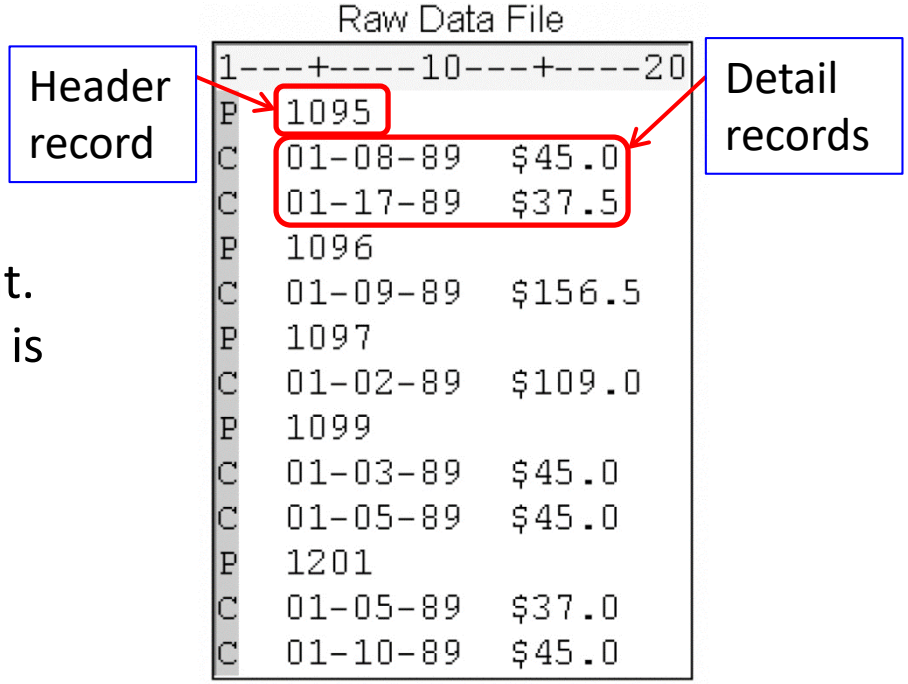
Chapter 21

Reading Hierarchical Files

Overview

Raw data files can be hierarchical in structure, consisting of a header record and one or more detail records. Typically, each record contains a field that identifies the record type.

For example, in the following data file the first column is a letter indicating the record type. Letter P indicates a header record that contains a patient's ID number. Letter C indicates a detail record that contains the date of the patient's appointment and the charge that the patient has incurred.

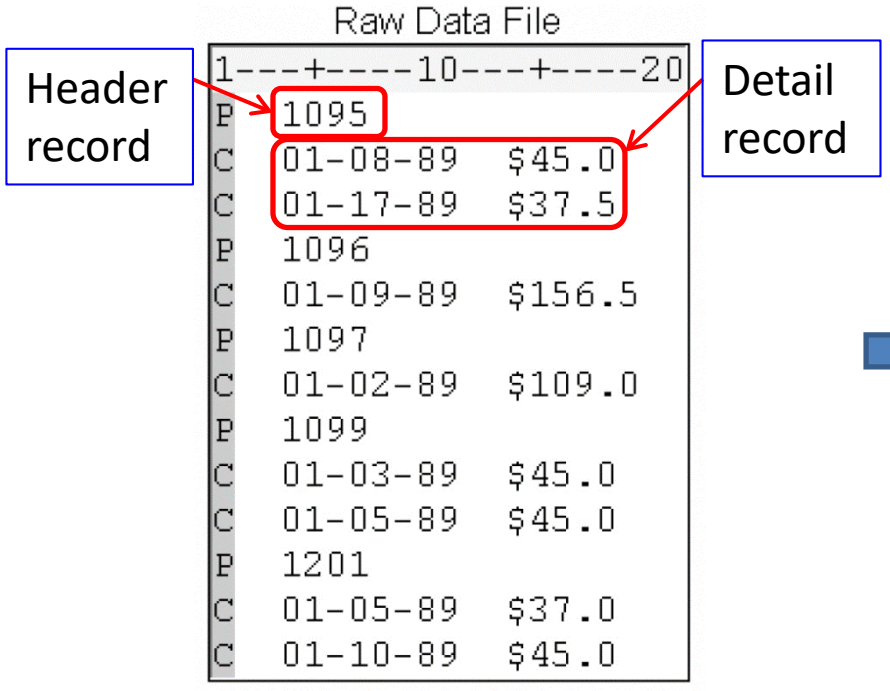


Chapter topics:

- Retain the value of a variable.
- Conditionally execute a SAS statement.
- Determine when the last observation is being processed.
- Conditionally execute multiple SAS statements to read hierarchical raw data.

Creating One Observation per Detail Record

You can build a SAS data set from a hierarchical file by creating one observation per detail record and storing each header record as part of the observation.



Obs	ID	Date	Amount
1	1095	01/08/89	$45.00
2	1095	01/17/89	$37.50
3	1096	01/09/89	$156.50
4	1097	01/02/89	$109.00
5	1099	01/03/89	$45.00
6	1099	01/05/89	$45.00
7	1201	01/05/89	$37.00
8	1201	01/10/89	$45.00

Creating One Observation per Detail Record

Another example: a partial census data file, in which **H** indicates a header record that contains a street address, and **P** indicates a detail record that contains information about a person who lives at that address.

Header record

Raw Data File

```
1---+----10---+----
H 321 S. MAIN ST
P MARY E     21 F
P WILLIAM M  23 M
P SUSAN K     3 F
H 324 S. MAIN ST
P THOMAS H   79 M
P WALTER S   46 M
P ALICE A    42 F
P MARYANN A  20 F
P JOHN S     16 M
H 325A S. MAIN ST
```

Detail records

Obs	Address	Name	Age	Gender
1	321 S. MAIN ST	MARY E	21	F
2	321 S. MAIN ST	WILLIAM M	23	M
3	321 S. MAIN ST	SUSAN K	3	F
4	324 S. MAIN ST	THOMAS H	79	M
5	324 S. MAIN ST	WALTER S	46	M
6	324 S. MAIN ST	ALICE A	42	F
7	324 S. MAIN ST	MARYANN A	20	F
8	324 S. MAIN ST	JOHN S	16	M
9	325A S. MAIN ST	JAMES L	34	M
10	325A S. MAIN ST	LIZA A	31	F
11	325B S. MAIN ST	MARGO K	27	F

Creating One Observation per Detail Record

To achieve the result, you need to do the following:

- Use a **retain** statement to retain the values for Address until the next header record is encountered.
- Use an **@ line-hold specifier** to hold the current record so that the other values in the record can be read later.
- Conditionally execute INPUT statements based on the value of first column.

Raw Data File

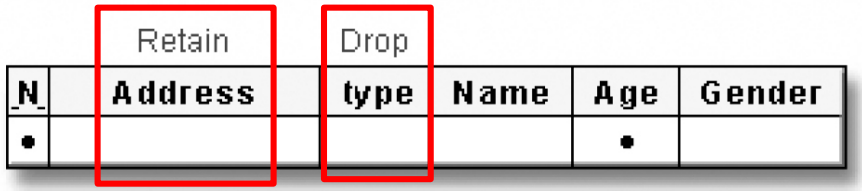
```
1---+----10---+----
H 321 S. MAIN ST
P MARY E    21 F
P WILLIAM M 23 M
P SUSAN K    3 F
H 324 S. MAIN ST
P THOMAS H  79 M
P WALTER S  46 M
P ALICE A   42 F
P MARYANN A 20 F
P JOHN S    16 M
H 325A S. MAIN ST
```

```
DATA perm.people (drop=type);
        infile census;
        retain Address;
        input type $1. @;
         if type='H' then input @3 Address $15.;
         if type='P';  /*subsetting IF statement*/
                input @3 Name $10. @13 Age 3.
                @16 Gender $1.;
run;
```

Processing of the DATA Step

At compilation time, the variable **type** is flagged so that its values are not written to the data set. **Address** is flagged so that its value is retained across iterations of the DATA step.

```
>----+----10---+----
H 321 S. MAIN ST
P MARY E    21  F
P WILLIAM M 23  M
P SUSAN K    3  F
H 324 S. MAIN ST
```




```
DATA perm.people (drop=type);
        infile census;
        retain Address;
        input type $1. @;
         if type='H' then input @3 Address $15.;
         if type='P';  /*subsetting IF statement*/
                input @3 Name $10. @13 Age 3.
                @16 Gender $1.;
run;
```

Processing of the DATA Step

As the DATA step begins to execute, the INPUT statement reads the value for type and holds the first record.

```
>----+----10---+----
H 321 S. MAIN ST
P MARY E    21 F
P WILLIAM M 23 M
P SUSAN K    3 F
H 324 S. MAIN ST
```

```
DATA perm.people (drop=type);
        infile census;
        retain Address;
        input type $1. @;
         if type='H' then input @3 Address $15.;
         if type='P';  /*subsetting IF statement*/
                input @3 Name $10. @13 Age 3.
                @16 Gender $1.;
run;
```

Retain Drop

_N_	Address	type	Name	Age	Gender
1		H		.	

Processing of the DATA Step

The condition type='H' is checked and found to be true, so the INPUT statement reads the value for Address in the first record.

```
>----+----10---+----
 H 321 S. MAIN ST
 P MARY E    21  F
 P WILLIAM M 23  M
 P SUSAN K    3  F
 H 324 S. MAIN ST
```

```
DATA perm.people (drop=type);
        infile census;
        retain Address;
        input type $1. @;
         if type='H' then input @3 Address $15.;
         if type='P';  /*subsetting IF statement*/
                input @3 Name $10. @13 Age 3.
                @16 Gender $1.;
run;
```

Retain Drop

_N_	Address	type	Name	Age	Gender
1	321 S. MAIN ST	H		.	

Processing of the DATA Step

Next, the subsetting IF statement checks for the condition type='P'. Because the condition is not true, the remaining statements are not executed and control returns to the top of the DATA step. The PDV is initialized but Address is retained

```
>----+----10---+----
 H 321 S. MAIN ST
 P MARY E    21  F
 P WILLIAM M 23  M
 P SUSAN K    3  F
 H 324 S. MAIN ST
```

```
DATA perm.people (drop=type);
        infile census;
        retain Address;
        input type $1. @;
         if type='H' then input @3 Address $15.;
         if type='P';  /*subsetting IF statement*/
                input @3 Name $10. @13 Age 3.
                 @16 Gender $1.;
run;
```

Retain Drop

_N_	Address	type	Name	Age	Gender
2	321 S. MAIN ST			•	

Processing of the DATA Step

As the second iteration begins, the input pointer moves to the next record and a new value for type is read. The condition expressed in the IF-THEN statement is not true, so the statement following the THEN keyword is not executed.

```
>----+----10---+----
H 321 S. MAIN ST
P MARY E    21  F
P WILLIAM M 23  M
P SUSAN K    3  F
H 324 S. MAIN ST
```

```
DATA perm.people (drop=type);
        infile census;
        retain Address;
        input type $1. @;
         if type='H' then input @3 Address $15.;
         if type='P';  /*subsetting IF statement*/
                input @3 Name $10. @13 Age 3.
                @16 Gender $1.;
run;
```

Retain Drop

_N_	Address	type	Name	Age	Gender
2	321 S. MAIN ST	P		.	

Processing of the DATA Step

Now the subsetting IF statement checks for the condition type='P'. In this iteration, the condition is true, so the final INPUT statement reads the values for Name, Age, and Gender.

```
>----+----10---+----
 H 321 S. MAIN ST
 P MARY E    21 F
 P WILLIAM M 23 M
 P SUSAN K    3 F
 H 324 S. MAIN ST
```

```
DATA perm.people (drop=type);
        infile census;
        retain Address;
        input type $1. @;
         if type='H' then input @3 Address $15.;
         if type='P';  /*subsetting IF statement*/
                input @3 Name $10. @13 Age 3.
                @16 Gender $1.;
run;
```

Retain Drop

N	Address	type	Name	Age	Gender
2	321 S. MAIN ST	P	MARY E	21	F

Processing of the DATA Step

Then the values in the program data vector are written as the first observation, and control returns to the top of the DATA step. Notice that the value for the variable type is not included.

```
>----+----10---+----
 H 321 S. MAIN ST
 P MARY E     21 F
 P WILLIAM M 23 M
 P SUSAN K     3 F
 H 324 S. MAIN ST
```

Retain Drop

_N_	Address	type	Name	Age	Gender
2	321 S. MAIN ST	P	MARY E	21	F

SAS Data Set Perm.People

Obs	Address	Name	Age	Gender
1	321 S. MAIN ST	MARY E	21	F

```
DATA perm.people (drop=type);
        infile census;
        retain Address;
        input type $1. @;
         if type='H' then input @3 Address $15.;
         if type='P';  /*subsetting IF statement*/
                input @3 Name $10. @13 Age 3.
                @16 Gender $1.;
run;
```

Processing of the DATA Step

As execution continues, observations are produced from the third and fourth records. However, notice that the fifth record is a header record. During the fifth iteration, the condition type='H' is true, so a new Address is read into the program data vector, overwriting the previous value.

```
DATA perm.people (drop=type);
        infile census;
        retain Address;
        input type $1. @;
           if type='H' then input @3 Address $15.;
           if type='P';  /*subsetting IF statement*/
                   input @3 Name $10. @13 Age 3.
                    @16 Gender $1.;
run;
```

```
>----+----10---+----
 H 321 S. MAIN ST
 P MARY E    21 F
 P WILLIAM M 23 M
 P SUSAN K    3 F
 H 324 S. MAIN ST
```

Retain Drop

_N_	Address	type	Name	Age	Gender
5	324 S. MAIN ST	H		.	

SAS Data Set Perm.People

Obs	Address	Name	Age	Gender
1	321 S. MAIN ST	MARY E	21	F
2	321 S. MAIN ST	WILLIAM M	23	M
3	321 S. MAIN ST	SUSAN K	3	F

The execution continues till the end of the raw data file is reached.

Creating One Observation per Header Record

In the previous example, we created one observation per detail record, now suppose we only want to know how many people reside at each address. We can create a data set that reads each detail record, counts the number of people, and stores this number in a summary variable.

Raw Data File

```
1---+----10---+----20
H 321 S. MAIN ST
P MARY E    21 F
P WILLIAM M 23 M
P SUSAN K    3 F
H 324 S. MAIN ST
P THOMAS H  79 M
P WALTER S  46 M
P ALICE A   42 F
P MARYANN A 20 F
P JOHN S    16 M
H 325A S. MAIN ST
P JAMES L 34 M
P LIZA A 31 F
H 325B S. MAIN ST
P MARGO K 27 F
P WILLIAM R 27 M
P ROBERT W 1 M
```

Address	total
321 S MAIN ST	3
324 S MAIN ST	5
325A S MAIN ST	2
325B S MAIN ST	3

Creating One Observation per Header Record

In writing the DATA step to create such a data set, there are several tasks:

- The value of Address must be retained as detail records are read and summarized. (**retain Address;**)
- The value of type must be read in order to determine whether the current record is a header record or a detail record. Add an @ to hold the record so that another INPUT statement can read the remaining values. (**input type $1. @;**)
- When the value of type indicates a header record, several statements need to be executed. (**if type='H' then do;**)
- When the value of type indicates a detail record, you need to define an alternative set of actions. (**else if type='P' then**)

Creating One Observation per Header Record

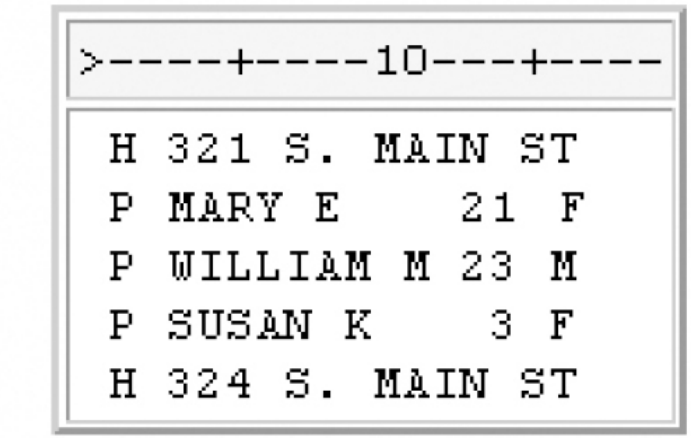
```
data perm.residnts (drop=type);
  infile census end=last; /*To determine the end of a file*/
  retain Address;
  input type $1. @;
  if type='H' then do; /*DO group*/
       if _n_ > 1 then output;
       total=0;      /*Initialize the summary variable*/
       input address $ 3-17;
     end;
  else if type='P' then total+1; /*Sum statement*/
  if last then output;
run;
```

```
1---+----10---+----20
H 321 S. MAIN ST
P MARY E     21 F
P WILLIAM M 23 M
P SUSAN K     3 F
H 324 S. MAIN ST
P THOMAS H   79 M
P WALTER S   46 M
P ALICE A    42 F
P MARYANN A 20 F
P JOHN S     16 M
H 325A S. MAIN ST
P JAMES L 34 M
P LIZA A 31 F
H 325B S. MAIN ST
P MARGO K 27 F
P WILLIAM R 27 M
P ROBERT W 1 M
```

Processing a DATA Step That Creates One Observation per Header Record

During the compilation phase, the variable **type** is flagged so that later it can be dropped. The values for **Address** and **Total** (SUM statement) are retained.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
      if _n_ > 1 then output;
      total=0;
      input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```



Processing a DATA Step That Creates One Observation per Header Record

As the execution begins, _N_ is 1 and last is 0. Total is initialized to 0 because of the sum statement.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
      if _n_ > 1 then output;
      total=0;
      input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```

```
>V---+----10---+----
H 321 S. MAIN ST
P MARY E     21  F
P WILLIAM M 23  M
P SUSAN K     3  F
H 324 S. MAIN ST
```

	Retain		Drop	Retain
_N_	last	Address	type	Total
1	0			0

Processing a DATA Step That Creates One Observation per Header Record

Now the value for type is read, the condition type='H' is true, and therefore the statements in the DO group execute.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
      if _n_ > 1 then output;
      total=0;
      input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```

```
>V---+----10---+----
H 321 S. MAIN ST
P MARY E     21  F
P WILLIAM M 23  M
P SUSAN K     3  F
H 324 S. MAIN ST
```

	Retain		Drop	Retain
_N_	last	Address	type	Total
1	0		H	0

Processing a DATA Step That Creates One Observation per Header Record

The condition _N_>1 is not true, so the OUTPUT statement is not executed. Total is assigned the value of 0, and the value for Address is read.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
      if _n_ > 1 then output;
      total=0;
      input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```

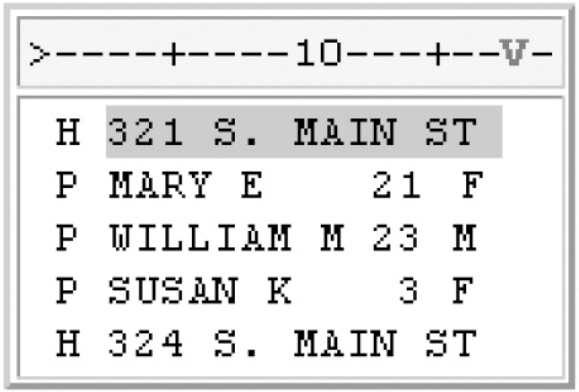
```
>----+----10---+--V-
H 321 S. MAIN ST
P MARY E     21  F
P WILLIAM M 23  M
P SUSAN K     3  F
H 324 S. MAIN ST
```

	Retain		Drop	Retain
_N_	last	Address	type	Total
1	0	321 S. MAIN ST	H	0

Processing a DATA Step That Creates One Observation per Header Record

The END statement closes the DO group. The alternative condition expressed in the ELSE statement is not checked because the first condition, type='H', was true.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
      if _n_ > 1 then output;
      total=0;
      input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```



		Retain		Drop	Retain
_N_	last	Address		type	Total
1	0	321 S. MAIN ST		H	0

Processing a DATA Step That Creates One Observation per Header Record

The value of **last** is still **0**, so the OUTPUT statement is not executed. Control returns to the top of the DATA step. _N_ =2.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
      if _n_ > 1 then output;
      total=0;
      input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```

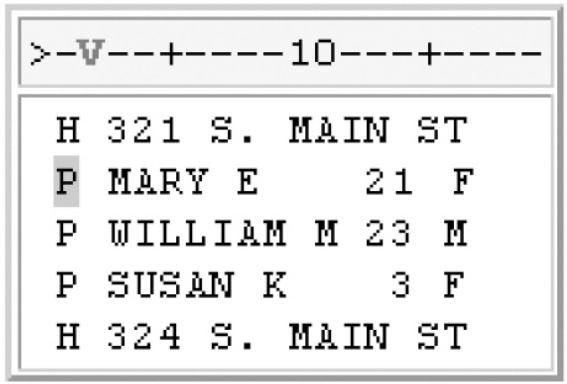
```
>----+----10---+--V-
 H 321 S. MAIN ST
 P MARY E     21  F
 P WILLIAM M 23  M
 P SUSAN K     3  F
 H 324 S. MAIN ST
```

	Retain		Drop	Retain
_N_	last	Address	type	Total
2	0	321 S. MAIN ST		0

Processing a DATA Step That Creates One Observation per Header Record

During the second iteration, the value of type is **'P'** and Total is incremented by 1. Again, the value of last is **0**, so control returns to the top of the DATA step and type is set to missing.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
      if _n_ > 1 then output;
      total=0;
      input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```



	Retain		Drop	Retain
_N_	last	Address	type	Total
2	0	321 S. MAIN ST	P	1

	Retain		Drop	Retain
_N_	last	Address	type	Total
2	0	321 S. MAIN ST		1

Processing a DATA Step That Creates One Observation per Header Record

During the third iteration, the value of type is **'P'** and Total is incremented by 1. Again, the value of last is **0**, so control returns to the top of the DATA step and type is set to missing.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
       if _n_ > 1 then output;
       total=0;
       input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```

```
>----+----10---+----
 H 321 S. MAIN ST
 P MARY E     21  F
 P WILLIAM M 23  M
 P SUSAN K     3  F
 H 324 S. MAIN ST
```

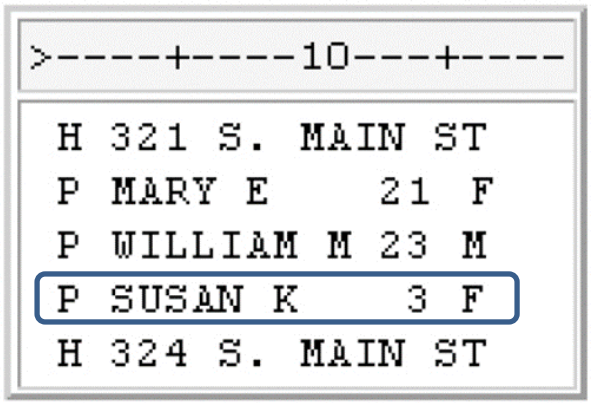
	Retain		Drop	Retain
_N_	last	Address	type	Total
3	0	321 S. MAIN ST	P	2

	Retain		Drop	Retain
_N_	last	Address	type	Total
3	0	321 S. MAIN ST		2

Processing a DATA Step That Creates One Observation per Header Record

During the fourth iteration, the value of type is **'P'** and Total is incremented by 1. Again, the value of last is **0**, so control returns to the top of the DATA step and type is set to missing.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
       if _n_ > 1 then output;
       total=0;
       input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```



```
>----+----10---+----
H 321 S. MAIN ST
P MARY E     21  F
P WILLIAM M 23  M
P SUSAN K     3  F
H 324 S. MAIN ST
```

			Retain	Drop	Retain
_N_	last	Address	type	Total
4	0	321 S. MAIN ST	P	3

			Retain	Drop	Retain
_N_	last	Address	type	Total
4	0	321 S. MAIN ST		3

Processing a DATA Step That Creates One Observation per Header Record

During the fifth iteration, the value of type is **'H'** and **_N_** =5 > 1, so the output statement is executed and the values for Address and Total are written to the data set as the first observation.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
      if _n_ > 1 then output;
      total=0;
      input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```

```
>-V--+----10---+----
 H 321 S. MAIN ST
 P MARY E     21  F
 P WILLIAM M 23  M
 P SUSAN K     3  F
 H 324 S. MAIN ST
```

Retain Drop Retain

_N_	last	Address	type	Total
5	0	321 S. MAIN ST	H	3

SAS Data Set Perm.People

Obs	Address	Total
1	321 S. MAIN ST	3

Processing a DATA Step That Creates One Observation per Header Record

As the last record in the file is read, the variable last is set to **1**. Now that the condition for last is true, the values in the program data vector are written to the data set as the final observation.

```
data perm.residnts (drop=type);
  infile census end=last;
  retain Address;
  input type $1. @;
  if type='H' then do;
      if _n_ > 1 then output;
      total=0;
      input address $ 3-17;
  end;
  else if type='P' then total+1;
  if last then output;
run;
```

```
>-V--+----10---+----
 P LIZA A     31 F
 H 325B S. MAIN ST
 P MARGO K    27 F
 P WILLIAM R 27 F
 P ROBERT W    1 M
```

Retain Drop Retain

_N_	last	Address	type	Total
17	1	325B S. MAIN ST	P	3

SAS Data Set Perm.People

Obs	Address	Total
1	321 S. MAIN ST	3
.		
4	325B S. MAIN ST	3