CHAPTER

4

Browsing and Updating ADABAS Data

Introduction

The SAS/ACCESS interface to ADABAS enables you to browse and update ADABAS data directly from a SAS session or program. This chapter shows you how to use SAS procedures to browse and update ADABAS data described by SAS/ACCESS view descriptors.

For definitions of the view descriptors used in this chapter as well as their associated access descriptors, the ADABAS files, NATURAL DDMs, and SAS data files used throughout the book, see Appendix 3, “Example Data,” on page 137.

Before you can browse or update ADABAS data, you must have access to the data through appropriate security options. ADABAS and NATURAL have several levels of security options, and you may be allowed to display or browse data but not update values. Check with your Database Administrator (DBA) or the ADABAS file’s or NATURAL DDM’s creator to see what security options you have. If you have been granted the appropriate ADABAS security options, you can use the SAS procedures described in this chapter to update ADABAS data with a SAS/ACCESS view descriptor. For more information on ADABAS and NATURAL security, see Chapter 2, “ADABAS Essentials,” on page 7, and Appendix 1, “Information for the Database Administrator,” on page 105.

Browsing and Updating with the SAS/FSP Procedures

If your site has SAS/FSP software as well as SAS/ACCESS software, you can browse and update ADABAS data from within a SAS program.

You can use three SAS/FSP procedures: FSBROWSE, FSEDIT, and FSVIEW. The FSBROWSE and FSEDIT procedures show you one ADABAS logical record at a time, whereas the FSVIEW procedure displays multiple logical records in a tabular format similar to the PRINT procedure. PROC FSVIEW enables you to both browse and update ADABAS data, depending on which option you choose.

Using the FSBROWSE Procedure

The FSBROWSE procedure enables you to look at ADABAS data but does not allow you to change it. To use PROC FSBROWSE, submit the following SAS statements:

```
proc fsbrowse data=vlib.usacust;
run;
```

The FSBROWSE procedure retrieves one logical record of ADABAS data at a time. To browse each logical record, issue the FORWARD and BACKWARD commands.

Using the FSEDIT Procedure

The FSEDIT procedure enables you to update ADABAS data described by a view descriptor if you have access to the data through the appropriate ADABAS or NATURAL security options. To use PROC FSEDIT, submit the following SAS statements:

```
proc fsedit data=vlib.usacust;
run;
```

A window similar to the FSBROWSE window opens to enable you to edit the ADABAS data one observation at a time.

Note: When using PROC FSEDIT, you can cancel an edit only before you scroll. The CANCEL command redisplays the observation as it was before you began to edit it and cancels your editing changes. After you scroll, the changes are saved. △

Using the FSVIEW Procedure

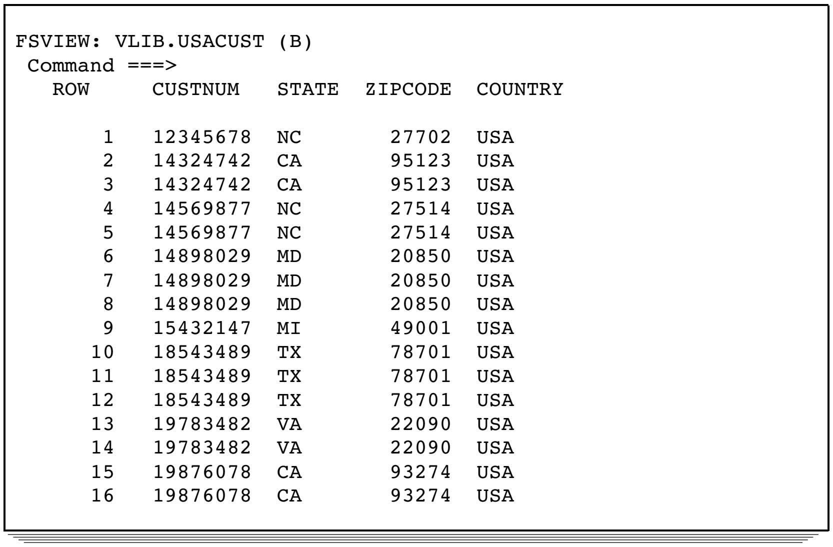
The FSVIEW procedure enables you to browse or update ADABAS data using a view descriptor, depending on how you submit the procedure.

Using the FSVIEW Procedure to Browse ADABAS Data

To browse ADABAS data, submit the PROC FSVIEW statement as follows. Browse mode is the default for the FSVIEW procedure. The statements display the data as shown in Output 4.1 on page 41:

```
proc fsview data=vlib.usacust;
run;
```

Output 4.1 FSVIEW Window Example



```
FSVIEW: VLIB.USACUST (B)
 Command ===>
   ROW     CUSTNUM   STATE  ZIPCODE  COUNTRY

       1   12345678  NC       27702  USA
       2   14324742  CA       95123  USA
       3   14324742  CA       95123  USA
       4   14569877  NC       27514  USA
       5   14569877  NC       27514  USA
       6   14898029  MD       20850  USA
       7   14898029  MD       20850  USA
       8   14898029  MD       20850  USA
       9   15432147  MI       49001  USA
      10   18543489  TX       78701  USA
      11   18543489  TX       78701  USA
      12   18543489  TX       78701  USA
      13   19783482  VA       22090  USA
      14   19783482  VA       22090  USA
      15   19876078  CA       93274  USA
      16   19876078  CA       93274  USA
```

To see the rest of the accessed ADABAS data, you must scroll the window to the right multiple times. You can do this by entering the RIGHT command on the command line or by pressing the function key assigned to this command.

Using the FSVIEW Procedure to Update ADABAS Data

You can also use the FSVIEW procedure to update ADABAS data. To edit the ADABAS data in a listing format, you have to add EDIT or MODIFY to the PROC FSVIEW statement, as shown in the following statement:

```
proc fsview data=vlib.usacust edit;
run;
```

The same window as shown in Output 4.1 on page 41 appears, except the window title contains an (E) for edit, not a (B). *SAS/FSP Software: Usage and Reference* discusses in detail how to edit data using the FSVIEW procedure.

Note: The CANCEL command in the FSVIEW window does *not* cancel your changes, whether you have scrolled or not. △

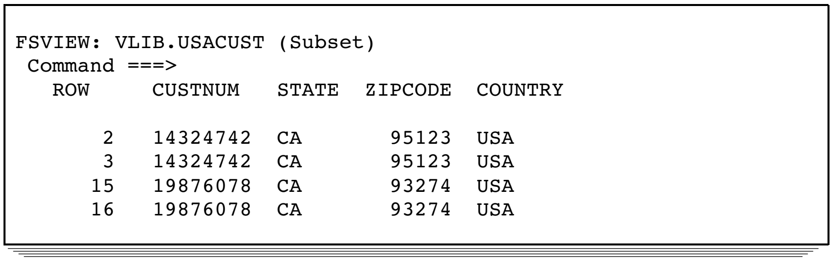
Specifying a SAS WHERE Expression While Browsing or Editing

You can specify a SAS WHERE statement or a SAS WHERE command to retrieve a subset of ADABAS data while using the FSP procedures. The WHERE statement is submitted when the FSP procedure is invoked and retrieves only the observations that meet the conditions of the WHERE statement. The other observations are not available until you exit the procedure. This is called a *permanent* WHERE clause. A SAS WHERE command is a WHERE expression that is invoked from the command line within a FSP procedure. You can clear the command to make all the observations available so it is known as a *temporary* WHERE clause.

The following example of a WHERE statement retrieves the customers from California. These customers are a subset of the customers for the CUSTOMERS DDM. Output 4.2 on page 42 shows the FSVIEW window after the following statements have been submitted:

```
proc fsview data=vlib.usacust edit;
   where state='CA';
run;
```

Output 4.2 FSVIEW Window after SAS WHERE Statement



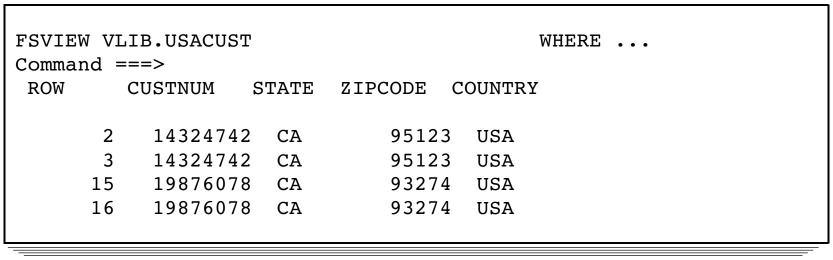
```
FSVIEW: VLIB.USACUST (Subset)
 Command ===>
   ROW     CUSTNUM   STATE  ZIPCODE  COUNTRY

       2   14324742  CA       95123  USA
       3   14324742  CA       95123  USA
      15   19876078  CA       93274  USA
      16   19876078  CA       93274  USA
```

Only the logical records with a STATE value of **CA** are retrieved for editing. Note that (Subset) appears after VLIB.USACUST in the window title to remind you that the data retrieved are a subset of the data described by the view descriptor. You can then edit each observation by typing over the information you want to modify. Issue the END command to end your editing session.

Output 4.3 on page 42 shows the FSVIEW window when the subset of data is generated by the WHERE command:

```
where state='CA'
```

Output 4.3 FSVIEW Window after SAS WHERE Command



```
FSVIEW VLIB.USACUST                       WHERE ...
Command ===>
 ROW     CUSTNUM   STATE  ZIPCODE  COUNTRY

       2   14324742  CA       95123  USA
       3   14324742  CA       95123  USA
      15   19876078  CA       93274  USA
      16   19876078  CA       93274  USA
```

Output 4.2 on page 42 and Output 4.3 on page 42 are identical, except (Subset) after the title is replaced with WHERE ... in the upper-right corner. You can then update each observation, as described earlier.

Although these examples have shown a SAS WHERE statement and a SAS WHERE command with the FSVIEW procedure, you can also retrieve a subset of data using the FSBROWSE and FSEDIT procedures. For more information on the SAS WHERE statement, refer to *SAS Language Reference: Dictionary*. For more information about using the SAS WHERE command within the SAS/FSP procedures, refer to *SAS/FSP Software: Usage and Reference*.

Adding and Deleting Data with the SAS/FSP Procedures

Adding and deleting ADABAS data with the SAS/FSP procedures is different for view descriptors than for SAS data files.

Adding Data

Adding ADABAS data as a result of any SAS System update operation can cause the interface view engine to decide whether to add a new ADABAS logical record or to modify an existing one, for example, to add an occurrence to a periodic group.

If there are no periodic group fields accessed by the view descriptor or within the ADABAS file, doing an insert is straightforward. However, if a periodic group does exist, then doing an insert is more complicated, because the interface view engine generates multiple SAS observations from a single ADABAS record that contains a periodic group.

Values in the observation to be added are compared to values in the previous observation. If the contents of the previous observation do not help determine whether to add or modify, a new logical record is added. However, it is possible that some of the new values might already reside in the ADABAS file, which would mean that a new logical record is not necessary. This occurs if a periodic group is selected by the view descriptor, and the new data occur only in variables corresponding to data fields that are part of that periodic group.

You can help the interface view engine resolve whether to add a new logical record or modify an existing one by specifying BY keys. For information and examples of using BY keys, see “Using a BY Key To Resolve Ambiguous Inserts” on page 122.

Deleting Data

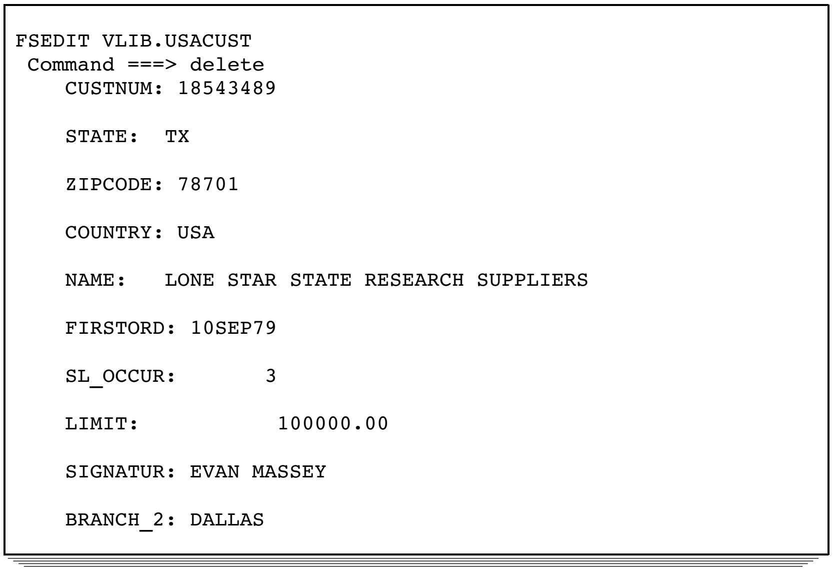
When you delete a logical record, the results depend on whether the observation is part of a periodic group. If the logical record is not part of a periodic group, deleting an observation causes a logical record to be deleted from the ADABAS file. However, if the logical record is part of a periodic group, the results of deleting an observation depend on the status of the ADBDEL systems option for the interface view engine, which is set in the ADBEUSE CSECT. For more information, see “Systems Options for the ACCESS Procedure and the Interface View Engine” on page 115.

- □ If ADBDEL=N (which is the default setting), the selected values for that occurrence in the periodic group are set to null (missing), but the logical record is not deleted.
- □ If ADBDEL=P, the entire logical record is deleted.

The following example illustrates using the DELETE command in the FSEDIT procedure. (Note that the ADBDEL systems option is set to N.)

Suppose you want to edit the ADABAS data described by VLIB.USACUST. You can use the FSEDIT procedure with a PROC FSEDIT statement. Scroll forward to the observation to be deleted. In this example, there are three occurrences for the periodic group SIGNATURE-LIST. Output 4.4 on page 44 shows the third occurrence, which you want to delete. (Notice that the variable SL_OCCUR displays the value 3, which tells you that this is the observation for the third occurrence.) Enter the DELETE command on the command line, as shown in Output 4.4 on page 44, and press ENTER.

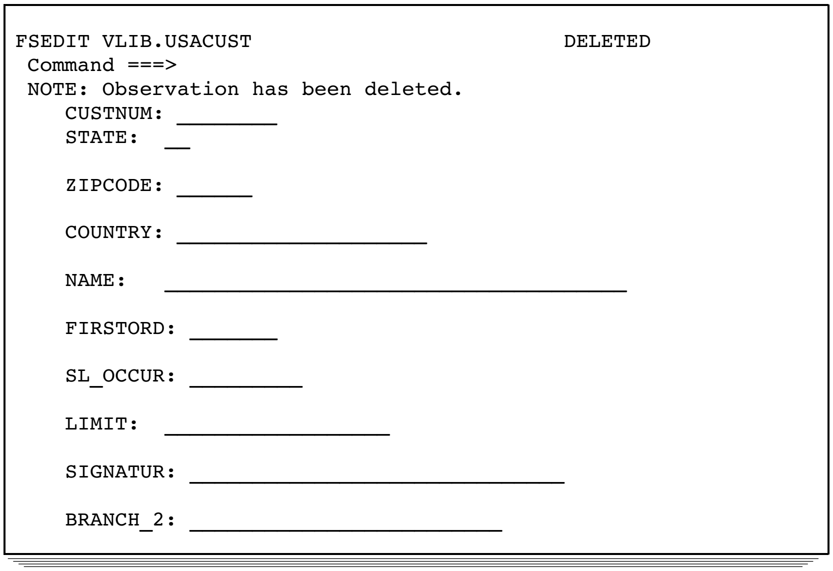
Output 4.4 Deleting an ADABAS Logical Record



```
FSEDIT VLIB.USACUST
 Command ===> delete
    CUSTNUM: 18543489

    STATE:  TX

    ZIPCODE: 78701

    COUNTRY: USA

    NAME:   LONE STAR STATE RESEARCH SUPPLIERS

    FIRSTORD: 10SEP79

    SL_OCCUR:       3

    LIMIT:           100000.00

    SIGNATUR: EVAN MASSEY

    BRANCH_2: DALLAS
```

The DELETE command processes the deletion and displays a message to that effect, as shown in Output 4.5 on page 45. There is no indication of what actions the interface view engine actually took.

Output 4.5 Deletion Message Displayed



```
FSEDIT VLIB.USACUST                          DELETED
 Command ===>
 NOTE: Observation has been deleted.
    CUSTNUM: ________
    STATE:  __

    ZIPCODE: ______

    COUNTRY: ____________________

    NAME:   _____________________________________

    FIRSTORD: _______

    SL_OCCUR: _________

    LIMIT:  __________________

    SIGNATUR: ______________________________

    BRANCH_2: _________________________
```

The entire observation seems to have been removed from the ADABAS file, but this is not the case. For the third occurrence, the interface view engine sets the values for data fields LIMIT and SIGNATUR to missing; the other data remain the same. Regardless of the actions though, the observation you deleted is no longer available for processing. For more information on using the SAS/FSP procedures, see *SAS/FSP Software: Usage and Reference.*

Browsing and Updating with the SQL Procedure

The SAS System SQL procedure also enables you to retrieve and update ADABAS data. You can retrieve and browse ADABAS data by specifying a view descriptor in a PROC SQL SELECT statement.

To update the data, you can specify view descriptors in the PROC SQL DELETE, INSERT, and UPDATE statements. You must have access to the data through appropriate ADABAS and/or NATURAL security options before you can edit ADABAS data. Here is a summary of these PROC SQL statements:

DELETE deletes logical records from an ADABAS file.

INSERT inserts logical records in an ADABAS file.

SELECT retrieves and displays data from an ADABAS file. A SELECT statement is usually referred to as a query because it queries the ADABAS file for information.

UPDATE updates values in an ADABAS file.

When using the SQL procedure, note that the data are displayed in the SAS OUTPUT window. The procedure displays output data automatically without using the

PRINT procedure and executes without using the RUN statement when an SQL procedure statement is executed.

You can use the SELECT statement to browse ADABAS data described by a view descriptor. The query in the following example retrieves and displays specified data fields and logical records in the CUSTOMERS DDM that are described by the VLIB.USACUST view descriptor. The LINESIZE= system option is used to reset the default output width to 120 columns.

Note: The following SQL procedure examples assume the CUSTOMERS DDM has not been updated by the earlier SAS/FSP examples. △

```
options linesize=120;

proc sql;
   title 'ADABAS Data Output by a
      SELECT Statement';
select custnum, state, name, limit,signatur
   from vlib.usacust;
```

Output 4.6 on page 47 displays the query's results. Notice in the output that the SQL procedure displays the ADABAS data field names, not the corresponding SAS variable names.

Output 4.6 ADABAS Data Output by a PROC SQL Query

```
                             ADABAS Data Output by a SELECT Statement

CUSTOMER  STATE  NAME                                                                       LIMIT
SIGNATURE
-------------------------------------------------------------------------------------------------
12345678  NC                                                                                 0.00


14324742  CA     SANTA CLARA VALLEY TECHNOLOGY SPECIALISTS                                5000.00
BOB HENSON

14324742  CA     SANTA CLARA VALLEY TECHNOLOGY SPECIALISTS                               25000.00
KAREN DRESSER

14569877  NC     PRECISION PRODUCTS                                                       5000.00
JEAN CRANDALL

14569877  NC     PRECISION PRODUCTS                                                     100000.00
STEVE BLUNTSEN

14898029  MD     UNIVERSITY BIOMEDICAL MATERIALS                                         10000.00
MASON FOXWORTH

14898029  MD     UNIVERSITY BIOMEDICAL MATERIALS                                         50000.00
DANIEL STEVENS

14898029  MD     UNIVERSITY BIOMEDICAL MATERIALS                                        100000.00
ELIZABETH PATTON

15432147  MI     GREAT LAKES LABORATORY EQUIPMENT MANUFACTURERS                          10000.00
JACK TREVANE

18543489  TX     LONE STAR STATE RESEARCH SUPPLIERS                                      10000.00
NANCY WALSH

18543489  TX     LONE STAR STATE RESEARCH SUPPLIERS                                      50000.00
TED WHISTLER

18543489  TX     LONE STAR STATE RESEARCH SUPPLIERS                                     100000.00
EVAN MASSEY

19783482  VA     TWENTY-FIRST CENTURY MATERIALS                                           5000.00
PETER THOMAS

19783482  VA     TWENTY-FIRST CENTURY MATERIALS                                          10000.00
LOUIS PICKERING

19876078  CA     SAN JOAQUIN SCIENTIFIC AND INDUSTRIAL SUPPLY, INC.                       7500.00
EDWARD LOWE

19876078  CA     SAN JOAQUIN SCIENTIFIC AND INDUSTRIAL SUPPLY, INC.                      25000.00
 E.F. JENSEN
```

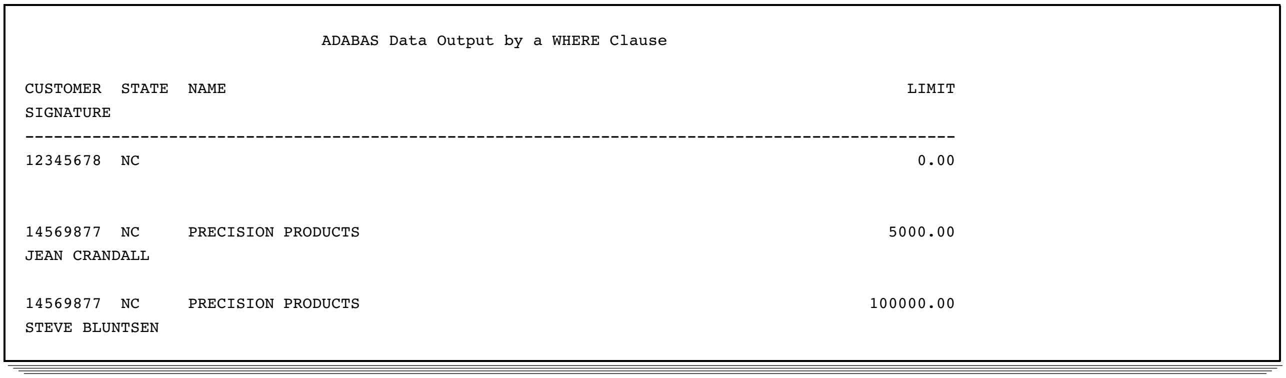
You can specify a WHERE clause as part of the SELECT statement to retrieve a subset of the logical records for display. The following example displays the companies that are located in North Carolina:

```
title 'ADABAS Data Output by a WHERE Clause';
select custnum, state, name, limit, signatur
```

```
    from vlib.usacust
    where state='NC';
```

Notice that the PROC SQL statement is not repeated in this query. With the SQL procedure, you do not need to repeat the PROC SQL statement unless you use another SAS procedure, a DATA step, or a QUIT statement between PROC SQL statements. Output 4.7 on page 48 displays the companies from North Carolina described by VLIB.USACUST.

Output 4.7 ADABAS Data Output Subset by a WHERE Clause



```
                               ADABAS Data Output by a WHERE Clause

CUSTOMER  STATE  NAME                                                                       LIMIT
SIGNATURE
-------------------------------------------------------------------------------------------------
12345678  NC                                                                                 0.00


14569877  NC     PRECISION PRODUCTS                                                       5000.00
JEAN CRANDALL

14569877  NC     PRECISION PRODUCTS                                                     100000.00
STEVE BLUNTSEN
```

You can use the UPDATE statement to update ADABAS data. Remember that when you reference a view descriptor in a PROC SQL statement, you are not updating the view descriptor, but rather the ADABAS data described by the view descriptor.

The following UPDATE statements update the values described by the logical record that meets the WHERE clause criteria. The SELECT statement then displays the view's output as shown in Output 4.8 on page 50. The ORDER BY clause in the SELECT statement causes the data to be presented in ascending order by the CUSTOMER data field. (Because you are referencing a view descriptor, you use the SAS variable names for data fields in the UPDATE statement; however, the SQL procedure displays the ADABAS data field names.)

```
update vlib.usacust
   set zipcode=27702
   where custnum='12345678';
update vlib.usacust
   set name='DURHAM SCIENTIFIC SUPPLY COMPANY'
   where custnum='12345678';
update vlib.usacust
   set firstord='02JAN88'd
   where custnum='12345678';
update vlib.usacust
   set limit=5000.00
   where custnum='12345678';
update vlib.usacust
```

```
   set signatur='MARC PLOUGHMAN'
   where custnum='12345678';
update vlib.usacust
   set branch_2='DURHAM'
   where custnum='12345678';
title 'Updated ADABAS Data in CUSTOMERS';
select custnum, state, name, limit, signatur
   from vlib.usacust;
```

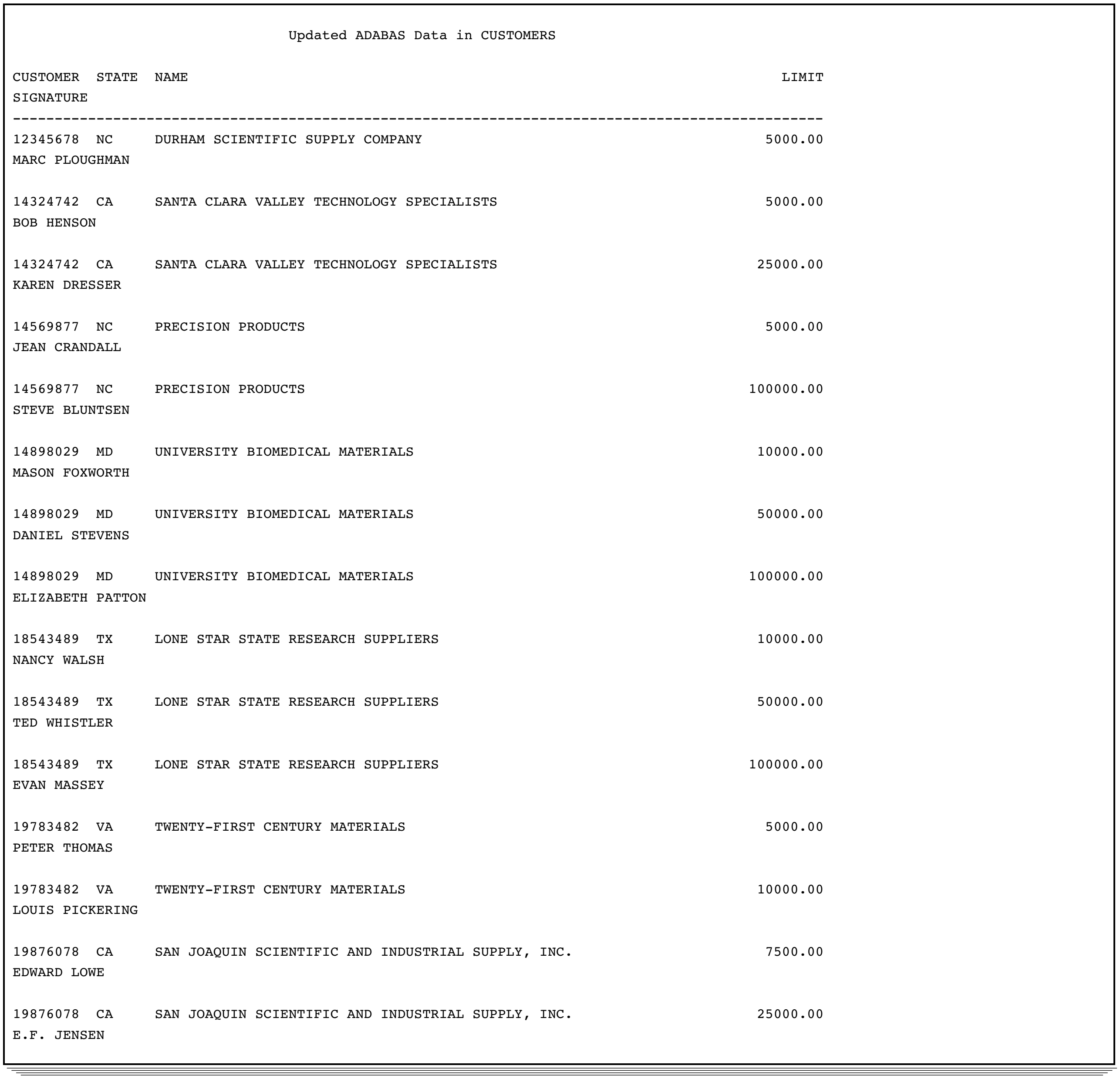
Output 4.8 ADABAS Data Updated by the UPDATE Statement

```
                                 Updated ADABAS Data in CUSTOMERS

CUSTOMER  STATE  NAME                                                                       LIMIT
SIGNATURE
-------------------------------------------------------------------------------------------------
12345678  NC     DURHAM SCIENTIFIC SUPPLY COMPANY                                         5000.00
MARC PLOUGHMAN

14324742  CA     SANTA CLARA VALLEY TECHNOLOGY SPECIALISTS                                5000.00
BOB HENSON

14324742  CA     SANTA CLARA VALLEY TECHNOLOGY SPECIALISTS                               25000.00
KAREN DRESSER

14569877 0 NC     PRECISION PRODUCTS                                                      5000.00
JEAN CRANDALL

14569877  NC     PRECISION PRODUCTS                                                     100000.00
STEVE BLUNTSEN

14898029  MD     UNIVERSITY BIOMEDICAL MATERIALS                                         10000.00
MASON FOXWORTH

14898029  MD     UNIVERSITY BIOMEDICAL MATERIALS                                         50000.00
DANIEL STEVENS

14898029  MD     UNIVERSITY BIOMEDICAL MATERIALS                                        100000.00
ELIZABETH PATTON

15432147  MI     GREAT LAKES LABORATORY EQUIPMENT MANUFACTURERS                          10000.00
JACK TREVANE

18543489  TX     LONE STAR STATE RESEARCH SUPPLIERS                                      10000.00
NANCY WALSH

18543489  TX     LONE STAR STATE RESEARCH SUPPLIERS                                      50000.00
TED WHISTLER

18543489  TX     LONE STAR STATE RESEARCH SUPPLIERS                                     100000.00
EVAN MASSEY

19783482  VA     TWENTY-FIRST CENTURY MATERIALS                                           5000.00
PETER THOMAS

19783482  VA     TWENTY-FIRST CENTURY MATERIALS                                          10000.00
LOUIS PICKERING

19876078  CA     SAN JOAQUIN SCIENTIFIC AND INDUSTRIAL SUPPLY, INC.                       7500.00
EDWARD LOWE

19876078  CA     SAN JOAQUIN SCIENTIFIC AND INDUSTRIAL SUPPLY, INC.                      25000.00
E.F. JENSEN
```

You can use the INSERT statement to add logical records to an ADABAS file or the DELETE statement to remove logical records. In the following example, the logical record containing the CUSTOMER value 15432147 is deleted by using the CUSTOMERS DDM. The SELECT statement then displays the VLIB.USACUST data in Output 4.9 on page 51, ordering them again by the CUSTOMER data field.

```
delete from vlib.usacust
   where custnum='15432147';
   title 'Logical Record Deleted from
      CUSTOMERS';
select custnum, state, name, limit, signatur
   from vlib.usacust;
```

Output 4.9 ADABAS Data with a Logical Record Deleted



```
                              Updated ADABAS Data in CUSTOMERS

CUSTOMER  STATE  NAME                                                                   LIMIT
SIGNATURE
-------------------------------------------------------------------------------------------------
12345678  NC     DURHAM SCIENTIFIC SUPPLY COMPANY                                     5000.00
MARC PLOUGHMAN

14324742  CA     SANTA CLARA VALLEY TECHNOLOGY SPECIALISTS                            5000.00
BOB HENSON

14324742  CA     SANTA CLARA VALLEY TECHNOLOGY SPECIALISTS                           25000.00
KAREN DRESSER

14569877  NC     PRECISION PRODUCTS                                                   5000.00
JEAN CRANDALL

14569877  NC     PRECISION PRODUCTS                                                 100000.00
STEVE BLUNTSEN

14898029  MD     UNIVERSITY BIOMEDICAL MATERIALS                                     10000.00
MASON FOXWORTH

14898029  MD     UNIVERSITY BIOMEDICAL MATERIALS                                     50000.00
DANIEL STEVENS

14898029  MD     UNIVERSITY BIOMEDICAL MATERIALS                                    100000.00
ELIZABETH PATTON

18543489  TX     LONE STAR STATE RESEARCH SUPPLIERS                                  10000.00
NANCY WALSH

18543489  TX     LONE STAR STATE RESEARCH SUPPLIERS                                  50000.00
TED WHISTLER

18543489  TX     LONE STAR STATE RESEARCH SUPPLIERS                                 100000.00
EVAN MASSEY

19783482  VA     TWENTY-FIRST CENTURY MATERIALS                                       5000.00
PETER THOMAS

19783482  VA     TWENTY-FIRST CENTURY MATERIALS                                      10000.00
LOUIS PICKERING

19876078  CA     SAN JOAQUIN SCIENTIFIC AND INDUSTRIAL SUPPLY, INC.                   7500.00
EDWARD LOWE

19876078  CA     SAN JOAQUIN SCIENTIFIC AND INDUSTRIAL SUPPLY, INC.                  25000.00
E.F. JENSEN
```

CAUTION:

Always use the WHERE clause in a DELETE statement. If you omit the WHERE clause from a DELETE statement, you delete *all* the data in the ADABAS file that is accessed by the view descriptor. △

For more information on the SAS System SQL procedure, see the SQL chapter in the *SAS Procedures Guide*.

Appending Data with the APPEND Procedure

In earlier releases of the SAS System, the APPEND procedure operated only on SAS data files. You can append data described by SAS/ACCESS view descriptors and PROC SQL views to SAS data files and vice versa. You can also append data described by view descriptors to each other.

In the following example, two personnel managers have kept separate employee records. One manager has kept records in a NATURAL DDM named EMPLOYEE, described by the view descriptor VLIB.ADAEMPS. The other manager has kept records in a SAS data file named MYDATA.SASEMPS. Due to a corporate reorganization, the two sources of data must be combined so that all employee data are stored in the EMPLOYEE DDM. The APPEND procedure can do this.

The data described by the view descriptor VLIB.ADAEMPS and the data in the SAS data file MYDATA.SASEMPS are printed with the following statements and displayed in Output 4.10 on page 53 and Output 4.11 on page 53:

```
options linesize=80;

proc print data=vlib.adaemps;
   title 'Data Described by VLIB.ADAEMPS';
run;

proc print data=mydata.sasemps;
   format birthdat date7.;
   title 'Data in MYDATA.SASEMPS Data File';
run;
```

Output 4.10 Data Described by VLIB.ADAEMPS

```
                  Data Described by VLIB.ADAEMPS

OBS   EMPID BIRTHDAT LASTNAME           FIRSTNAM        MIDDLENA

  1  119012 05JAN46  WOLF-PROVENZA      G.              ANDREA
  2  120591 12FEB46  HAMMERSTEIN        S.              RACHAEL
  3  123456       .  VARGAS             PAUL            JESUS
  4  127845 25DEC43  MEDER              VLADIMIR        JORAN
  5  129540 31JUL60  CHOULAI            CLARA           JANE
  6  135673 21MAR61  HEMESLY            STEPHANIE       J.
  7  212916 29MAY28  WACHBERGER         MARIE-LOUISE    TERESA
  8  216382 24JUL63  PURINTON           PRUDENCE        VALENTINE
  9  234967 21DEC67  SMITH              GILBERT         IRVINE
 10  237642 13MAR54  BATTERSBY          R.              STEPHEN
 11  239185 28AUG59  DOS REMEDIOS       LEONARD         WESLEY
 12  254896 06APR49  TAYLOR-HUNYADI     ITO             MISHIMA
 13  321783 03JUN35  GONZALES           GUILLERMO       RICARDO
 14  328140 02JUN51  MEDINA-SIDONIA     MARGARET        ROSE
 15  346917 15MAR50  SHIEKELESLAM       SHALA           Y.
 16  356134 25OCT60  DUNNETT            CHRISTINE       MARIE
 17  423286 31OCT64  MIFUNE             YUKIO           TOSHIRO
 18  456910 24SEP53  ARDIS              RICHARD         BINGHAM
 19  456921 12MAY62  KRAUSE             KARL-HEINZ      G.
 20  457232 15OCT63  LOVELL             WILLIAM         SINCLAIR
 21  459287 05JAN34  RODRIGUES          JUAN            M.
 22  677890 24APR65  NISHIMATSU-LYNCH   CAROL           ANNE
```

Output 4.11 Data in MYDATA. SASEMPS

```
               Data in MYDATA.SASEMPS Data File

OBS     EMPID    BIRTHDAT    LASTNAME     FIRSTNAM    MIDDLENA

 1     245962    30AUG64     BEDORTHA     KATHY        MARTHA
 2     765432    01MAR59     POWELL       FRANK        X.
 3     219223    13JUN47     HANSINGER    BENJAMIN     HAROLD
 4     326745    21FEB52     RAWN         BEATRICE     MAY
```

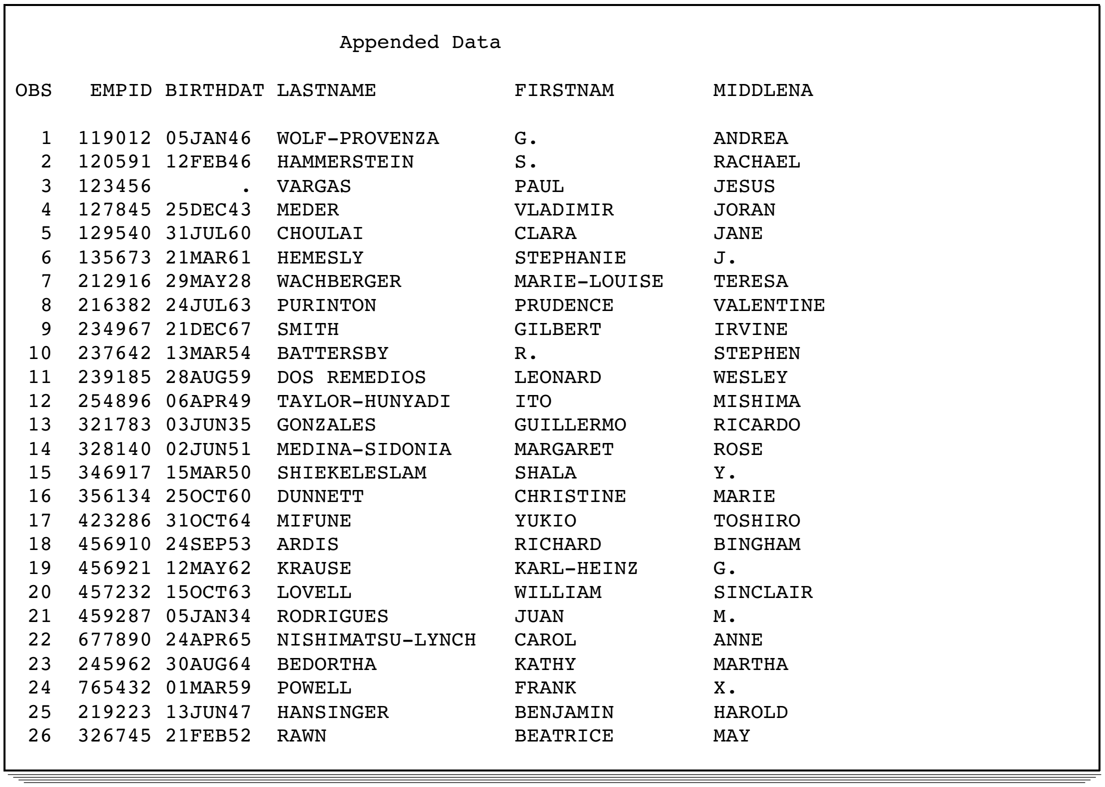
The following statements use the APPEND procedure to combine the data from these two sources:

```
proc append base=vlib.adaemps
   data=mydata.sasemps;
run;
```

```
proc print data=vlib.adaemps;
   title ’Appended Data’;
run;
```

Output 4.12 on page 54 displays the appended data described by the view descriptor VLIB.ADAEMPS. Notice that the data in MYDATA.SASEMPS follow the data described by VLIB.ADAEMPS.

Output 4.12 Appended Data



```
                          Appended Data

OBS   EMPID BIRTHDAT LASTNAME           FIRSTNAM        MIDDLENA

  1  119012 05JAN46  WOLF-PROVENZA      G.              ANDREA
  2  120591 12FEB46  HAMMERSTEIN        S.              RACHAEL
  3  123456       .  VARGAS             PAUL            JESUS
  4  127845 25DEC43  MEDER              VLADIMIR        JORAN
  5  129540 31JUL60  CHOULAI            CLARA           JANE
  6  135673 21MAR61  HEMESLY            STEPHANIE       J.
  7  212916 29MAY28  WACHBERGER         MARIE-LOUISE    TERESA
  8  216382 24JUL63  PURINTON           PRUDENCE        VALENTINE
  9  234967 21DEC67  SMITH              GILBERT         IRVINE
 10  237642 13MAR54  BATTERSBY          R.              STEPHEN
 11  239185 28AUG59  DOS REMEDIOS       LEONARD         WESLEY
 12  254896 06APR49  TAYLOR-HUNYADI     ITO             MISHIMA
 13  321783 03JUN35  GONZALES           GUILLERMO       RICARDO
 14  328140 02JUN51  MEDINA-SIDONIA     MARGARET        ROSE
 15  346917 15MAR50  SHIEKELESLAM       SHALA           Y.
 16  356134 25OCT60  DUNNETT            CHRISTINE       MARIE
 17  423286 31OCT64  MIFUNE             YUKIO           TOSHIRO
 18  456910 24SEP53  ARDIS              RICHARD         BINGHAM
 19  456921 12MAY62  KRAUSE             KARL-HEINZ      G.
 20  457232 15OCT63  LOVELL             WILLIAM         SINCLAIR
 21  459287 05JAN34  RODRIGUES          JUAN            M.
 22  677890 24APR65  NISHIMATSU-LYNCH   CAROL           ANNE
 23  245962 30AUG64  BEDORTHA           KATHY           MARTHA
 24  765432 01MAR59  POWELL             FRANK           X.
 25  219223 13JUN47  HANSINGER          BENJAMIN        HAROLD
 26  326745 21FEB52  RAWN               BEATRICE        MAY
```

The APPEND procedure also accepts a WHERE= SAS data set option or a SAS WHERE statement to retrieve a subset of the data. In the following example, a subset of the observations from the DATA= data set is added to the BASE= data set. The results are displayed in Output 4.13 on page 55.

```
proc append base=vlib.adaemps
   data=mydata.sasemps
      (where=(birthdat>=’01JAN60’d));
run;
proc print data=vlib.adaemps;
   title ’Appended Data with a WHERE= Data Set
```

```
        Option';
run;
```

Output 4.13 Appended Data with a WHERE= Data Set Option

```
            Appended Data with a WHERE= Data Set Option

OBS   EMPID BIRTHDAT LASTNAME           FIRSTNAM        MIDDLENA

  1  119012 05JAN46  WOLF-PROVENZA      G.              ANDREA
  2  120591 12FEB46  HAMMERSTEIN        S.              RACHAEL
  3  123456        .  VARGAS             PAUL            JESUS
  4  127845 25DEC43  MEDER              VLADIMIR        JORAN
  5  129540 31JUL60  CHOULAI            CLARA           JANE
  6  135673 21MAR61  HEMESLY            STEPHANIE       J.
  7  212916 29MAY28  WACHBERGER         MARIE-LOUISE    TERESA
  8  216382 24JUL63  PURINTON           PRUDENCE        VALENTINE
  9  234967 21DEC67  SMITH              GILBERT         IRVINE
 10  237642 13MAR54  BATTERSBY          R.              STEPHEN
 11  239185 28AUG59  DOS REMEDIOS       LEONARD         WESLEY
 12  254896 06APR49  TAYLOR-HUNYADI     ITO             MISHIMA
 13  321783 03JUN35  GONZALES           GUILLERMO       RICARDO
 14  328140 02JUN51  MEDINA-SIDONIA     MARGARET        ROSE
 15  346917 15MAR50  SHIEKELESLAM       SHALA           Y.
 16  356134 25OCT60  DUNNETT            CHRISTINE       MARIE
 17  423286 31OCT64  MIFUNE             YUKIO           TOSHIRO
 18  456910 24SEP53  ARDIS              RICHARD         BINGHAM
 19  456921 12MAY62  KRAUSE             KARL-HEINZ      G.
 20  457232 15OCT63  LOVELL             WILLIAM         SINCLAIR
 21  459287 05JAN34  RODRIGUES          JUAN            M.
 22  677890 24APR65  NISHIMATSU-LYNCH   CAROL           ANNE
 23  245962 30AUG64  BEDORTHA           KATHY           MARTHA
```

For more information on the APPEND procedure, see the *SAS Procedures Guide.*

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ACCESS® Interface to ADABAS Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/ACCESS® Interface to ADABAS Software: Reference, Version 8

ISBN 1–58025–546–9

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.