to return to the SQL QUERY TABLES window. SAMPLE.WAGE is now in the list of Available Tables.



Available Tables
SAMPLE.EMPINFO
SAMPLE.JOBCODES
SAMPLE.LEAVE
SAMPLE.SALARY
SAMPLE.WAGE

Table Sources
SAMPLE
WORK

Selected Tables
SAMPLE.SALARY

Adding the New Table to Selected Tables

Select **SAMPLE.WAGE** and ⇒ to add the new table to the list of Selected Tables.

Select

Tools ▶ Reset

to reset your query and return to the SQL QUERY TABLES window.

Joining Matching Data

An inner join returns a result table for all of the rows in a table that has one or more matching rows in another table. You can use an inner join to display the hourly wage for each employee identification number.

In the previous example, you added SAMPLE.WAGE to the list of Available Tables. Select SAMPLE.SALARY and SAMPLE.WAGE from the list of Available Tables and add them to the list of Selected Tables. Select **OK** to display the SQL QUERY COLUMNS window. Select **Identification Number**, **JOBCODE**, and **Hourly Rate** from the list of Available Columns and move them to the list of Selected Columns.

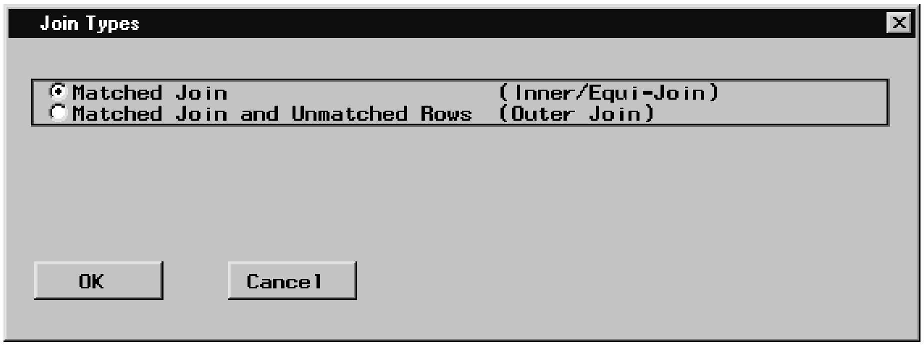
Join Types

You can choose to use the results of an inner join or an outer join. See "Creating and Using Outer Joins" on page 66 for more information on outer joins.

Select

View ▶ Join Type...

to display the Join Types window.



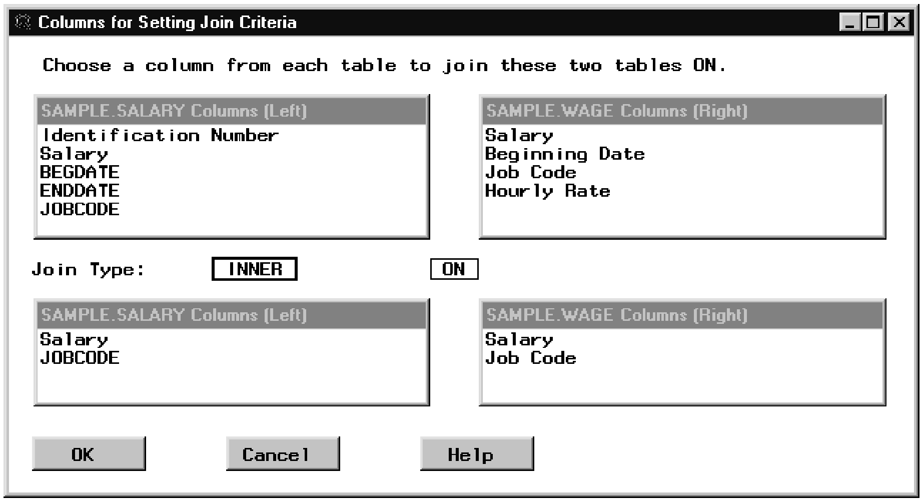
Join Types

● Matched Join (Inner/Equi-Join)
○ Matched Join and Unmatched Rows (Outer Join)

Select **Matched Join** and **OK**.

Setting Join Criteria

In the Columns for Setting Join Criteria window, select **Salary** from SAMPLE.SALARY Columns and SAMPLE.WAGE Columns. Select **JOBCODE** from SAMPLE.SALARY Columns and **JOBCODE** from SAMPLE.WAGE Columns.



Columns for Setting Join Criteria

Choose a column from each table to join these two tables ON.

SAMPLE.SALARY Columns (Left)
Identification Number
Salary
BEGDATE
ENDDATE
JOBCODE

SAMPLE.WAGE Columns (Right)
Salary
Beginning Date
Job Code
Hourly Rate

Join Type: INNER ON

SAMPLE.SALARY Columns (Left)
Salary
JOBCODE

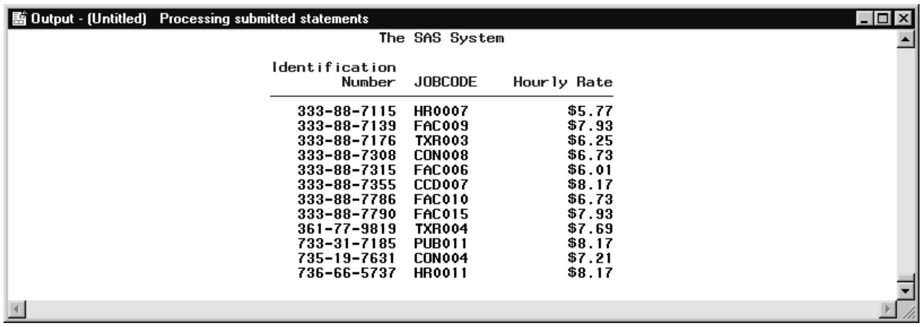
SAMPLE.WAGE Columns (Right)
Salary
Job Code

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

Viewing Your Output

To run your query and view the output in the OUTPUT window, select

Tools ▶ Run Query ▶ Run Immediate



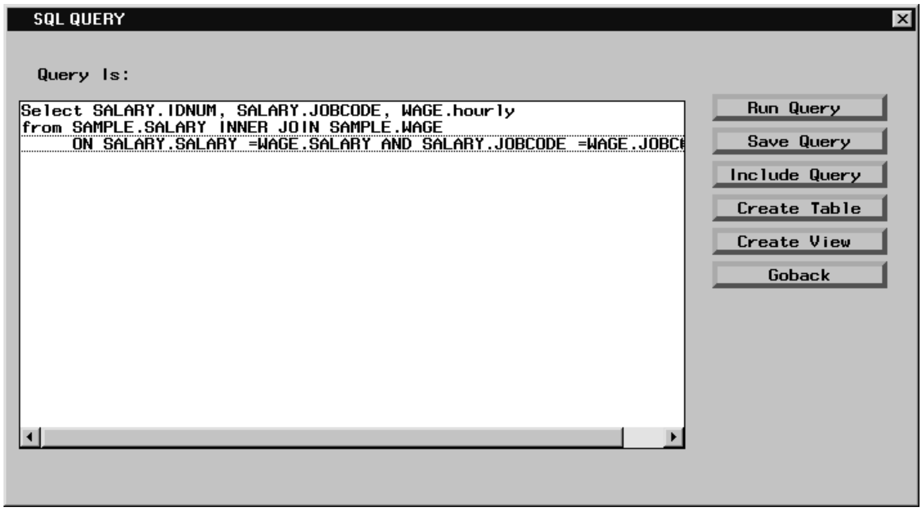
```
                              The SAS System

               Identification
                       Number   JOBCODE    Hourly Rate
               ___________________________________________

                  333-88-7115   HR0007           $5.77
                  333-88-7139   FAC009           $7.93
                  333-88-7176   TXR003           $6.25
                  333-88-7308   CON008           $6.73
                  333-88-7315   FAC006           $6.01
                  333-88-7355   CCD007           $8.17
                  333-88-7786   FAC010           $6.73
                  333-88-7790   FAC015           $7.93
                  361-77-9819   TXR004           $7.69
                  733-31-7185   PUB011           $8.17
                  735-19-7631   CON004           $7.21
                  736-66-5737   HR0011           $8.17
```

Saving Queries

To save the query that you created in the previous example in your SASUSER.PROFILE, select

Tools ► Show Query...

to display the SQL QUERY window.



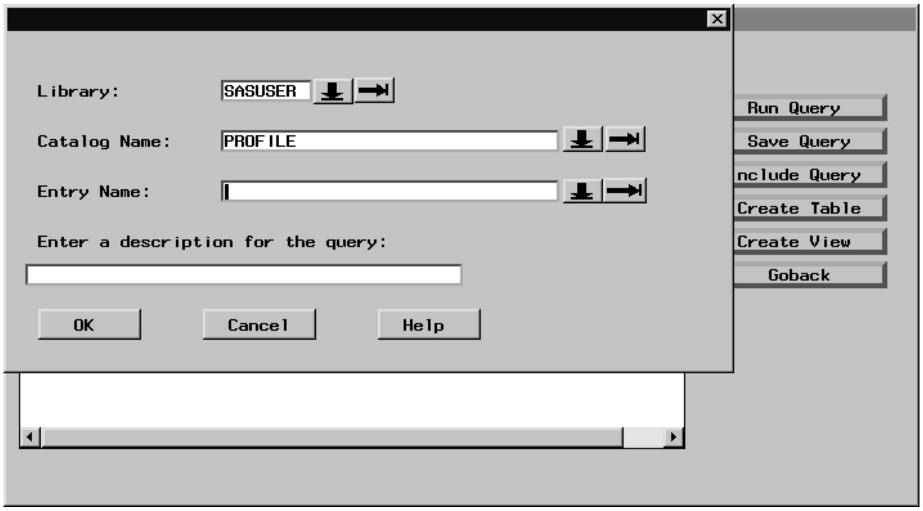
```
Select SALARY.IDNUM, SALARY.JOBCODE, WAGE.hourly
from SAMPLE.SALARY INNER JOIN SAMPLE.WAGE
      ON SALARY.SALARY =WAGE.SALARY AND SALARY.JOBCODE =WAGE.JOBC
```

Saving a Query to Include Later

Select

Save Query ► Save as QUERY to Include later...

to save your query to SASUSER.PROFILE or another catalog of your choosing.



Library: SASUSER

Catalog Name: PROFILE

Entry Name:

Enter a description for the query:

Type IDWAGE in the **Entry Name** field. Type **ID number and hourly wage** in the description field. Select **OK** to save your query as an entry in SASUSER.PROFILE and to return to the SQL QUERY window. Select **Goback** to return to the SQL QUERY COLUMNS window.

Saving Several Queries

You can save more than one query and then select from a list of queries that you have saved in the current Query Window session or in a previous Query Window session.

From the SQL QUERY COLUMNS window, select

View ▶ Tables...

to return to the SQL QUERY TABLES window.

Remove SAMPLE.WAGE from the list of Selected Tables. Select **OK** to display the SQL QUERY COLUMNS window.

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

View ▶ Where Conditions for Subset...

to display the WHERE EXPRESSION window.

Select **Salary** from the list of Available Columns. Select **GT** (Greater Than) from the list of Operators. Select **<LOOKUP distinct values>** from the list of Available Columns. Select **$25,000** from the Lookup Values window. Select **OK** to save your WHERE expression. Select

View ▶ Order By...

to display the ORDER BY COLUMNS window. Select **Salary** from the list of Available Columns and add it to the list of Selected Columns. Select **OK** to return to the SQL QUERY COLUMNS window.

You can also use the **File** menu to save queries. Select

File ▶ Save Query ▶ Save as Query to Include later

Type ABOVE25 in the **Entry Name:** field. Type **Salaries above $25,000** in the Description field.

You can also save queries that will be processed against different tables. For the next query that you will save, select

View ▶ Tables...

to return to the SQL QUERY TABLES window. Remove SAMPLE.SALARY from the list of Selected Tables, and select **SAMPLE.EMPINFO** from the list of Available Tables to add to the list of Selected Tables. Select **OK** to display the SQL QUERY COLUMNS window. Add **NAME**, **DIVISION**, and **Education Level** to the list of Selected Columns. Select

View ▶ Where Conditions for Subset...

Select **Education level** from the list of Available Columns. Select **GE** (Greater than or Equal to) from the list of Operators. Select **<LOOKUP distinct values>**, and select **20** from the list of Lookup Values. Select **OK** to return to the SQL QUERY COLUMNS window.

Select

File ▶ Save Query ▶ Save as Query to Include later

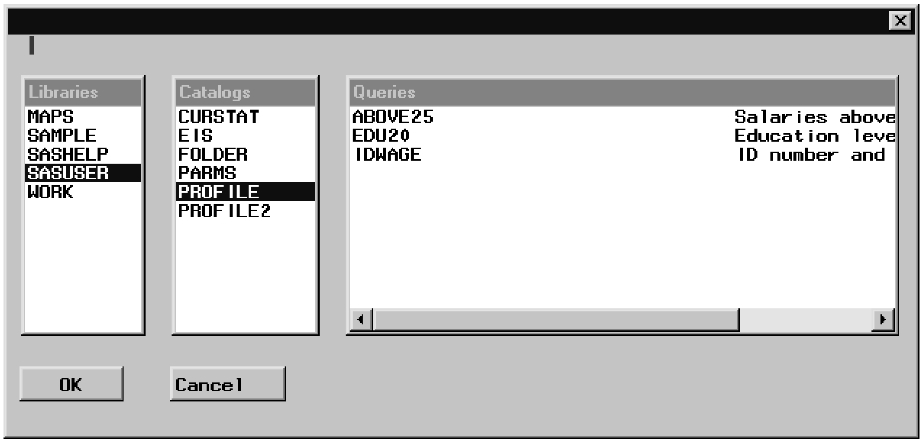
Type EDU20 in the **Entry Name** field. Type **Education level above 20 years** in the Description field. Select **OK** to save the query.

Listing Saved Queries

You can now display a list of the queries that you have saved, and include one of the queries. Select

File ▶ List/Include Saved Queries

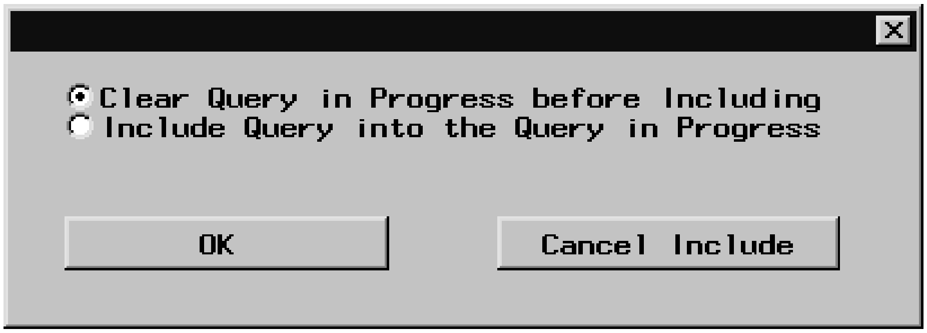
The queries that you have created are listed in the Saved Queries window.



```
Libraries      Catalogs      Queries
MAPS           CURSTAT       ABOVE25                        Salaries above
SAMPLE         EIS           EDU20                          Education leve
SASHELP        FOLDER        IDWAGE                         ID number and
SASUSER        PARMS
WORK           PROFILE
               PROFILE2
```

Including a Saved Query

Select **SASUSER.PROFILE.EDU20** and select **OK**. A pop-up window asks whether or not you want to clear the previous query or include the previous query with the new one. Select **OK**.



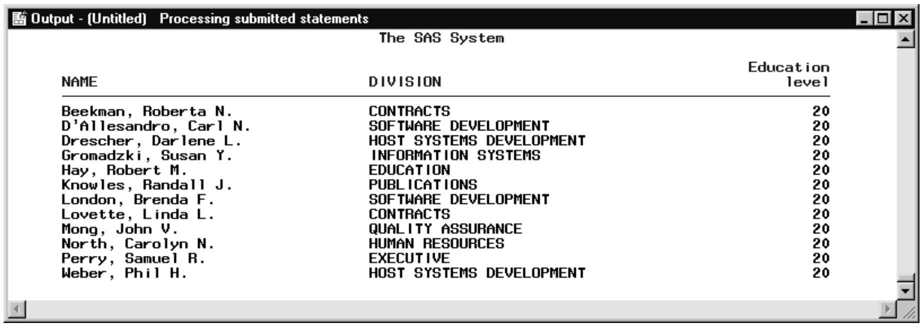
● Clear Query in Progress before Including
○ Include Query into the Query in Progress

Viewing Your Output

You can run SASUSER.PROFILE.EDU20 by selecting

Tools ► Run Query ► Run Immediate

The results are displayed in the OUTPUT window.



```
                                  The SAS System

                                                                        Education
NAME                              DIVISION                                  level
---------------------------------------------------------------------------------

Beekman, Roberta N.               CONTRACTS                                    20
D'Allesandro, Carl N.             SOFTWARE DEVELOPMENT                         20
Drescher, Darlene L.              HOST SYSTEMS DEVELOPMENT                     20
Gromadzki, Susan Y.               INFORMATION SYSTEMS                          20
Hay, Robert M.                    EDUCATION                                    20
Knowles, Randall J.               PUBLICATIONS                                 20
London, Brenda F.                 SOFTWARE DEVELOPMENT                         20
Lovette, Linda L.                 CONTRACTS                                    20
Mong, John V.                     QUALITY ASSURANCE                            20
North, Carolyn N.                 HUMAN RESOURCES                              20
Perry, Samuel R.                  EXECUTIVE                                    20
Weber, Phil H.                    HOST SYSTEMS DEVELOPMENT                     20
```

Using Parentheses and Other Operators

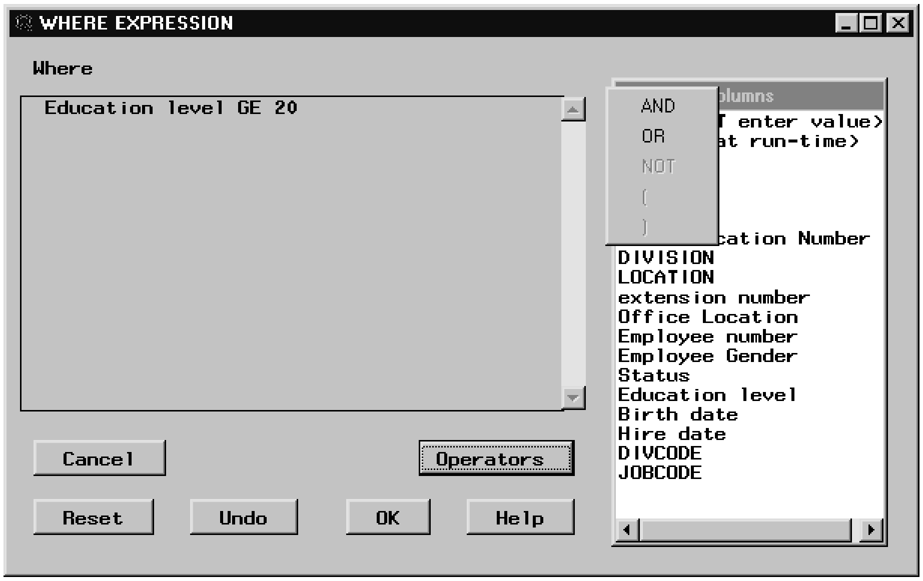
You can use operators other than the comparison operators to subset your data for querying; you can easily change a WHERE condition that has been previously set.

Changing a WHERE Expression

You can change the WHERE expression in SASUSER.PROFILE.EDU20 from the previous example. In the SQL QUERY TABLES window, select

View ► Where Conditions for Subset...

Select **Operators** to display the list of valid operators.



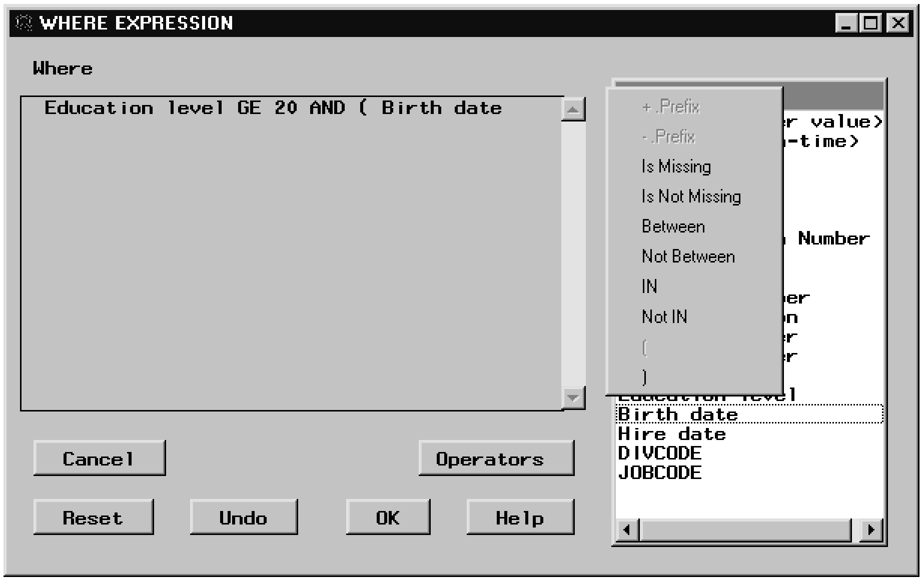
Where

Education level GE 20

AND

Select **AND** from the list.

Select **Operators**. Select **(** from the list. Select **Birth date** from the list of Available Columns. Select **OTHER Operators** from the Operators list to display a second menu of operators.



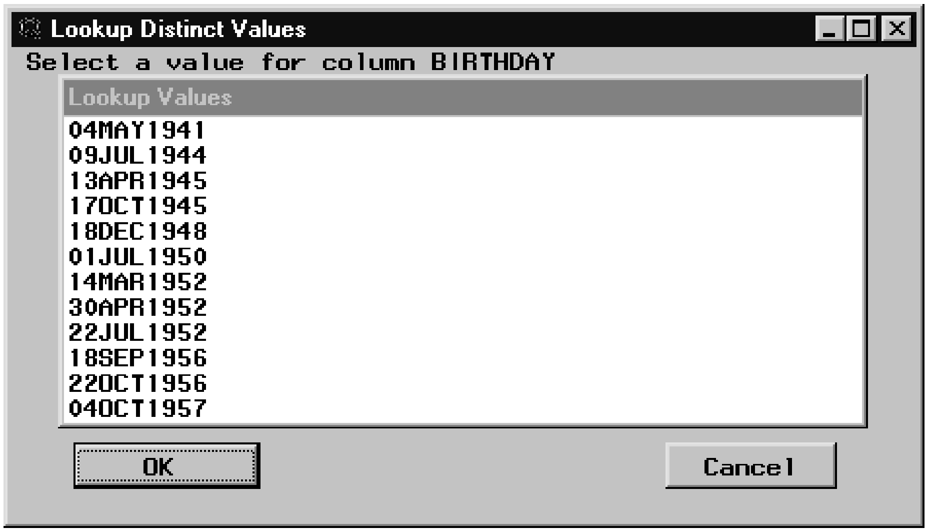
Where

Education level GE 20 AND (Birth date

Between

Select **Between** from the list of **OTHER Operators**.

Select **<LOOKUP distinct values>** from the list of Available Columns.



Lookup Distinct Values

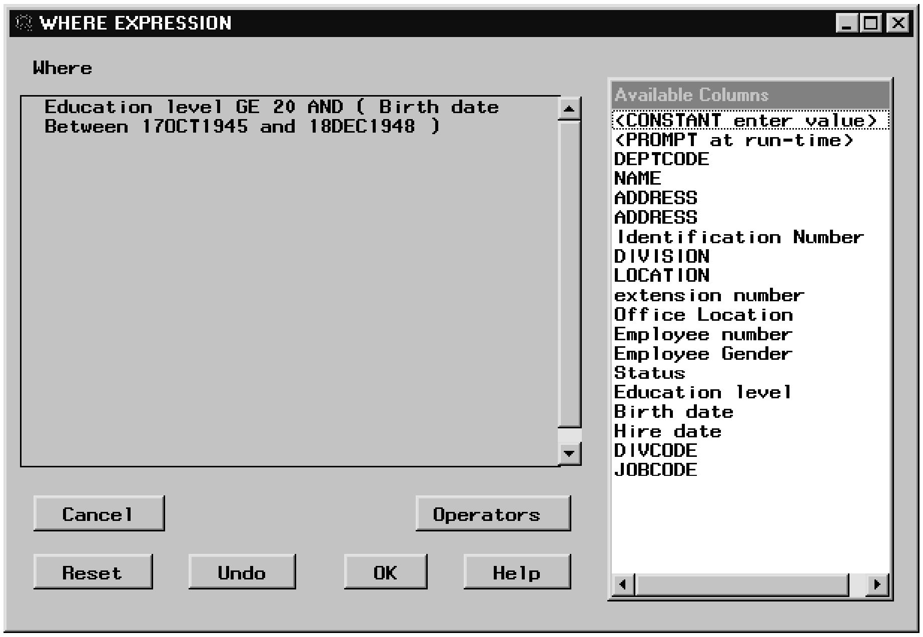
Select a value for column BIRTHDAY

Lookup Values

```
04MAY1941
09JUL1944
13APR1945
17OCT1945
18DEC1948
01JUL1950
14MAR1952
30APR1952
22JUL1952
18SEP1956
22OCT1956
04OCT1957
```

Select **17OCT1945** from the list of Lookup Values. Because the BETWEEN operator requires a second value, the Lookup Distinct Values window appears again after you have selected a value. Select **18DEC1948** from the list of Lookup Values.

In the WHERE EXPRESSION window, select **Operators**. Select **)**from the list of operators to complete the expression that will be evaluated first when the query is run.



WHERE EXPRESSION

Where

```
Education level GE 20 AND ( Birth date
Between 17OCT1945 and 18DEC1948 )
```

Available Columns

```
<CONSTANT enter value>
<PROMPT at run-time>
DEPTCODE
NAME
ADDRESS
ADDRESS
Identification Number
DIVISION
LOCATION
extension number
Office Location
Employee number
Employee Gender
Status
Education level
Birth date
Hire date
DIVCODE
JOBCODE
```

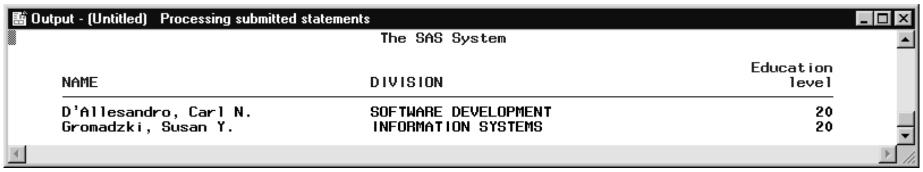
Select **OK** to return to the SQL QUERY COLUMNS window.

Viewing Your Output

Select

Tools ▶ Run Query ▶ Run Immediate

to display the output of your query.



```
                                  The SAS System

                                                                   Education
    NAME                              DIVISION                         level
    -------------------------------------------------------------------------
    D'Allesandro, Carl N.             SOFTWARE DEVELOPMENT                20
    Gromadzki, Susan Y.               INFORMATION SYSTEMS                 20
```

From the SQL QUERY COLUMNS window, select

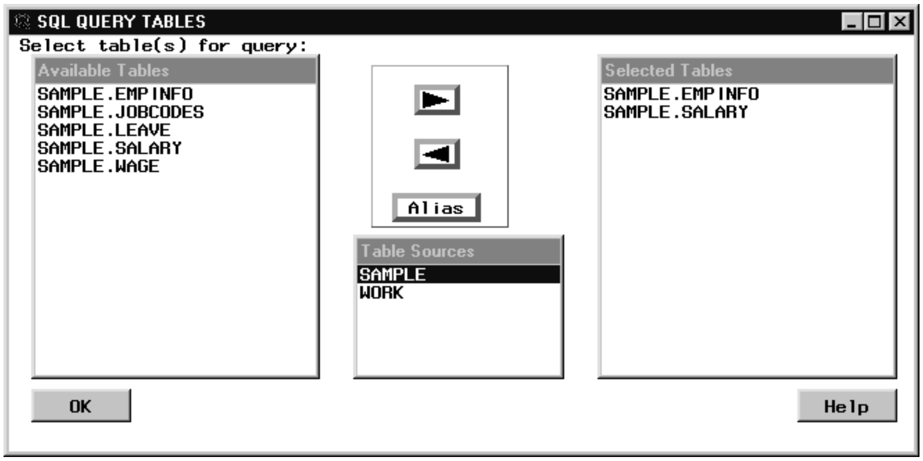
Tools ► Reset

to reset your query.

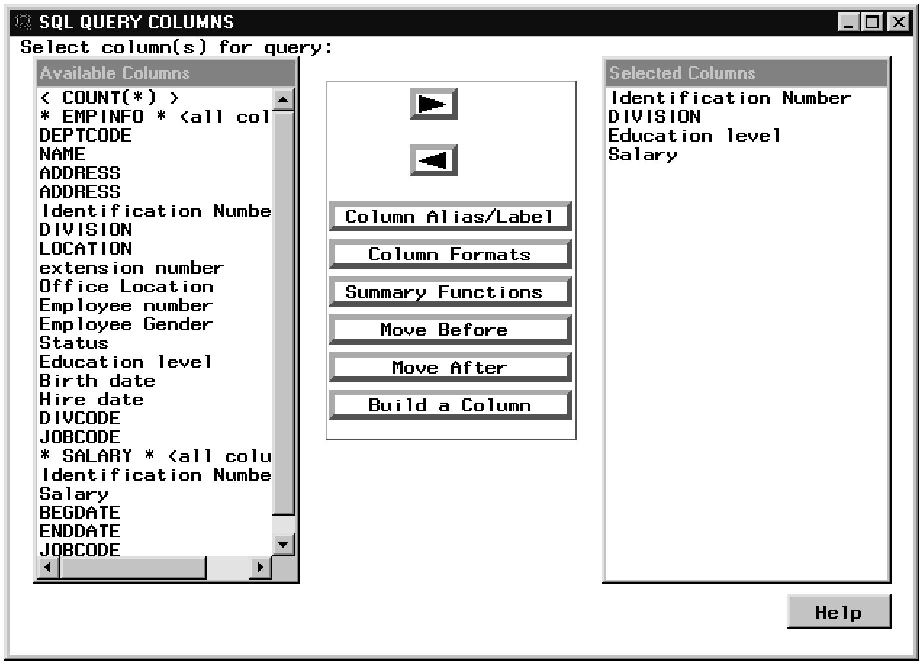
Designing and Saving a Report

When you run your query, you can use the REPORT procedure to modify your output.

In the SQL QUERY window, select **SAMPLE.EMPINFO** and **SAMPLE.SALARY** from the list of Available Tables and add them to the list of Selected Tables.



Select **OK** to display the SQL QUERY COLUMNS window. Select **Identification Number**, **DIVISION**, **Education Level**, and **Salary** from the list of Available Columns.



SQL QUERY COLUMNS

Select column(s) for query:

Available Columns

```
< COUNT(*) >
* EMPINFO * <all col
DEPTCODE
NAME
ADDRESS
ADDRESS
Identification Numbe
DIVISION
LOCATION
extension number
Office Location
Employee number
Employee Gender
Status
Education level
Birth date
Hire date
DIVCODE
JOBCODE
* SALARY * <all colu
Identification Numbe
Salary
BEGDATE
ENDDATE
JOBCODE
```

Selected Columns

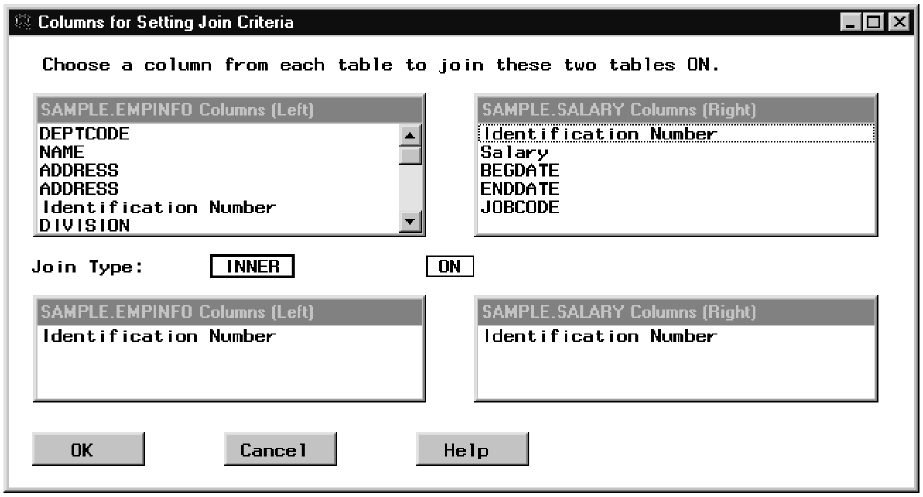
```
Identification Number
DIVISION
Education level
Salary
```

Select

View ▶ Join Type

to create an inner join. Select **Matched Join** from the Join Types window. Select **OK**.

Select **Identification Number** from both columns in the Columns for Setting Join Criteria window.



Columns for Setting Join Criteria

Choose a column from each table to join these two tables ON.

SAMPLE.EMPINFO Columns (Left)

```
DEPTCODE
NAME
ADDRESS
ADDRESS
Identification Number
DIVISION
```

SAMPLE.SALARY Columns (Right)

```
Identification Number
Salary
BEGDATE
ENDDATE
JOBCODE
```

Join Type: INNER ON

SAMPLE.EMPINFO Columns (Left): Identification Number

SAMPLE.SALARY Columns (Right): Identification Number

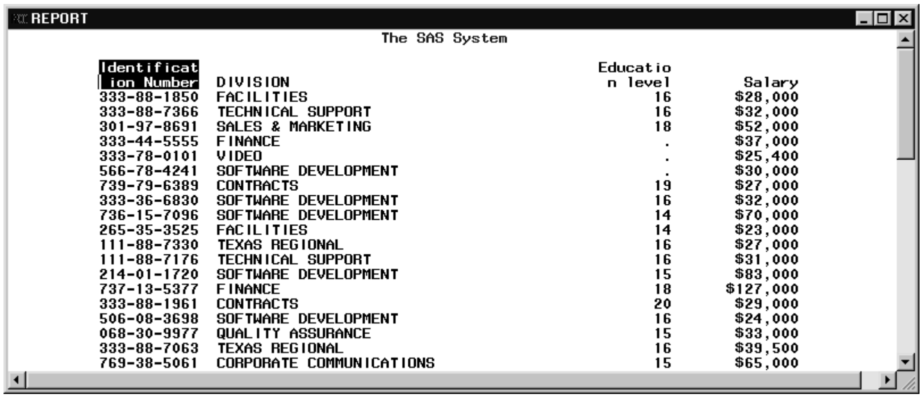
Select **OK**.

Producing Output with the REPORT Procedure

Select

Tools ▶ Run Query ▶ Design a Report ▶ Begin with default report

The output from your query appears in a PROC REPORT window.



```
                                    The SAS System

         Identificat                                          Educatio
          ion Number  DIVISION                                 n level        Salary
         333-88-1850  FACILITIES                                    16       $28,000
         333-88-7366  TECHNICAL SUPPORT                             16       $32,000
         301-97-8691  SALES & MARKETING                             18       $52,000
         333-44-5555  FINANCE                                        .       $37,000
         333-78-0101  VIDEO                                          .       $25,400
         566-78-4241  SOFTWARE DEVELOPMENT                           .       $30,000
         739-79-6389  CONTRACTS                                     19       $27,000
         333-36-6830  SOFTWARE DEVELOPMENT                          16       $32,000
         736-15-7096  SOFTWARE DEVELOPMENT                          14       $70,000
         265-35-3525  FACILITIES                                    14       $23,000
         111-88-7330  TEXAS REGIONAL                                16       $27,000
         111-88-7176  TECHNICAL SUPPORT                             16       $31,000
         214-01-1720  SOFTWARE DEVELOPMENT                          15       $83,000
         737-13-5377  FINANCE                                       18      $127,000
         333-88-1961  CONTRACTS                                     20       $29,000
         506-08-3698  SOFTWARE DEVELOPMENT                          16       $24,000
         068-30-9977  QUALITY ASSURANCE                             15       $33,000
         333-88-7063  TEXAS REGIONAL                                16       $39,500
         769-38-5061  CORPORATE COMMUNICATIONS                      15       $65,000
```

Modifying the Format of Your Report

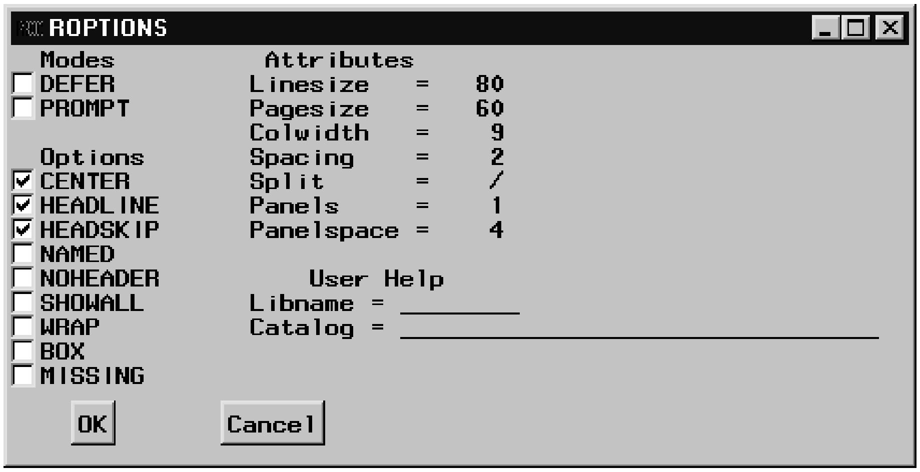
You can now modify your report.

Set Report Options

In the REPORT window, select

Tools ▶ Options ▶ Report

To set the width of the output, in the ROPTIONS window, Type 80 in the **Linesize** field. Type 60 in the **Pagesize** field. Select **HEADLINE** and **HEADSKIP**.



Modes
☐ DEFER
☐ PROMPT

Options
☑ CENTER
☑ HEADLINE
☑ HEADSKIP
☐ NAMED
☐ NOHEADER
☐ SHOWALL
☐ WRAP
☐ BOX
☐ MISSING

Attributes
Linesize = 80
Pagesize = 60
Colwidth = 9
Spacing = 2
Split = /
Panels = 1
Panelspace = 4

User Help
Libname = ________
Catalog = ________

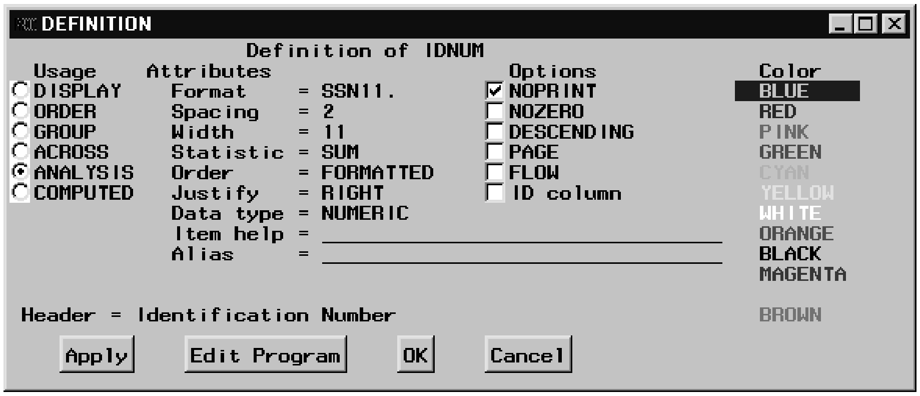
Select **OK**.

Define Selected Item

Select the **Identification Number** heading. Select

Edit ▶ Define

Select **NOPRINT** in the **DEFINITION** window to prevent the identification number from being displayed.



```
                    Definition of IDNUM
  Usage      Attributes                    Options              Color
○ DISPLAY      Format    = SSN11.        ☑ NOPRINT              BLUE
○ ORDER        Spacing   = 2             □ NOZERO               RED
○ GROUP        Width     = 11            □ DESCENDING           PINK
○ ACROSS       Statistic = SUM           □ PAGE                 GREEN
● ANALYSIS     Order     = FORMATTED     □ FLOW                 CYAN
○ COMPUTED     Justify   = RIGHT         □ ID column            YELLOW
               Data type = NUMERIC                              WHITE
               Item help = ________________________________     ORANGE
               Alias     = ________________________________     BLACK
                                                                MAGENTA

 Header = Identification Number                                 BROWN
```

Select **OK**.

Move Selected Item

In the REPORT window, select **Education Level**. Select

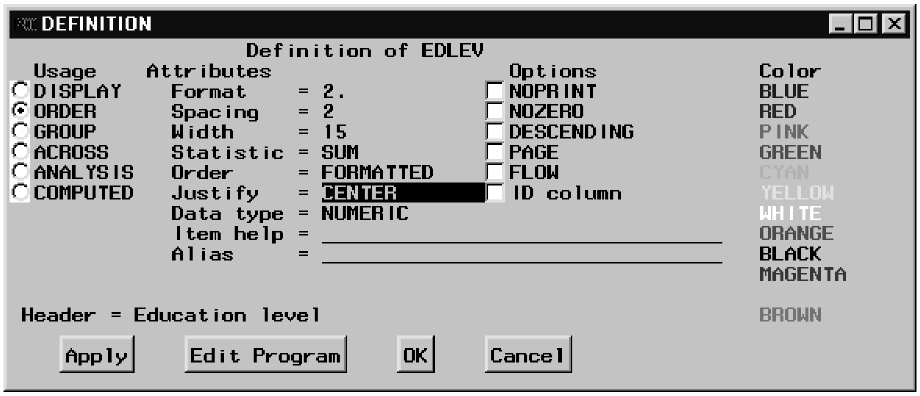
Edit ▶ Move ▶ Left of next selected item.

Select the **DIVISION** heading in the REPORT window. Education Level will appear as the first column in the window.

Select the **Education Level** in the REPORT window. Select

Edit ▶ Define

In the DEFINITION window, select **ORDER**. Type **2.** in the Format field. Type **15** in the Width field. Type **CENTER** in the Justify field.



```
                    Definition of EDLEV
  Usage      Attributes                    Options              Color
○ DISPLAY      Format    = 2.            □ NOPRINT              BLUE
● ORDER        Spacing   = 2             □ NOZERO               RED
○ GROUP        Width     = 15            □ DESCENDING           PINK
○ ACROSS       Statistic = SUM           □ PAGE                 GREEN
○ ANALYSIS     Order     = FORMATTED     □ FLOW                 CYAN
○ COMPUTED     Justify   = CENTER        □ ID column            YELLOW
               Data type = NUMERIC                              WHITE
               Item help = ________________________________     ORANGE
               Alias     = ________________________________     BLACK
                                                                MAGENTA

 Header = Education level                                       BROWN
```

Select **OK**.

Select the **SALARY** heading in the REPORT window. Select

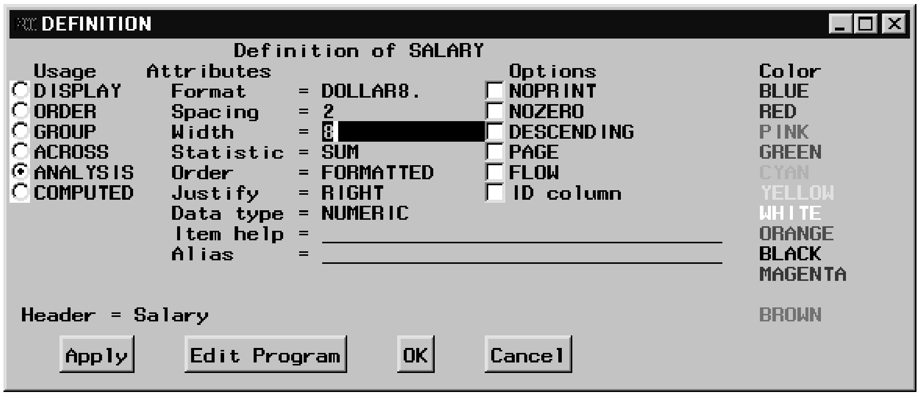
Edit ▶ Move ▶ Left of next selected item

Select the **Division** heading. Salary will appear as the second column in the window.

Select the **Salary** heading. Select

Edit ▶ Define

In the DEFINITION window, type **DOLLAR8.** in the Format field. Type **8** in the Width field.



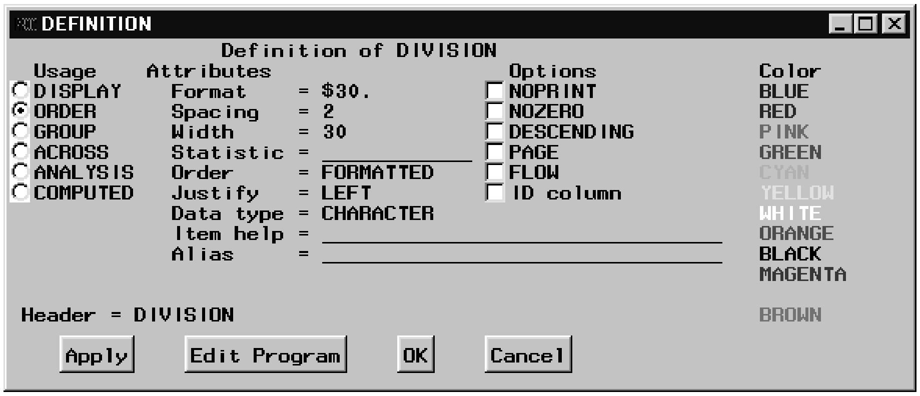
```
                    Definition of SALARY
 Usage    Attributes                    Options            Color
○ DISPLAY   Format     = DOLLAR8.       □ NOPRINT          BLUE
○ ORDER     Spacing    = 2              □ NOZERO           RED
○ GROUP     Width      = 8              □ DESCENDING       PINK
○ ACROSS    Statistic  = SUM            □ PAGE             GREEN
● ANALYSIS  Order      = FORMATTED      □ FLOW             CYAN
○ COMPUTED  Justify    = RIGHT          □ ID column        YELLOW
            Data type  = NUMERIC                           WHITE
            Item help  = ________________________________  ORANGE
            Alias      = ________________________________  BLACK
                                                           MAGENTA

Header = Salary                                            BROWN
```

Select **OK**.

Select the **DIVISION** heading in the REPORT window. Select

Edit ▶ Define

Type **$30.** in the **Format** field of the DEFINITION window. Type **30** in the **Width** field.

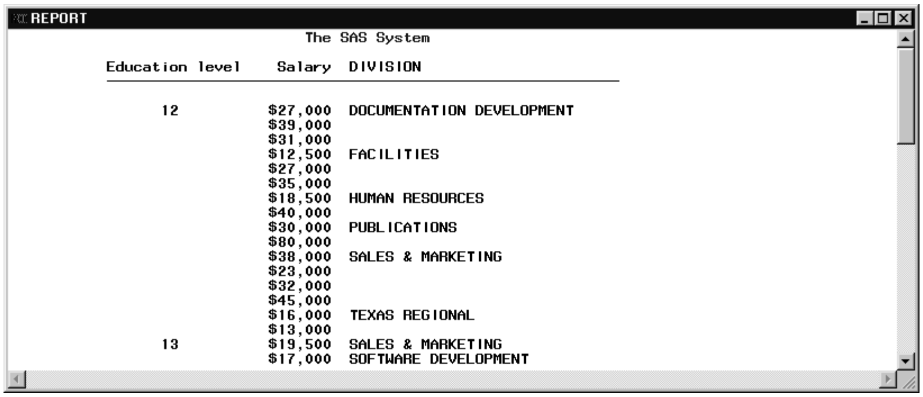


```
                    Definition of DIVISION
 Usage    Attributes                    Options            Color
○ DISPLAY   Format     = $30.           □ NOPRINT          BLUE
● ORDER     Spacing    = 2              □ NOZERO           RED
○ GROUP     Width      = 30             □ DESCENDING       PINK
○ ACROSS    Statistic  = ____________   □ PAGE             GREEN
○ ANALYSIS  Order      = FORMATTED      □ FLOW             CYAN
○ COMPUTED  Justify    = LEFT           □ ID column        YELLOW
            Data type  = CHARACTER                         WHITE
            Item help  = ________________________________  ORANGE
            Alias      = ________________________________  BLACK
                                                           MAGENTA

Header = DIVISION                                          BROWN
```

Select **OK**.

The Formatted Report

Your completed report compares the salaries of different divisions for employees with the same education level.

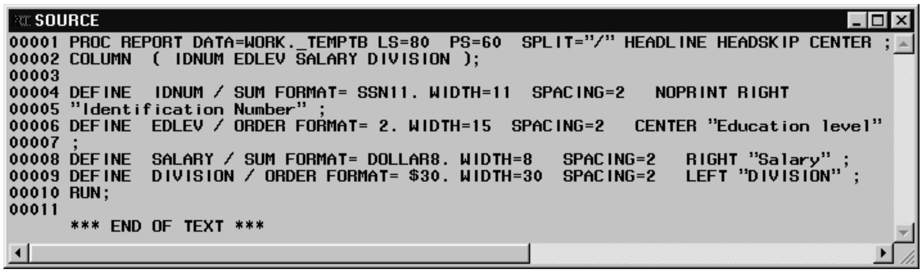


```
                              The SAS System

        Education level    Salary  DIVISION
        ___________________________________________________

              12          $27,000  DOCUMENTATION DEVELOPMENT
                          $39,000
                          $31,000
                          $12,500  FACILITIES
                          $27,000
                          $35,000
                          $18,500  HUMAN RESOURCES
                          $40,000
                          $30,000  PUBLICATIONS
                          $80,000
                          $38,000  SALES & MARKETING
                          $23,000
                          $32,000
                          $45,000
                          $16,000  TEXAS REGIONAL
                          $13,000
              13          $19,500  SALES & MARKETING
                          $17,000  SOFTWARE DEVELOPMENT
```

Viewing the Report Statements

You can view your report statements in the SOURCE window by selecting

Tools ► REPORT Statements



```
PROC REPORT DATA=WORK._TEMPTB LS=80  PS=60  SPLIT="/" HEADLINE HEADSKIP CENTER ;
COLUMN  ( IDNUM EDLEV SALARY DIVISION );

DEFINE  IDNUM / SUM FORMAT= SSN11. WIDTH=11  SPACING=2   NOPRINT RIGHT
"Identification Number" ;
DEFINE  EDLEV / ORDER FORMAT= 2. WIDTH=15  SPACING=2   CENTER "Education level"
;
DEFINE  SALARY / SUM FORMAT= DOLLAR8. WIDTH=8   SPACING=2   RIGHT "Salary" ;
DEFINE  DIVISION / ORDER FORMAT= $30. WIDTH=30  SPACING=2   LEFT "DIVISION" ;
RUN;

     *** END OF TEXT ***
```

Select

File ► End

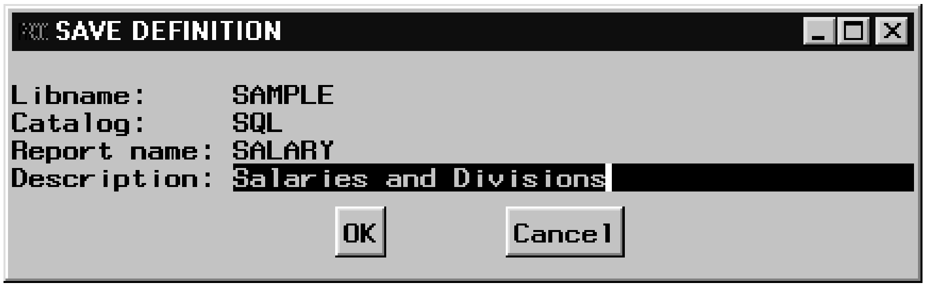
to close the SOURCE window and return to the REPORT window.

Saving Your Report

You can save your customized report to a catalog entry for use with later queries by selecting

File ► Save Report...

to display the SAVE DEFINITION window. Type SAMPLE in the **`Libname`** field. Type SQL in the **`Catalog:`** field. Type SALARY in the **`Report Name:`** field. Type Salaries and Divisions in the **`Description:`** field.



Libname: SAMPLE
Catalog: SQL
Report name: SALARY
Description: Salaries and Divisions

Select **`OK`**. A dialog window appears that notifies you of the creation of a new catalog. Select **`OK`**.

Select

File ► Close

to exit the REPORT window. Select **`OK`** in the dialog window that appears.

You can also save your report definition in the SQL Query Window when you save the query.

Use Definition from Last Report

You can use your customized report definition. In the SQL QUERY COLUMNS window, select

Tools ▶ Run Query ▶ Design a Report ▶ Use definition from last Report

The results of the query are presented using your predefined report.

Creating Summary Reports

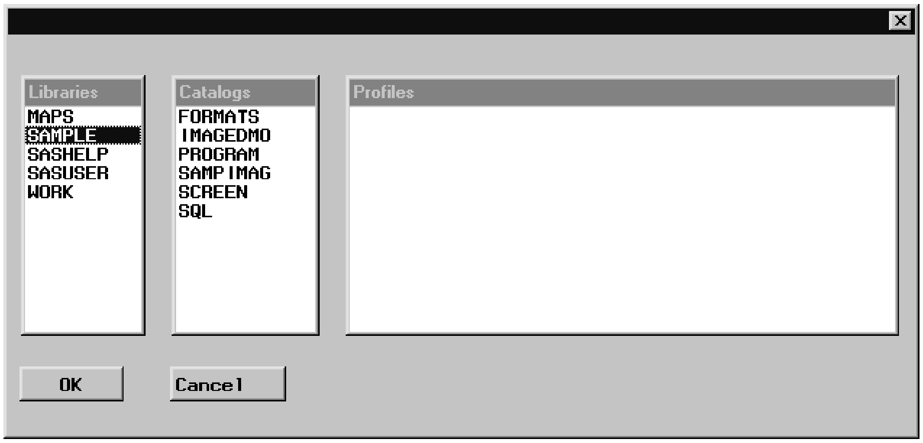
You can use the SQL QUERY Window in conjunction with the REPORT Procedure to create a summary report with totals.

Using a Saved Report Definition

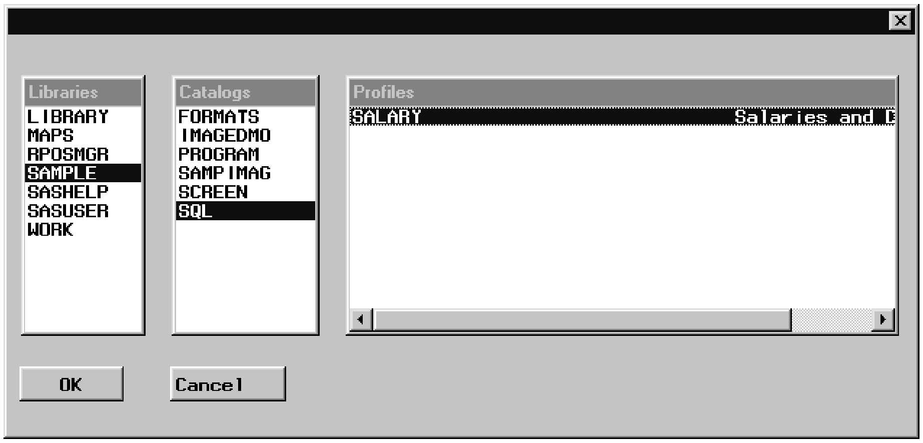
For this example you are modifying the report that you created in the previous example to display the total salaries for each division. In the SQL QUERY COLUMNS window, select

Tools ▶ Run Query ▶ Design a Report ▶ Name a predefined Report...

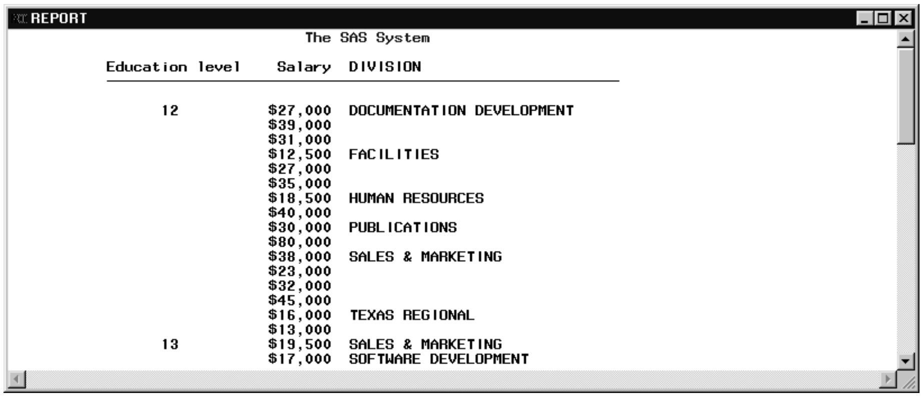
When the dialog displays, select **SAMPLE** from the list of libraries displayed. The libraries and catalogs that are listed in your display may differ from the ones in the example.



Select **SQL** from the list of catalogs. Select the **SALARY** report definition.



Click **OK**.



```
                         The SAS System

 Education level    Salary  DIVISION
 ------------------------------------------------------------

       12          $27,000  DOCUMENTATION DEVELOPMENT
                   $39,000
                   $31,000
                   $12,500  FACILITIES
                   $27,000
                   $35,000
                   $18,500  HUMAN RESOURCES
                   $40,000
                   $30,000  PUBLICATIONS
                   $80,000
                   $38,000  SALES & MARKETING
                   $23,000
                   $32,000
                   $45,000
                   $16,000  TEXAS REGIONAL
                   $13,000
       13          $19,500  SALES & MARKETING
                   $17,000  SOFTWARE DEVELOPMENT
```

Deleting a Heading

You will not need to display education level for this report. In the REPORT window, select the EDUCATION LEVEL heading. Select

Edit ► Delete

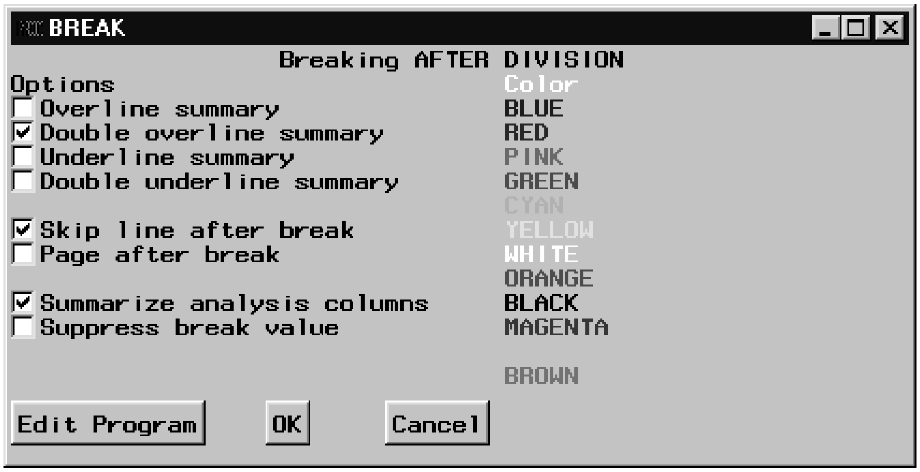
to delete the EDUCATION LEVEL heading from the report. You are not deleting EDUCATION LEVEL from the query.

Summarizing Information

Select the DIVISION heading. Select

Edit ► Summarize information ► After Item

to display the BREAK window. Select **Double overline summary** to print a double line over the summary total. Select **Skip line after break**. Select **Summarize analysis columns**.



Breaking AFTER DIVISION

Options

□ Overline summary
☑ Double overline summary
□ Underline summary
□ Double underline summary

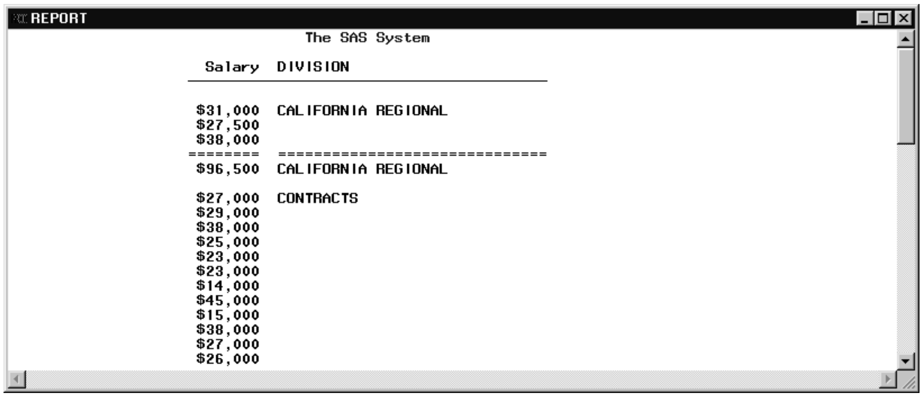
☑ Skip line after break
□ Page after break

☑ Summarize analysis columns
□ Suppress break value

Color
BLUE
RED
PINK
GREEN
CYAN
YELLOW
WHITE
ORANGE
BLACK
MAGENTA

BROWN

Select **OK** to return to the REPORT window and display the total salaries for each division.



```
                The SAS System

  Salary  DIVISION
 --------------------------------

 $31,000  CALIFORNIA REGIONAL
 $27,500
 $38,000
========  ==============================
 $96,500  CALIFORNIA REGIONAL

 $27,000  CONTRACTS
 $29,000
 $38,000
 $25,000
 $23,000
 $23,000
 $14,000
 $45,000
 $15,000
 $38,000
 $27,000
 $26,000
```

Select

File ► Close

Select **OK** in the pop-up dialog to close the REPORT window and to return to the SQL QUERY TABLES window.

Select

Tools ► Reset

to reset the query and return to the SQL QUERY COLUMNS window.

Counting and Grouping Data Automatically

You can count and report the total number of rows that have the same value for one or more columns. You can use the automatic group by feature to group the values according to their columns.

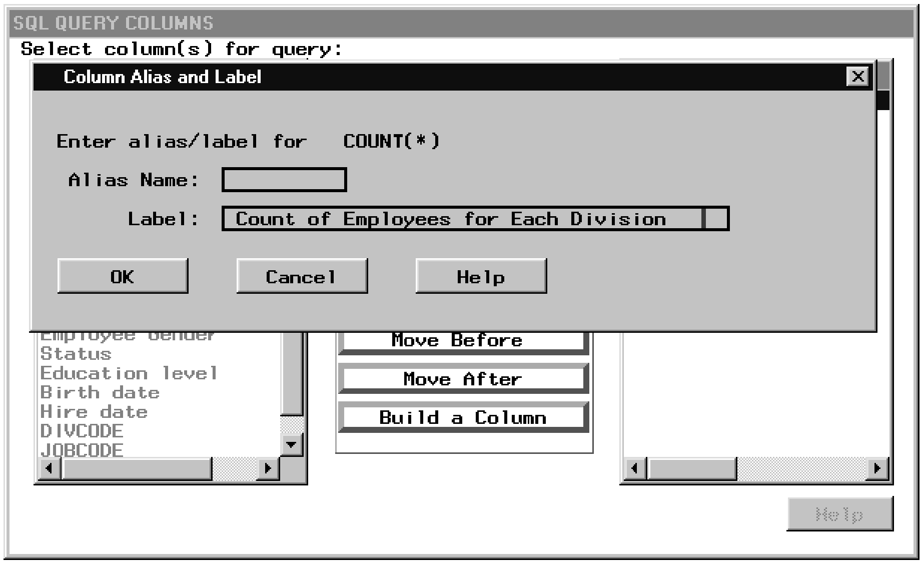
The following query displays the number of employees in each division.

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

In the SQL QUERY COLUMNS window, select **DIVISION** and **< COUNT(*) >** from the Available Columns list and add them to the list of Selected Columns.

Count

Select **COUNT(*)** from the Selected Columns List. Select **Move After** to move the column. Reselect **COUNT (*)**. Select **COLUMN ALIAS/LABEL**. Type **Count of Employees for Each Division** in the LABEL field of the Column Alias and Label window.



SQL QUERY COLUMNS

Select column(s) for query:

Column Alias and Label

Enter alias/label for COUNT(*)

Alias Name:

Label: Count of Employees for Each Division

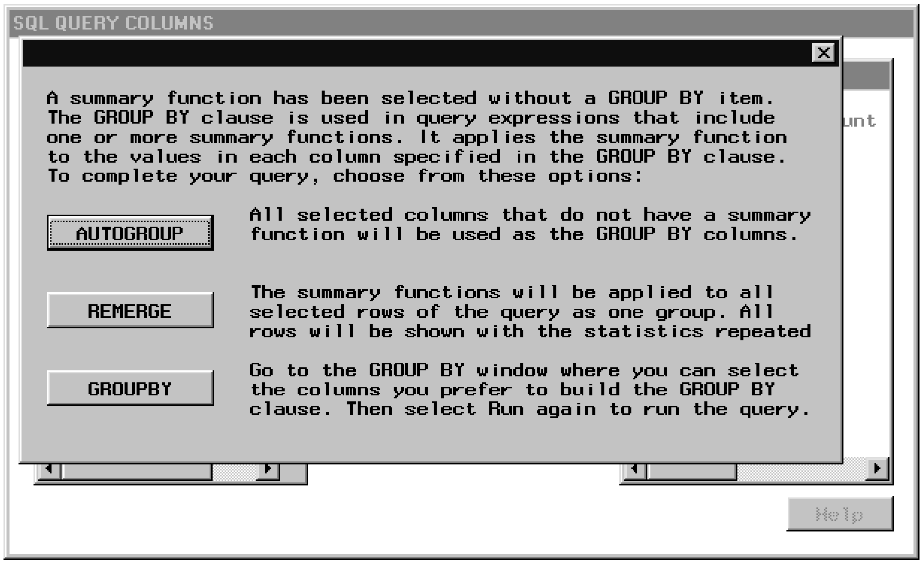
Select **OK**.

Grouping Columns Automatically

Select

Tools ► Run Query ► Run Immediate

A dialog is displayed.



SQL QUERY COLUMNS

A summary function has been selected without a GROUP BY item. The GROUP BY clause is used in query expressions that include one or more summary functions. It applies the summary function to the values in each column specified in the GROUP BY clause. To complete your query, choose from these options:

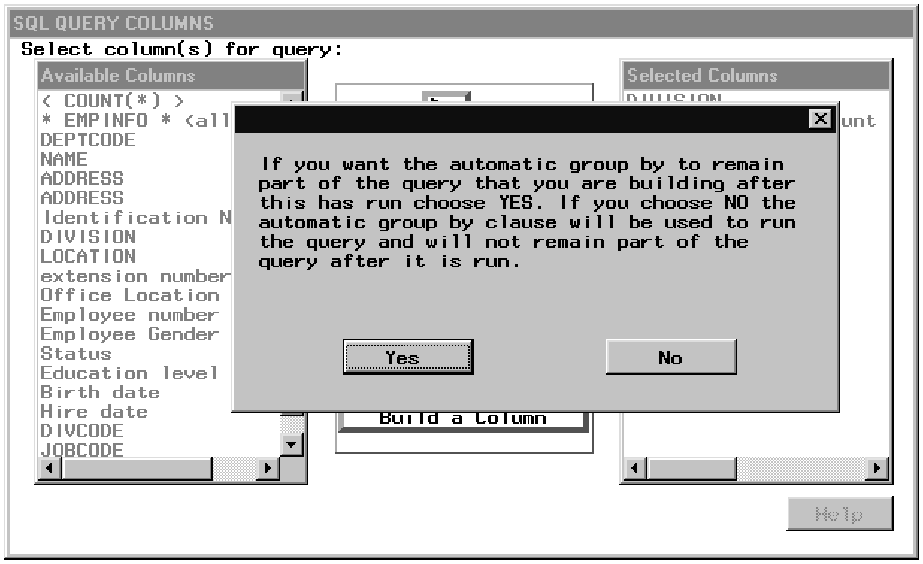
AUTOGROUP All selected columns that do not have a summary function will be used as the GROUP BY columns.

REMERGE The summary functions will be applied to all selected rows of the query as one group. All rows will be shown with the statistics repeated

GROUPBY Go to the GROUP BY window where you can select the columns you prefer to build the GROUP BY clause. Then select Run again to run the query.

Select **AUTOGROUP** to automatically select the correct columns. Selected columns that do not have summary functions applied to them will be the group(s) that the summary functions are computed for.

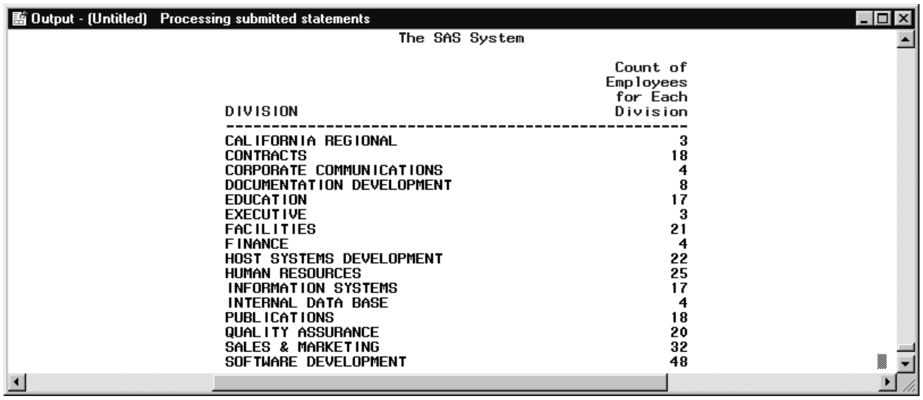
A second dialog is displayed.



If you want the automatic group by to remain part of the query that you are building after this has run choose YES. If you choose NO the automatic group by clause will be used to run the query and will not remain part of the query after it is run.

Select **NO**. The automatic Group By clause will be part of the query syntax while the query runs, but will not be retained. You can select or remove columns after the query is executed and use **AUTOGROUP** to automatically select the columns again.

The count of employees for each division is displayed in the OUTPUT window.

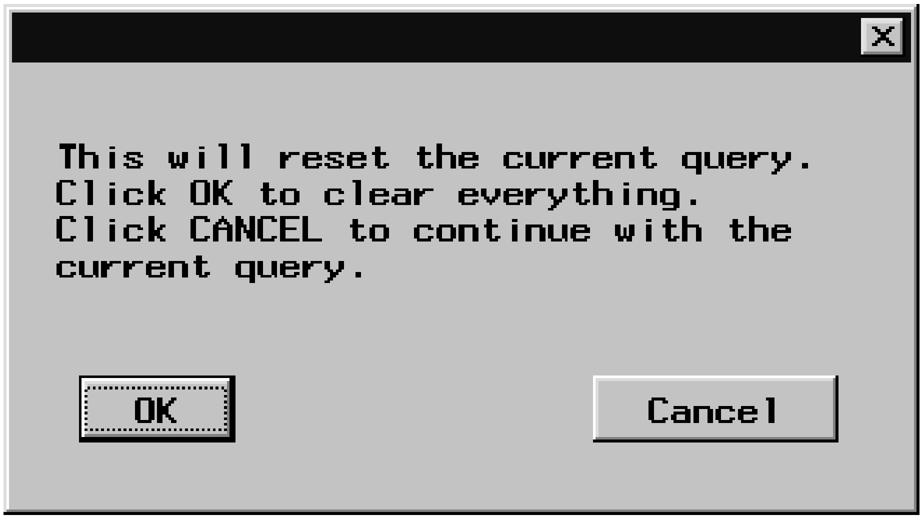


```
                                   The SAS System

                                                       Count of
                                                      Employees
                                                       for Each
                  DIVISION                             Division
                  --------------------------------------------------
                  CALIFORNIA REGIONAL                          3
                  CONTRACTS                                   18
                  CORPORATE COMMUNICATIONS                     4
                  DOCUMENTATION DEVELOPMENT                    8
                  EDUCATION                                   17
                  EXECUTIVE                                    3
                  FACILITIES                                  21
                  FINANCE                                      4
                  HOST SYSTEMS DEVELOPMENT                    22
                  HUMAN RESOURCES                             25
                  INFORMATION SYSTEMS                         17
                  INTERNAL DATA BASE                           4
                  PUBLICATIONS                                18
                  QUALITY ASSURANCE                           20
                  SALES & MARKETING                           32
                  SOFTWARE DEVELOPMENT                        48
```

In the SQL QUERY COLUMNS window, select

Tools ► Reset

to reset your query. A dialog appears.



This will reset the current query.
Click OK to clear everything.
Click CANCEL to continue with the
current query.

Select **OK** to return to the SQL QUERY TABLES window. The components of the current query are cleared.

Automatic Group By with More than One Table

The next query joins two tables to display the number of employees for each job title. The JOBCODES table contains the job title for each job code.

Select **SAMPLE.JOBCODES** and **SAMPLE.EMPINFO** from the list of Available Tables and add them to the list of Selected Tables.

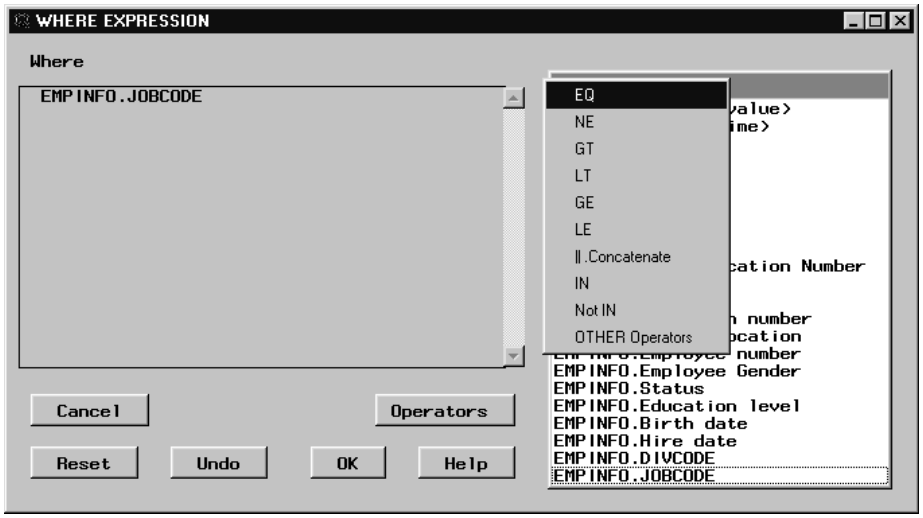
Select **OK**.

In the SQL QUERY COLUMNS window, select **TITLE** and **< COUNT(*) >** from the list of Available Columns and add them to the list of Selected Columns.

Select

View ▶ Where Conditions for Subset...

In the WHERE EXPRESSION window, select **EMPINFO.JOBCODE** from the Available Columns list. Select **EQ** from the list of comparison operators.

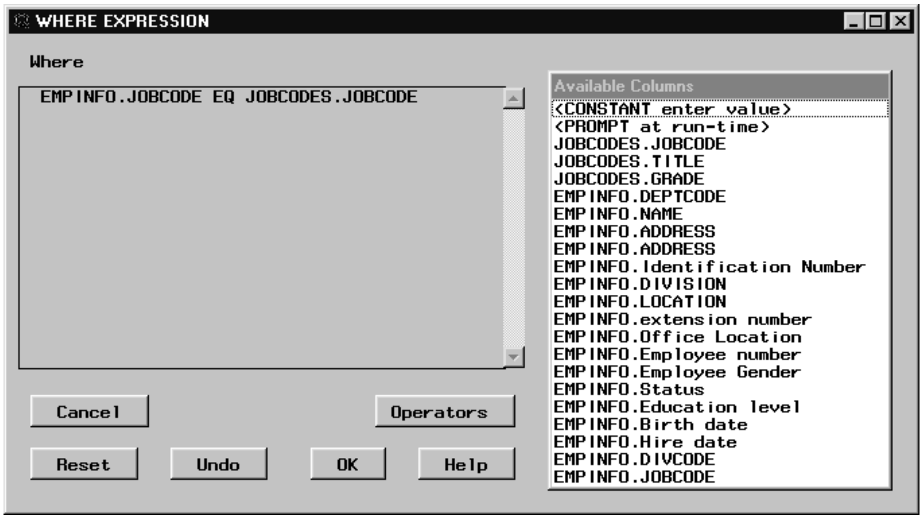


WHERE EXPRESSION

Where

EMPINFO.JOBCODE

Select **JOBCODES.JOBCODE** from the Available Columns List.

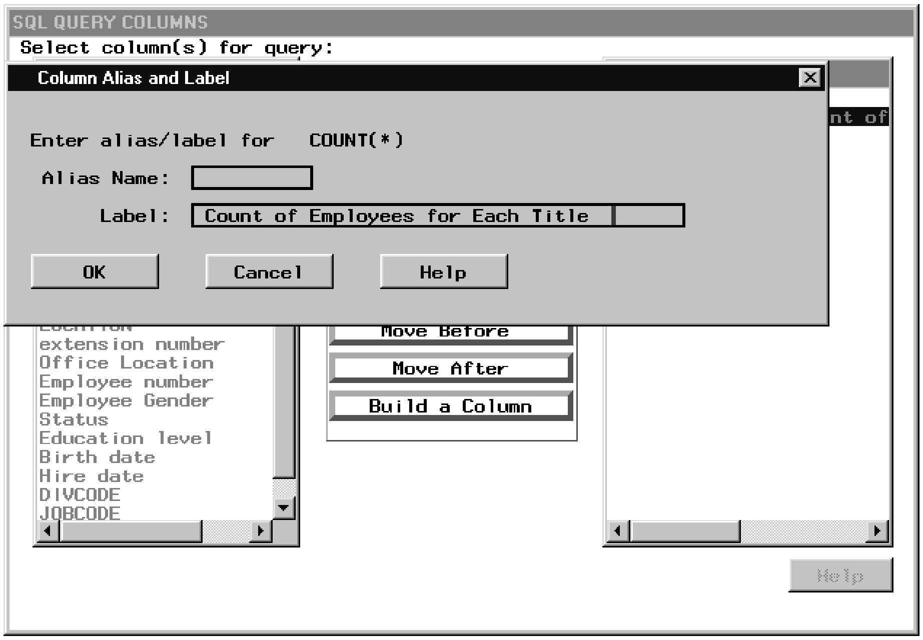


Where

EMPINFO.JOBCODE EQ JOBCODES.JOBCODE

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

Select **COUNT(*)** from the Selected Columns List. Select **Move After** to move the column. Reselect **COUNT (*)**. Select **Column Alias/Label**. Type **Count of Employees for Each Title** in the LABEL field of the Column Alias and Label window.



Enter alias/label for COUNT(*)

Alias Name:

Label: Count of Employees for Each Title

Select **OK**.

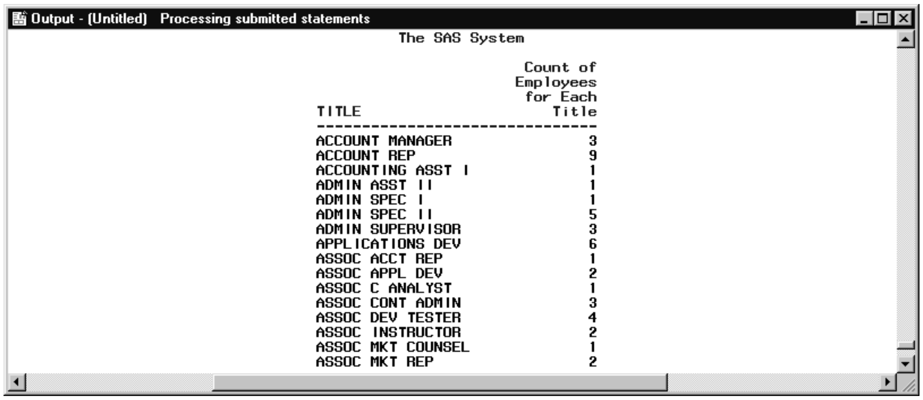
Retaining an Automatic Group By as Part of a Query

Select

Tools ► Run Query ► Run Immediate

A dialog is displayed. Select **AUTOGROUP** in the dialog window to use JOBCODES.TITLE as the Group By column. A second dialog is displayed. Select **YES** in the second dialog window to retain the Group By column as part of the query.

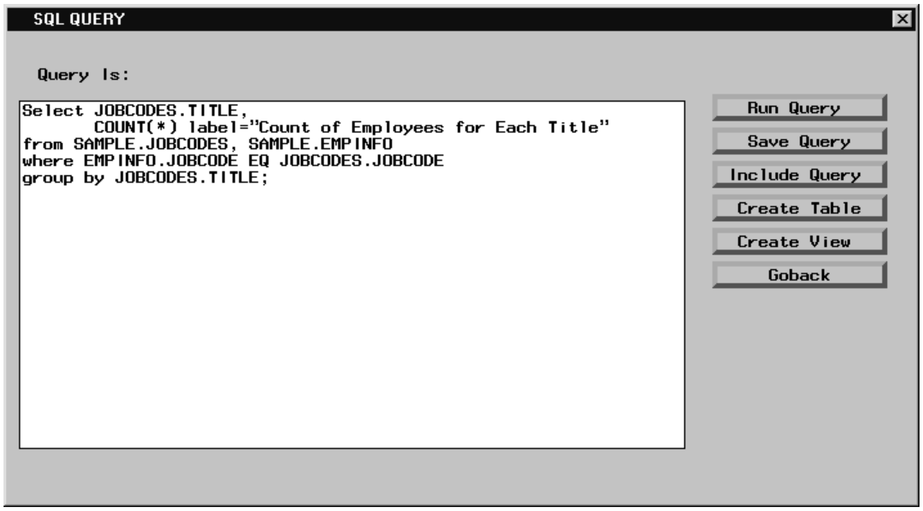
The OUTPUT window displays the number of employees for each job title.



```
                           The SAS System

                                         Count of
                                        Employees
                                         for Each
                     TITLE                  Title
                     ------------------------------
                     ACCOUNT MANAGER            3
                     ACCOUNT REP                9
                     ACCOUNTING ASST I          1
                     ADMIN ASST II              1
                     ADMIN SPEC I               1
                     ADMIN SPEC II              5
                     ADMIN SUPERVISOR           3
                     APPLICATIONS DEV           6
                     ASSOC ACCT REP             1
                     ASSOC APPL DEV             2
                     ASSOC C ANALYST            1
                     ASSOC CONT ADMIN           3
                     ASSOC DEV TESTER           4
                     ASSOC INSTRUCTOR           2
                     ASSOC MKT COUNSEL          1
                     ASSOC MKT REP              2
```

In the SQL QUERY COLUMNS window, select

Tools ▶ Show Query...



```
Select JOBCODES.TITLE,
       COUNT(*) label="Count of Employees for Each Title"
from SAMPLE.JOBCODES, SAMPLE.EMPINFO
where EMPINFO.JOBCODE EQ JOBCODES.JOBCODE
group by JOBCODES.TITLE;
```

The automatic Group By will be retained as part of the query syntax when the query is run again, saved, or used to create a table or view. Select **Goback** to return to the SQL QUERY COLUMNS window.

In the SQL QUERY COLUMNS window, select

File ▶ Save Query ▶ Save as Query to Include later

In the **Entry Name** field, type COUNTS as the name of the query. In the **Enter a description for the query field** type COUNT OF EMPINFO BY TITLE. Select **OK** to save the query and return to the SQL QUERY COLUMNS window.

Select

View ▶ Tables...

to return to the SQL QUERY TABLES window. Remove SALARY.JOBCODES from the list of Selected Tables. Select **OK** in the pop-up dialog.

Summarizing Groups of Data

Summary functions produce a statistical summary of a table or of group(s) of data. The following example displays the minimum, average, and maximum level of education within each division. Use the GROUPBY clause and a summary function to summarize information about a group of data. If you omit a GROUPBY, one summary value is produced for the entire table.

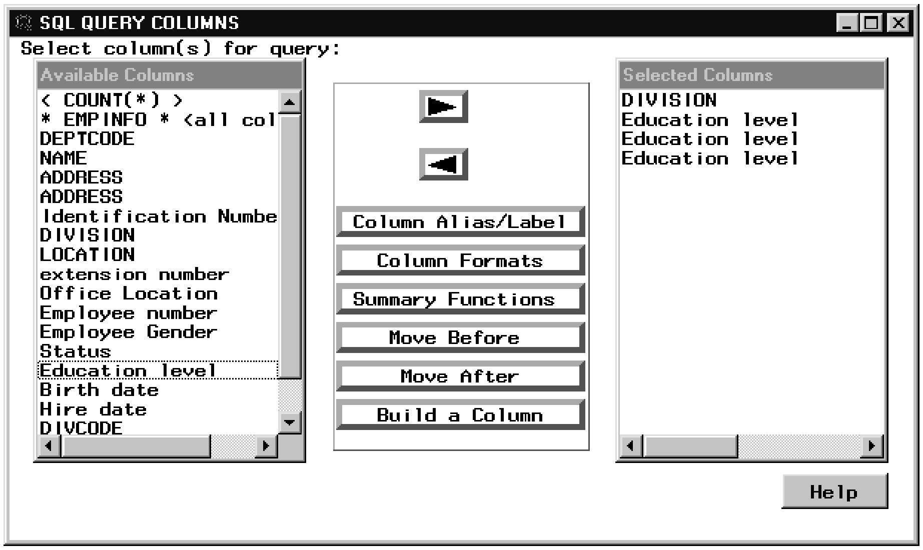
Summary Functions

The list of Selected Tables in the SQL QUERY TABLES window contains SAMPLE.EMPINFO from the previous example. Select **OK**.

In the SQL QUERY COLUMNS window, remove **COUNT(*)** from the list of Selected Columns. Select **DIVISION** and **Education level** from the Available Columns list and add them to the list of Selected Columns.

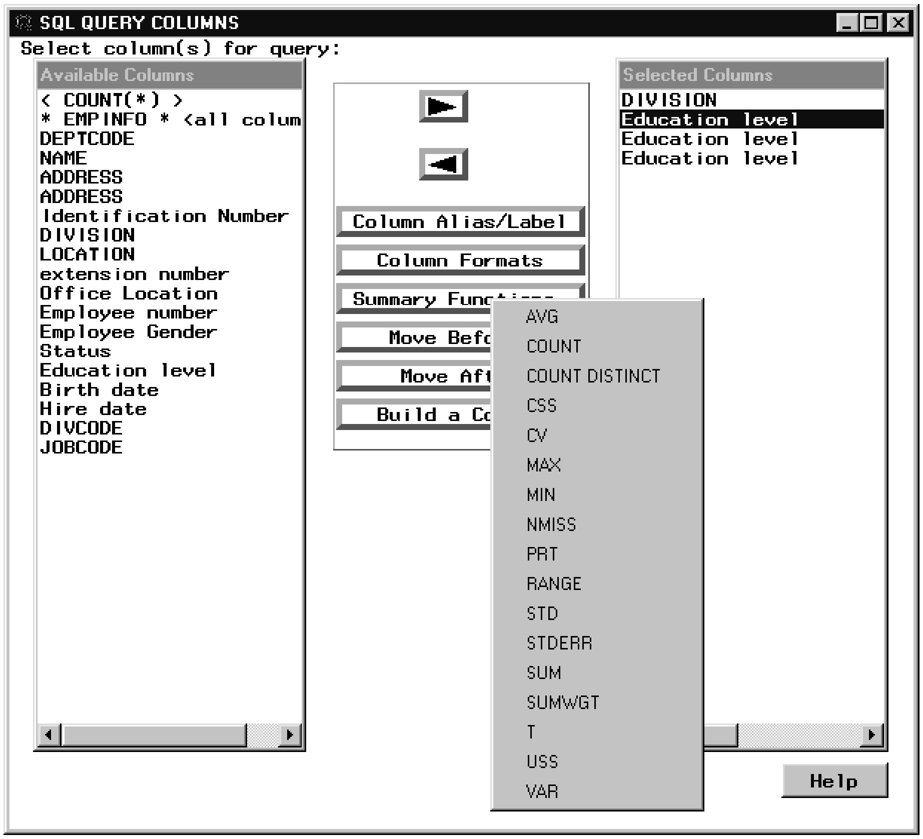
Select **Education level** a second time from the Available Columns List and add it to the list of Selected Columns.

Select **Education level** a third time from the Available Columns list and add it to the list of Selected Columns.



Selected Columns
DIVISION
Education level
Education level
Education level

Select the first **Education level** from the Selected Columns list. Select **Summary Functions**.

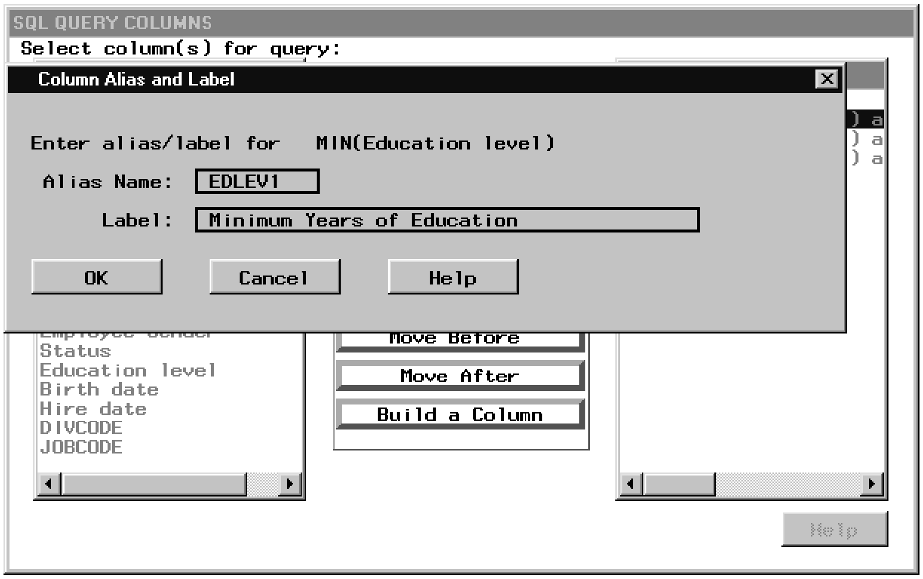


Select **MIN** from the list of Summary Functions. A summary function is applied to the selected column and a default unique column alias is automatically generated. The summary function and the selected column name are automatically set as the label. You can use this default label in the report, or you can set a new alias or label.

Select the second **Education level** from the Selected Columns List. Select **Summary Functions**. Select **AVG** from the list of Summary Functions.

Select the third **Education level** from the Selected Columns list. Select **Summary Functions**. Select **MAX** from the list of Summary Functions.

Select the first **Education level** from the Selected Columns List. Select **Column Alias/Label**. Type **Minimum Years of Education** in the LABEL field of the Column Alias and Label window.



SQL QUERY COLUMNS

Select column(s) for query:

Column Alias and Label

Enter alias/label for MIN(Education level)

Alias Name: EDLEV1

Label: Minimum Years of Education

Select **OK**.

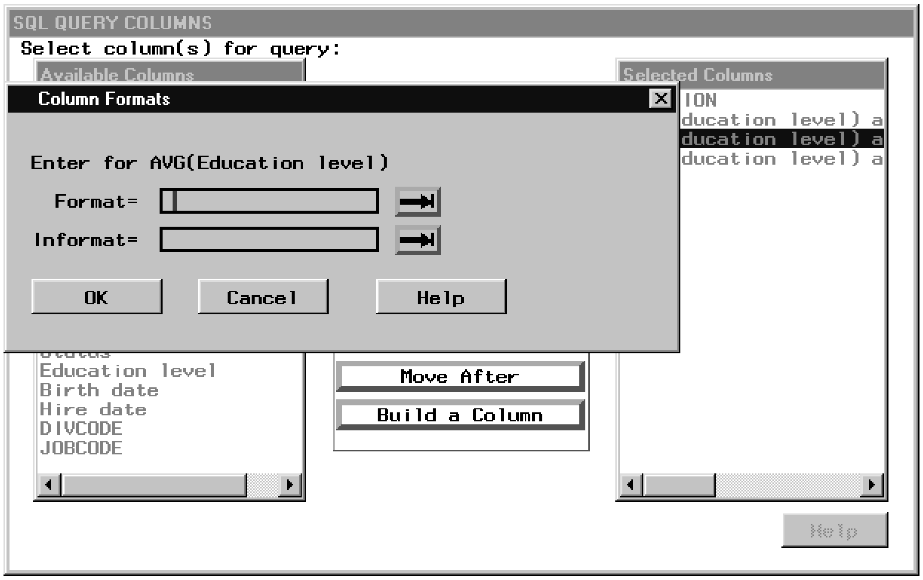
Select the second **Education level** from the Selected Columns List. Select **Column Alias/Label**. Type **Average Years of Education** in the LABEL field of the Column Alias and Label window.

Select **OK**.

Select the third **Education level** from the Selected Columns List. Select **Column Alias/Label**. Type **Maximum Years of Education** in the LABEL field of the Column Alias and Label window.

Select **OK**.

Select the second **Education level** from the Selected Columns List. Select **Column Formats**.



SQL QUERY COLUMNS

Select column(s) for query:

Column Formats

Enter for AVG(Education level)

Format=

Informat=

Type **comma4.0** in the Format= field. Select **OK**.

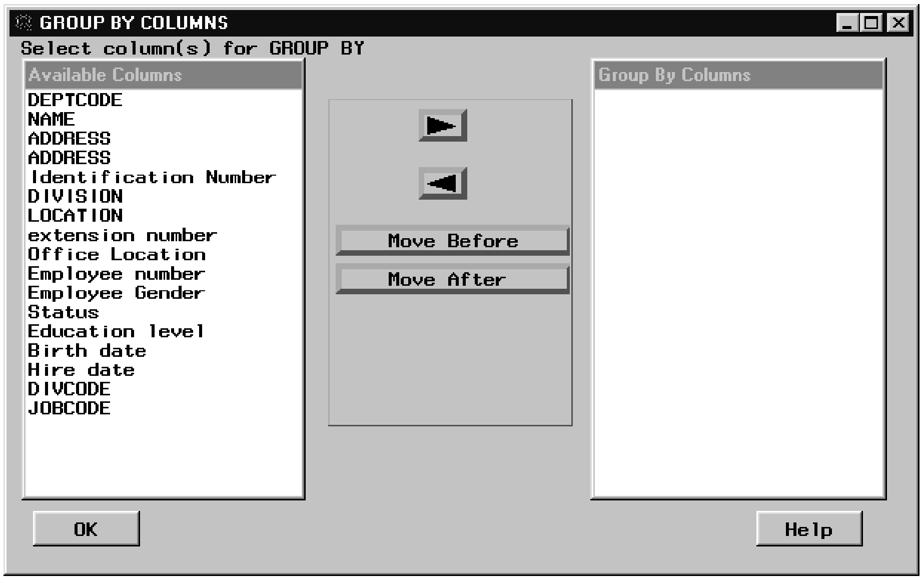
Select

Tools ► Run Immediate

A dialog is displayed.

Group By Columns

Select **Group(s) for Summary Functions** to display the GROUP BY COLUMNS window.



Select column(s) for GROUP BY

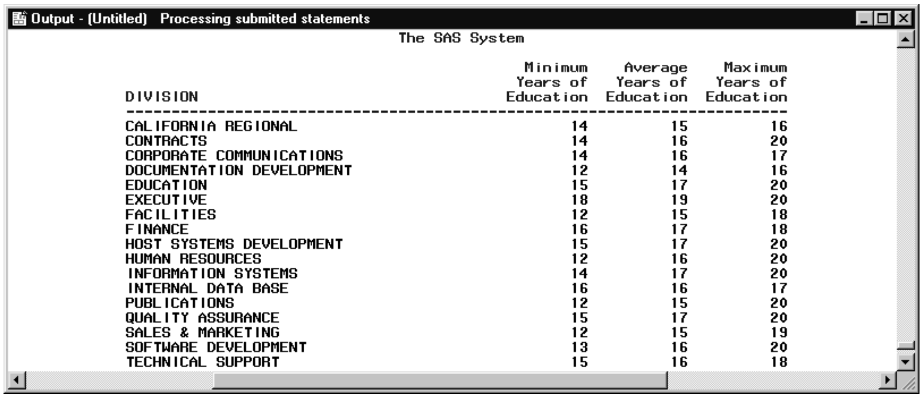
```
DEPTCODE
NAME
ADDRESS
ADDRESS
Identification Number
DIVISION
LOCATION
extension number
Office Location
Employee number
Employee Gender
Status
Education level
Birth date
Hire date
DIVCODE
JOBCODE
```

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

Select

Tools ► Run Immediate

The maximum, minimum, and average education levels for each division are displayed in the OUTPUT window.



```
                                 The SAS System

                                             Minimum    Average    Maximum
                                            Years of   Years of   Years of
DIVISION                                   Education  Education  Education
--------------------------------------------------------------------------
CALIFORNIA REGIONAL                               14         15         16
CONTRACTS                                         14         16         20
CORPORATE COMMUNICATIONS                          14         16         17
DOCUMENTATION DEVELOPMENT                         12         14         16
EDUCATION                                         15         17         20
EXECUTIVE                                         18         19         20
FACILITIES                                        12         15         18
FINANCE                                           16         17         18
HOST SYSTEMS DEVELOPMENT                          15         17         20
HUMAN RESOURCES                                   12         16         20
INFORMATION SYSTEMS                               14         17         20
INTERNAL DATA BASE                                16         16         17
PUBLICATIONS                                      12         15         20
QUALITY ASSURANCE                                 15         17         20
SALES & MARKETING                                 12         15         19
SOFTWARE DEVELOPMENT                              13         16         20
TECHNICAL SUPPORT                                 15         16         18
```

Select

Tools ► Reset

to reset your query and return to the SQL QUERY TABLES window.

Removing Duplicate Rows

You can remove duplicate rows from your query output. To display each distinct division and location, select **SAMPLE.EMPINFO** and add it to the list of Selected Tables. Select **OK**.

Select **DIVISION** and **LOCATION** in the SQL QUERY COLUMNS window and add them to the list of Selected Columns.

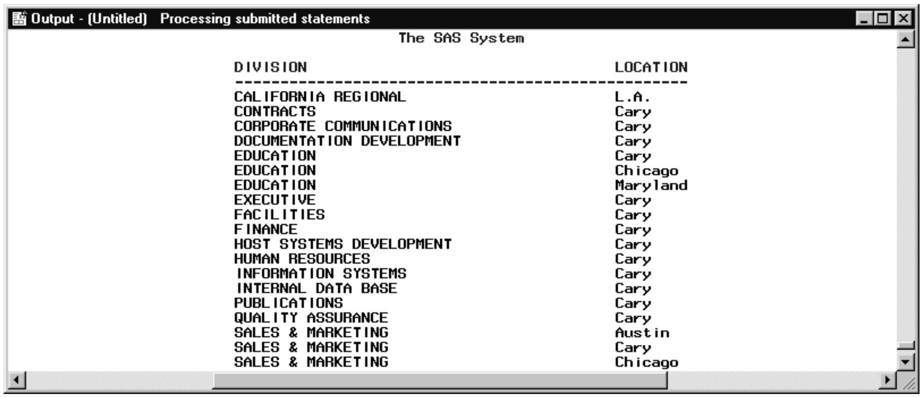
Select

View ► Distinct

Select

Tools ► Run Query ► Run Immediate

Lines in the OUTPUT window which contain the same division and location are not repeated.



```
                         The SAS System

              DIVISION                                LOCATION
              ---------------------------------------------------
              CALIFORNIA REGIONAL                     L.A.
              CONTRACTS                               Cary
              CORPORATE COMMUNICATIONS                Cary
              DOCUMENTATION DEVELOPMENT               Cary
              EDUCATION                               Cary
              EDUCATION                               Chicago
              EDUCATION                               Maryland
              EXECUTIVE                               Cary
              FACILITIES                              Cary
              FINANCE                                 Cary
              HOST SYSTEMS DEVELOPMENT                Cary
              HUMAN RESOURCES                         Cary
               INFORMATION SYSTEMS                    Cary
               INTERNAL DATA BASE                     Cary
              PUBLICATIONS                            Cary
              QUALITY ASSURANCE                       Cary
              SALES & MARKETING                       Austin
              SALES & MARKETING                       Cary
              SALES & MARKETING                       Chicago
```

Subsetting Groups of Data with the Having Condition

The Having condition specifies the condition(s) that each group must satisfy in order to be included in the query output. You can use a Having condition to subset grouped data by using HAVING in the same query with a GROUPBY and a summary function.

Which divisions in the previous example have a minimum education level that is greater than 15 years?

To find out, remove **LOCATION** from the Selected Columns list in the SQL QUERY COLUMNS window. Remove duplicate values by selecting

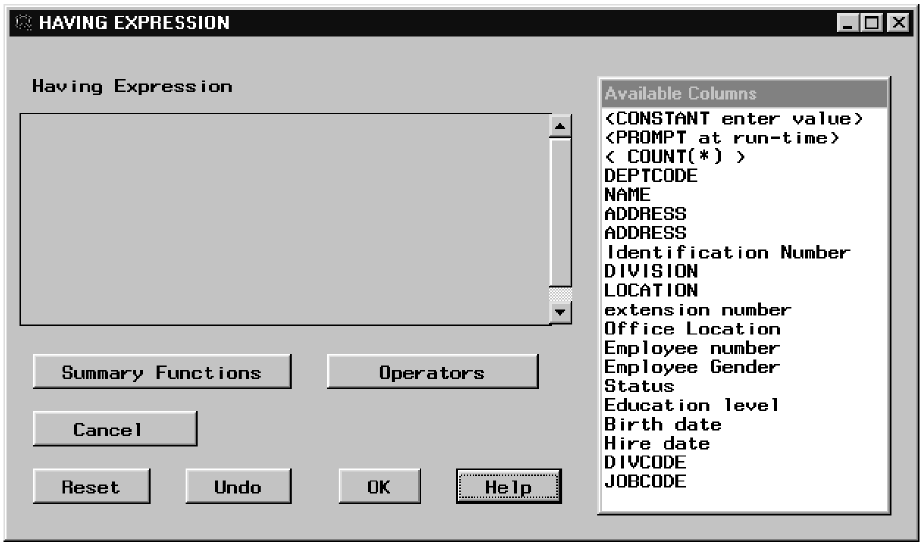
View ► Distinct

Having Expression Window

To create a condition that each group must satisfy, select

View ► Having Condition for Group...

to display the HAVING EXPRESSION window.

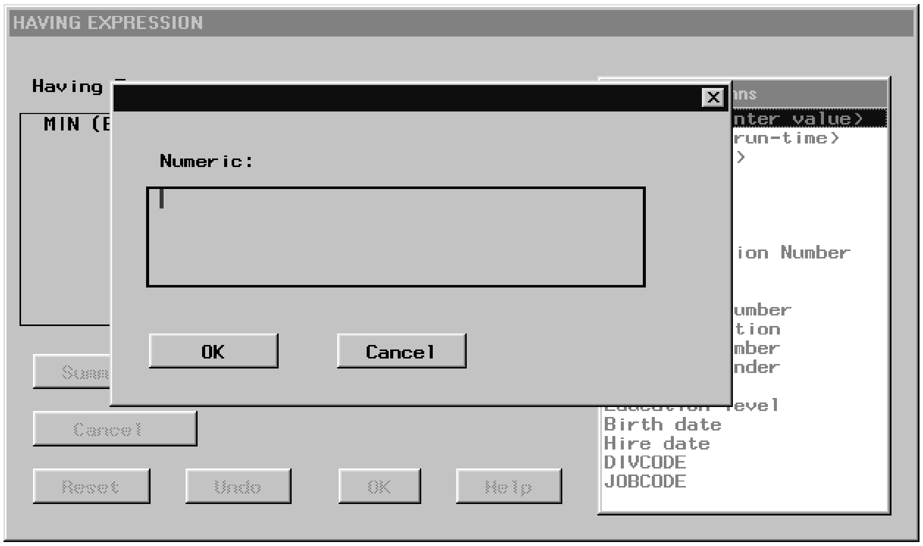


Select **Summary Functions**. Select **MIN** from the list of summary functions.

Select **Education level** from the list of Available Columns.

Select **GT** from the list of operators that appears.

Select **<CONSTANT enter value>** from the list of Available Columns. The Numeric Values window appears.



Type 15 in the Numeric Values window and select **OK**.

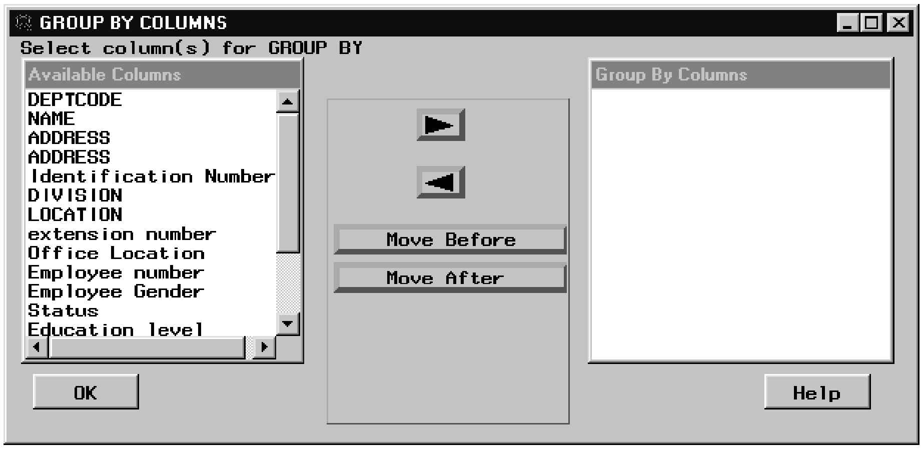
In the HAVING EXPRESSION window, select **OK** to return to the SQL QUERY COLUMNS window.

Viewing the Results of the HAVING Condition

Select

View ► Group(s) for Summary Functions...

to display the GROUP BY COLUMNS window.



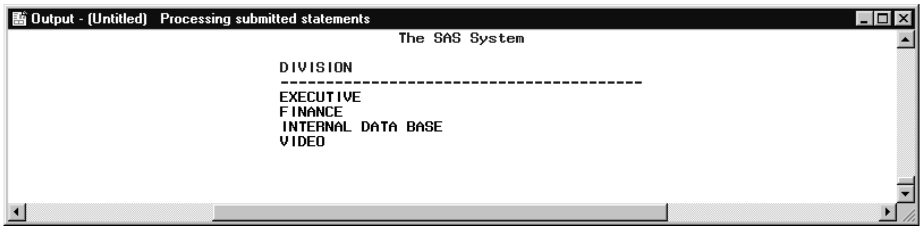
Select **DIVISION** from the list of Available Columns and add it to the list of Group By Columns.

Select **OK**.

Select

Tools ► Run Immediate

to display the divisions whose minimum employee education level is greater than 15.



```
                              The SAS System

                 DIVISION
                 ----------------------------------------
                 EXECUTIVE
                 FINANCE
                 INTERNAL DATA BASE
                 VIDEO
```

Select

Tools ► Reset

to reset the query and return to the SQL QUERY TABLES window.

Using the Automatic Lookup Feature

You can implement automatic lookup for any column in a table that can be accessed from the SQL Query window. An action automatically occurs when that column and an operator is selected from the Where Expression window.

Implement automatic lookup by creating a SAS data set called a lookup table. Insert a set of values into the lookup table for each column for which you want a Lookup Values window to be displayed.

Lookup Strategies

You can specify any one of five lookup strategies for each column:

V (Value)
: Automatically retrieves the distinct values of the column that has been specified in the lookup table. The distinct values appear in a Lookup Values window in the

Where Expression window when you have selected both the specified column from the Available Columns window and an operator from the menu that subsequently appears. When you select one or more values, these values are inserted into the Where expression. The EQ operator is converted to the IN operator to allow multiple selections.

T (Table)
Reads a table and displays the values of all the columns in the Lookup Values window. The first column in the table must contain the values that are needed in the Where expression. You can use other columns to provide descriptive information.

If the first column contains a small number of distinct rows in comparison to the number of rows in the table, the distinct values and their descriptions can be stored in a separate table. This table can be used to display automatic lookup values for the subset conditions.

L (List)
Enables you to select specific columns from a table for display in the Lookup Values window. The first column that you specify must contain the values that are needed for the Where expression. You can use other columns to provide descriptive data values.

F (Format)
Displays column data values and their corresponding formatted values that have been created with the FORMAT procedure.

P (Program)
Invokes a user-written SAS/AF program. A list that contains the currently pending Where expression is passed to the program, where it can be either used or ignored.

Creating an Empty Lookup Table

You can use the following PROC SQL statements to create an empty lookup table.

```
proc sql;
create table sasuser.lookup
  (lookltc  char(100) label='library.table.column',
   lookinfo char(200) label='varies depending on strategy',
   strategy char(8)   label='lookup strategy to use'
  );
```

SASUSER.LOOKUP is the default name of the lookup table.

Adding a Row to the Lookup Table

After you create the empty lookup table, you can submit additional PROC SQL statements to insert values into the table's LOOKLTC, LOOKINFO, and STRATEGY columns. You can also invoke PROC FSEDIT to add this information. The syntax for inserting values into the table is:

```
proc sql;
insert into lookup.table
  values('lookltc-value','lookinfo-value','strategy-value');
```

Add a row to the SASUSER.LOOKUP data set by submitting the following code in the PROGRAM EDITOR window:

```
proc sql;
insert into sasuser.lookup
  values(’sample.empinfo.location’,’sample.program.region.frame’,’P’);
quit;
```

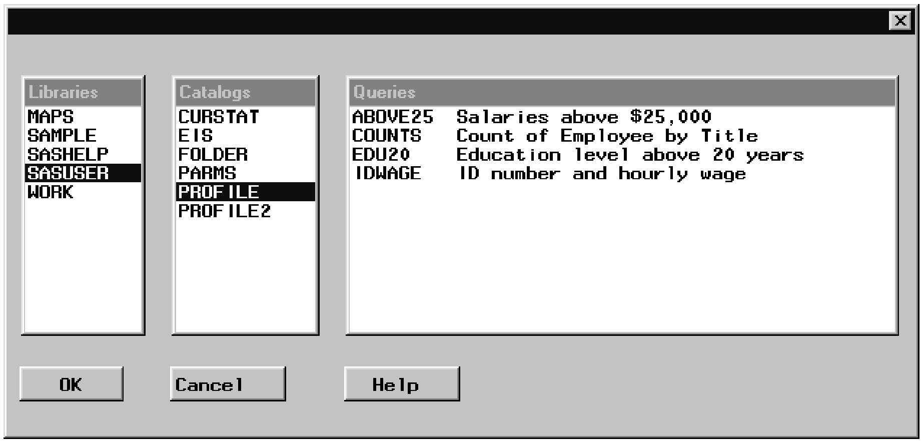
SAMPLE.PROGRAM.REGION.FRAME is a FRAME entry that is part of the sample library you are using for these examples.

Using the Lookup Table

To display the number of employees in each division within a specific geographic region, from the SQL QUERY TABLES window, select

File ▶ List/Include Saved Queries...

to display the Saved Queries window.



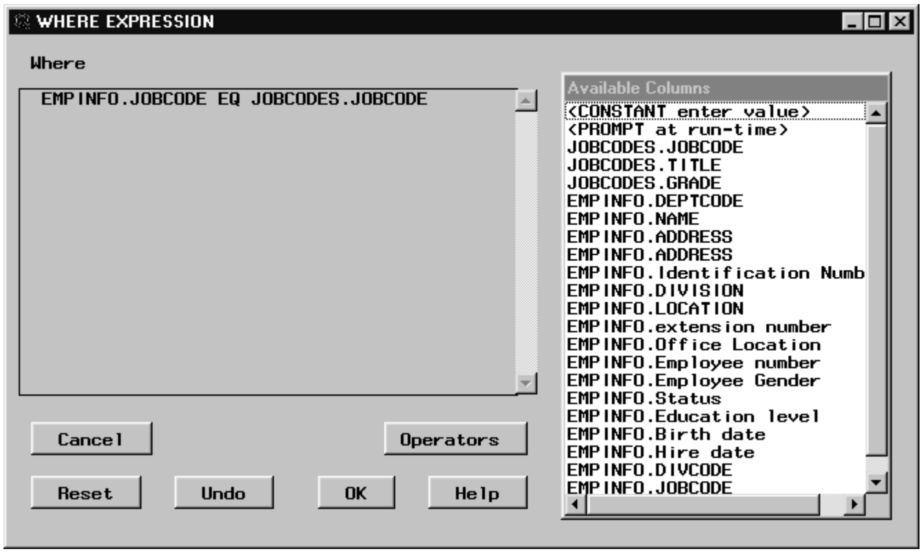
Queries	
ABOVE25	Salaries above $25,000
COUNTS	Count of Employee by Title
EDU20	Education level above 20 years
IDWAGE	ID number and hourly wage

Select **SASUSER.PROFILE.COUNTS**, which was created in“Counting and Grouping Data Automatically” on page 46. Select **Include** to include the query and to return to the SQL QUERY TABLES window.

Select

View ▶ Where Conditions for Subset...

to display the WHERE EXPRESSION window.

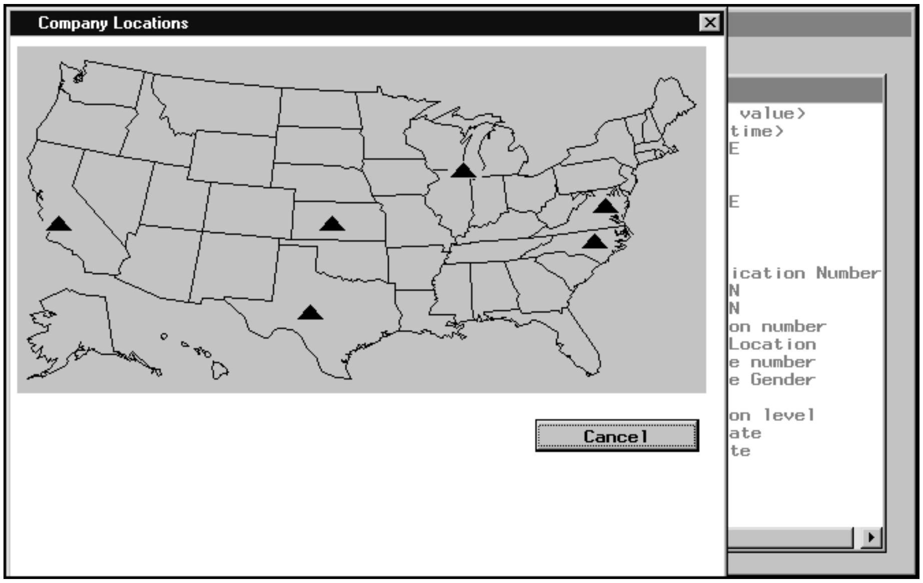


```
EMPINFO.JOBCODE EQ JOBCODES.JOBCODE
```

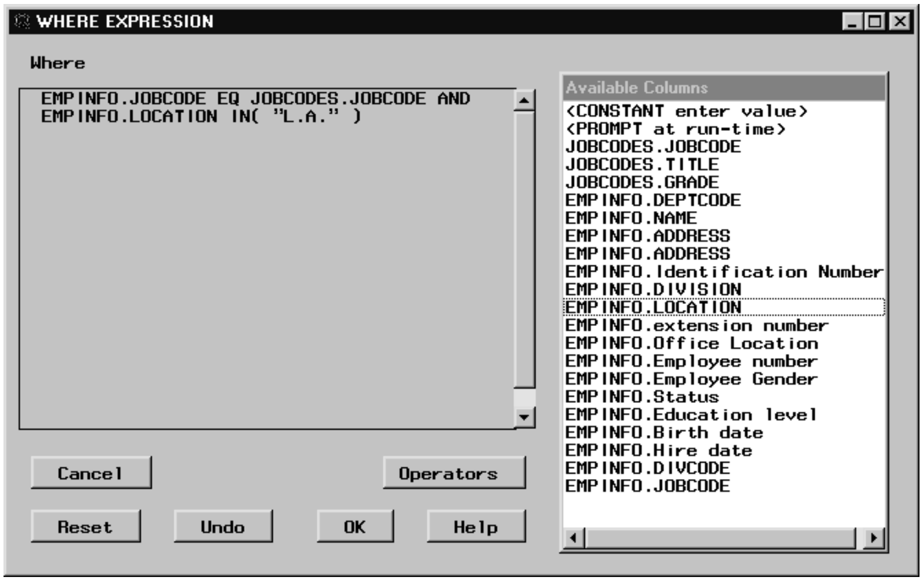
```
<CONSTANT enter value>
<PROMPT at run-time>
JOBCODES.JOBCODE
JOBCODES.TITLE
JOBCODES.GRADE
EMPINFO.DEPTCODE
EMPINFO.NAME
EMPINFO.ADDRESS
EMPINFO.ADDRESS
EMPINFO.Identification Numb
EMPINFO.DIVISION
EMPINFO.LOCATION
EMPINFO.extension number
EMPINFO.Office Location
EMPINFO.Employee number
EMPINFO.Employee Gender
EMPINFO.Status
EMPINFO.Education level
EMPINFO.Birth date
EMPINFO.Hire date
EMPINFO.DIVCODE
EMPINFO.JOBCODE
```

Select **Operators**. Select **AND** from the list of operators.

Select **EMPINFO.LOCATION** from the list of Available Columns. Select **EQ** from the list of comparison operators that appears. Because you have defined EMPINFO.LOCATION with an automatic lookup, the Company Locations window will automatically appear.



Select the westernmost site to complete the WHERE clause.



```
EMPINFO.JOBCODE EQ JOBCODES.JOBCODE AND
EMPINFO.LOCATION IN( "L.A." )
```

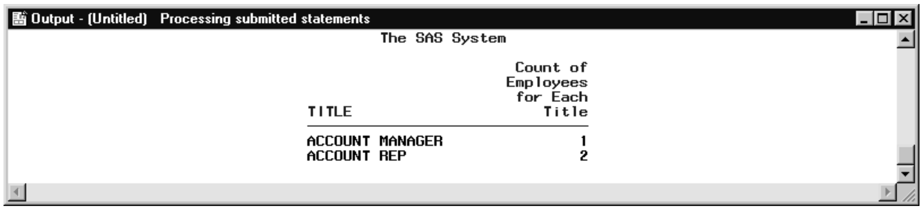
Viewing Your Output

Select **OK**.

Select

Tools ► Run Query ► Run Immediate

to display the results of your query.



```
                          The SAS System

                                     Count of
                                    Employees
                                     for Each
           TITLE                        Title
           ----------------------------------
           ACCOUNT MANAGER                  1
           ACCOUNT REP                      2
```

Select

Tools ► Reset

to reset your query.

Creating a Slider Bar to Indicate a Range

You can use a slider bar to select a range of lookup values in a query.

In this example, you will associate the slider with EMPINFO.salary. Because you may not want to permanently associate these lookup values with the EMPINFO.salary column, you can insert the lookup table into a different profile and switch to that profile when you want to use the slider bar.

Creating a New Lookup Table

Use the following PROC SQL statements to create an empty lookup table in the SAMPLE directory.

```
proc sql;
create table sample.lookup
  (lookltc  char(100) label='library.table.column',
   lookinfo char(200) label='varies depending on strategy',
   strategy char(8)   label='lookup strategy to use'
  );
```

Add a row to the SAMPLE.LOOKUP data set by submitting the following code in the PROGRAM EDITOR window:

```
proc sql;
insert into sample.lookup
  values('sample.salary.salary','sample.program.salrange.frame','P');
quit;
```

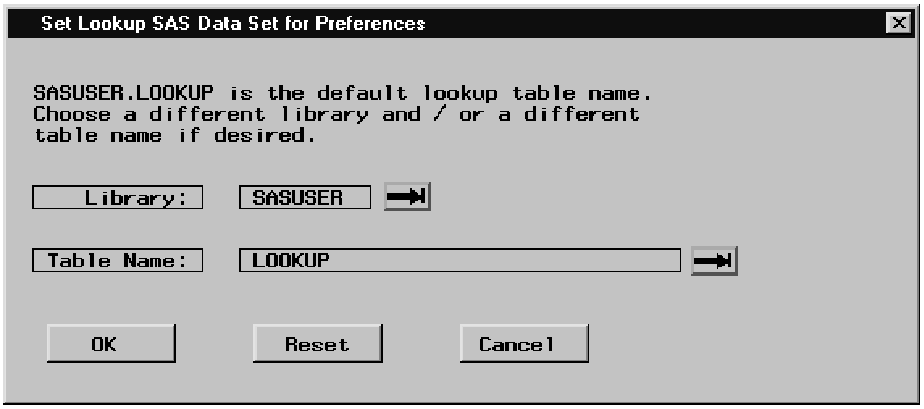
SAMPLE.PROGRAM.SALRANGE.FRAME is a FRAME entry that is part of the sample library you are using for these examples.

Creating a New Profile

Create an SQL Query Window profile that specifies SAMPLE.LOOKUP as the automatic lookup table. From the PMENU, select

Profile ▶ Set Preferences...

Select the ⇒ next to **Automatic Lookup** to display the Set Lookup SAS Data Set for Preferences window.



Set Lookup SAS Data Set for Preferences

SASUSER.LOOKUP is the default lookup table name. Choose a different library and / or a different table name if desired.

Library: SASUSER

Table Name: LOOKUP

Select **SAMPLE** from the list of Libraries. Select **OK** to return to the Preference Settings for Profile window.

Select the ⇒ next to **Data Restrictions** to display the Data Restrictions for Profile window. Select **SAMPLE** from the list of Table Sources. Select **Add entire Table Source to preferences** from the pop-up menu that appears. Select **WORK** from the list of Table Sources. Select **Add entire Table Source to preferences** from the pop-up menu that appears.

Select **OK** to return to the Preference Settings for Profile window.

Select **Save** to save your new profile setting. Type LOOKUP in the **Entry Name:** field of the Name Catalog Entry for Profile window. Type **Slider Bar for Salary Range** in the **Entry description for the profile:** field.

Select **OK** to return to the Preference Settings for Profile window. Select **Close**.

From the SQL QUERY TABLES window PMENU, select

Tools ▶ Switch to New Profile

Select the ⇒ next to **Profile Name:** to display a list of profiles. Select **SASUSER.PROFILE.LOOKUP** from the Preference Profiles in Catalog window.

Select **OK** to return to the SQL QUERY TABLES window and to complete the switch to the new profile.

See “Setting Your Profile” on page 73 for more information on the SQL Query Window user profile.

A Demonstration of the Slider Bar

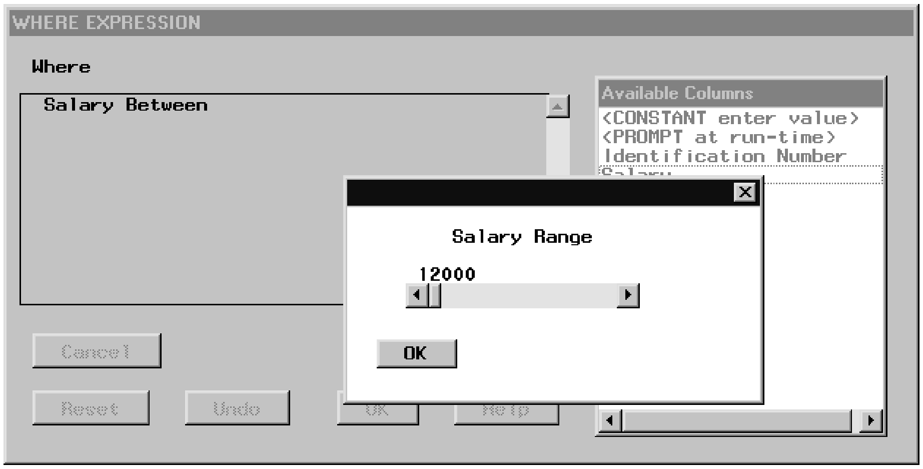
To show how the slider works, you can construct a simple WHERE expression that displays the range of salaries. In the SQL QUERY TABLES window, select SAMPLE.SALARY from the list of Available Tables and add it to the list of Selected Tables. Select **OK** to display the SQL QUERY COLUMNS window.

In the SQL QUERY COLUMNS window, select **Salary** and **Identification Number** from the list of Available Columns and add them to the list of Selected Columns.

Select

View ► Where Conditions for Subset...

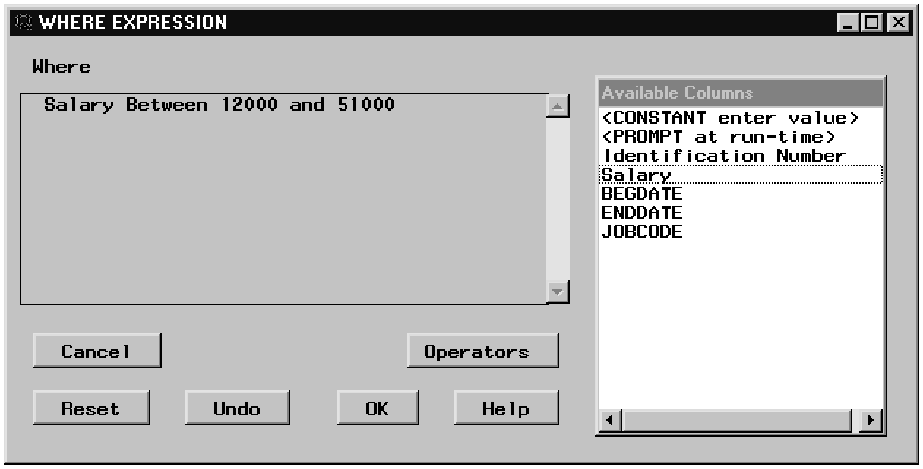
In the WHERE EXPRESSION window, select **Salary** from the list of Available Columns. Select **Between** from the list of **OTHER Operators**. Because the lookup table is associated with the Salary column, the slider bar that is the FRAME entry appears.



Where

Salary Between

Select **OK** to accept the value of **12000**. The slider bar appears again because the **Between** requires a second value. Move the slider to the right until **51000** is displayed. Select **OK** to complete the WHERE expression.



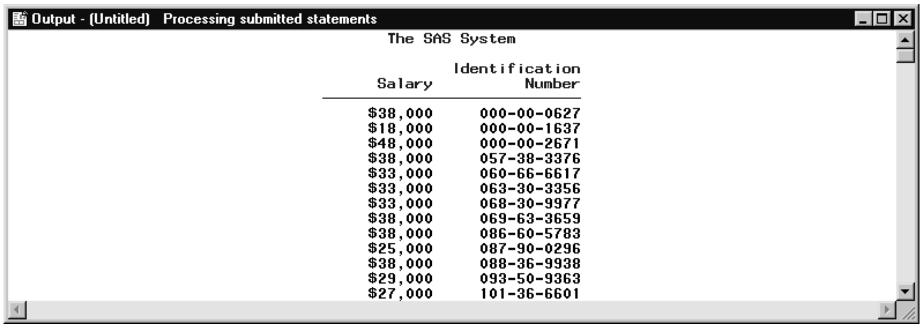
```
Salary Between 12000 and 51000
```

```
<CONSTANT enter value>
<PROMPT at run-time>
Identification Number
Salary
BEGDATE
ENDDATE
JOBCODE
```

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

Tools ▶ Run Query ▶ Run Immediate

to display the employee identification numbers whose salaries are between $12,000 and $51,000.



```
                The SAS System

                    Identification
         Salary             Number
      ----------------------------

        $38,000        000-00-0627
        $18,000        000-00-1637
        $48,000        000-00-2671
        $38,000        057-38-3376
        $33,000        060-66-6617
        $33,000        063-30-3356
        $33,000        068-30-9977
        $38,000        069-63-3659
        $38,000        086-60-5783
        $25,000        087-90-0296
        $38,000        088-36-9938
        $29,000        093-50-9363
        $27,000        101-36-6601
```

Select

Tools ▶ Reset

to reset the query and return to the SQL QUERY TABLES window.

How to Use SCL to Call a FRAME Entry

If your site is licensed to use SAS/AF software, you can use SAS Screen Control Language (SCL) to create a lookup table that uses the SAMPLE.PROGRAM.SALRANGE.FRAME entry or another FRAME entry that you design.

```
entry looklst 8 lkuptype $1 rc 8 msg $40 wherelst 8;

init:
 salrange =12000;
 lkuptype = 'N';
return;


main:
```

```
return;

term:
return;

range:
  call notify('range', '_GET_VALUE_', value);
  call notify('salrange', '_SET_VALUE_', value);
return;

ok:
 call notify('salrange', '_GET_VALUE_', value);
 looklst  = insertn(looklst, value, 1);
 rc       = 0;
 _status_ = 'H';
 link term;
return;
```

Refer to *SAS Component Language: Reference* for more information on SCL.

Creating and Using Outer Joins

An outer join combines rows of data from two tables. There are three types of outer joins:

left join
: returns all matching rows in both tables, in addition to rows in the left table that have no matching rows in the right table.

right join
: returns all matching rows in both tables, in addition to rows in the right table that have no matching rows in the left table.

full join
: returns all matching and unmatching rows from both tables.

In all three types of outer joins, the columns in the result row that are from the unmatched row are set to missing values.

In this example, you will first create an inner join that relates employee identification number and salary. Then, you will create an outer join that combines this data with data from another table to compute the gross monthly pay for employees who have taken leave.

Creating a Query View

You can create an SQL view that contains the syntax of your query. You will use this view to create an outer join query.

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

In the SQL QUERY COLUMNS window, select **NAME**, the two **ADDRESS** items, **Identification Number**, **Employee number**, **Salary**, **BEGDATE**, and **ENDDATE** and add them to the list of Selected Columns.

Select

View ► Where Conditions for Subset...

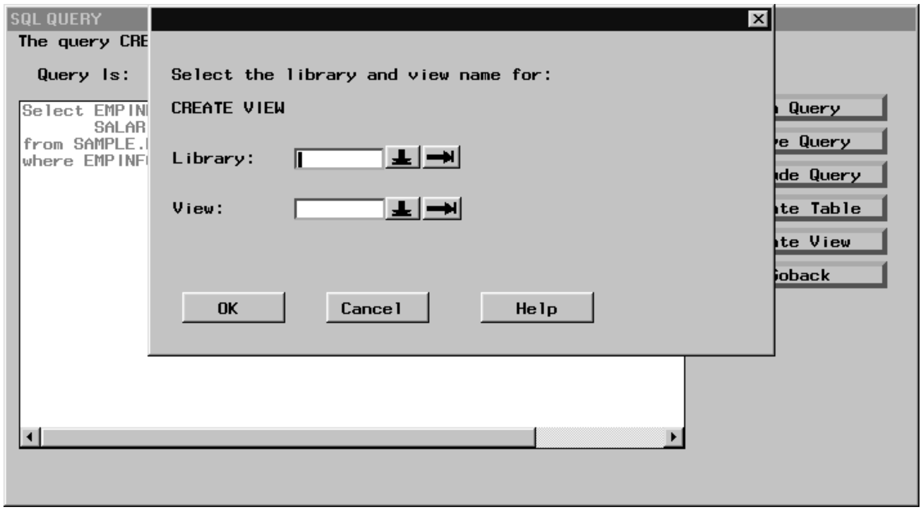
to display the WHERE EXPRESSION window.

Select **EMPINFO.Identification Number** from the list of Available Columns. Select **EQ** from the list of Operators. Select **Salary.Identification Number** from the list of Available Columns. Select **OK**.

This WHERE expression creates an inner join of EMPINFO and Salary based on Identification Number. To save the query as a view, select

Tools ► Show Query...

to display the SQL QUERY window. Select **Create View**.



SQL QUERY

The query CRE

Query Is:

```
Select EMPIN
       SALAR
from SAMPLE.I
where EMPINF
```

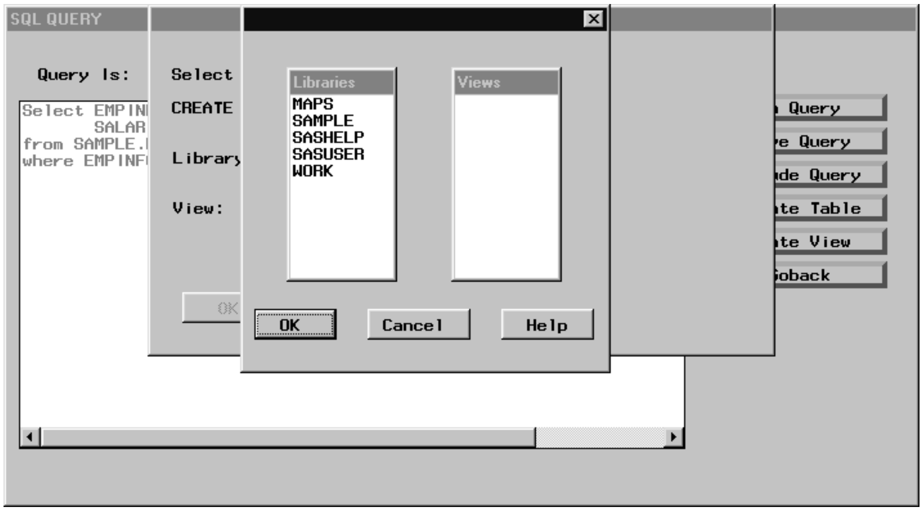
Select the library and view name for:

CREATE VIEW

Library:

View:

Select the → next to the **Library:** field to display a list of SAS libraries.



SQL QUERY

Query Is:

```
Select EMPIN
       SALAR
from SAMPLE.I
where EMPINF
```

Select

CREATE

Library

View:

Libraries	Views
MAPS SAMPLE SASHELP SASUSER WORK	

The list of libraries displayed at your site may be different from the ones in the illustration. Select **SAMPLE** from the list of libraries. Select **OK**.

Type MYVIEW in the **View:** field. Select **OK** to return to the SQL QUERY window. Select **Goback** to return to the SQL QUERY COLUMNS window.

Creating an Outer Join

You can now create an outer join with other tables. Select

Tools ▶ Reset

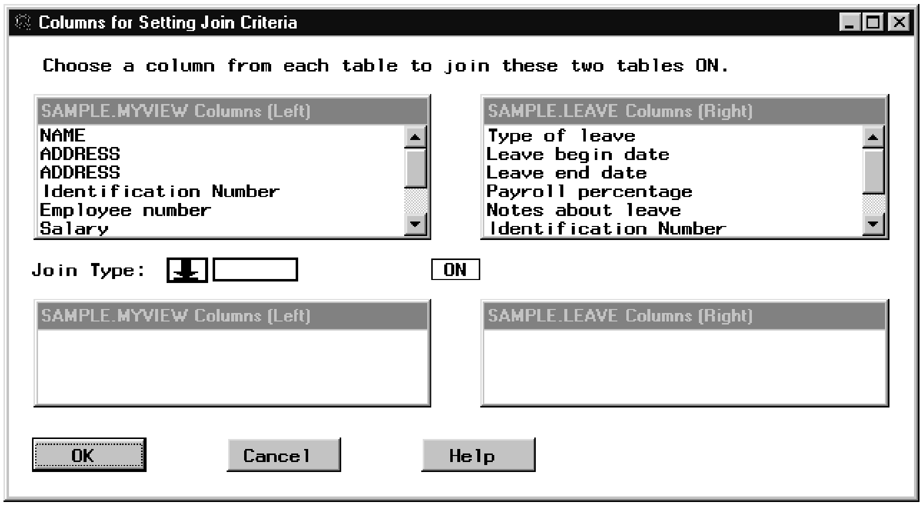
Select **OK** in the pop-up dialog to reset the query.

Select **SAMPLE.MYVIEW** and **SAMPLE.LEAVE** from the list of Available Tables and add them to the list of Selected Tables. Select **OK** to display the SQL QUERY COLUMNS window.

Select

View ▶ Join Type

Select **Matched Join and Unmatched rows (Outer Join)**. Select **OK** to display the Columns for Setting Join Criteria window.



Choose a column from each table to join these two tables ON.

SAMPLE.MYVIEW Columns (Left)

NAME
ADDRESS
ADDRESS
Identification Number
Employee number
Salary

SAMPLE.LEAVE Columns (Right)

Type of leave
Leave begin date
Leave end date
Payroll percentage
Notes about leave
Identification Number

Columns for Setting Join Criteria
Join Type:
ON
SAMPLE.MYVIEW Columns (Left)
SAMPLE.LEAVE Columns (Right)
OK
Cancel
Help

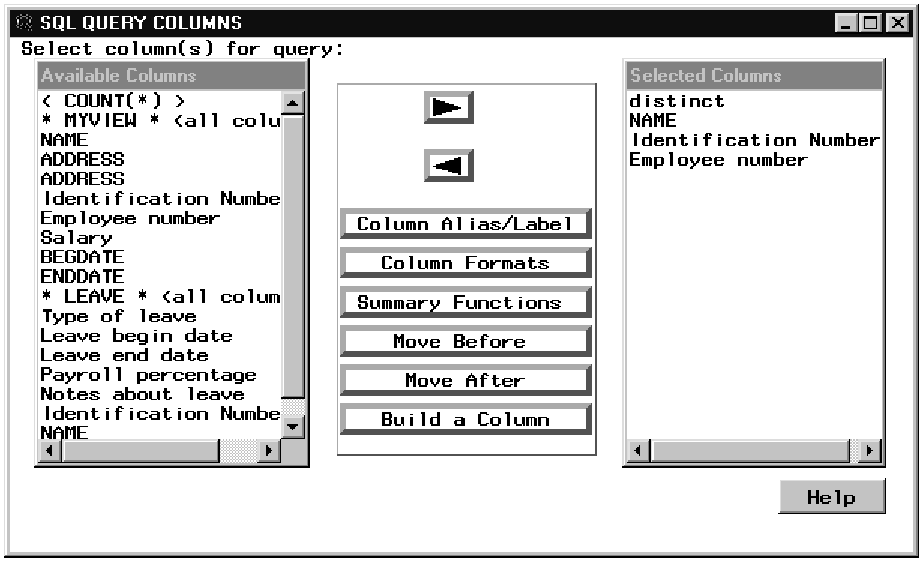
Select **Identification Number** from SAMPLE.MYVIEW Columns (Left). Select **Identification Number** from SAMPLE.LEAVE Columns (Right). Select the arrow next to **Join Type:**. Select **Left** from the pop-up menu. Select **OK** to return to the SQL QUERY COLUMNS window.

Select

View ▶ Distinct

to eliminate duplicate values from your output.

Select **NAME**, **Identification Number**, and **Employee Number** from the list of Available Columns and add them to the list of Selected Columns.



SQL QUERY COLUMNS

Select column(s) for query:

Available Columns

```
< COUNT(*) >
* MYVIEW * <all colu
NAME
ADDRESS
ADDRESS
Identification Numbe
Employee number
Salary
BEGDATE
ENDDATE
* LEAVE * <all colum
Type of leave
Leave begin date
Leave end date
Payroll percentage
Notes about leave
Identification Numbe
NAME
```

Selected Columns

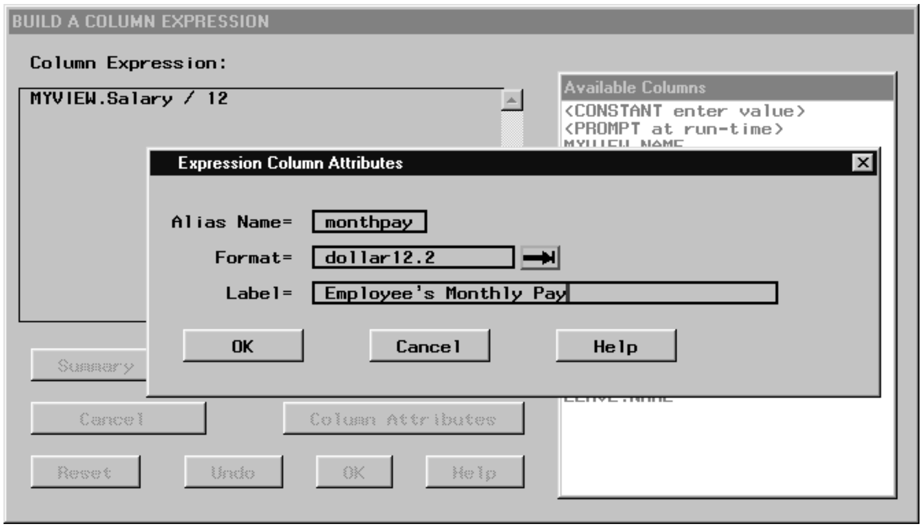
```
distinct
NAME
Identification Number
Employee number
```

Building a Column Expression

Select **Build a Column** to display the BUILD A COLUMN EXPRESSION window.

Select **MYVIEW.Salary** from the list of Available Columns. Select **/** from the list of Operators. Select **<CONSTANT enter value>** from the list of Available columns. Type **12** in the **Numeric:** field. Select **OK**. Click outside the list of operators to make it disappear.

Select **Column Attributes** to display the Expression Column Attributes window. Enter **monthpay** in the **Alias Name=** field. Enter **dollar12.2** in the **Format=** field. Enter **Employee's Monthly Pay** in the **Label=** field.



BUILD A COLUMN EXPRESSION

Column Expression:

```
MYVIEW.Salary / 12
```

Available Columns

```
<CONSTANT enter value>
<PROMPT at run-time>
```

Expression Column Attributes

Alias Name= monthpay

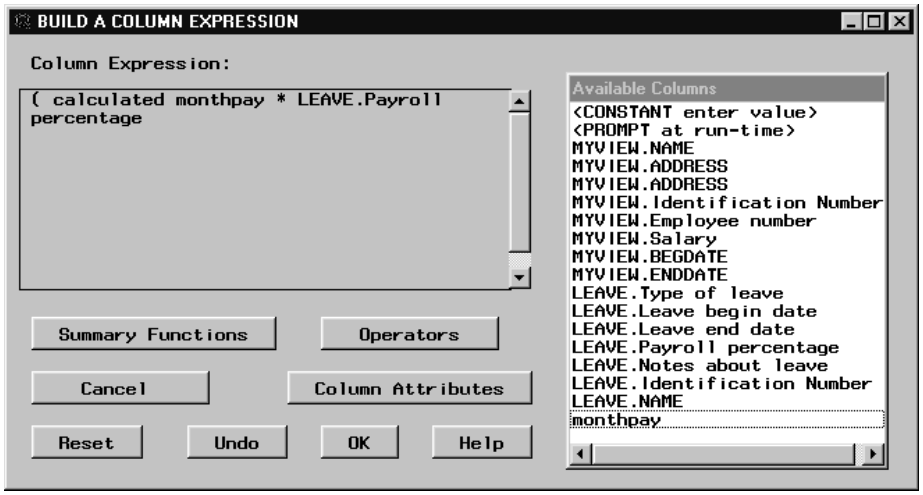
Format= dollar12.2

Label= Employee's Monthly Pay

Select **OK** to return to the BUILD A COLUMN EXPRESSION window. Select **OK** to return to the SQL QUERY COLUMNS window.

In the SQL QUERY COLUMNS window, select **Build a Column** to display the BUILD A COLUMN EXPRESSION window. Select **Operators**. Select **(** from the list of Operators.

Select **monthpay** from the list of Available Columns. Select ***** from the list of Operators. Select **LEAVE.Payroll percentage** from the list of Available Columns. Select **)**from the list of Operators. Click outside the list of operators to make it disappear.



Column Expression:

```
( calculated monthpay * LEAVE.Payroll
percentage
```

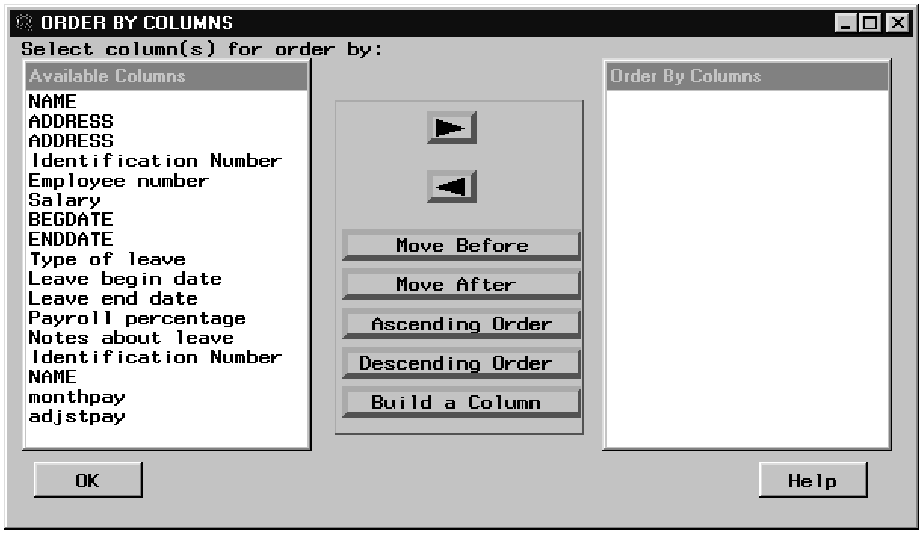
Select **Column Attributes** to display the Expression Column Attributes window. Enter **adjstpay** in the **Alias Name=** field. Enter **dollar12.2** in the **Format=** field. Enter **Employee's Gross Pay** in the **Label=** field. Select **OK** to return to the BUILD A COLUMN EXPRESSION window. Select **OK** to return to the SQL QUERY COLUMNS window.

Order By Columns

In the SQL QUERY COLUMNS window, select

View ▶ Order By...

to display the ORDER BY COLUMNS window



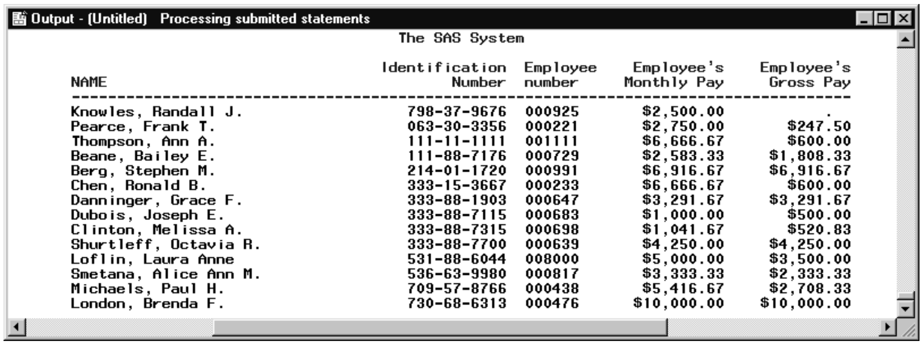
Select the second **Identification Number** from the list of Available Columns and add it to the list of Selected Columns. Select **OK** to return to the SQL QUERY COLUMNS window.

Viewing Your Output

Select

Tools ▶ Run Query ▶ Run Immediate

to display the results of the query in the OUTPUT window.



```
                                    The SAS System

                                  Identification Employee    Employee's    Employee's
NAME                                      Number number     Monthly Pay     Gross Pay
-------------------------------------------------------------------------------------
Knowles, Randall J.                  798-37-9676 000925       $2,500.00             .
Pearce, Frank T.                     063-30-3356 000221       $2,750.00       $247.50
Thompson, Ann A.                     111-11-1111 001111       $6,666.67       $600.00
Beane, Bailey E.                     111-88-7176 000729       $2,583.33     $1,808.33
Berg, Stephen M.                     214-01-1720 000991       $6,916.67     $6,916.67
Chen, Ronald B.                      333-15-3667 000233       $6,666.67       $600.00
Danninger, Grace F.                  333-88-1903 000647       $3,291.67     $3,291.67
Dubois, Joseph E.                    333-88-7115 000683       $1,000.00       $500.00
Clinton, Melissa A.                  333-88-7315 000698       $1,041.67       $520.83
Shurtleff, Octavia R.                333-88-7700 000639       $4,250.00     $4,250.00
Loflin, Laura Anne                   531-88-6044 008000       $5,000.00     $3,500.00
Smetana, Alice Ann M.                536-63-9980 000817       $3,333.33     $2,333.33
Michaels, Paul H.                    709-57-8766 000438       $5,416.67     $2,708.33
London, Brenda F.                    730-68-6313 000476      $10,000.00    $10,000.00
```

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

SAS SQL Query Window User's Guide, Version 8

ISBN 1–58025–554–X

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

1st printing, October 1999

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