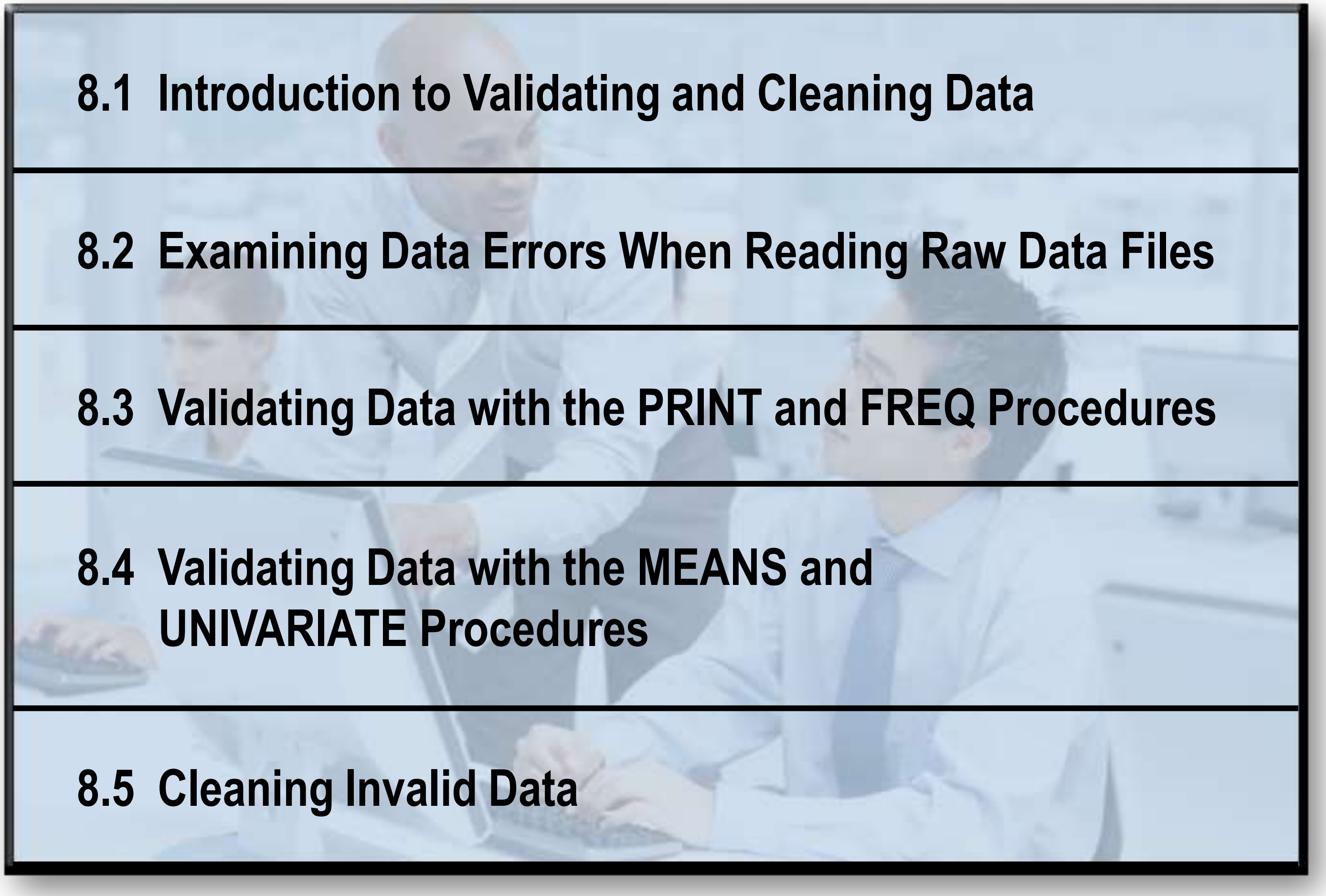
STAT604

Lesson SAS 16

Chapter 8: Validating and Cleaning Data



8.1 Introduction to Validating and Cleaning Data

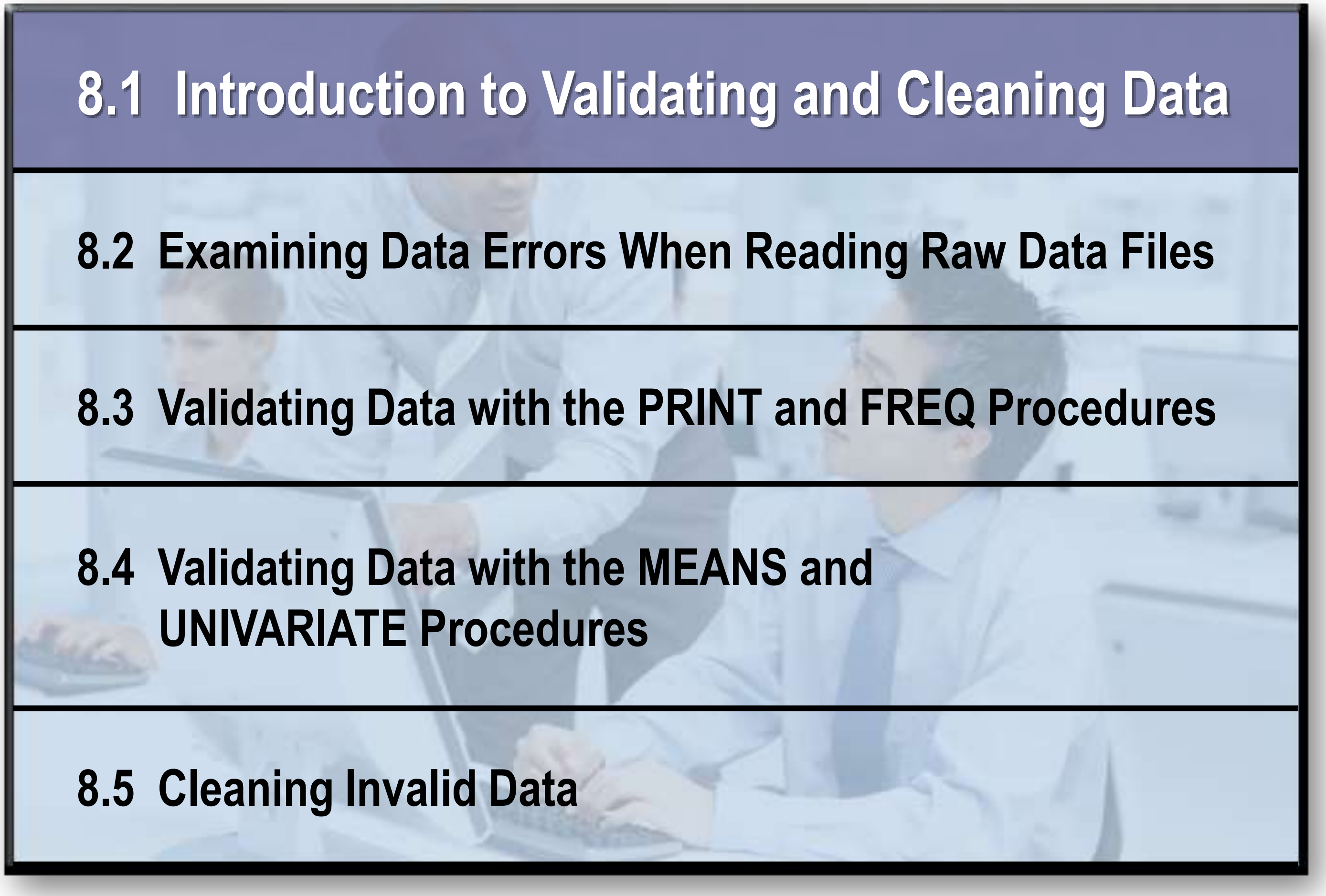
8.2 Examining Data Errors When Reading Raw Data Files

8.3 Validating Data with the PRINT and FREQ Procedures

8.4 Validating Data with the MEANS and UNIVARIATE Procedures

8.5 Cleaning Invalid Data

Chapter 8: Validating and Cleaning Data



8.1 Introduction to Validating and Cleaning Data

8.2 Examining Data Errors When Reading Raw Data Files

8.3 Validating Data with the PRINT and FREQ Procedures

8.4 Validating Data with the MEANS and UNIVARIATE Procedures

8.5 Cleaning Invalid Data

Objectives

- Identify procedures for validating data.
- Identify techniques for cleaning data.
- Define the business scenario that will be used with validating and cleaning data.

Business Scenario

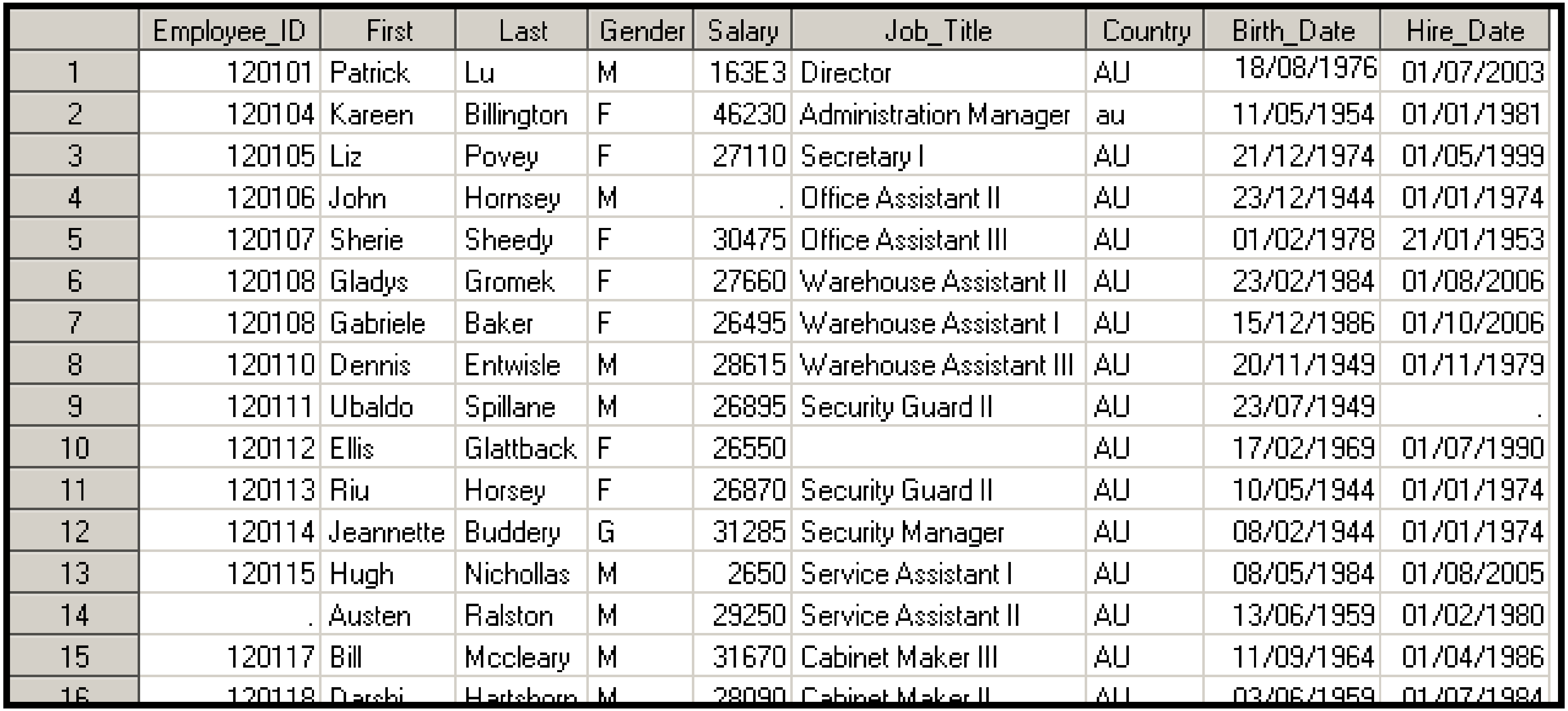
Additional requirements of non-sales employee data:

- **`Employee_ID`** must be unique and not missing.
- **`Gender`** must have a value of `F` or `M`.
- **`Salary`** must be in the numeric range of 24000 – 500000.
- **`Job_Title`** must not be missing.
- **`Country`** must have a value of `AU` or `US`.
- **`Birth_Date`** value must occur before **`Hire_Date`** value.
- **`Hire_Date`** must have a value of 01/01/1974 or later.

Poll
Quiz

8.02 Quiz

What problems exist with the data in this partial data set?

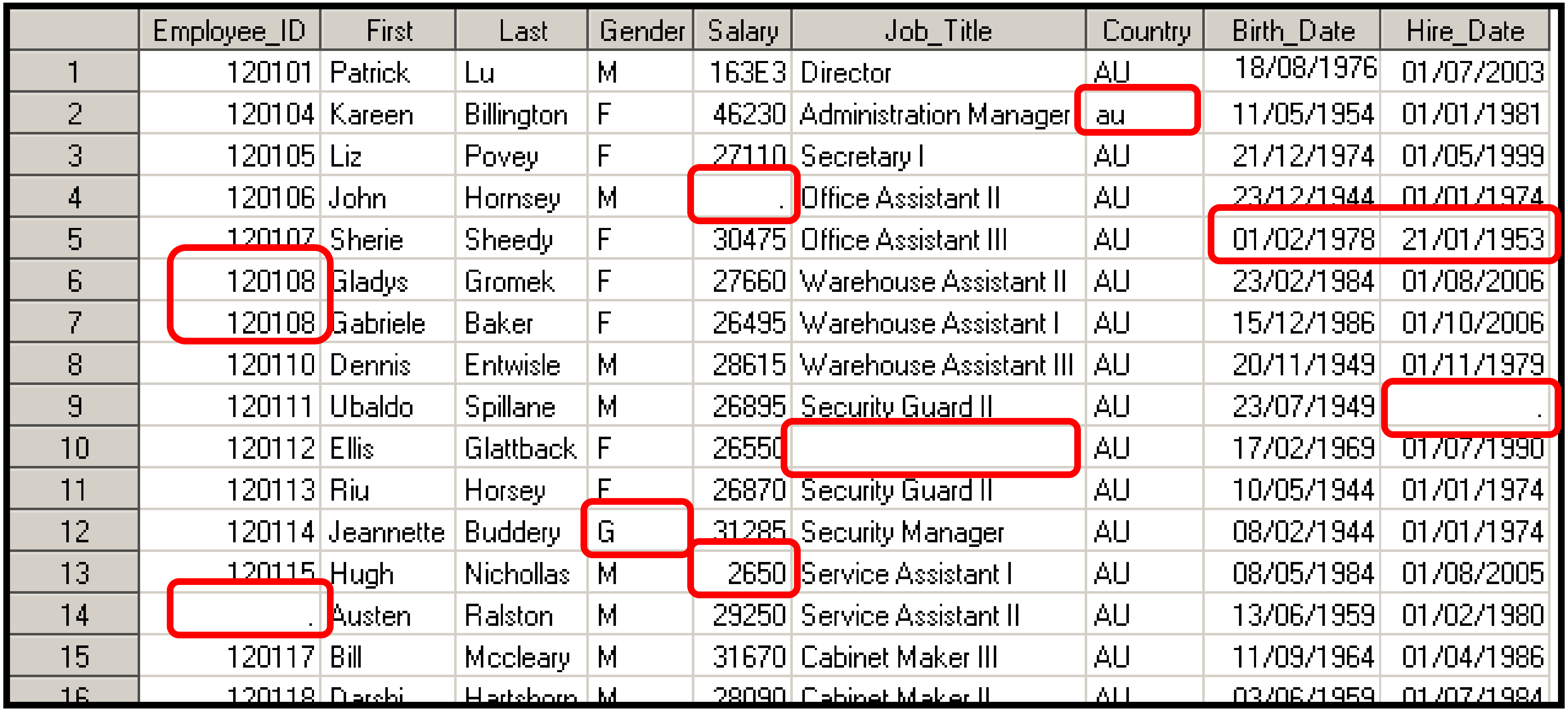


	Employee_ID	First	Last	Gender	Salary	Job_Title	Country	Birth_Date	Hire_Date
1	120101	Patrick	Lu	M	163E3	Director	AU	18/08/1976	01/07/2003
2	120104	Kareen	Billington	F	46230	Administration Manager	au	11/05/1954	01/01/1981
3	120105	Liz	Povey	F	27110	Secretary I	AU	21/12/1974	01/05/1999
4	120106	John	Hornsey	M	.	Office Assistant II	AU	23/12/1944	01/01/1974
5	120107	Sherie	Sheedy	F	30475	Office Assistant III	AU	01/02/1978	21/01/1953
6	120108	Gladys	Gromek	F	27660	Warehouse Assistant II	AU	23/02/1984	01/08/2006
7	120108	Gabriele	Baker	F	26495	Warehouse Assistant I	AU	15/12/1986	01/10/2006
8	120110	Dennis	Entwisle	M	28615	Warehouse Assistant III	AU	20/11/1949	01/11/1979
9	120111	Ubaldo	Spillane	M	26895	Security Guard II	AU	23/07/1949	.
10	120112	Ellis	Glattback	F	26550		AU	17/02/1969	01/07/1990
11	120113	Riu	Horsey	F	26870	Security Guard II	AU	10/05/1944	01/01/1974
12	120114	Jeannette	Buddery	G	31285	Security Manager	AU	08/02/1944	01/01/1974
13	120115	Hugh	Nichollas	M	2650	Service Assistant I	AU	08/05/1984	01/08/2005
14	.	Austen	Ralston	M	29250	Service Assistant II	AU	13/06/1959	01/02/1980
15	120117	Bill	Mccleary	M	31670	Cabinet Maker III	AU	11/09/1964	01/04/1986
16	120118	Darshi	Hartshorn	M	28090	Cabinet Maker II	AU	03/06/1959	01/07/1984

Hint: There are nine data problems.

8.02 Quiz – Correct Answer

What problems exist with the data in this partial data set?



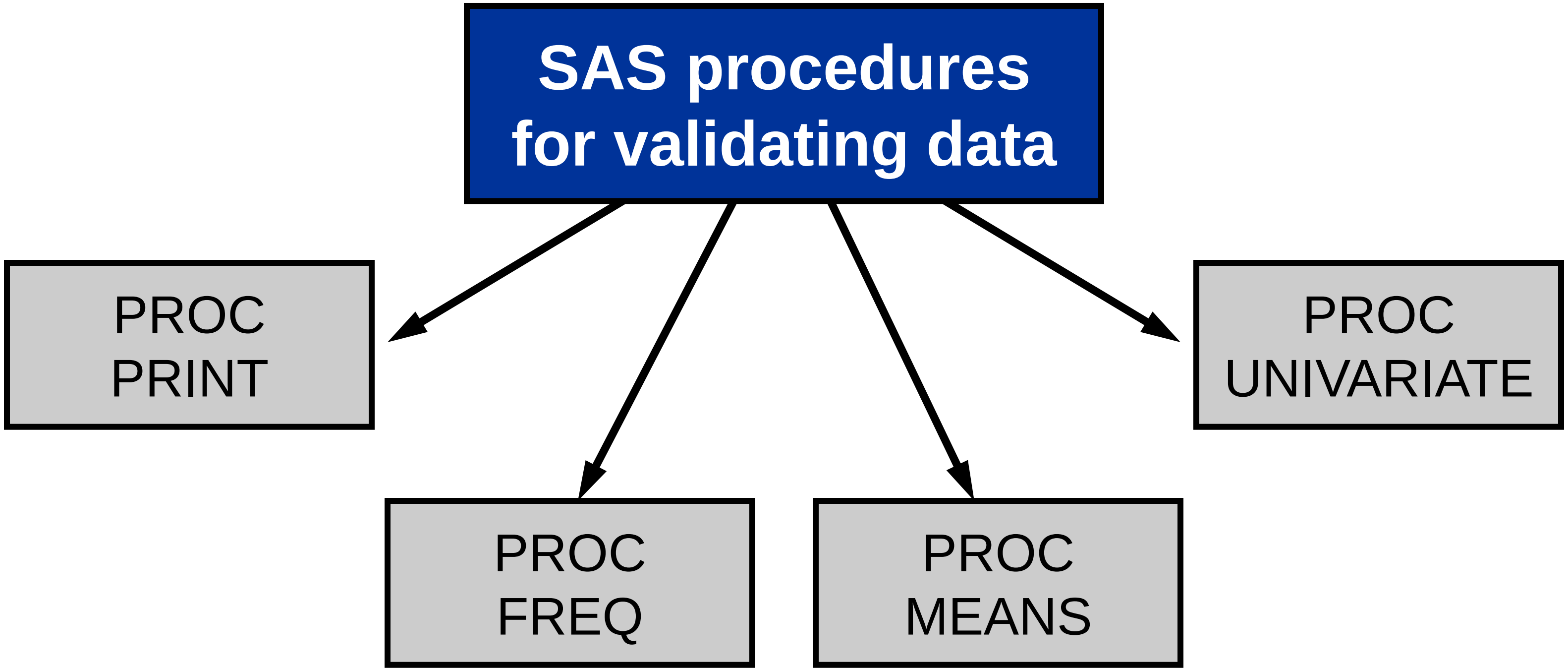
	Employee_ID	First	Last	Gender	Salary	Job_Title	Country	Birth_Date	Hire_Date
1	120101	Patrick	Lu	M	163E3	Director	AU	18/08/1976	01/07/2003
2	120104	Kareen	Billington	F	46230	Administration Manager	au	11/05/1954	01/01/1981
3	120105	Liz	Povey	F	27110	Secretary I	AU	21/12/1974	01/05/1999
4	120106	John	Hornsey	M	.	Office Assistant II	AU	23/12/1944	01/01/1974
5	120107	Sherie	Sheedy	F	30475	Office Assistant III	AU	01/02/1978	21/01/1953
6	120108	Gladys	Gromek	F	27660	Warehouse Assistant II	AU	23/02/1984	01/08/2006
7	120108	Gabriele	Baker	F	26495	Warehouse Assistant I	AU	15/12/1986	01/10/2006
8	120110	Dennis	Entwisle	M	28615	Warehouse Assistant III	AU	20/11/1949	01/11/1979
9	120111	Ubaldo	Spillane	M	26895	Security Guard II	AU	23/07/1949	.
10	120112	Ellis	Glattback	F	26550		AU	17/02/1969	01/07/1990
11	120113	Riu	Horsey	F	26870	Security Guard II	AU	10/05/1944	01/01/1974
12	120114	Jeannette	Buddery	G	31285	Security Manager	AU	08/02/1944	01/01/1974
13	120115	Hugh	Nichollas	M	2650	Service Assistant I	AU	08/05/1984	01/08/2005
14	.	Austen	Ralston	M	29250	Service Assistant II	AU	13/06/1959	01/02/1980
15	120117	Bill	Mccleary	M	31670	Cabinet Maker III	AU	11/09/1964	01/04/1986
16	120118	Darshi	Hartshorn	M	28090	Cabinet Maker II	AU	03/06/1959	01/07/1984

Hint: There are nine data problems.

Validating the Data

In general, SAS procedures analyze data, produce output, or manage SAS files.

In addition, SAS procedures can be used to detect invalid data.



The PRINT Procedure

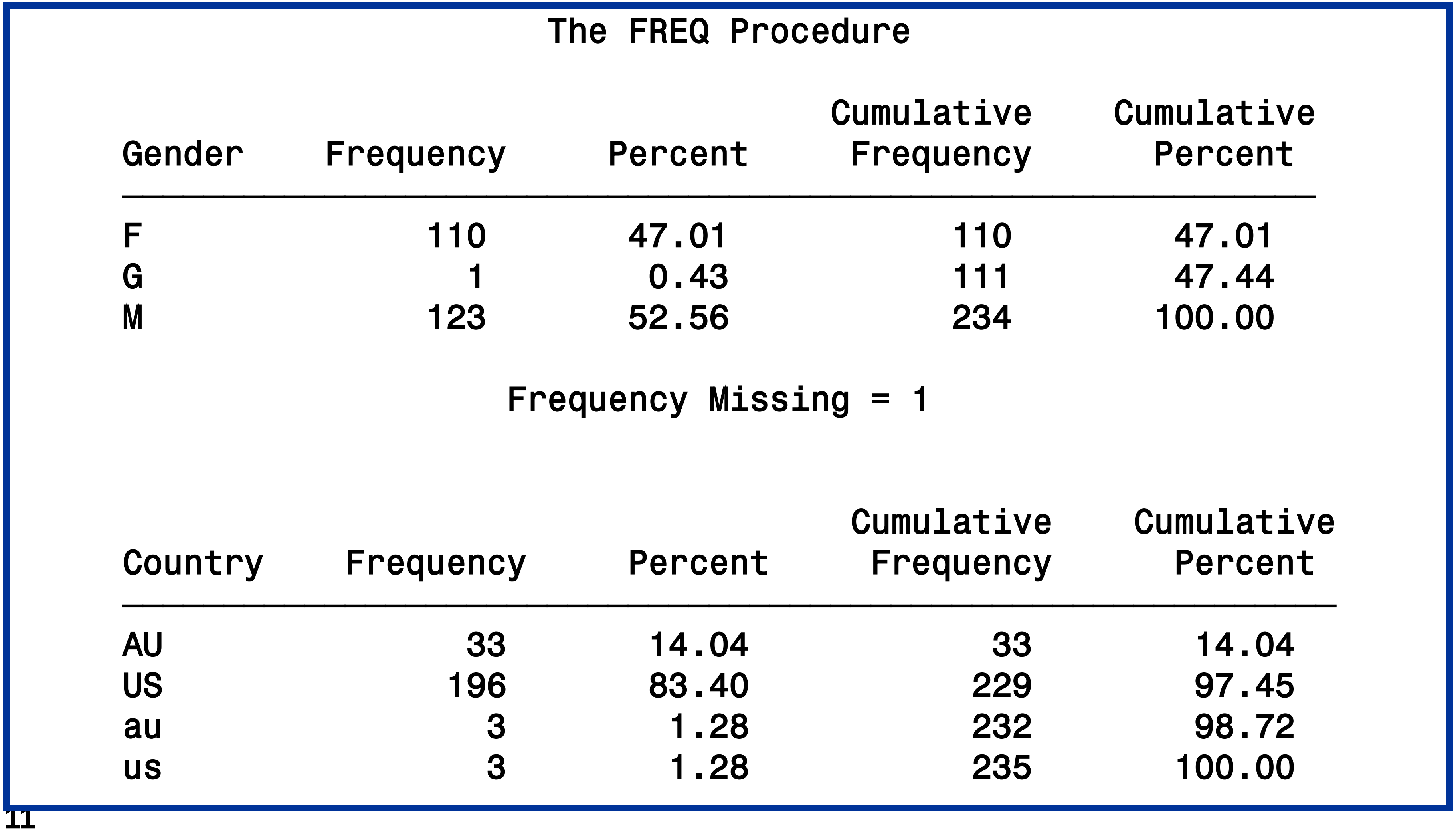
The PRINT procedure can show the job titles that are missing and the hire dates that occur before the birth dates.

```
       Employee_
Obs       ID               Job_Title            Birth_Date      Hire_Date

  5     120107      Office Assistant III    01/02/1978     21/01/1953
  9     120111      Security Guard II       23/07/1949              .
 10     120112                              17/02/1969     01/07/1990
```

p108d01

The FREQ Procedure

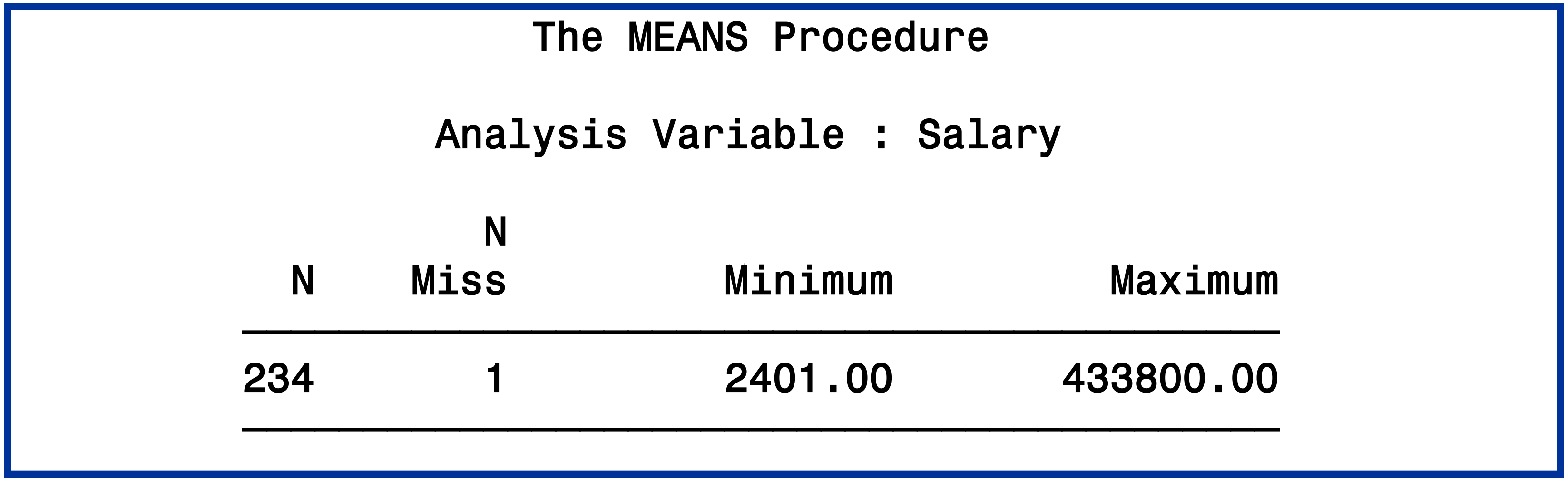
The FREQ procedure can show if any genders are not `F` or `M` and if any countries are not `AU` or `US`.



```
                          The FREQ Procedure

                                         Cumulative    Cumulative
Gender     Frequency     Percent          Frequency       Percent
--------------------------------------------------------------------
F                110       47.01               110         47.01
G                  1        0.43               111         47.44
M                123       52.56               234        100.00

                        Frequency Missing = 1


                                          Cumulative    Cumulative
Country     Frequency     Percent          Frequency       Percent
---------------------------------------------------------------------
AU                 33       14.04                 33         14.04
US                196       83.40                229         97.45
au                  3        1.28                232         98.72
us                  3        1.28                235        100.00
```

The MEANS Procedure

The MEANS procedure can show if any salaries are not in the range of 24000 to 500000.



```
                  The MEANS Procedure

             Analysis Variable : Salary

             N
  N       Miss           Minimum           Maximum
 ______________________________________________________
 234         1           2401.00         433800.00
 ______________________________________________________
```

p108d01

The UNIVARIATE Procedure

The UNIVARIATE procedure can show if any salaries are not in the range of 24000 to 500000.

Partial PROC UNIVARIATE Output

```
                 The UNIVARIATE Procedure
                    Variable:  Salary

                  Extreme Observations

       -----Lowest----        -----Highest----

        Value      Obs          Value      Obs

         2401       20         163040        1
         2650       13         194885      231
        24025       25         207885       28
        24100       19         268455       29
        24390      228         433800       27
```

p108d01

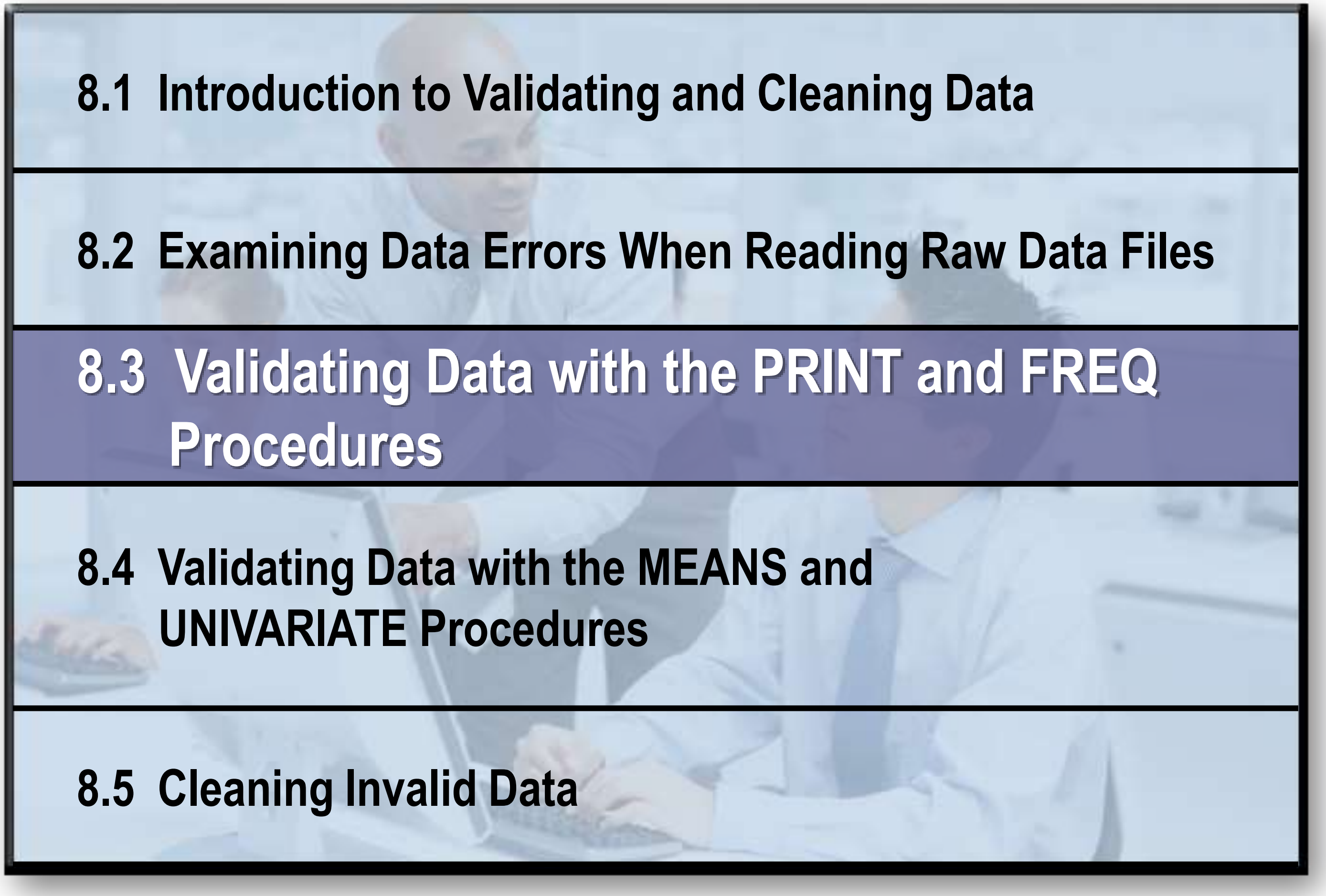
Cleaning the Data

After the data is validated, the invalid data needs to be cleaned.

Techniques for cleaning data:

- Editing raw data file outside of SAS
- Interactively editing data set using VIEWTABLE
- Programmatically editing data set using the DATA step
- Programmatically editing data set using the SQL procedure
- Using the SAS DataFlux product dfPower Studio

Chapter 8: Validating and Cleaning Data



8.1 Introduction to Validating and Cleaning Data

8.2 Examining Data Errors When Reading Raw Data Files

8.3 Validating Data with the PRINT and FREQ Procedures

8.4 Validating Data with the MEANS and UNIVARIATE Procedures

8.5 Cleaning Invalid Data

Objectives

- Validate data by using the PRINT procedure with the WHERE statement.
- Validate data by using the FREQ procedure with the TABLES statement.

Business Scenario

Additional requirements of non-sales employee data:

- **`Employee_ID`** must be unique and not missing.
- **`Gender`** must have a value of `F` or `M`.
- **`Salary`** must be in the numeric range of 24000 – 500000.
- **`Job_Title`** must not be missing.
- **`Country`** must have a value of `AU` or `US`.
- **`Birth_Date`** value must occur before **`Hire_Date`** value.
- **`Hire_Date`** must have a value of 01/01/1974 or later.

SAS Procedures for Validating Data

SAS procedures can be used to detect invalid data.

PROC PRINT step with **VAR** and **WHERE** statements	detects invalid character and numeric values by subsetting observations based on conditions.
PROC FREQ step with **TABLES** statement	detects invalid character and numeric values by looking at distinct values.
PROC MEANS step with **VAR** statement	detects invalid numeric values by using summary statistics.
PROC UNIVARIATE step with **VAR** statement	detects invalid numeric values by looking at extreme values.

The PRINT Procedure

The PRINT procedure produces detail reports based on SAS data sets.

General form of the PRINT procedure:

```
PROC PRINT DATA=SAS-data-set ;
      VAR variable(s) ;
      WHERE where-expression ;
RUN;
```

- The VAR statement selects variables to include in the report and determines their order in the report.
- The WHERE statement is used to obtain a subset of observations.

The WHERE Statement

For validating data, the WHERE statement is used to retrieve the observations that do not meet the data requirements.

General form of the WHERE statement:

```
WHERE where-expression ;
```

The *where-expression* is a sequence of operands and operators that form a set of instructions that define a condition for selecting observations.

- Operands include constants and variables.
- Operators are symbols that request a comparison, arithmetic calculation, or logical operation.

The WHERE Statement

The following PROC PRINT step retrieves observations that have missing values for **`Job_Title`**.

```
proc print data=orion.nonsales;
   var Employee_ID Last Job_Title;
   where Job_Title = ' ';
run;
```

```
                   Employee_                 Job_
            Obs        ID         Last       Title

             10      120112     Glattback
```

p108d04

The WHERE Statement

A WHERE statement might need to reference a SAS date value.

For example, the PRINT procedure needs to retrieve observations that have values of **`Hire_Date`** less than January 1, 1974.

What is the numeric SAS date value for January 1, 1974?

A *SAS date constant* is used to convert a calendar date to a SAS date value.

SAS Date Constant

To write a SAS date constant, enclose a date in quotation marks in the form ***ddMMMyyyy*** and immediately follow the final quotation mark with the letter **d**.

dd	is a one- or two-digit value for the day.
MMM	is a three-letter abbreviation for the month.
yyyy	is a four-digit value for the year.
d	is required to convert the quoted string to a SAS date.

Example:

The date constant for January 1, 1974, is **`'01JAN1974'd`** .

SAS Date Constant

The following PROC PRINT step retrieves observations that have values of **Hire_Date** that are less than January 1, 1974.

```
proc print data=orion.nonsales;
   var Employee_ID Birth_Date Hire_Date;
   where Hire_Date < '01JAN1974'd;
run;
```

```
             Employee_
     Obs        ID        Birth_Date      Hire_Date

       5      120107      01/02/1978     21/01/1953
       9      120111      23/07/1949              .
     214      121011      11/03/1944     01/01/1968
```

p108d04

Poll

Quiz

8.05 Multiple Choice Poll

Which data requirement cannot be achieved with the PRINT procedure using a WHERE statement?

a. **Employee_ID** must be unique and not missing.

b. **Gender** must have a value of F or M.

c. **Salary** must be in the numeric range of 24000 – 500000.

d. **Job_Title** must not be missing.

e. **Country** must have a value of AU or US.

f. **Birth_Date** value must occur before **Hire_Date** value.

g. **Hire_Date** must have a value of 01/01/1974 or later.

8.05 Multiple Choice Poll – Correct Answer

Which data requirement cannot be achieved with the PRINT procedure using a WHERE statement?

[✓ a.] **Employee_ID** must be unique and not missing.

b. **Gender** must have a value of F or M.

c. **Salary** must be in the numeric range of 24000 – 500000.

d. **Job_Title** must not be missing.

e. **Country** must have a value of AU or US.

f. **Birth_Date** value must occur before **Hire_Date** value.

g. **Hire_Date** must have a value of 01/01/1974 or later.

Data Requirements

Data Requirement	*where-expression* to obtain invalid data
`Employee_ID` must be unique and not missing.	`Employee_ID = .` Does not account for uniqueness.
`Gender` must have a value of `F` or `M`.	`Gender not in ('F','M')`
`Salary` must be in the range of 24000 – 500000.	`Salary not between 24000 and 500000`
`Job_Title` must not be missing.	`Job_Title = ' '`
`Country` must have a value of `AU` or `US`.	`Country not in ('AU','US')`
`Birth_Date` must occur before `Hire_Date`.	`Birth_Date > Hire_Date`
`Hire_Date` must have a value of 01/01/1974 or later.	`Hire_Date < '01JAN1974'd`

Data Requirements

The following PROC PRINT step accounts for all of the data requirements except the **Employee_ID** being unique.

```
proc print data=orion.nonsales;
   var Employee_ID Gender Salary Job_Title
       Country Birth_Date Hire_Date;
   where Employee_ID = . or
         Gender not in ('F','M') or
         Salary not between 24000 and 500000 or
         Job_Title = ' ' or
         Country not in ('AU','US') or
         Birth_Date > Hire_Date or
         Hire_Date < '01JAN1974'd;
run;
```

The OR operator is used between expressions. Only one expression needs to be true to account for an observation with invalid data.

p108d04

Data Requirements

Sixteen observations need the data cleaned.

```
Obs   Employee_ID Gender Salary Job_Title                    Country Birth_Date  Hire_Date

  2        120104    F     46230 Administration Manager        au     11/05/1954 01/01/1981
  4        120106    M         . Office Assistant II           AU     23/12/1944 01/01/1974
  5        120107    F     30475 Office Assistant III          AU     01/02/1978 21/01/1953
  9        120111    M     26895 Security Guard II             AU     23/07/1949          .
 10        120112    F     26550                               AU     17/02/1969 01/07/1990
 12        120114    G     31285 Security Manager              AU     08/02/1944 01/01/1974
 13        120115    M      2650 Service Assistant I           AU     08/05/1984 01/08/2005
 14             .    M     29250 Service Assistant II          AU     13/06/1959 01/02/1980
 20        120191    F      2401 Trainee                       AU     17/01/1959 01/01/2003
 84        120695    M     28180 Warehouse Assistant II        au     13/07/1964 01/07/1989
 87        120698    M     26160 Warehouse Assistant I         au     17/05/1954 01/08/1976
101        120723          33950 Corp. Comm. Specialist II     US     10/08/1949 01/01/1974
125        120747    F     43590 Financial Controller I        us     20/06/1974 01/08/1995
197        120994    F     31645 Office Administrator I        us     16/06/1974 01/11/1994
200        120997    F     27420 Shipping Administrator I      us     21/11/1974 01/09/1996
214        121011    M     25735 Service Assistant I           US     11/03/1944 01/01/1968
```

The FREQ Procedure

The FREQ procedure produces one-way to *n*-way frequency tables.

General form of the FREQ procedure:

```
PROC FREQ DATA=SAS-data-set <NLEVELS>;
      TABLES variable(s);
RUN;
```

- The TABLES statement specifies the frequency tables to produce.
- The NLEVELS option displays a table that provides the number of distinct values for each variable named in the TABLES statement.

The FREQ Procedure

The following PROC FREQ step will show whether there are any invalid values for **Gender** and **Country**.

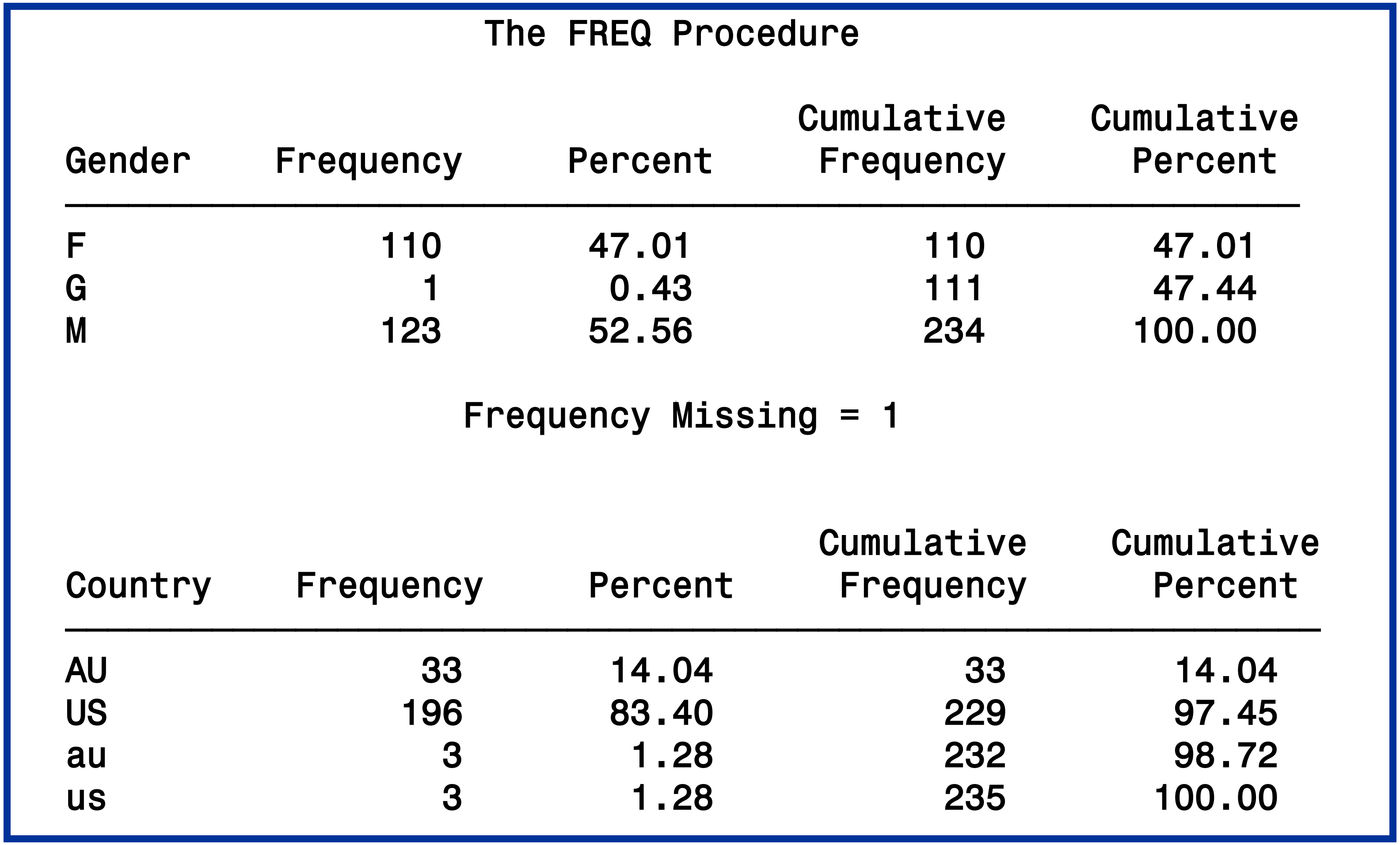
```
proc freq data=orion.nonsales;
   tables Gender Country;
run;
```

✎ Without the TABLES statement, PROC FREQ produces a frequency table for each variable.

p108d05

The FREQ Procedure

Two observations need the data cleaned for **Gender** and six observations need the data cleaned for **Country**.



```
                        The FREQ Procedure

                                           Cumulative    Cumulative
Gender     Frequency     Percent            Frequency       Percent
-------------------------------------------------------------------
F                110       47.01                  110         47.01
G                  1        0.43                  111         47.44
M                123       52.56                  234        100.00

                       Frequency Missing = 1


                                            Cumulative    Cumulative
Country     Frequency     Percent            Frequency       Percent
--------------------------------------------------------------------
AU                 33       14.04                   33         14.04
US                196       83.40                  229         97.45
au                  3        1.28                  232         98.72
us                  3        1.28                  235        100.00
```

The FREQ Procedure

This PROC FREQ step will show whether there are any duplicates for **`Employee_ID`**.

```
proc freq data=orion.nonsales;
   tables Employee_ID;
run;
```

p108d05

The FREQ Procedure

Partial PROC FREQ Output

```
                     The FREQ Procedure

                                           Cumulative    Cumulative
Employee_ID    Frequency     Percent        Frequency       Percent
_______________________________________________________________________

     120101            1        0.43                1          0.43
     120104            1        0.43                2          0.86
     120105            1        0.43                3          1.29
     120106            1        0.43                4          1.72
     120107            1        0.43                5          2.15
     120108            2        0.85                7          2.99
     120110            1        0.43                8          3.43
     120111            1        0.43                9          3.86
     120112            1        0.43               10          4.29
     120113            1        0.43               11          4.72
```

```
                                                          ative
                                                          cent
                                                       _______

                                                          .57
                                                          .00
                                                          .42
                                                          .85
                                                          .28
                                                          .71
     121146            1        0.43              232     99.14
     121147            1        0.43              233     99.57
     121148            1        0.43              234    100.00

             Frequency Missing = 1
```

The NLEVELS Option

If the number of desired distinct values is known, the NLEVELS option can help to determine whether there are any duplicates.

```
proc freq data=orion.nonsales nlevels;
   tables Gender Country Employee_ID;
run;
```

The *NLEVELS option* displays a table that provides the number of distinct values for each variable named in the TABLES statement.

p108d05

The NLEVELS Option

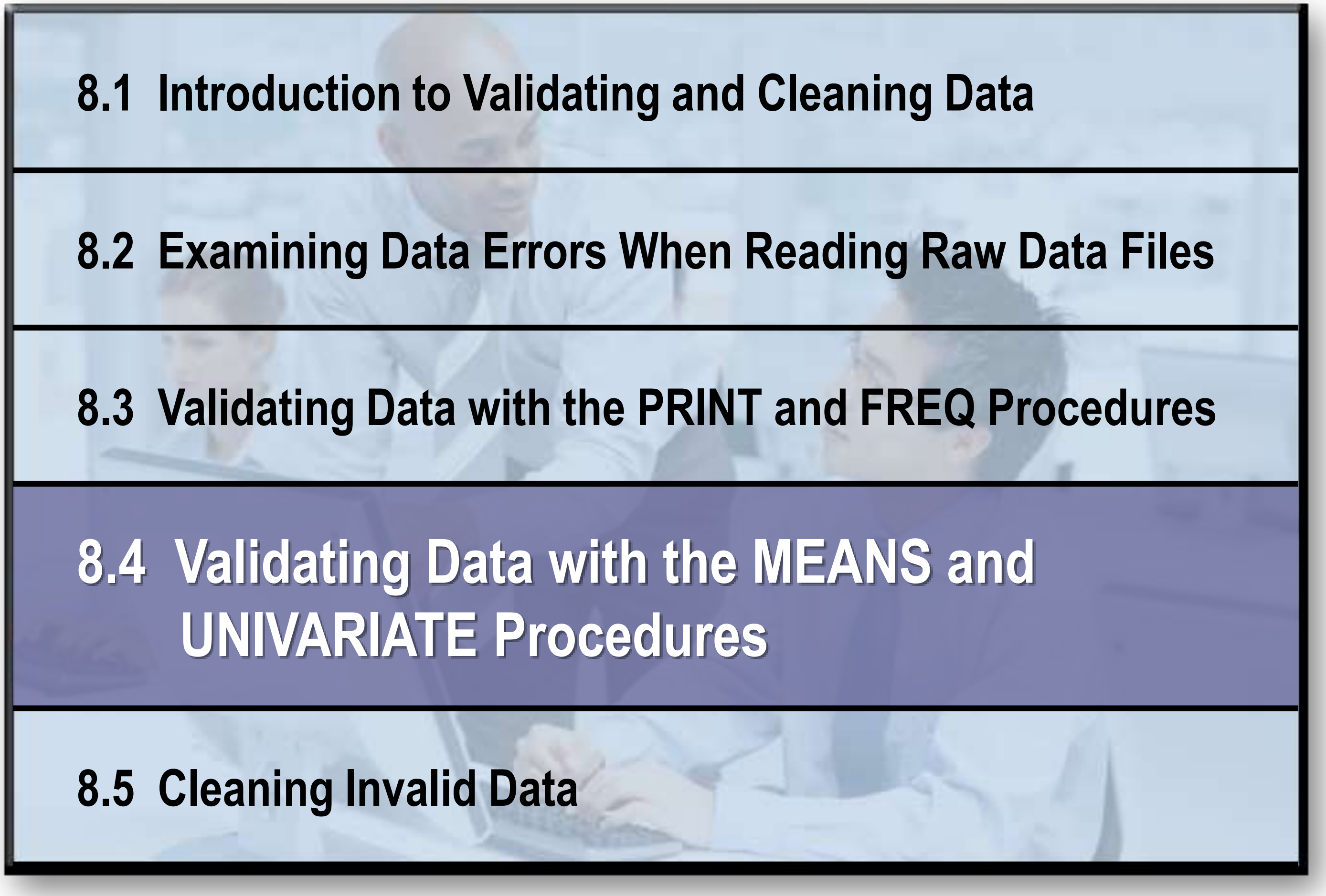
The Number of Variable Levels table appears before the individual frequency tables.

Partial PROC FREQ Output

```
                     The FREQ Procedure

                 Number of Variable Levels

                                  Missing    Nonmissing
Variable               Levels      Levels        Levels
-------------------------------------------------------
Gender                      4           1             3
Country                     4           0             4
Employee_ID               234           1           233
```

There are 235 employees but there are only 234 distinct **Employee_ID** values. Therefore, there is one duplicate value for **Employee_ID**.

Chapter 8: Validating and Cleaning Data



8.1 Introduction to Validating and Cleaning Data

8.2 Examining Data Errors When Reading Raw Data Files

8.3 Validating Data with the PRINT and FREQ Procedures

8.4 Validating Data with the MEANS and UNIVARIATE Procedures

8.5 Cleaning Invalid Data

Objectives

- Validate data by using the MEANS procedure with the VAR statement.
- Validate data by using the UNIVARIATE procedure with the VAR statement.

The MEANS Procedure

The MEANS procedure produces summary reports that display descriptive statistics.

General form of the MEANS procedure:

```
PROC MEANS DATA=SAS-data-set <statistics>;
      VAR variable(s);
RUN;
```

- The VAR statement specifies the analysis variables and their order in the results.
- The statistics to display can be specified in the PROC MEANS statement.

The MEANS Procedure

This PROC MEANS step shows default descriptive statistics for **Salary**.

```
proc means data=orion.nonsales;
   var Salary;
run;
```

```
                        The MEANS Procedure

                   Analysis Variable : Salary

  N            Mean         Std Dev         Minimum         Maximum
 ------------------------------------------------------------------
 234       43954.60        38354.77         2401.00       433800.00
 ------------------------------------------------------------------
```

Without the VAR statement, PROC MEANS analyzes all numeric variables in the data set.

p108d06

The MEANS Procedure

By default, the MEANS procedure creates a report with N (number of nonmissing values), MEAN, STDDEV, MIN, and MAX.

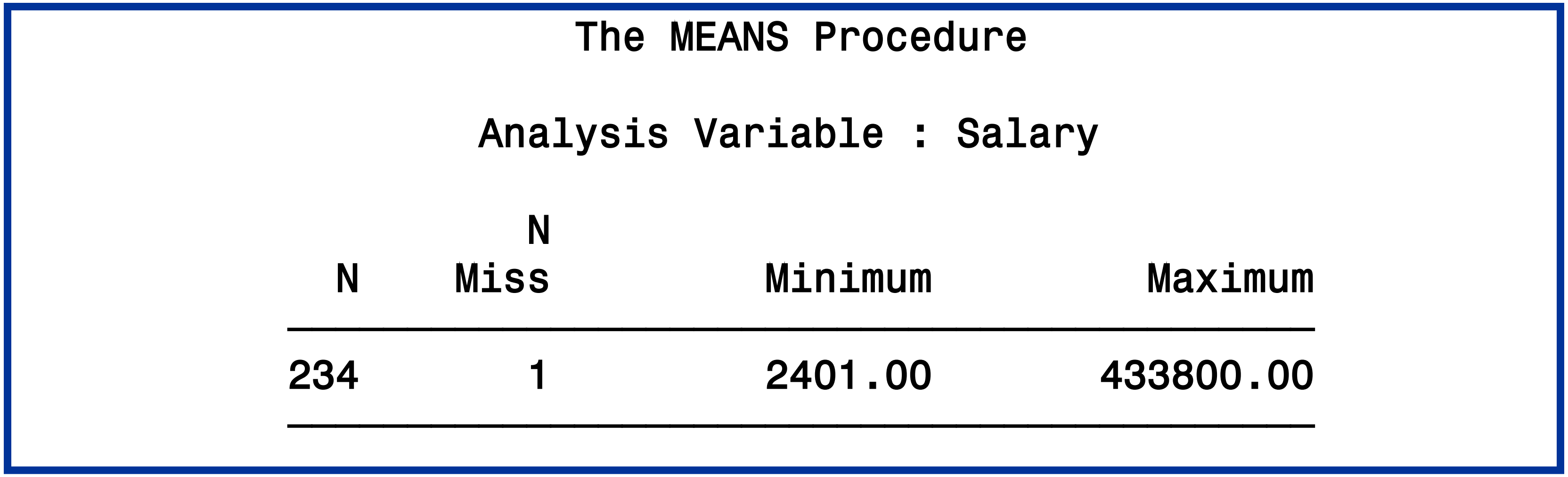
For validating data, the following descriptive statistics are beneficial:

- N, number of nonmissing values
- NMISS, number of missing values
- MIN
- MAX

The MEANS Procedure

The following PROC MEANS step shows whether there are any **Salary** values not in the range of 24000 through 500000.

```
proc means data=orion.nonsales n nmiss min max;
   var Salary;
run;
```



```
                  The MEANS Procedure

             Analysis Variable : Salary

              N
      N    Miss          Minimum          Maximum
    ------------------------------------------------
    234       1          2401.00        433800.00
    ------------------------------------------------
```

p108d06

The UNIVARIATE Procedure

The UNIVARIATE procedure produces summary reports that display descriptive statistics.

General form of the UNIVARIATE procedure:

```
PROC UNIVARIATE DATA=SAS-data-set;
      VAR variable(s);
RUN;
```

The VAR statement specifies the analysis variables and their order in the results.

The UNIVARIATE Procedure

The following PROC UNIVARIATE step shows default descriptive statistics for **`Salary`**.

```
proc univariate data=orion.nonsales;
   var Salary;
run;
```

✎ Without the VAR statement, SAS will analyze all numeric variables.

p108d06

The UNIVARIATE Procedure

The UNIVARIATE procedure can produce the following sections of output:

- Moments
- Basic Statistical Measures
- Tests for Locations
- Quantiles
- Extreme Observations
- Missing Values

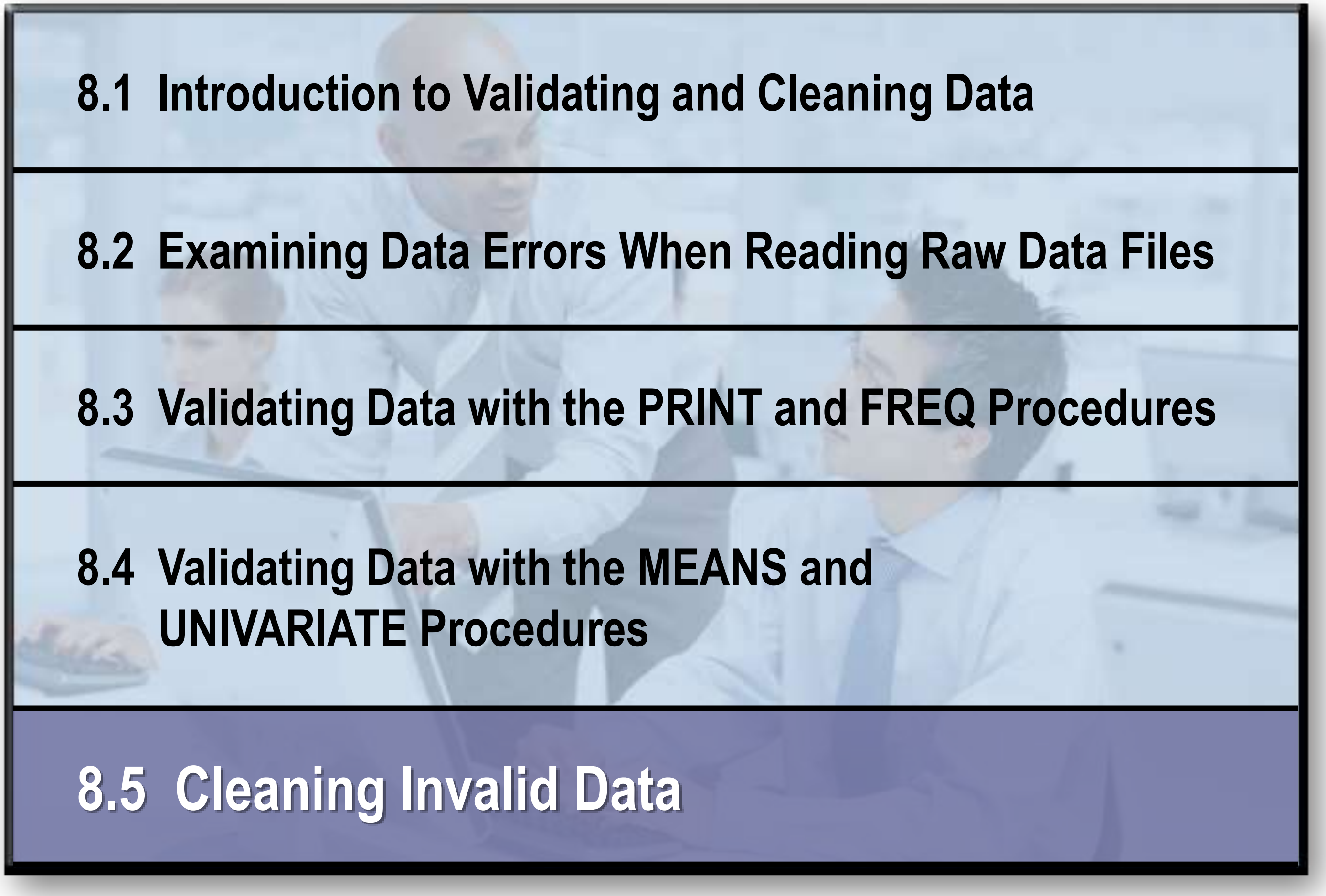
For validating data, the Extreme Observations and Missing Values sections are beneficial.

The UNIVARIATE Procedure

Partial PROC UNIVARIATE Output

```
                  Extreme Observations

         -----Lowest----          -----Highest----

          Value        Obs           Value        Obs

           2401         20          163040          1
           2650         13          194885        231
          24025         25          207885         28
          24100         19          268455         29
          24390        228          433800         27


                     Missing Values

                                -----Percent Of-----
      Missing                                Missing
        Value         Count       All Obs        Obs

            .             1          0.43     100.00
```

Chapter 8: Validating and Cleaning Data



8.1 Introduction to Validating and Cleaning Data

8.2 Examining Data Errors When Reading Raw Data Files

8.3 Validating Data with the PRINT and FREQ Procedures

8.4 Validating Data with the MEANS and UNIVARIATE Procedures

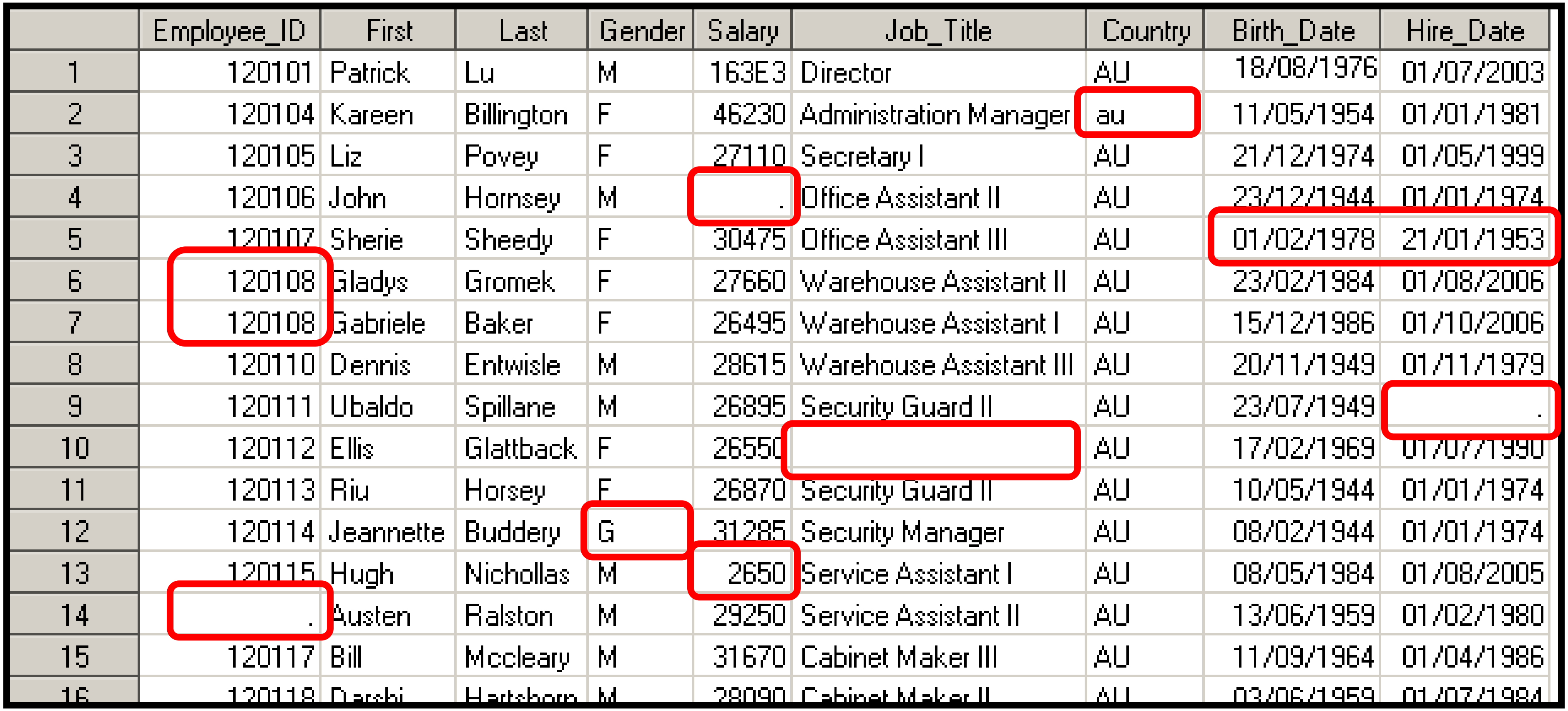
8.5 Cleaning Invalid Data

Objectives

- Clean data by using the Viewtable window.
- Clean data by using assignment statements in the DATA step.
- Clean data by using IF-THEN/ELSE statements in the DATA step.

Invalid Data to Clean

The **`orion.nonsales`** data set contains invalid data that needs to be cleaned.



	Employee_ID	First	Last	Gender	Salary	Job_Title	Country	Birth_Date	Hire_Date
1	120101	Patrick	Lu	M	163E3	Director	AU	18/08/1976	01/07/2003
2	120104	Kareen	Billington	F	46230	Administration Manager	au	11/05/1954	01/01/1981
3	120105	Liz	Povey	F	27110	Secretary I	AU	21/12/1974	01/05/1999
4	120106	John	Hornsey	M	.	Office Assistant II	AU	23/12/1944	01/01/1974
5	120107	Sherie	Sheedy	F	30475	Office Assistant III	AU	01/02/1978	21/01/1953
6	120108	Gladys	Gromek	F	27660	Warehouse Assistant II	AU	23/02/1984	01/08/2006
7	120108	Gabriele	Baker	F	26495	Warehouse Assistant I	AU	15/12/1986	01/10/2006
8	120110	Dennis	Entwisle	M	28615	Warehouse Assistant III	AU	20/11/1949	01/11/1979
9	120111	Ubaldo	Spillane	M	26895	Security Guard II	AU	23/07/1949	.
10	120112	Ellis	Glattback	F	26550		AU	17/02/1969	01/07/1990
11	120113	Riu	Horsey	F	26870	Security Guard II	AU	10/05/1944	01/01/1974
12	120114	Jeannette	Buddery	G	31285	Security Manager	AU	08/02/1944	01/01/1974
13	120115	Hugh	Nichollas	M	2650	Service Assistant I	AU	08/05/1984	01/08/2005
14	.	Austen	Ralston	M	29250	Service Assistant II	AU	13/06/1959	01/02/1980
15	120117	Bill	Mccleary	M	31670	Cabinet Maker III	AU	11/09/1964	01/04/1986
16	120118	Darshi	Hartshorn	M	28090	Cabinet Maker II	AU	03/06/1959	01/07/1984

After you validate the data and find the invalid data, the correct data values are needed.

Variable	Obs	Invalid Value	Correct Value
Employee_ID	7	120108	120109
	14	.	120116
Gender	12	G	F
	101		F
Job_Title	10		Security Guard I
Country	2, 84, 87, 125, 197, and 200	au or us	AU or US
Salary	4	.	26960
	13	2650	26500
	20	2401	24015
Hire_Date	5	21/01/1953	21/01/1995
	9	.	01/11/1978
	214	01/01/1968	01/01/1998

Interactively Cleaning Data

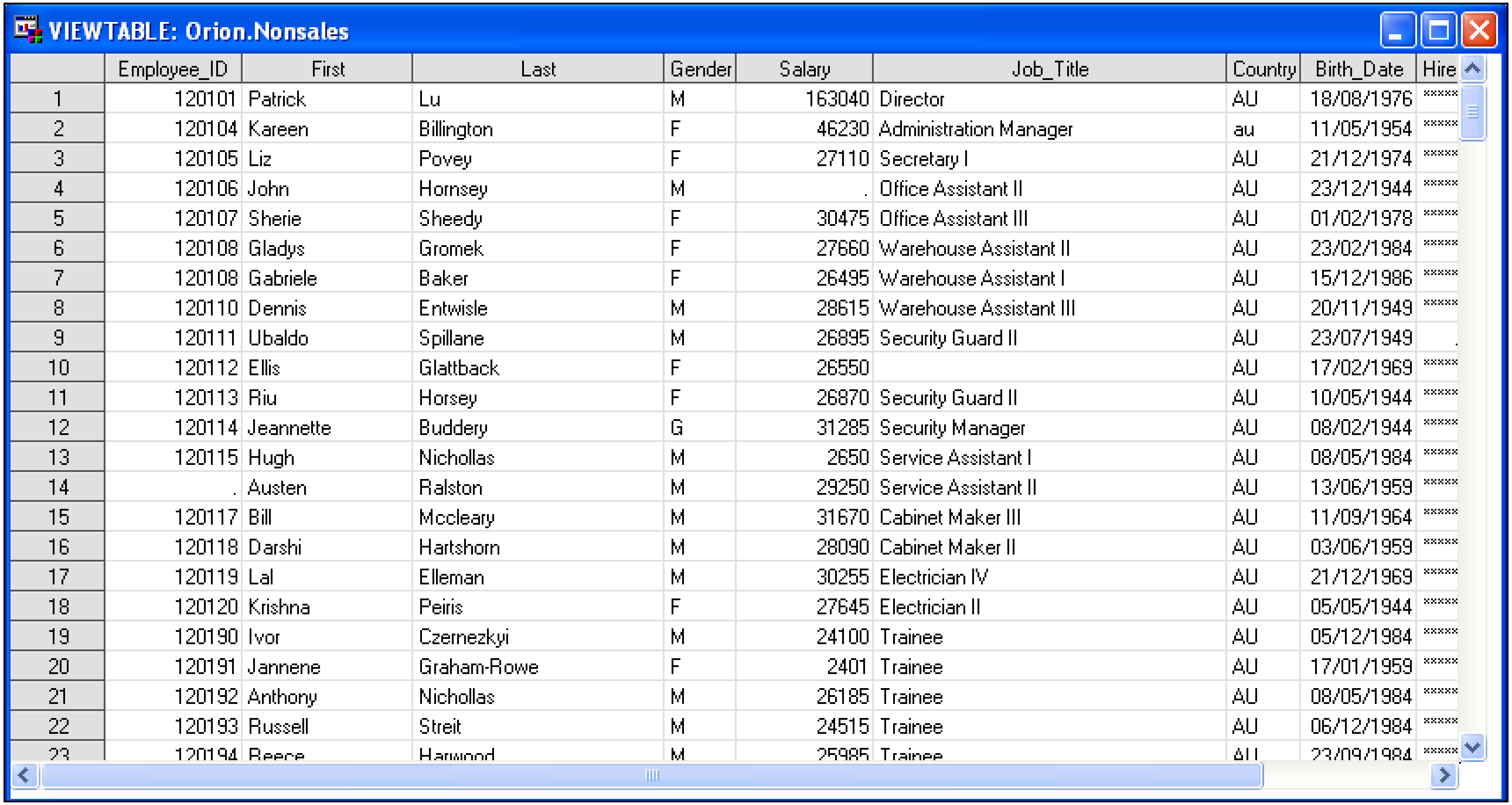
If you are using the SAS windowing environment, the Viewtable window can be used to interactively clean data.

Use the Viewtable window to interactively clean the following five observations:

Variable	Obs	Invalid Value	Correct Value
Employee_ID	7	120108	120109
	14	.	120116
Gender	12	G	F
	101		F
Job_Title	10		Security Guard I

Interactively Cleaning Data

The Viewtable window enables you to browse, edit, or create SAS data sets.



VIEWTABLE: Orion.Nonsales

	Employee_ID	First	Last	Gender	Salary	Job_Title	Country	Birth_Date	Hire
1	120101	Patrick	Lu	M	163040	Director	AU	18/08/1976	*****
2	120104	Kareen	Billington	F	46230	Administration Manager	au	11/05/1954	*****
3	120105	Liz	Povey	F	27110	Secretary I	AU	21/12/1974	*****
4	120106	John	Hornsey	M	.	Office Assistant II	AU	23/12/1944	*****
5	120107	Sherie	Sheedy	F	30475	Office Assistant III	AU	01/02/1978	*****
6	120108	Gladys	Gromek	F	27660	Warehouse Assistant II	AU	23/02/1984	*****
7	120108	Gabriele	Baker	F	26495	Warehouse Assistant I	AU	15/12/1986	*****
8	120110	Dennis	Entwisle	M	28615	Warehouse Assistant III	AU	20/11/1949	*****
9	120111	Ubaldo	Spillane	M	26895	Security Guard II	AU	23/07/1949	.
10	120112	Ellis	Glattback	F	26550		AU	17/02/1969	*****
11	120113	Riu	Horsey	F	26870	Security Guard II	AU	10/05/1944	*****
12	120114	Jeannette	Buddery	G	31285	Security Manager	AU	08/02/1944	*****
13	120115	Hugh	Nichollas	M	2650	Service Assistant I	AU	08/05/1984	*****
14	.	Austen	Ralston	M	29250	Service Assistant II	AU	13/06/1959	*****
15	120117	Bill	Mccleary	M	31670	Cabinet Maker III	AU	11/09/1964	*****
16	120118	Darshi	Hartshorn	M	28090	Cabinet Maker II	AU	03/06/1959	*****
17	120119	Lal	Elleman	M	30255	Electrician IV	AU	21/12/1969	*****
18	120120	Krishna	Peiris	F	27645	Electrician II	AU	05/05/1944	*****
19	120190	Ivor	Czernezkyi	M	24100	Trainee	AU	05/12/1984	*****
20	120191	Jannene	Graham-Rowe	F	2401	Trainee	AU	17/01/1959	*****
21	120192	Anthony	Nichollas	M	26185	Trainee	AU	08/05/1984	*****
22	120193	Russell	Streit	M	24515	Trainee	AU	06/12/1984	*****
23	120194	Reece	Harwood	M	25985	Trainee	AU	23/09/1984	*****

Using the Viewtable Window to Clean Data

This demonstration illustrates using the Viewtable window to clean the values of four observations.

Poll

Quiz

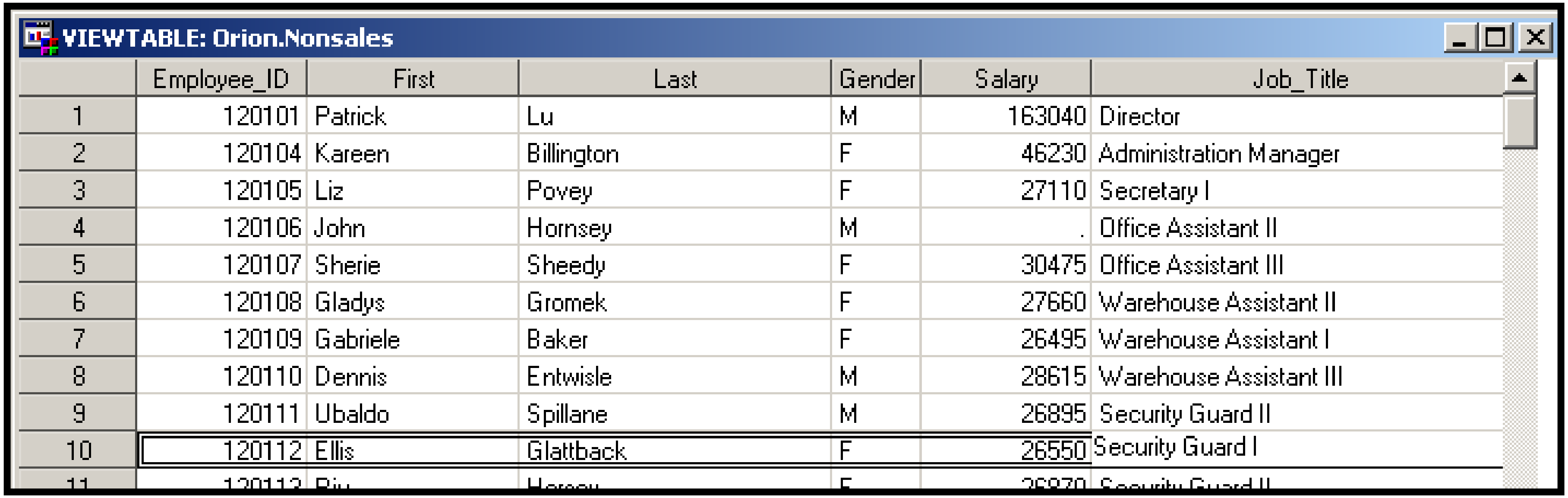
8.06 Quiz

- Open the VIEWTABLE window for **`orion.nonsales`**.
- Use the VIEWTABLE window to interactively clean the following observation:

Variable	Obs	Invalid Value	Correct Value
Job_Title	**10**		**Security Guard I**

8.06 Quiz – Correct Answer

- Open the VIEWTABLE window for **`orion.nonsales`**.
- Use the VIEWTABLE window to interactively clean the following observation:



VIEWTABLE: Orion.Nonsales

	Employee_ID	First	Last	Gender	Salary	Job_Title
1	120101	Patrick	Lu	M	163040	Director
2	120104	Kareen	Billington	F	46230	Administration Manager
3	120105	Liz	Povey	F	27110	Secretary I
4	120106	John	Hornsey	M	.	Office Assistant II
5	120107	Sherie	Sheedy	F	30475	Office Assistant III
6	120108	Gladys	Gromek	F	27660	Warehouse Assistant II
7	120109	Gabriele	Baker	F	26495	Warehouse Assistant I
8	120110	Dennis	Entwisle	M	28615	Warehouse Assistant III
9	120111	Ubaldo	Spillane	M	26895	Security Guard II
10	120112	Ellis	Glattback	F	26550	Security Guard I
11	120113	Riu	Horsey	F	26870	Security Guard II

Programmatically Cleaning Data

The DATA step can be used to programmatically clean the invalid data.

Use the DATA step to clean the following observations:

Variable	Obs	Invalid Value	Correct Value
Country	2, 84, 87, 125, 197, and 200	au or us	AU or US
Salary	4	.	26960
	13	2650	26500
	20	2401	24015
Hire_Date	5	21/01/1953	21/01/1995
	9	.	01/11/1978
	214	01/01/1968	01/01/1998

The Assignment Statement

The *assignment statement* evaluates an expression and assigns the resulting value to a variable.

General form of the assignment statement:

> *variable* = *expression*;

- *variable* names an existing or new variable.
- *expression* is a sequence of operands and operators that form a set of instructions that produce a value.

The Assignment Statement Expression

Operands are

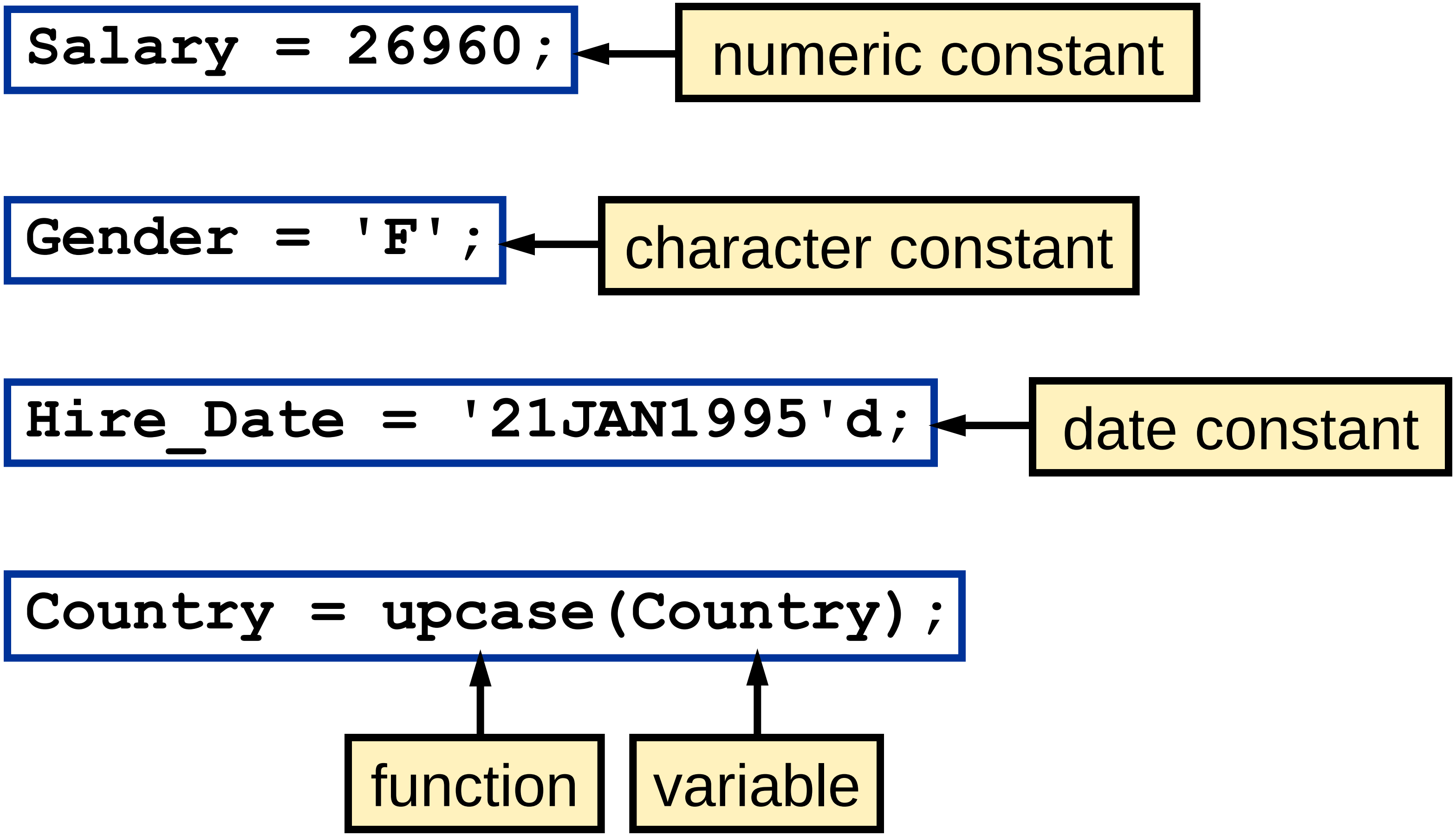
- character constants
- numeric constants
- date constants
- character variables
- numeric variables.

Operators are

- symbols that represent an arithmetic calculation
- SAS functions.

The Assignment Statement Expression

Examples:



```
Salary = 26960;
```

numeric constant

```
Gender = 'F';
```

character constant

```
Hire_Date = '21JAN1995'd;
```

date constant

```
Country = upcase(Country);
```

function

variable

SAS Functions

A SAS *function* is a routine that returns a value that is determined from specified arguments.

The *UPCASE function* converts all letters in an argument to uppercase.

General form of the UPCASE function:

UPCASE(*argument***)**

The *argument* specifies any SAS character expression.

The Assignment Statement

All the values of **Country** in the data set **orion.nonsales** need to be uppercase.

```
data work.clean;
   set orion.nonsales;
   Country=upcase(Country);
run;
```

PDV

Employee_ID		Job_Title	Country	
120101	...	Director	AU	...

p108d07

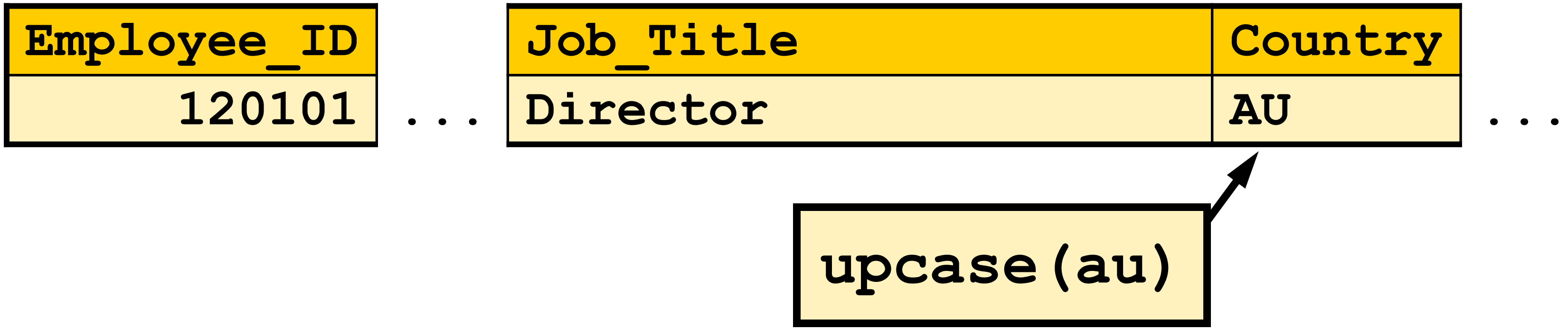
...

The Assignment Statement

All the values of **Country** in the data set **orion.nonsales** need to be uppercase.

```
data work.clean;
   set orion.nonsales;
   Country=upcase(Country);
run;
```

PDV



Employee_ID		Job_Title	Country	
120101	...	Director	AU	...

upcase(au)

The Assignment Statement

All the values of **Country** in the data set **orion.nonsales** need to be uppercase.

```
data work.clean;
   set orion.nonsales;
   Country=upcase(Country);
run;
```

PDV

Employee_ID		Job_Title	Country	
120104	...	Administration Manager	au	...

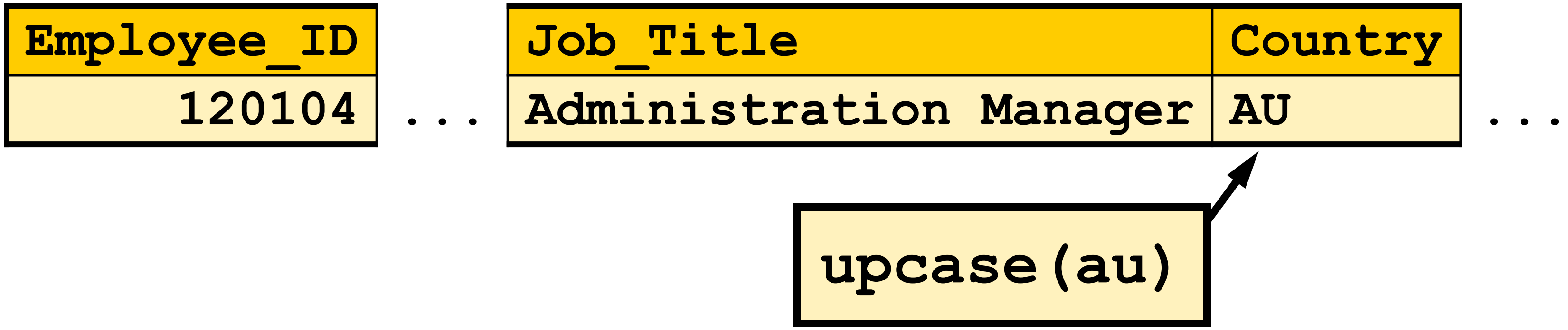
...

The Assignment Statement

All the values of **Country** in the data set **orion.nonsales** need to be uppercase.

```
data work.clean;
   set orion.nonsales;
   Country=upcase(Country);
run;
```

PDV



Employee_ID		Job_Title	Country	
120104	...	Administration Manager	AU	...

upcase(au)

The Assignment Statement

```
proc print data=work.clean;
   var Employee_ID Job_Title Country;
run;
```

Partial PROC PRINT Output

```
        Employee_
 Obs       ID       Job_Title                      Country

  84     120695     Warehouse Assistant II           AU
  85     120696     Warehouse Assistant I            AU
  86     120697     Warehouse Assistant IV           AU
  87     120698     Warehouse Assistant I            AU
  88     120710     Business Analyst II              US
  89     120711     Business Analyst III             US
  90     120712     Marketing Manager                US
  91     120713     Marketing Assistant III          US
```

The assignment statement executed for every observation regardless of whether the value needed to be uppercased or not.

Programmatically Cleaning Data

The DATA step can be used to programmatically clean the invalid data.

Use the DATA step to clean the following observations:

Variable	Obs	Invalid Value	Correct Value
Country			
Salary	4	.	26960
	13	2650	26500
	20		
Hire_Date	5		
	9	.	01/11/1978
	214	01/01/1968	01/01/1998

The assignment statement was applied to all observations.

The assignment statement needs to be applied to specific observations.

Poll

Quiz

8.07 Quiz

Which variable can be used to specifically identify the observations with invalid salary values?

```
Obs  Employee_ID Gender Salary Job_Title                Country Birth_Date  Hire_Date

  2       120104   F     46230 Administration Manager     au    11/05/1954 01/01/1981
  4       120106   M         . Office Assistant II        AU    23/12/1944 01/01/1974
  5       120107   F     30475 Office Assistant III       AU    01/02/1978 21/01/1953
  9       120111   M     26895 Security Guard II          AU    23/07/1949          .
 10       120112   F     26550                            AU    17/02/1969 01/07/1990
 12       120114   G     31285 Security Manager           AU    08/02/1944 01/01/1974
 13       120115   M      2650 Service Assistant I        AU    08/05/1984 01/08/2005
 14            .   M     29250 Service Assistant II       AU    13/06/1959 01/02/1980
 20       120191   F      2401 Trainee                    AU    17/01/1959 01/01/2003
 84       120695   M     28180 Warehouse Assistant II     au    13/07/1964 01/07/1989
 87       120698   M     26160 Warehouse Assistant I      au    17/05/1954 01/08/1976
101       120723         33950 Corp. Comm. Specialist II  US    10/08/1949 01/01/1974
125       120747   F     43590 Financial Controller I     us    20/06/1974 01/08/1995
197       120994   F     31645 Office Administrator I     us    16/06/1974 01/11/1994
200       120997   F     27420 Shipping Administrator I   us    21/11/1974 01/09/1996
214       121011   M     25735 Service Assistant I        US    11/03/1944 01/01/1968
```

8.07 Quiz – Correct Answer

Which variable can be used to specifically identify the observations with invalid salary values?

```
Obs   Employee_ID Gender Salary Job_Title                       Country Birth_Date  Hire_Date

  2        120104    F      46230 Administration Manager         au     11/05/1954 01/01/1981
  4        120106    M          . Office Assistant II            AU     23/12/1944 01/01/1974
  5        120107    F      30475 Office Assistant III           AU     01/02/1978 21/01/1953
  9        120111    M      26895 Security Guard II              AU     23/07/1949          .
 10        120112    F      26550                                AU     17/02/1969 01/07/1990
 12        120114    G      31285 Security Manager               AU     08/02/1944 01/01/1974
 13        120115    M       2650 Service Assistant I            AU     08/05/1984 01/08/2005
 14             .    M      29250 Service Assistant II           AU     13/06/1959 01/02/1980
 20        120191    F       2401 Trainee                        AU     17/01/1959 01/01/2003
 84        120695    M      28180 Warehouse Assistant II         au     13/07/1964 01/07/1989
 87        120698    M      26160 Warehouse Assistant I          au     17/05/1954 01/08/1976
101        120723           33950 Corp. Comm. Specialist II      US     10/08/1949 01/01/1974
125        120747    F      43590 Financial Controller I         us     20/06/1974 01/08/1995
197        120994    F      31645 Office Administrator I         us     16/06/1974 01/11/1994
200        120997    F      27420 Shipping Administrator I       us     21/11/1974 01/09/1996
214        121011    M      25735 Service Assistant I            US     11/03/1944 01/01/1968
```

`Employee_ID` because the values are unique.

Programmatically Cleaning Data

The DATA step can be used to programmatically clean the invalid data.

Use the DATA step to clean the following observations:

Variable	Obs	Invalid Value	Correct Value
`Country`	2, 84, 87, 125, 197, and 200	au or us	AU or US
`Salary`	4	.	26960
	13	2650	26500
	20	2401	24015
`Hire_Date`	5	21/01/1953	21/01/1995
	9	.	01/11/1978
	214	01/01/1968	01/01/1998

IF-THEN Statements

The *IF-THEN statement* executes a SAS statement for observations that meet specific conditions.

General form of the IF-THEN statement:

IF *expression* **THEN** *statement* **;**

- *expression* is a sequence of operands and operators that form a set of instructions that define a condition for selecting observations.
- *statement* is any executable statement such as the assignment statement.

IF-THEN Statements

All the values of **Salary** must be in the range of 24000 – 500000.

```
data work.clean;
   set orion.nonsales;
   if Employee_ID=120106 then Salary=26960;
   if Employee_ID=120115 then Salary=26500;
   if Employee_ID=120191 then Salary=24015;
run;
```

PDV

Employee_ID		Salary	Job_Title	
120105	. . .	27110	Secretary I	. . .

p108d07

...

IF-THEN Statements

When an IF expression is TRUE in this IF-THEN statement series, there is no reason to check the remaining IF-THEN statements when checking **`Employee_ID`**.

TRUE

```
data work.clean;
   set orion.nonsales;
   if Employee_ID=120106 then Salary=26960;
   if Employee_ID=120115 then Salary=26500;
   if Employee_ID=120191 then Salary=24015;
run;
```

The word ELSE can be placed before the word IF, causing SAS to execute conditional statements until it encounters the first true statement.

IF-THEN/ELSE Statements

All the values of **Salary** must be in the range of 24000 – 500000.

```
data work.clean;
   set orion.nonsales;
   if Employee_ID=120106 then Salary=26960;
   else if Employee_ID=120115 then Salary=26500;
   else if Employee_ID=120191 then Salary=24015;
run;
```

PDV

Employee_ID		Salary	Job_Title	
120106	...	.	Office Assistant II	...

p108d07

...

IF-THEN/ELSE Statements

All the values of **`Salary`** must be in the range of 24000 – 500000.

TRUE

```
data work.clean;
   set orion.nonsales;
   if Employee_ID=120106 then Salary=26960;
   else if        yee_ID=120115 then Salary=26500;
   else if        yee_ID=120191 then Salary=24015;
run;
```

SKIP

PDV

Employee_ID
120106

...

Salary	Job_Title
26960	Office Assistant II

...

Programmatically Cleaning Data

The DATA step can be used to programmatically clean the invalid data.

Use the DATA step to clean the following observations:

Variable	Obs	Invalid Value	Correct Value
`Country`	2, 84, 87, 125, 197, and 200	au or us	AU or US
`Salary`	4	.	26960
	13	2650	26500
	20	2401	24015
`Hire_Date`	5	21/01/1953	21/01/1995
	9	.	01/11/1978
	214	01/01/1968	01/01/1998

IF-THEN/ELSE Statements

All the values of **Hire_Date** must have a value of 01/01/1974 or later.

```
data work.clean;
   set orion.nonsales;
   Country=upcase(Country);
   if Employee_ID=120106 then Salary=26960;
   else if Employee_ID=120115 then Salary=26500;
   else if Employee_ID=120191 then Salary=24015;
   else if Employee_ID=120107 then
               Hire_Date='21JAN1995'd;
   else if Employee_ID=120111 then
               Hire_Date='01NOV1978'd;
   else if Employee_ID=121011 then
               Hire_Date='01JAN1998'd;
run;
```

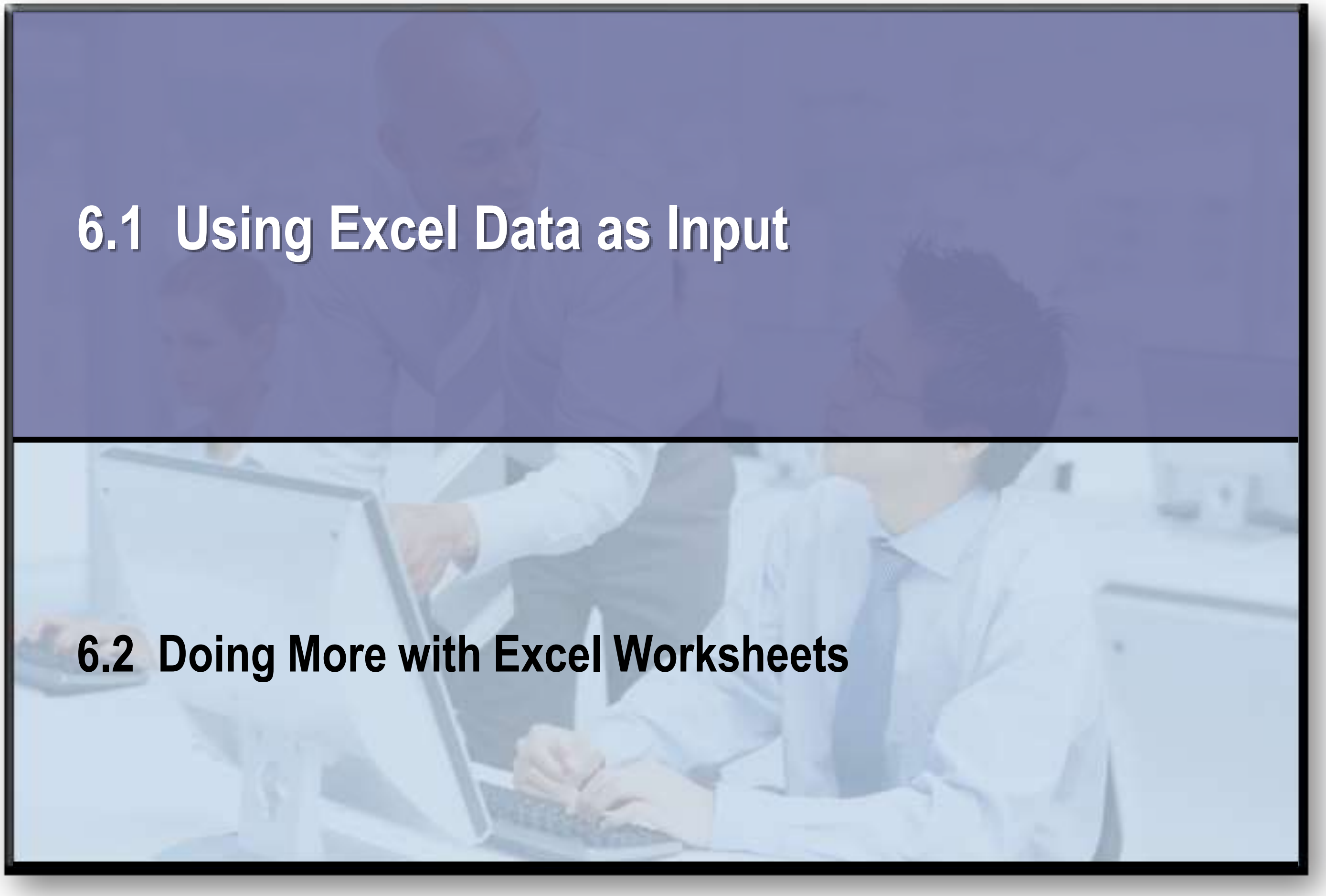
p108d07

Chapter 6: Reading Excel Worksheets

6.1 Using Excel Data as Input

6.2 Doing More with Excel Worksheets

Chapter 6: Reading Excel Worksheets



6.1 Using Excel Data as Input

6.2 Doing More with Excel Worksheets

Objectives

- Use the DATA step to create a SAS data set from an Excel worksheet.
- Use the SAS/ACCESS LIBNAME statement with PC Files Server to read from an Excel worksheet as though it were a SAS data set.

Business Scenario

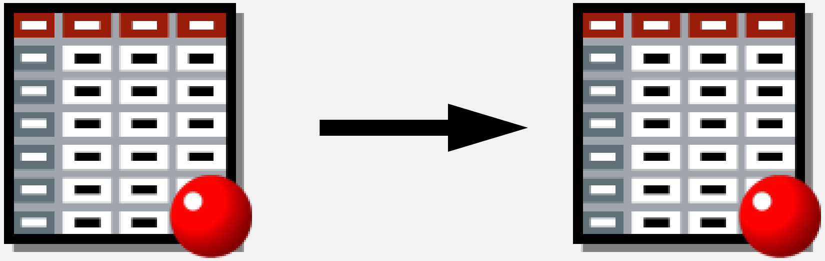
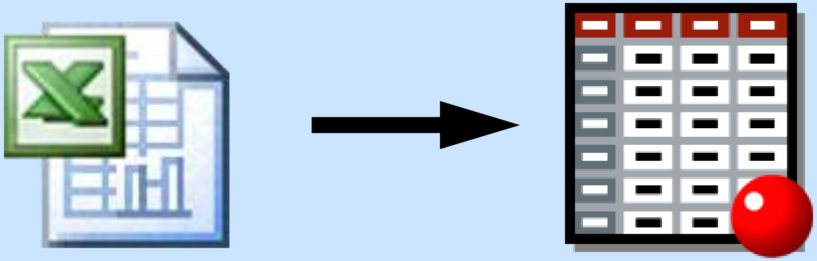
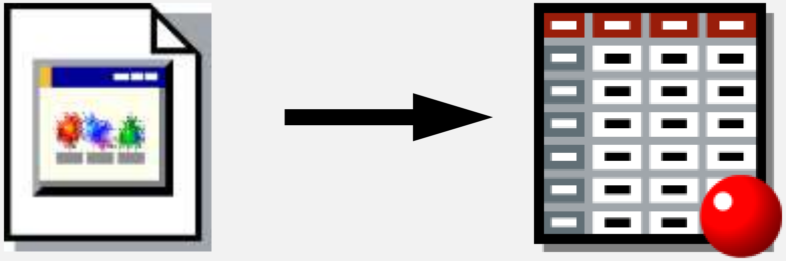
An existing data source contains information on Orion Star sales employees from Australia and the United States.

A new SAS data set needs to be created that contains a subset of this existing data source.

This new SAS data set must contain the following:

- only the employees from Australia who are Sales Representatives
- the employee's first name, last name, salary, job title, and hired date
- labels and formats in the descriptor portion

Business Scenario

Reading SAS Data Sets	
Reading Excel Worksheets	
Reading Delimited Raw Data Files	

Business Scenario

Reading SAS Data Sets	libname ______________________;
	data __________________;
	set __________________;
	. . .
	run;

Reading Excel Worksheets	libname ______________________;
	data __________________;
	set __________________;
	. . .
	run;

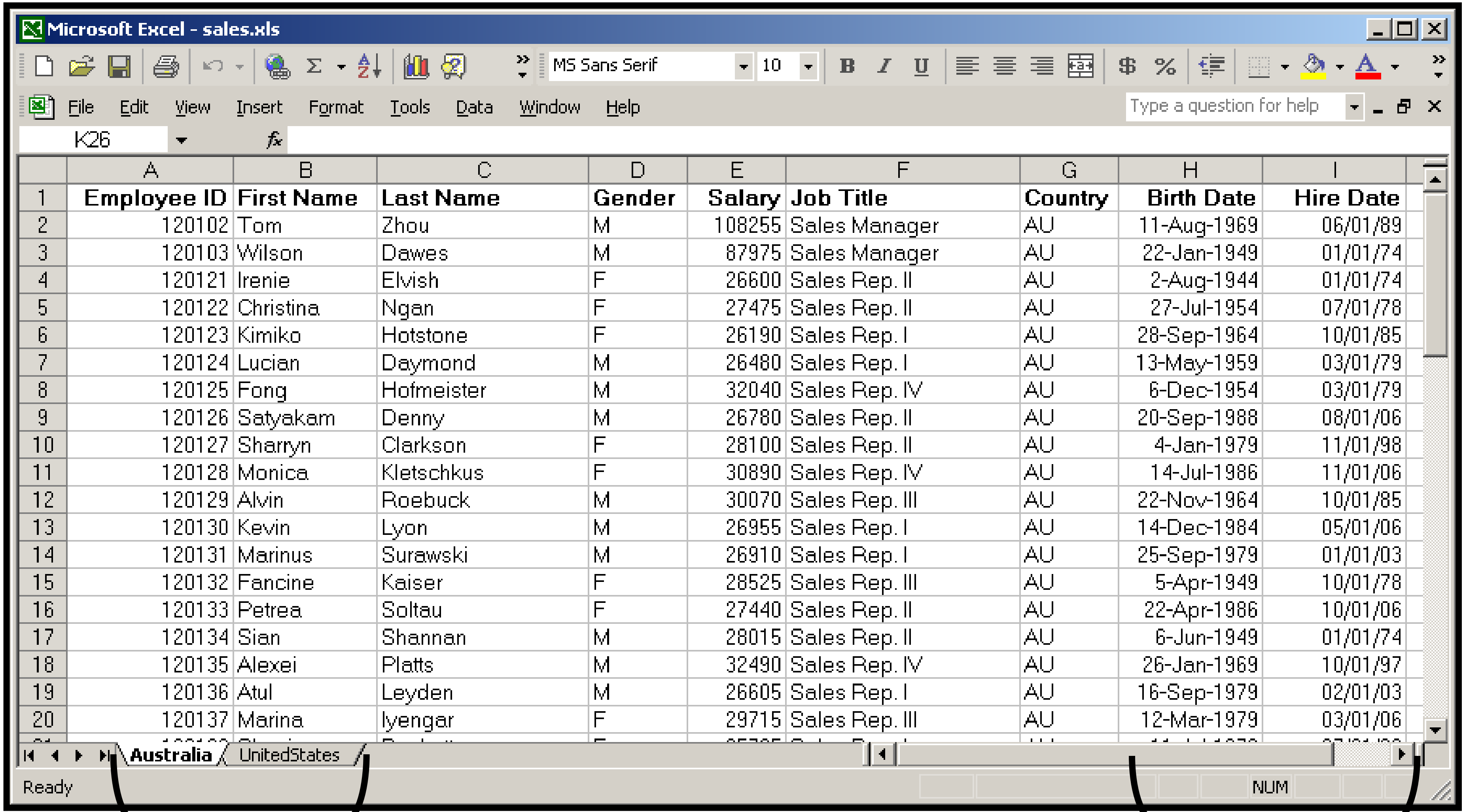
Reading Delimited Raw Data Files	data __________________;
	infile __________________;
	input __________________;
	. . .
	run;

Business Scenario Syntax

Use the following statements to complete the scenario:

```
LIBNAME libref  'physical-file-name';

DATA output-SAS-data-set;
      SET input-SAS-data-set;
      WHERE where-expression;
      KEEP variable-list;
      LABEL variable = 'label'
                  variable = 'label'
                  variable = 'label';
      FORMAT variable(s) format ;
RUN;
```

sales.xls



	A	B	C	D	E	F	G	H	I
1	Employee ID	First Name	Last Name	Gender	Salary	Job Title	Country	Birth Date	Hire Date
2	120102	Tom	Zhou	M	108255	Sales Manager	AU	11-Aug-1969	06/01/89
3	120103	Wilson	Dawes	M	87975	Sales Manager	AU	22-Jan-1949	01/01/74
4	120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	2-Aug-1944	01/01/74
5	120122	Christina	Ngan	F	27475	Sales Rep. II	AU	27-Jul-1954	07/01/78
6	120123	Kimiko	Hotstone	F	26190	Sales Rep. I	AU	28-Sep-1964	10/01/85
7	120124	Lucian	Daymond	M	26480	Sales Rep. I	AU	13-May-1959	03/01/79
8	120125	Fong	Hofmeister	M	32040	Sales Rep. IV	AU	6-Dec-1954	03/01/79
9	120126	Satyakam	Denny	M	26780	Sales Rep. II	AU	20-Sep-1988	08/01/06
10	120127	Sharryn	Clarkson	F	28100	Sales Rep. II	AU	4-Jan-1979	11/01/98
11	120128	Monica	Kletschkus	F	30890	Sales Rep. IV	AU	14-Jul-1986	11/01/06
12	120129	Alvin	Roebuck	M	30070	Sales Rep. III	AU	22-Nov-1964	10/01/85
13	120130	Kevin	Lyon	M	26955	Sales Rep. I	AU	14-Dec-1984	05/01/06
14	120131	Marinus	Surawski	M	26910	Sales Rep. I	AU	25-Sep-1979	01/01/03
15	120132	Fancine	Kaiser	F	28525	Sales Rep. III	AU	5-Apr-1949	10/01/78
16	120133	Petrea	Soltau	F	27440	Sales Rep. II	AU	22-Apr-1986	10/01/06
17	120134	Sian	Shannan	M	28015	Sales Rep. II	AU	6-Jun-1949	01/01/74
18	120135	Alexei	Platts	M	32490	Sales Rep. IV	AU	26-Jan-1969	10/01/97
19	120136	Atul	Leyden	M	26605	Sales Rep. I	AU	16-Sep-1979	02/01/03
20	120137	Marina	Iyengar	F	29715	Sales Rep. III	AU	12-Mar-1979	03/01/06

two worksheets

cells formatted as dates

The LIBNAME Statement (Review)

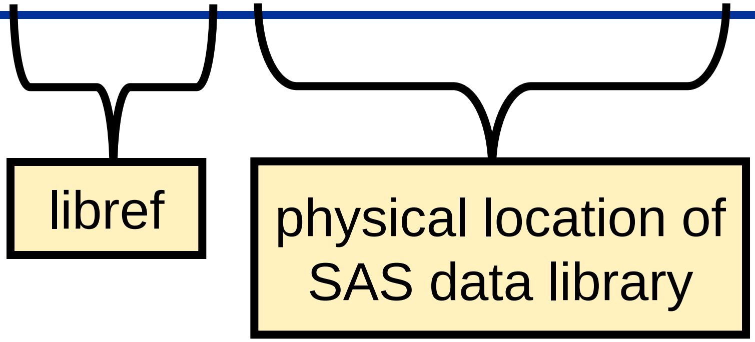
The *LIBNAME statement* assigns a library reference name (libref) to a SAS data library.

General form of the LIBNAME statement:

> **LIBNAME** *libref* '*SAS-data-library*' <*options*>**;**

Example:

```
libname orion 's:\workshop';
```



libref

physical location of SAS data library

The SAS/ACCESS LIBNAME Statement

The *SAS/ACCESS LIBNAME statement* extends the LIBNAME statement to support assigning a library reference name (libref) to Microsoft Excel workbooks.

General form of the SAS/ACCESS LIBNAME statement:

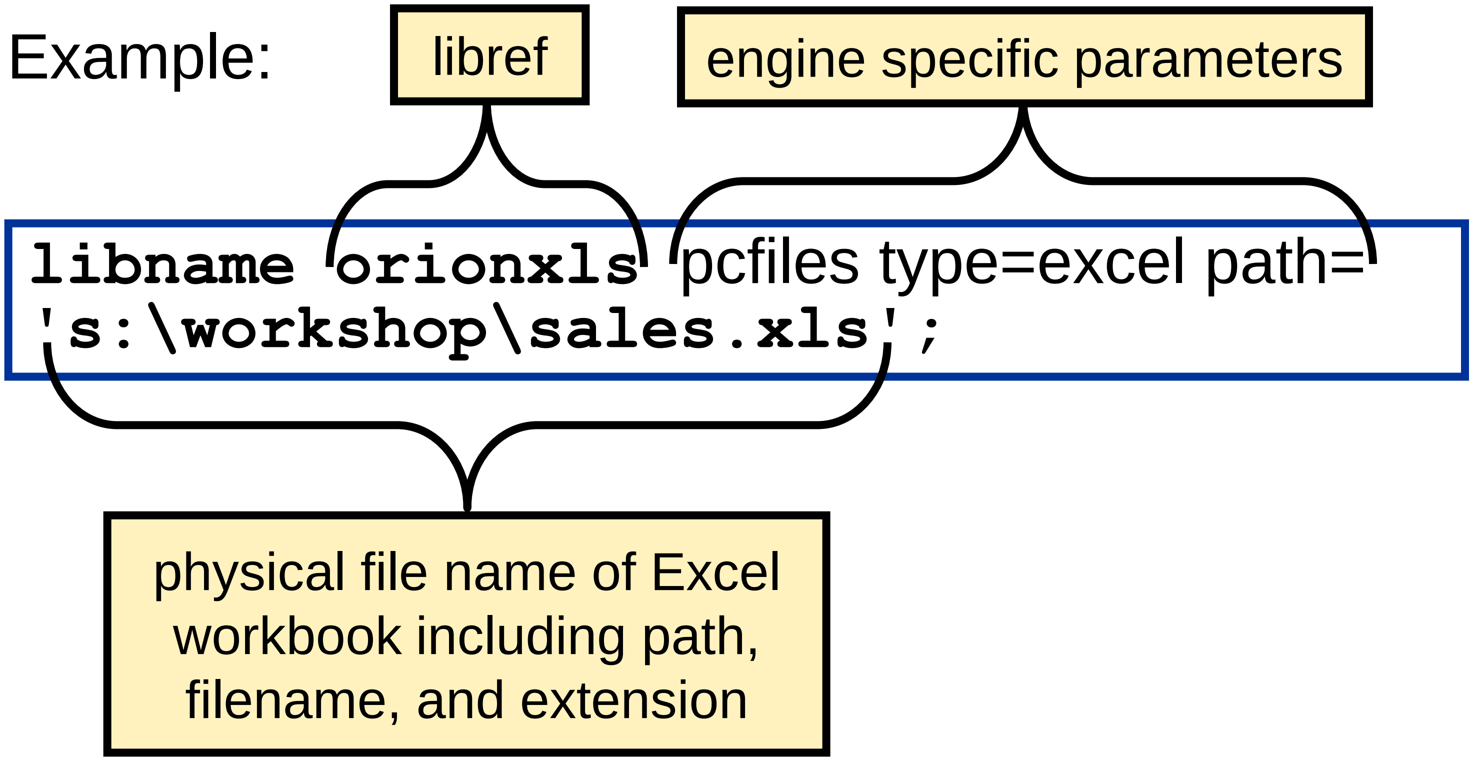
LIBNAME *libref* <*engine*> '*physical-file-name*' <*options*>**;**

This enables you to reference worksheets directly in a DATA step or SAS procedure, and to read from and write to a Microsoft Excel worksheet as though it were a SAS data set.

The SAS/ACCESS LIBNAME Statement

SAS/ACCESS Interface to PC File Formats and PC Files Server is required in order to use the SAS/ACCESS LIBNAME statement to access Excel workbooks.

Example:



libref

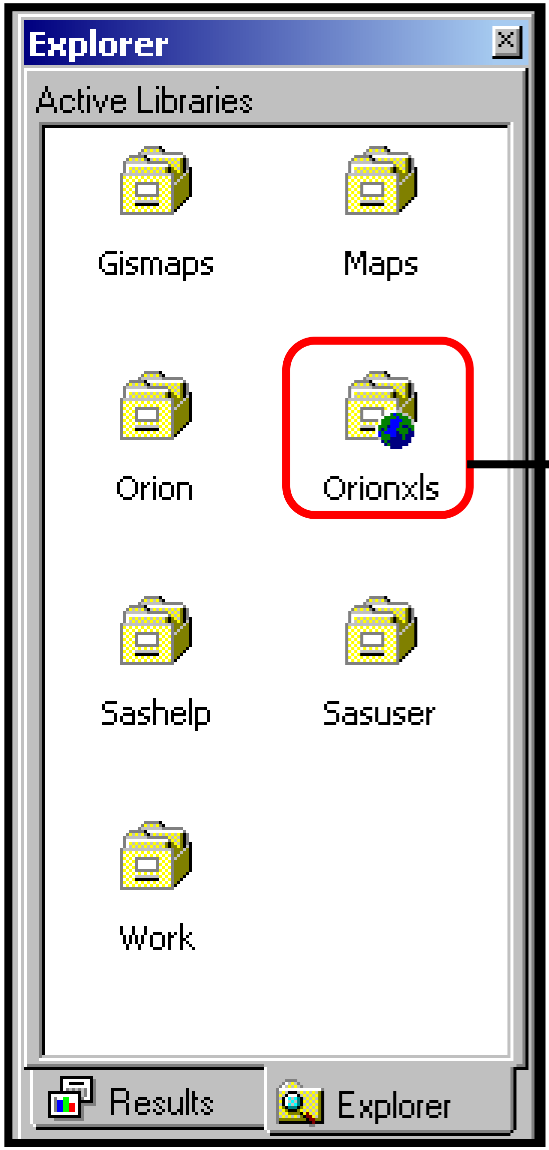
engine specific parameters

```
libname orionxls pcfiles type=excel path=
's:\workshop\sales.xls';
```

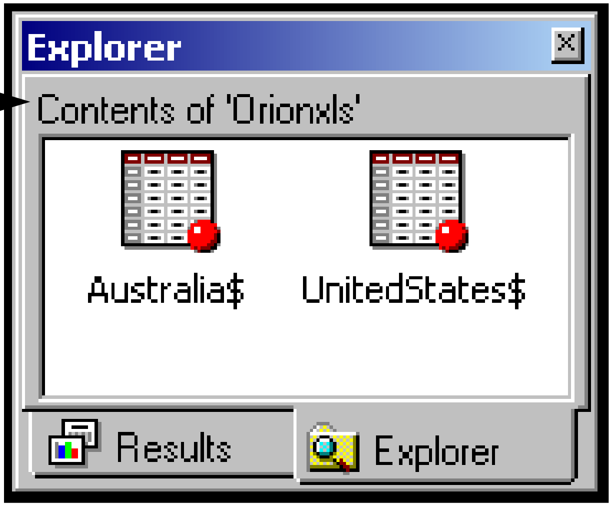
physical file name of Excel workbook including path, filename, and extension

p106d01

SAS Explorer Window



Each worksheet in the Excel workbook is treated as though it is a SAS data set.



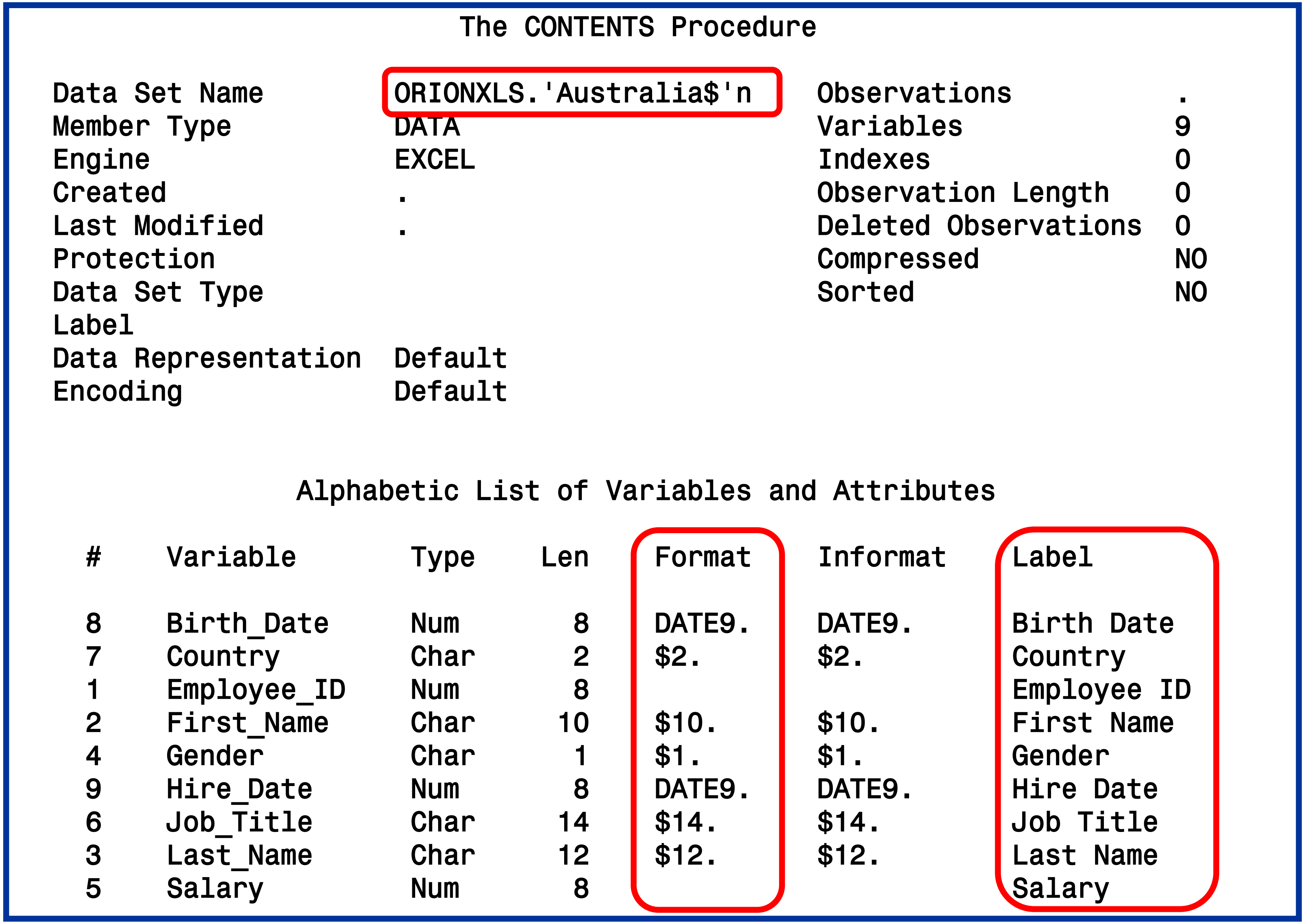
Worksheet names appear with a dollar sign at the end of the name.

The CONTENTS Procedure

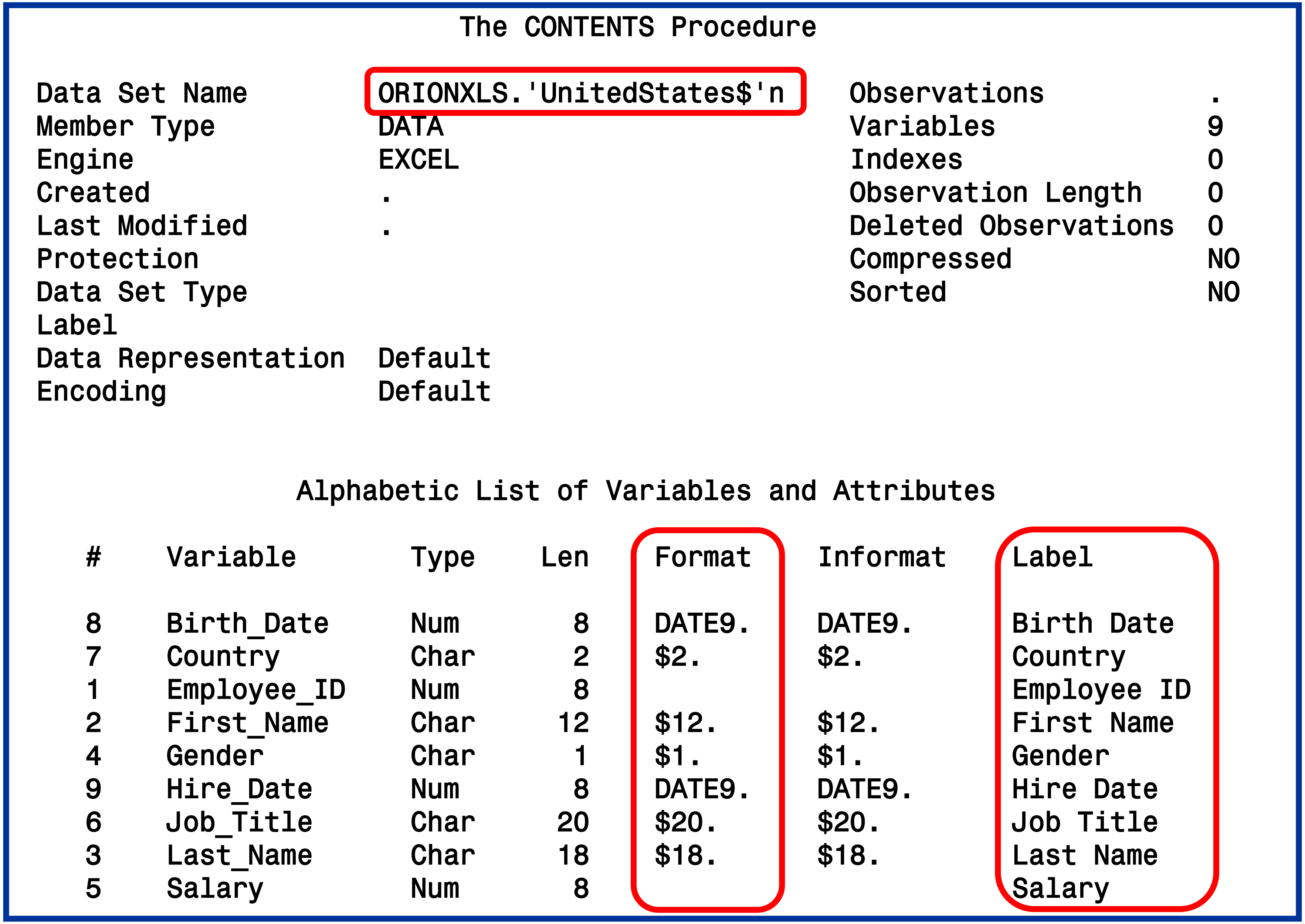
```
proc contents data=orionxls._all_;
run;
```

```
                    The CONTENTS Procedure

                          Directory

                Libref           ORIONXLS
                Engine           EXCEL
                Physical Name    sales.xls
                User             Admin



                                       DBMS
                               Member  Member
            #  Name            Type    Type

            1  Australia$      DATA    TABLE
            2  UnitedStates$   DATA    TABLE
```

p106d01



```
                    The CONTENTS Procedure

Data Set Name          ORIONXLS.'Australia$'n    Observations          .
Member Type            DATA                      Variables             9
Engine                 EXCEL                     Indexes               0
Created                .                         Observation Length    0
Last Modified          .                         Deleted Observations  0
Protection                                       Compressed            NO
Data Set Type                                    Sorted                NO
Label
Data Representation    Default
Encoding               Default


           Alphabetic List of Variables and Attributes

 #    Variable       Type    Len    Format     Informat    Label

 8    Birth_Date     Num       8    DATE9.     DATE9.      Birth Date
 7    Country        Char      2    $2.        $2.         Country
 1    Employee_ID    Num       8                           Employee ID
 2    First_Name     Char     10    $10.       $10.        First Name
 4    Gender         Char      1    $1.        $1.         Gender
 9    Hire_Date      Num       8    DATE9.     DATE9.      Hire Date
 6    Job_Title      Char     14    $14.       $14.        Job Title
 3    Last_Name      Char     12    $12.       $12.        Last Name
 5    Salary         Num       8                           Salary
```



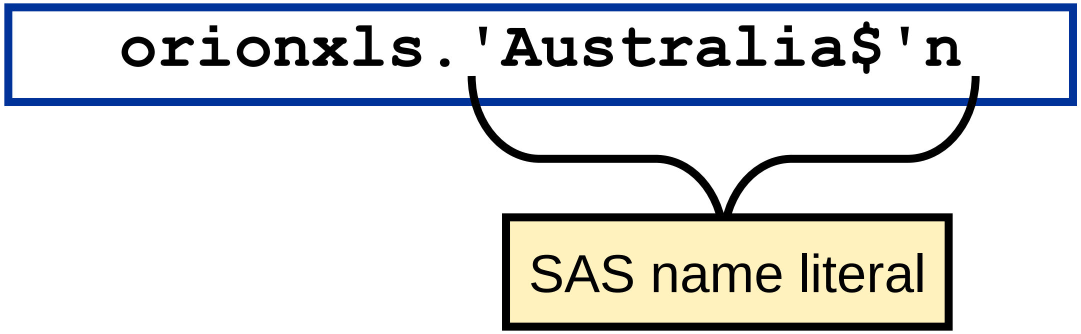
```
                          The CONTENTS Procedure

Data Set Name            ORIONXLS.'UnitedStates$'n    Observations             .
Member Type              DATA                         Variables                9
Engine                   EXCEL                        Indexes                  0
Created                  .                            Observation Length       0
Last Modified            .                            Deleted Observations     0
Protection                                            Compressed               NO
Data Set Type                                         Sorted                   NO
Label
Data Representation      Default
Encoding                 Default


                  Alphabetic List of Variables and Attributes

    #    Variable       Type    Len    Format     Informat    Label

    8    Birth_Date     Num       8    DATE9.     DATE9.      Birth Date
    7    Country        Char      2    $2.        $2.         Country
    1    Employee_ID    Num       8                           Employee ID
    2    First_Name     Char     12    $12.       $12.        First Name
    4    Gender         Char      1    $1.        $1.         Gender
    9    Hire_Date      Num       8    DATE9.     DATE9.      Hire Date
    6    Job_Title      Char     20    $20.       $20.        Job Title
    3    Last_Name      Char     18    $18.       $18.        Last Name
    5    Salary         Num       8                           Salary
```

SAS Name Literals

By default, special characters such as the $ are not allowed in data set names.

SAS name literals enable special characters to be included in data set names.

A *SAS name literal* is a name token that is expressed as a string within quotation marks, followed by the letter n.



```
orionxls.'Australia$'n
```

SAS name literal

The PRINT Procedure

```
proc print data=orionxls.'Australia$'n;
run;
```

Partial PROC PRINT Output

```
     Employee_                                                                            Birth_
Obs      ID     First_Name Last_Name    Gender Salary Job_Title       Country           Date Hire_Date

  1    120102   Tom        Zhou           M     108255 Sales Manager     AU      11AUG1969 01JUN1989
  2    120103   Wilson     Dawes          M      87975 Sales Manager     AU      22JAN1949 01JAN1974
  3    120121   Irenie     Elvish         F      26600 Sales Rep. II     AU      02AUG1944 01JAN1974
  4    120122   Christina  Ngan           F      27475 Sales Rep. II     AU      27JUL1954 01JUL1978
  5    120123   Kimiko     Hotstone       F      26190 Sales Rep. I      AU      28SEP1964 01OCT1985
  6    120124   Lucian     Daymond        M      26480 Sales Rep. I      AU      13MAY1959 01MAR1979
  7    120125   Fong       Hofmeister     M      32040 Sales Rep. IV     AU      06DEC1954 01MAR1979
  8    120126   Satyakam   Denny          M      26780 Sales Rep. II     AU      20SEP1988 01AUG2006
  9    120127   Sharryn    Clarkson       F      28100 Sales Rep. II     AU      04JAN1979 01NOV1998
 10    120128   Monica     Kletschkus     F      30890 Sales Rep. IV     AU      14JUL1986 01NOV2006
 11    120129   Alvin      Roebuck        M      30070 Sales Rep. III    AU      22NOV1964 01OCT1985
 12    120130   Kevin      Lyon           M      26955 Sales Rep. I      AU      14DEC1984 01MAY2006
 13    120131   Marinus    Surawski       M      26910 Sales Rep. I      AU      25SEP1979 01JAN2003
 14    120132   Fancine    Kaiser         F      28525 Sales Rep. III    AU      05APR1949 01OCT1978
 15    120133   Petrea     Soltau         F      27440 Sales Rep. II     AU      22APR1986 01OCT2006
```

p106d01

Poll
Quiz

6.01 Quiz

Which PROC PRINT step displays the worksheet containing employees from the United States?

a.

```
proc print data=orionxls.'UnitedStates';
run;
```

b.

```
proc print data=orionxls.'UnitedStates$';
run;
```

c.

```
proc print data=orionxls.'UnitedStates'n;
run;
```

d.

```
proc print data=orionxls.'UnitedStates$'n;
run;
```

6.01 Quiz – Correct Answer

Which PROC PRINT step displays the worksheet containing employees from the United States?

a.

```
proc print data=orionxls.'UnitedStates';
run;
```

b.

```
proc print data=orionxls.'UnitedStates$';
run;
```

c.

```
proc print data=orionxls.'UnitedStates'n;
run;
```

[✓ d.]

```
proc print data=orionxls.'UnitedStates$'n;
run;
```

Business Scenario

Create a temporary SAS data set named **Work.subset2** from the Excel workbook named **sales.xls**.

```
libname orionxls pcfiles type=excel
path='s:\workshop\sales.xls';

data work.subset2;
   set orionxls.'Australia$'n;
   where Job_Title contains 'Rep';
   keep First_Name Last_Name Salary
        Job_Title Hire_Date;
   label Job_Title='Sales Title'
         Hire_Date='Date Hired';
   format Salary comma10. Hire_Date weekdate.;
run;
```

p106d02

Business Scenario

```
proc contents data=work.subset2;
run;
```

Partial PROC CONTENTS Output

```
                  Alphabetic List of Variables and Attributes

 #    Variable      Type    Len    Format        Informat    Label

 1    First_Name    Char     10    $10.          $10.        First Name
 5    Hire_Date     Num       8    WEEKDATE.     DATE9.      Date Hired
 4    Job_Title     Char     14    $14.          $14.        Sales Title
 2    Last_Name     Char     12    $12.          $12.        Last Name
 3    Salary        Num       8    COMMA10.                  Salary
```

p106d02

Business Scenario

```
proc print data=work.subset2 label;
run;
```

Partial PROC PRINT Output

```
Obs First Name Last Name      Salary Sales Title            Date Hired

   1 Irenie     Elvish         26,600 Sales Rep. II     Tuesday, January 1, 1974
   2 Christina  Ngan           27,475 Sales Rep. II       Saturday, July 1, 1978
   3 Kimiko     Hotstone       26,190 Sales Rep. I      Tuesday, October 1, 1985
   4 Lucian     Daymond        26,480 Sales Rep. I       Thursday, March 1, 1979
   5 Fong       Hofmeister     32,040 Sales Rep. IV      Thursday, March 1, 1979
   6 Satyakam   Denny          26,780 Sales Rep. II      Tuesday, August 1, 2006
   7 Sharryn    Clarkson       28,100 Sales Rep. II     Sunday, November 1, 1998
   8 Monica     Kletschkus     30,890 Sales Rep. IV  Wednesday, November 1, 2006
   9 Alvin      Roebuck        30,070 Sales Rep. III    Tuesday, October 1, 1985
  10 Kevin      Lyon           26,955 Sales Rep. I          Monday, May 1, 2006
  11 Marinus    Surawski       26,910 Sales Rep. I    Wednesday, January 1, 2003
  12 Fancine    Kaiser         28,525 Sales Rep. III     Sunday, October 1, 1978
```

p106d02

Disassociating a Libref

If SAS has a libref assigned to an Excel workbook, the workbook cannot be opened in Excel. To disassociate a libref, use a LIBNAME statement and specify the libref and the CLEAR option.

```
libname orionxls pcfiles type=excel
path='s:\workshop\sales.xls';

data work.subset2;
   set orionxls.'Australia$'n;
   ...
run;

libname orionxls clear;
```

SAS disconnects from the data source and closes any resources that are associated with that libref's connection.

p106d02

Chapter 6: Reading Excel Worksheets

6.1 Using Excel Data as Input

6.2 Doing More with Excel Worksheets

Objectives

- Use the DATA step to create an Excel worksheet from a SAS data set.
- Use the COPY procedure to create an Excel worksheet from a SAS data set.
- Use the IMPORT Wizard and procedure to read an Excel worksheet.
- Use the EXPORT Wizard and procedure to create an Excel worksheet.

Creating Excel Worksheets

In addition to reading an Excel worksheet, the SAS/ACCESS LIBNAME statement with the DATA step can be used to create an Excel worksheet.

```
libname orionxls pcfiles type=excel
path='s:\workshop\qtr2007a.xls';

data orionxls.qtr1_2007;
   set orion.qtr1_2007;
run;

data orionxls.qtr2_2007;
   set orion.qtr2_2007;
run;

proc contents data=orionxls._all_;
run;

libname orionxls clear;
```

p106d03

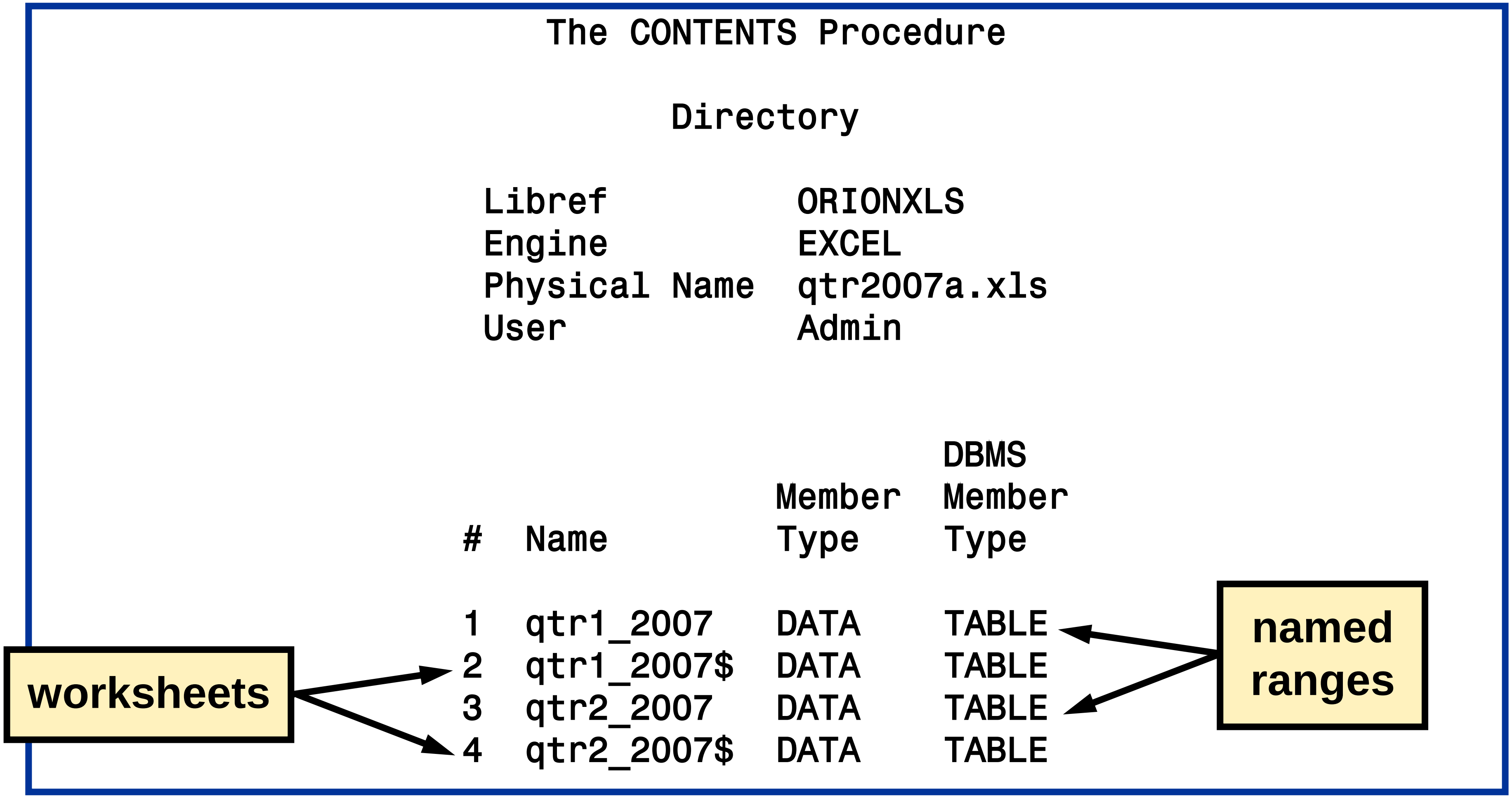
Creating Excel Worksheets

Partial SAS Log

```
70   data orionxls.qtr1_2007;
71      set orion.qtr1_2007;
72
73   run;

NOTE: SAS variable labels, formats, and lengths are not written to DBMS tables.
NOTE: There were 22 observations read from the data set ORION.QTR1_2007.
NOTE: The data set ORIONXLS.qtr1_2007 has 22 observations and 5 variables.


74   data orionxls.qtr2_2007;
75      set orion.qtr2_2007;
76   run;

NOTE: SAS variable labels, formats, and lengths are not written to DBMS tables.
NOTE: There were 36 observations read from the data set ORION.QTR2_2007.
NOTE: The data set ORIONXLS.qtr2_2007 has 36 observations and 6 variables.
```

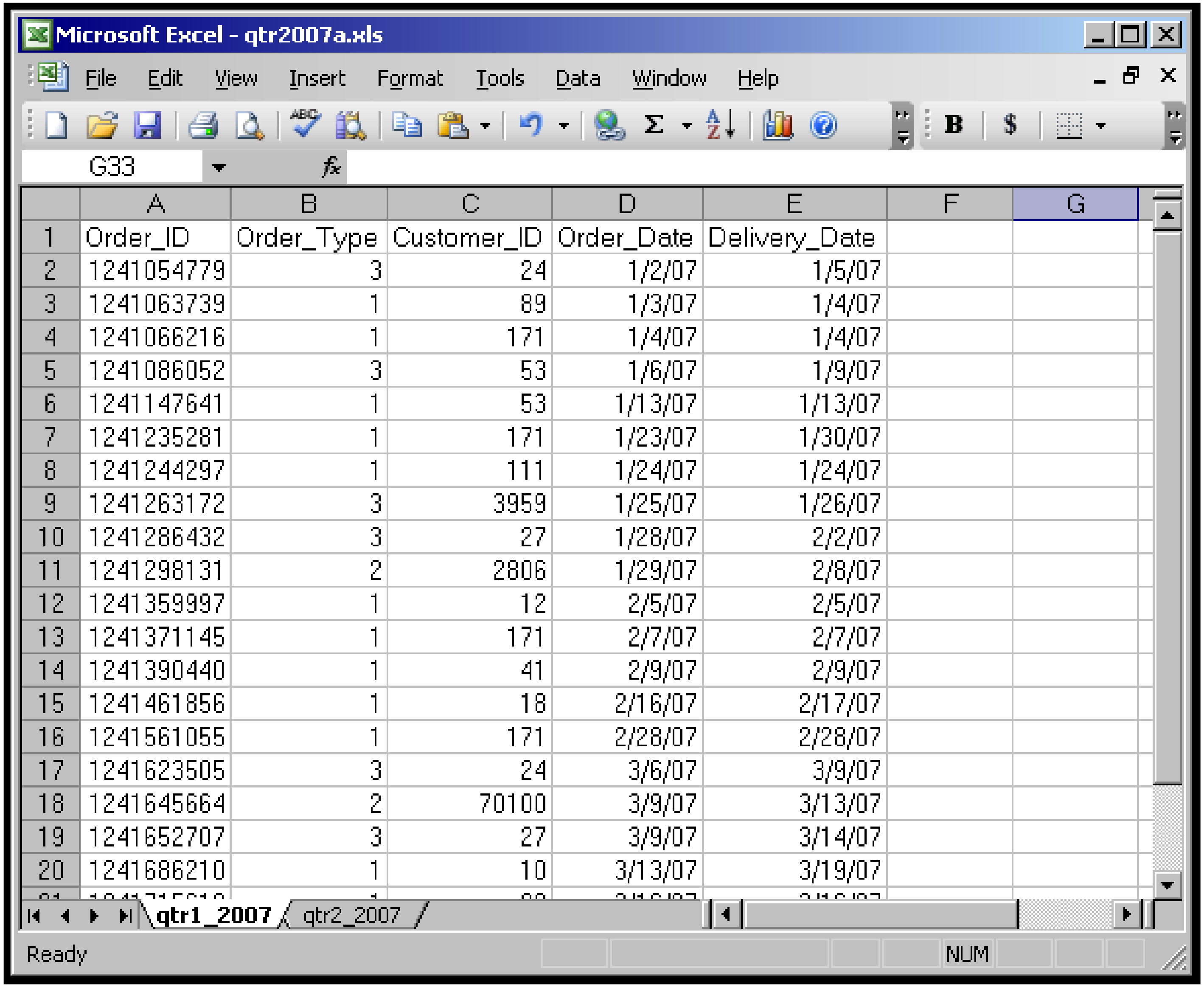
Creating Excel Worksheets

Partial PROC CONTENTS Output



```
                  The CONTENTS Procedure

                        Directory

          Libref           ORIONXLS
          Engine           EXCEL
          Physical Name    qtr2007a.xls
          User             Admin



                                       DBMS
                             Member    Member
         #   Name            Type      Type

         1   qtr1_2007       DATA      TABLE
         2   qtr1_2007$      DATA      TABLE
         3   qtr2_2007       DATA      TABLE
         4   qtr2_2007$      DATA      TABLE
```

Creating Excel Worksheets



	A	B	C	D	E	F	G
1	Order_ID	Order_Type	Customer_ID	Order_Date	Delivery_Date		
2	1241054779	3	24	1/2/07	1/5/07		
3	1241063739	1	89	1/3/07	1/4/07		
4	1241066216	1	171	1/4/07	1/4/07		
5	1241086052	3	53	1/6/07	1/9/07		
6	1241147641	1	53	1/13/07	1/13/07		
7	1241235281	1	171	1/23/07	1/30/07		
8	1241244297	1	111	1/24/07	1/24/07		
9	1241263172	3	3959	1/25/07	1/26/07		
10	1241286432	3	27	1/28/07	2/2/07		
11	1241298131	2	2806	1/29/07	2/8/07		
12	1241359997	1	12	2/5/07	2/5/07		
13	1241371145	1	171	2/7/07	2/7/07		
14	1241390440	1	41	2/9/07	2/9/07		
15	1241461856	1	18	2/16/07	2/17/07		
16	1241561055	1	171	2/28/07	2/28/07		
17	1241623505	3	24	3/6/07	3/9/07		
18	1241645664	2	70100	3/9/07	3/13/07		
19	1241652707	3	27	3/9/07	3/14/07		
20	1241686210	1	10	3/13/07	3/19/07		

Creating Excel Worksheets

As an alternative to the DATA step, the COPY procedure can be used to create an Excel worksheet.

```
libname orionxls pcfiles type=excel
path='s:\workshop\qtr2007b.xls';

proc copy  in=orion out=orionxls;
   select qtr1_2007 qtr2_2007;
run;

proc contents data=orionxls._all_;
run;

libname orionxls clear;
```

p106d03

Creating Excel Worksheets

Partial SAS Log

```
82   proc copy  in=orion out=orionxls;
83      select qtr1_2007 qtr2_2007;
84   run;

NOTE: Copying ORION.QTR1_2007 to ORIONXLS.QTR1_2007 (memtype=DATA).
NOTE: SAS variable labels, formats, and lengths are not written to DBMS tables.
NOTE: There were 22 observations read from the data set ORION.QTR1_2007.
NOTE: The data set ORIONXLS.QTR1_2007 has 22 observations and 5 variables.
NOTE: Copying ORION.QTR2_2007 to ORIONXLS.QTR2_2007 (memtype=DATA).
NOTE: SAS variable labels, formats, and lengths are not written to DBMS tables.
NOTE: There were 36 observations read from the data set ORION.QTR2_2007.
NOTE: The data set ORIONXLS.QTR2_2007 has 36 observations and 6 variables.
```

Import/Export Wizards and Procedures

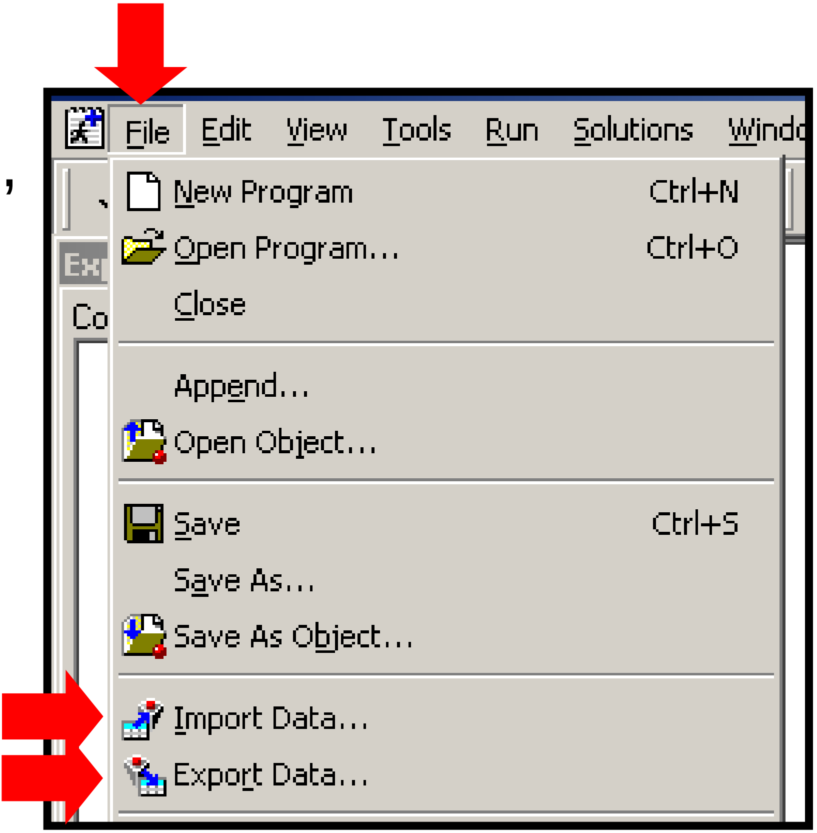
The Import/Export Wizards and IMPORT/EXPORT procedures enable you to read and write data between SAS data sets and external PC files.

The Import/Export Wizards and procedures are part of Base SAS and enable access to delimited files. If you have a license to SAS/ACCESS Interface to PC File Formats, you can also access Microsoft Excel, Microsoft Access, dBASE, JMP, Lotus 1-2-3, SPSS, Stata, and Paradox files.

Import/Export Wizards and Procedures

The wizards and procedures have similar capabilities; the wizards are point-and-click interfaces and the procedures are code-based.

To invoke the wizards from the SAS windowing environment, select **File** and **Import Data** or **Export Data**.



The Import Wizard

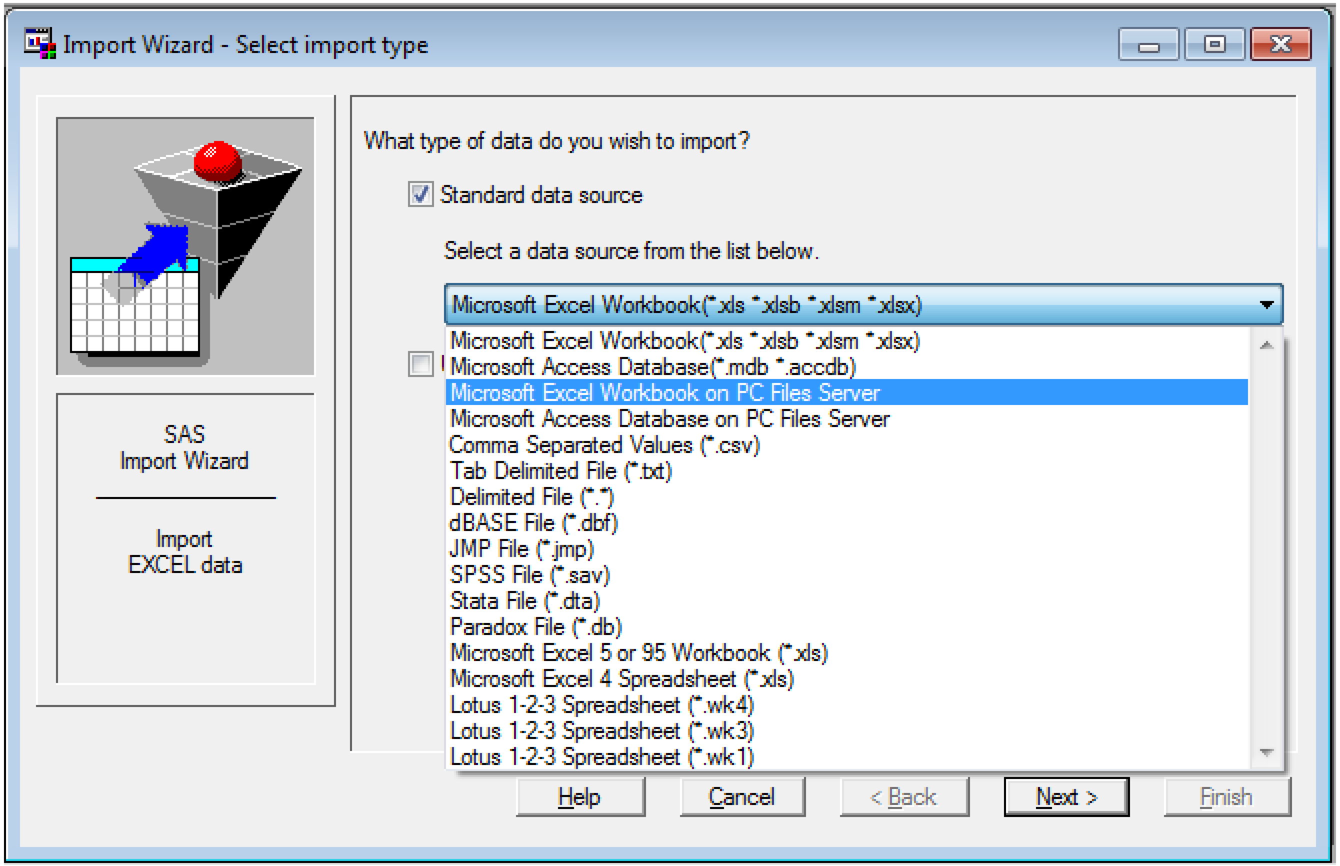
The Import Wizard enables you to read data from an external data source and write it to a SAS data set.

Steps of the Import Wizard:

1. Select the type of file you are importing.
2. Locate the input file.
3. Select the table range or worksheet from which to import data.
4. Select a location to store the imported file.
5. Save the generated PROC IMPORT code. (Optional)

The Import Wizard

1. Select the type of file you are importing.



Import Wizard - Select import type

SAS
Import Wizard

Import
EXCEL data

What type of data do you wish to import?

☑ Standard data source

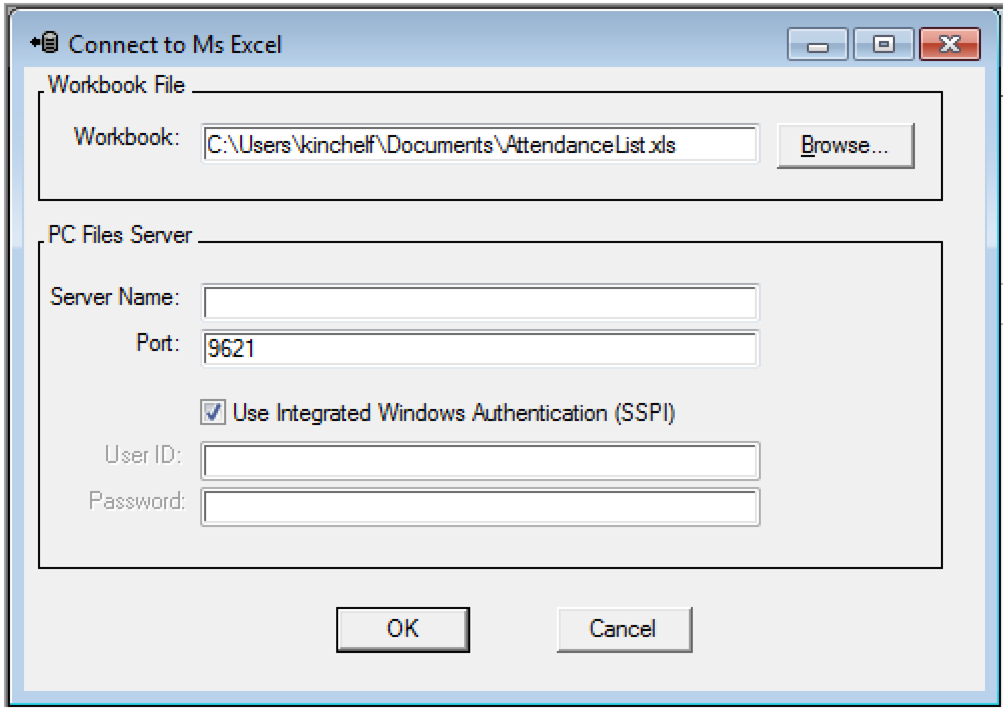
Select a data source from the list below.

Microsoft Excel Workbook(*.xls *.xlsb *.xlsm *.xlsx)

- Microsoft Excel Workbook(*.xls *.xlsb *.xlsm *.xlsx)
- Microsoft Access Database(*.mdb *.accdb)
- Microsoft Excel Workbook on PC Files Server
- Microsoft Access Database on PC Files Server
- Comma Separated Values (*.csv)
- Tab Delimited File (*.txt)
- Delimited File (*.*)
- dBASE File (*.dbf)
- JMP File (*.jmp)
- SPSS File (*.sav)
- Stata File (*.dta)
- Paradox File (*.db)
- Microsoft Excel 5 or 95 Workbook (*.xls)
- Microsoft Excel 4 Spreadsheet (*.xls)
- Lotus 1-2-3 Spreadsheet (*.wk4)
- Lotus 1-2-3 Spreadsheet (*.wk3)
- Lotus 1-2-3 Spreadsheet (*.wk1)

The Import Wizard

2. Locate the input file.



Connect to Ms Excel

Workbook File

Workbook: C:\Users\kinchelf\Documents\AttendanceList.xls

PC Files Server

Server Name:

Port: 9621

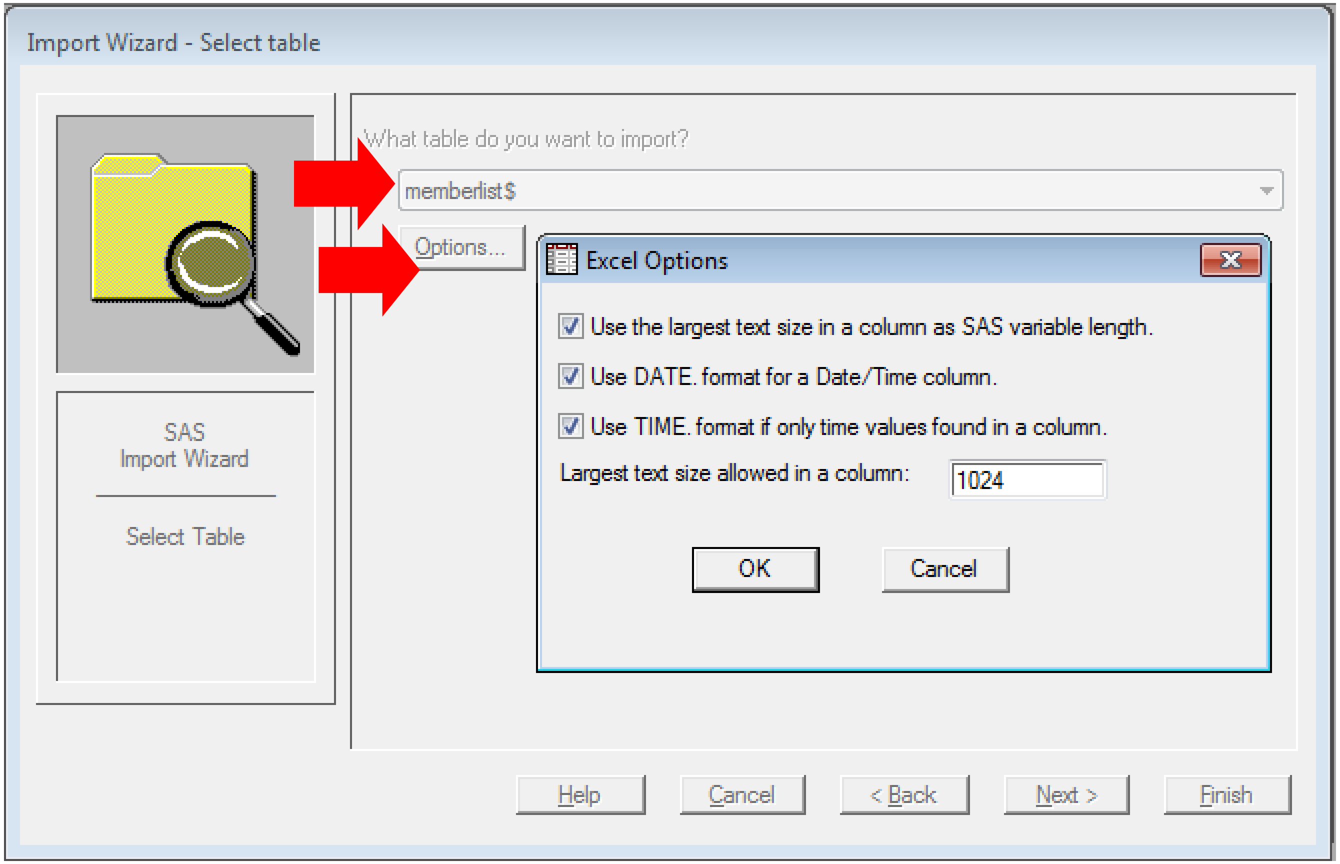
☑ Use Integrated Windows Authentication (SSPI)

User ID:

Password:

The Import Wizard

3. Select the table range or worksheet from which to import data.



Import Wizard - Select table

What table do you want to import?

memberlist$

Excel Options

☑ Use the largest text size in a column as SAS variable length.

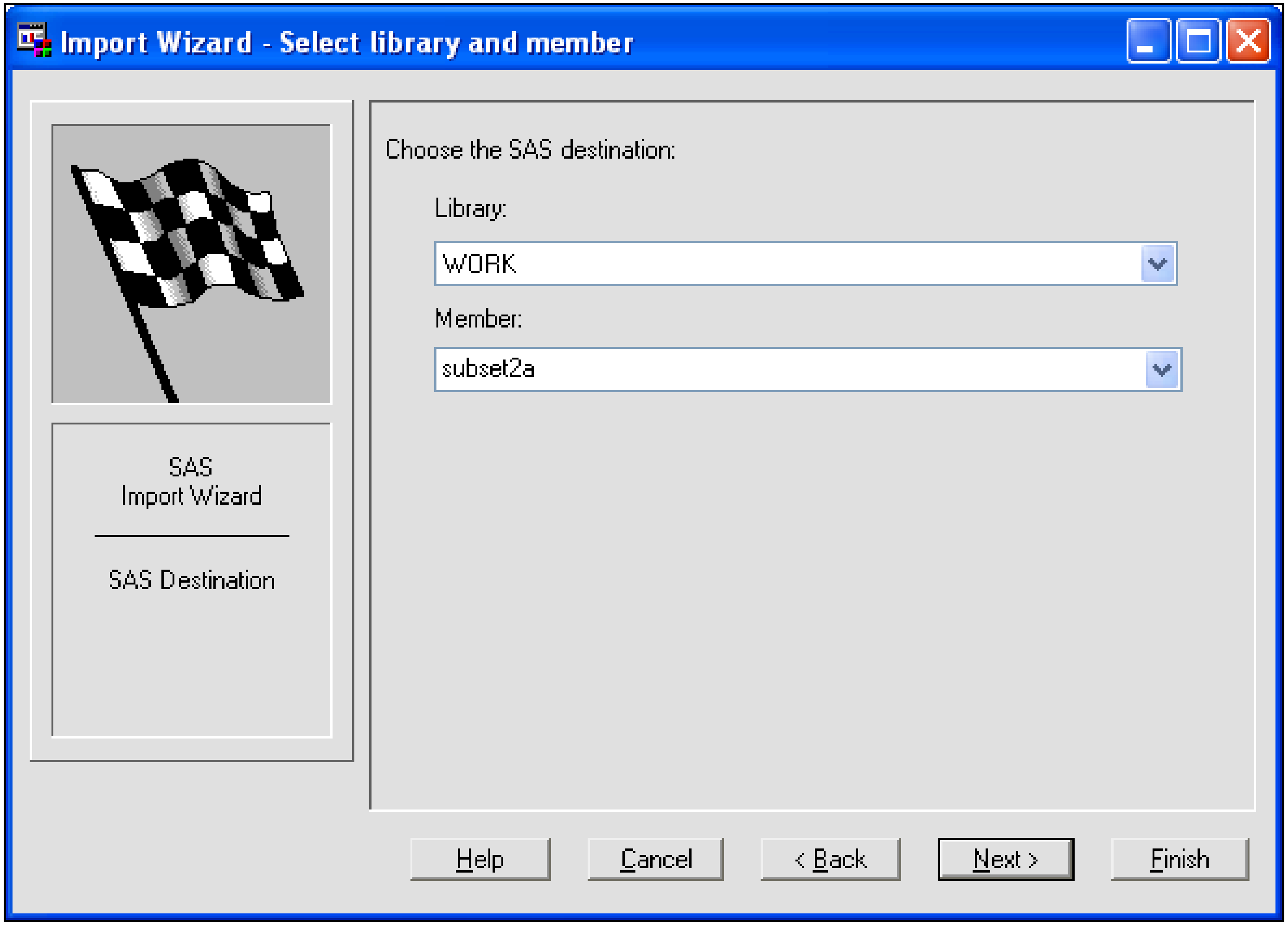
☑ Use DATE. format for a Date/Time column.

☑ Use TIME. format if only time values found in a column.

Largest text size allowed in a column: 1024

The Import Wizard

4. Select a location to store the imported file.



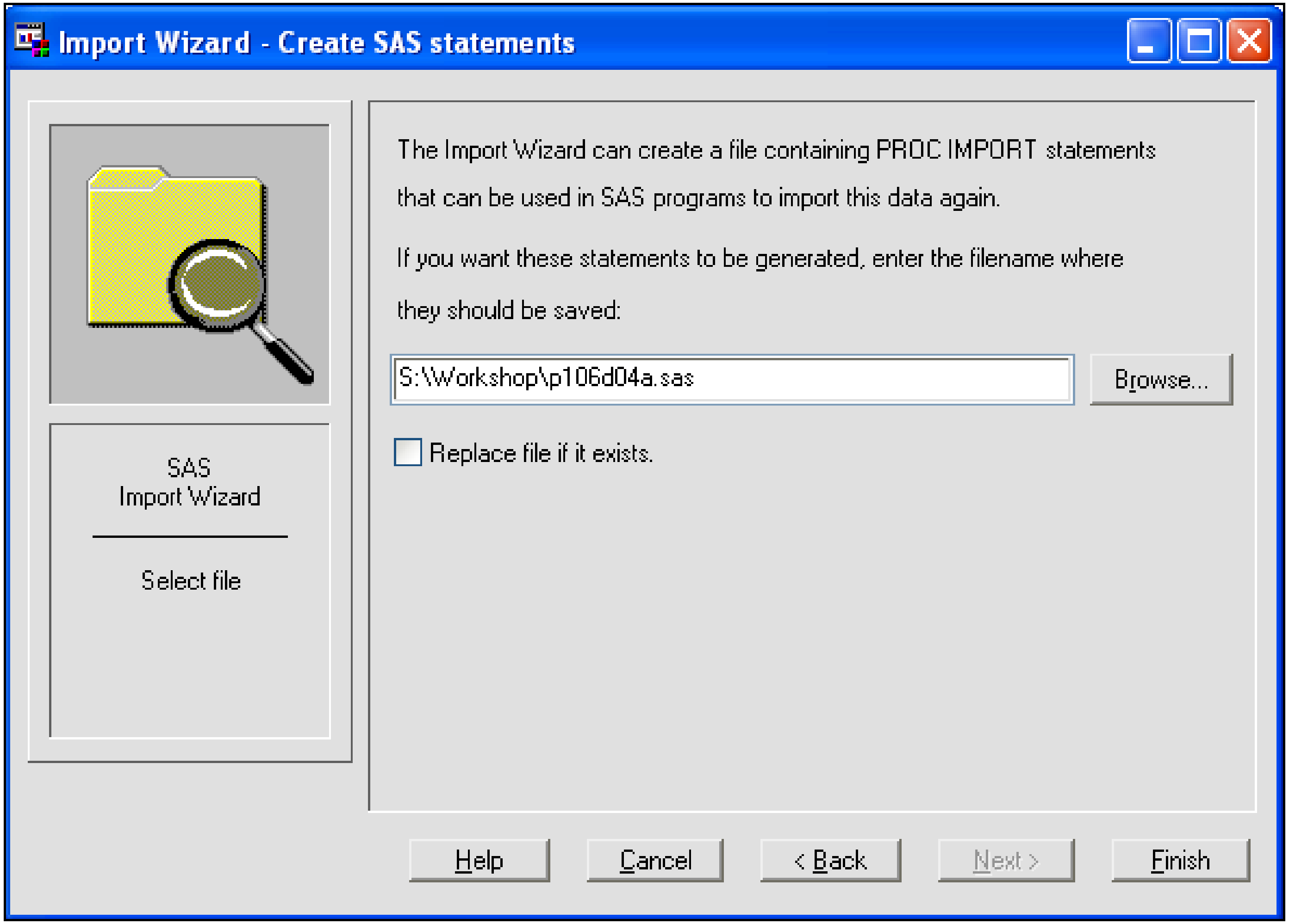
Choose the SAS destination:

Library: WORK

Member: subset2a

The Import Wizard

5. Save the generated PROC IMPORT code. (Optional)



The Import Wizard can create a file containing PROC IMPORT statements that can be used in SAS programs to import this data again.

If you want these statements to be generated, enter the filename where they should be saved:

S:\Workshop\p106d04a.sas

□ Replace file if it exists.

The Import Wizard

SAS Log

```
NOTE: WORK.SUBSET2A data set was successfully created.
```

```
proc print data=work.subset2a;
run;
```

Partial PROC PRINT Output

```
    Employee_                                                                 Birth_
Obs     ID    First_Name Last_Name    Gender Salary Job_Title      Country      Date Hire_Date

  1   120102  Tom        Zhou           M    108255 Sales Manager    AU    11AUG1969 01JUN1989
  2   120103  Wilson     Dawes          M     87975 Sales Manager    AU    22JAN1949 01JAN1974
  3   120121  Irenie     Elvish         F     26600 Sales Rep. II    AU    02AUG1944 01JAN1974
  4   120122  Christina  Ngan           F     27475 Sales Rep. II    AU    27JUL1954 01JUL1978
  5   120123  Kimiko     Hotstone       F     26190 Sales Rep. I     AU    28SEP1964 01OCT1985
```

p106d04

The Import Wizard

```
proc contents data=work.subset2a;
run;
```

Partial PROC CONTENTS Output

```
                    Alphabetic List of Variables and Attributes

     #    Variable       Type     Len     Format     Informat     Label

     8    Birth_Date     Num        8     DATE9.     DATE9.       Birth Date
     7    Country        Char       2     $2.        $2.          Country
     1    Employee_ID    Num        8                             Employee ID
     2    First_Name     Char      10     $10.       $10.         First Name
     4    Gender         Char       1     $1.        $1.          Gender
     9    Hire_Date      Num        8     DATE9.     DATE9.       Hire Date
     6    Job_Title      Char      14     $14.       $14.         Job Title
     3    Last_Name      Char      12     $12.       $12.         Last Name
     5    Salary         Num        8                             Salary
```

p106d04

The IMPORT Procedure

The program **p106d04a** was created from the Import Wizard.

```
PROC IMPORT OUT= WORK.subset2a
            DATAFILE= "S:\Workshop\sales.xls"
            DBMS=EXCELCS REPLACE;
     RANGE="Australia$";
     SCANTEXT=YES;
     USEDATE=YES;
     SCANTIME=YES;
RUN;
```

p106d04a

The Export Wizard

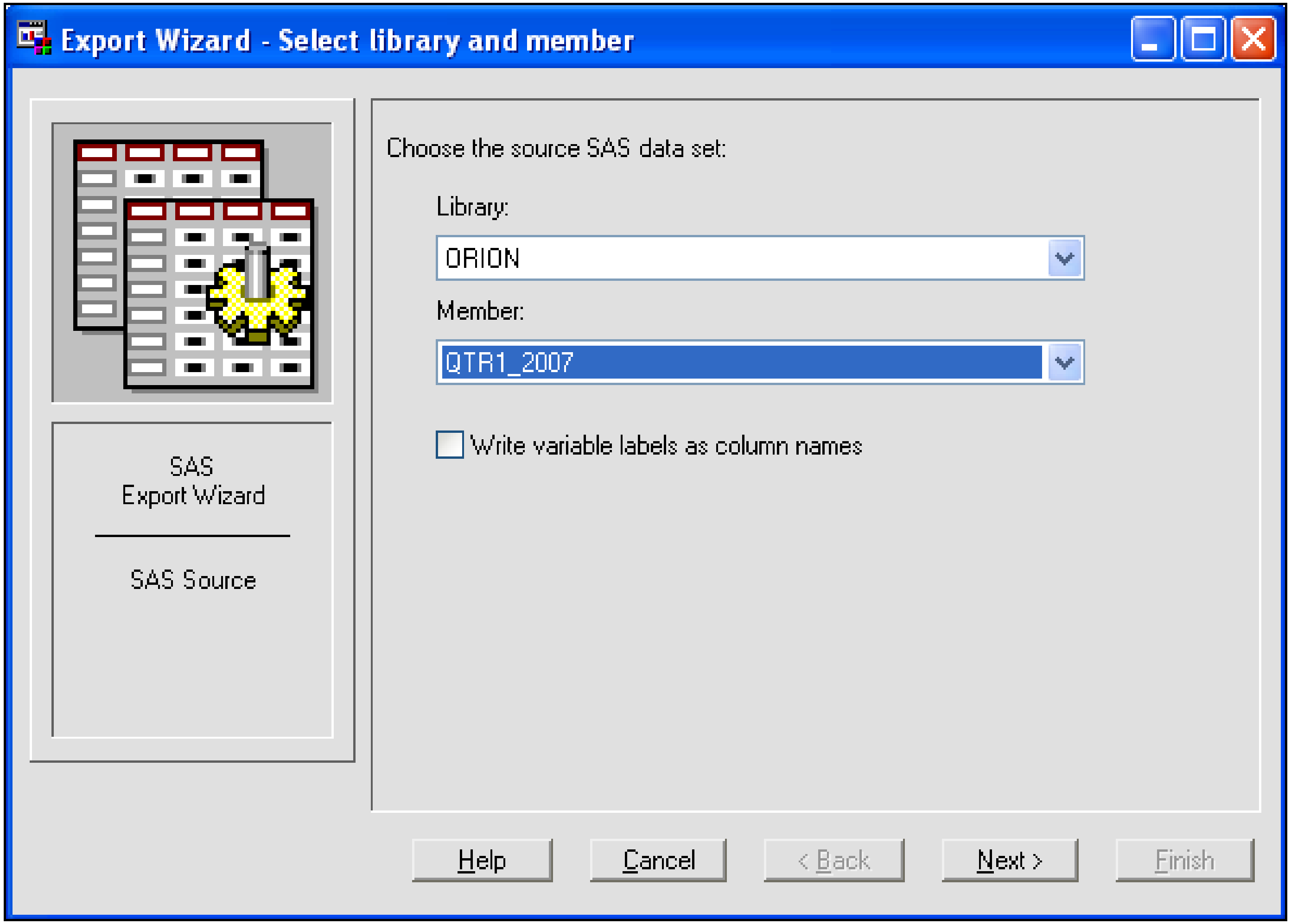
The Export Wizard reads data from a SAS data set and writes it to an external file source.

Steps of the Export Wizard:

1. Select the data set from which you want to export data.
2. Select the type of data source to which you want to export files.
3. Assign the output file.
4. Assign the table name.
5. Save the generated PROC EXPORT code. (Optional)

The Export Wizard

1. Select the data set from which you want to export data.



Export Wizard - Select library and member

SAS Export Wizard

SAS Source

Choose the source SAS data set:

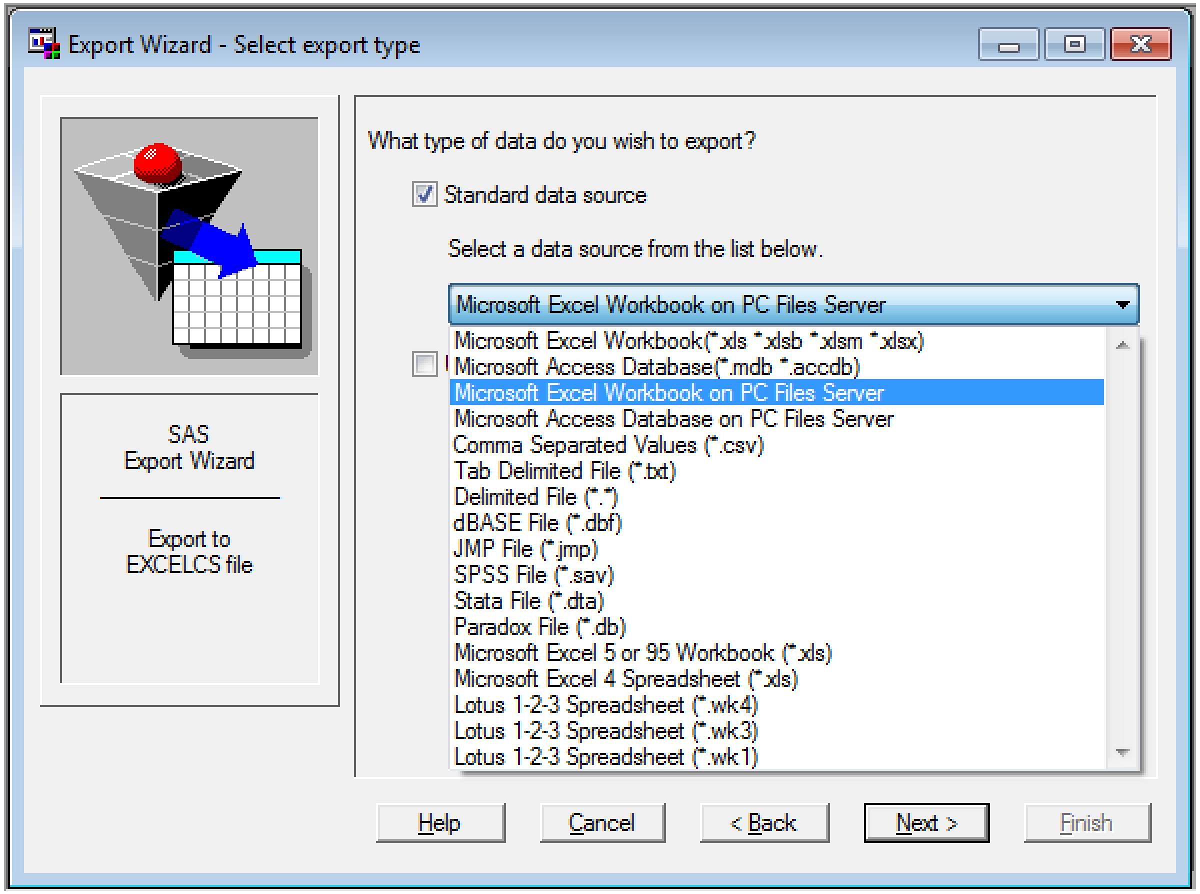
Library: ORION

Member: QTR1_2007

□ Write variable labels as column names

The Export Wizard

2. Select the type of data source to which you want to export files.



Export Wizard - Select export type

SAS Export Wizard

Export to EXCELCS file

What type of data do you wish to export?

☑ Standard data source

Select a data source from the list below.

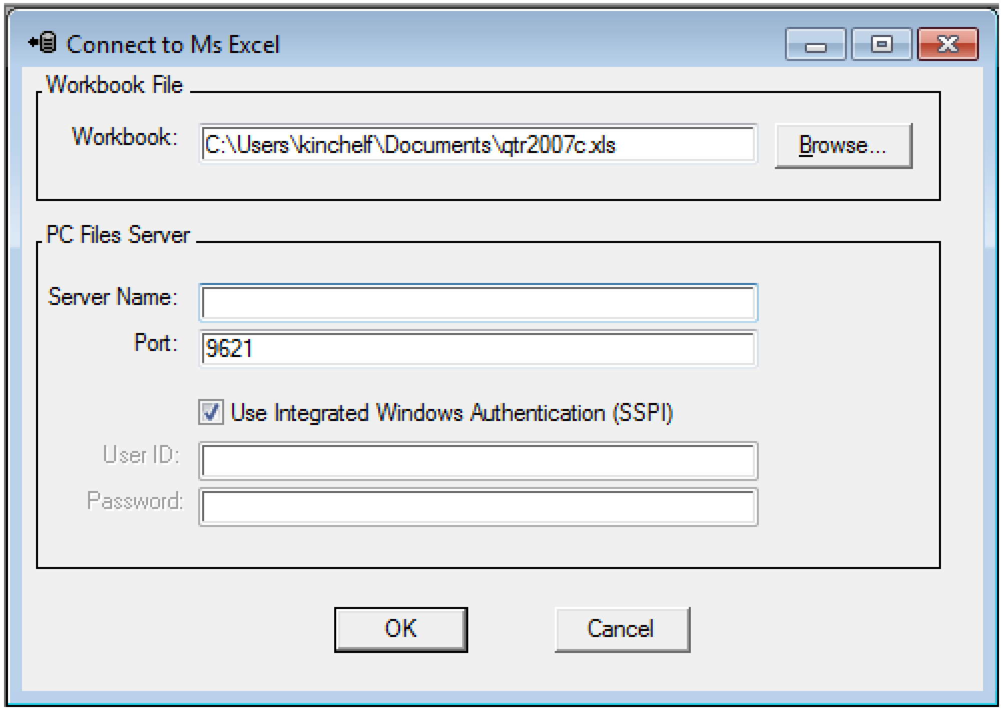
Microsoft Excel Workbook on PC Files Server

□

- Microsoft Excel Workbook(*.xls *.xlsb *.xlsm *.xlsx)
- Microsoft Access Database(*.mdb *.accdb)
- Microsoft Excel Workbook on PC Files Server
- Microsoft Access Database on PC Files Server
- Comma Separated Values (*.csv)
- Tab Delimited File (*.txt)
- Delimited File (*.*)
- dBASE File (*.dbf)
- JMP File (*.jmp)
- SPSS File (*.sav)
- Stata File (*.dta)
- Paradox File (*.db)
- Microsoft Excel 5 or 95 Workbook (*.xls)
- Microsoft Excel 4 Spreadsheet (*.xls)
- Lotus 1-2-3 Spreadsheet (*.wk4)
- Lotus 1-2-3 Spreadsheet (*.wk3)
- Lotus 1-2-3 Spreadsheet (*.wk1)

The Export Wizard

3. Assign the output file.



Connect to Ms Excel

Workbook File

Workbook: C:\Users\kinchelf\Documents\qtr2007c.xls

PC Files Server

Server Name:

Port: 9621

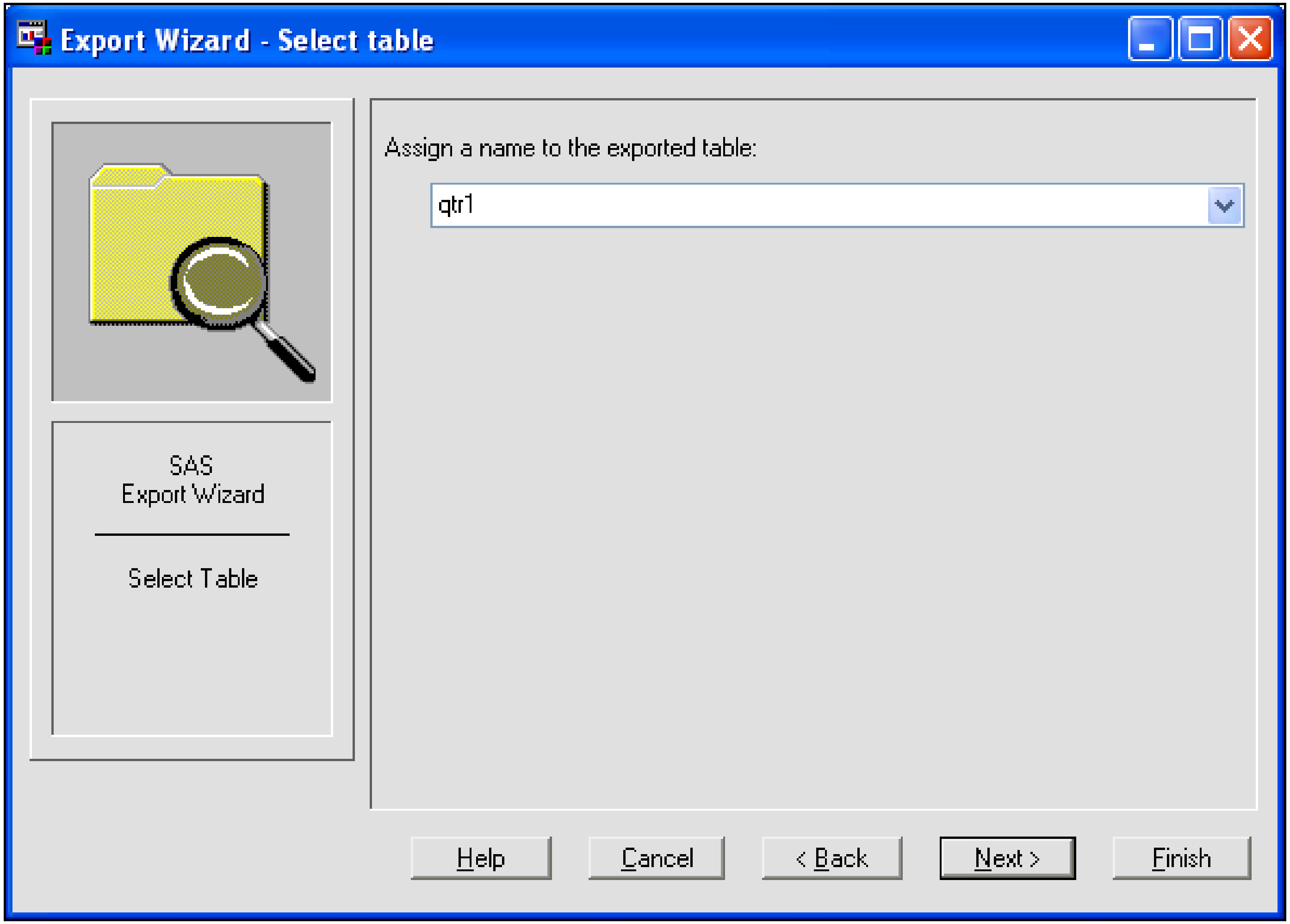
☑ Use Integrated Windows Authentication (SSPI)

User ID:

Password:

The Export Wizard

4. Assign the table name.

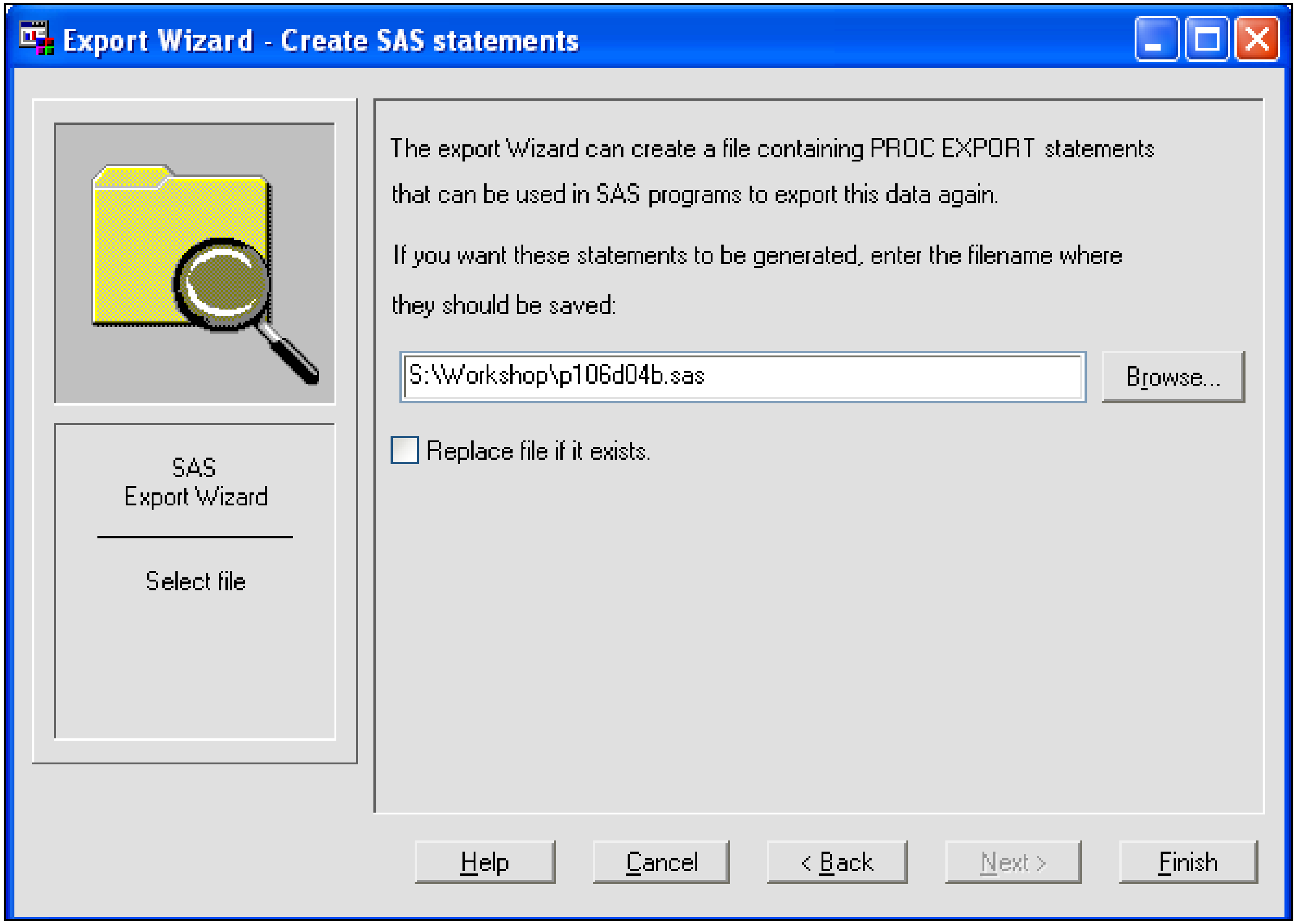


Assign a name to the exported table:

qtr1

The Export Wizard

5. Save the generated PROC EXPORT code. (Optional)



Export Wizard - Create SAS statements

The export Wizard can create a file containing PROC EXPORT statements that can be used in SAS programs to export this data again.

If you want these statements to be generated, enter the filename where they should be saved:

S:\Workshop\p106d04b.sas

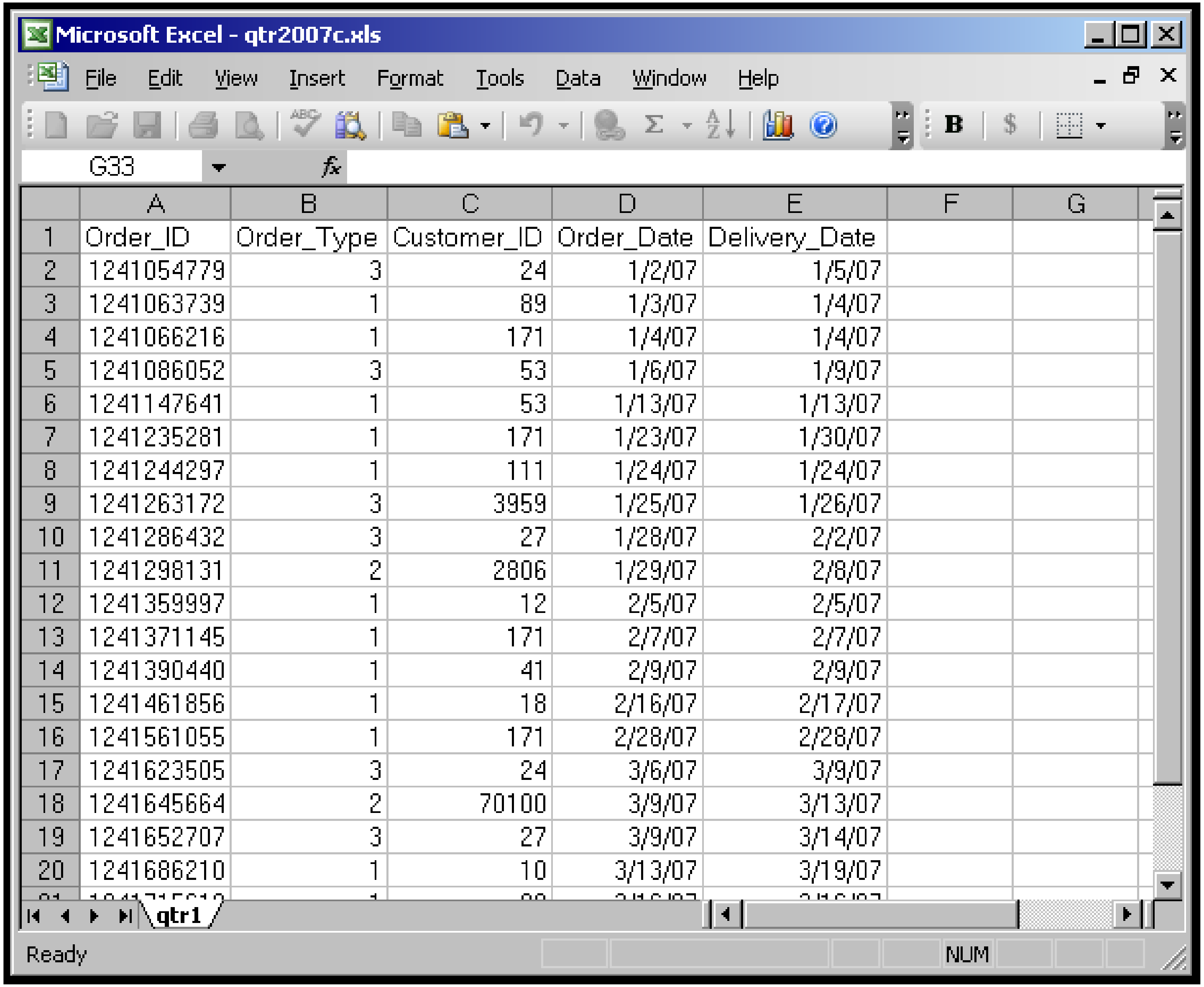
□ Replace file if it exists.

The Export Wizard

SAS Log

```
NOTE: File "S:\Workshop\qtr2007c.xls" will be created if the export
      process succeeds.
NOTE: "qtr1" table was successfully created.
```

The Export Wizard



	A	B	C	D	E	F	G
1	Order_ID	Order_Type	Customer_ID	Order_Date	Delivery_Date		
2	1241054779	3	24	1/2/07	1/5/07		
3	1241063739	1	89	1/3/07	1/4/07		
4	1241066216	1	171	1/4/07	1/4/07		
5	1241086052	3	53	1/6/07	1/9/07		
6	1241147641	1	53	1/13/07	1/13/07		
7	1241235281	1	171	1/23/07	1/30/07		
8	1241244297	1	111	1/24/07	1/24/07		
9	1241263172	3	3959	1/25/07	1/26/07		
10	1241286432	3	27	1/28/07	2/2/07		
11	1241298131	2	2806	1/29/07	2/8/07		
12	1241359997	1	12	2/5/07	2/5/07		
13	1241371145	1	171	2/7/07	2/7/07		
14	1241390440	1	41	2/9/07	2/9/07		
15	1241461856	1	18	2/16/07	2/17/07		
16	1241561055	1	171	2/28/07	2/28/07		
17	1241623505	3	24	3/6/07	3/9/07		
18	1241645664	2	70100	3/9/07	3/13/07		
19	1241652707	3	27	3/9/07	3/14/07		
20	1241686210	1	10	3/13/07	3/19/07		

The EXPORT Procedure

The program **p106d04b** was created from the Export Wizard.

```
PROC EXPORT DATA= ORION.QTR1_2007
            OUTFILE= "S:\Workshop\qtr2007c.xls"
            DBMS=EXCELCS REPLACE;
     SHEET="qtr1";
RUN;
```

✏ The RANGE statement is not supported and is ignored in the EXPORT procedure.

p106d04b