STAT604

Lesson SAS 12

Outputting to Multiple Data Sets

The DATA statement can specify multiple output data sets.

```
data EmpsAUC EmpsOnly PhoneOnly;
   merge EmpsAU(in=Emps) PhoneC(in=Cell);
   by EmpID;
   if Emps=1 and Cell=1
        then output EmpsAUC;
   else if Emps=1 and Cell=0
        then output EmpsOnly;
   else if Emps=0 and Cell=1
        then output PhoneOnly;
run;
```

p110d07

Outputting to Multiple Data Sets

An OUTPUT statement can be used in a conditional statement to write the current observation to a specific data set that is listed in the DATA statement.

```
data EmpsAUC EmpsOnly PhoneOnly;
   merge EmpsAU(in=Emps) PhoneC(in=Cell);
   by EmpID;
   if Emps=1 and Cell=1
        then output EmpsAUC;
   else if Emps=1 and Cell=0
        then output EmpsOnly;
   else if Emps=0 and Cell=1
        then output PhoneOnly;
run;
```

p110d07

Outputting to Multiple Data Sets

EmpsAUC

First	Gender	EmpID	Phone
Togar	M	121150	+61(2)5555-1795
Birin	M	121152	+61(2)5555-1667

EmpsOnly

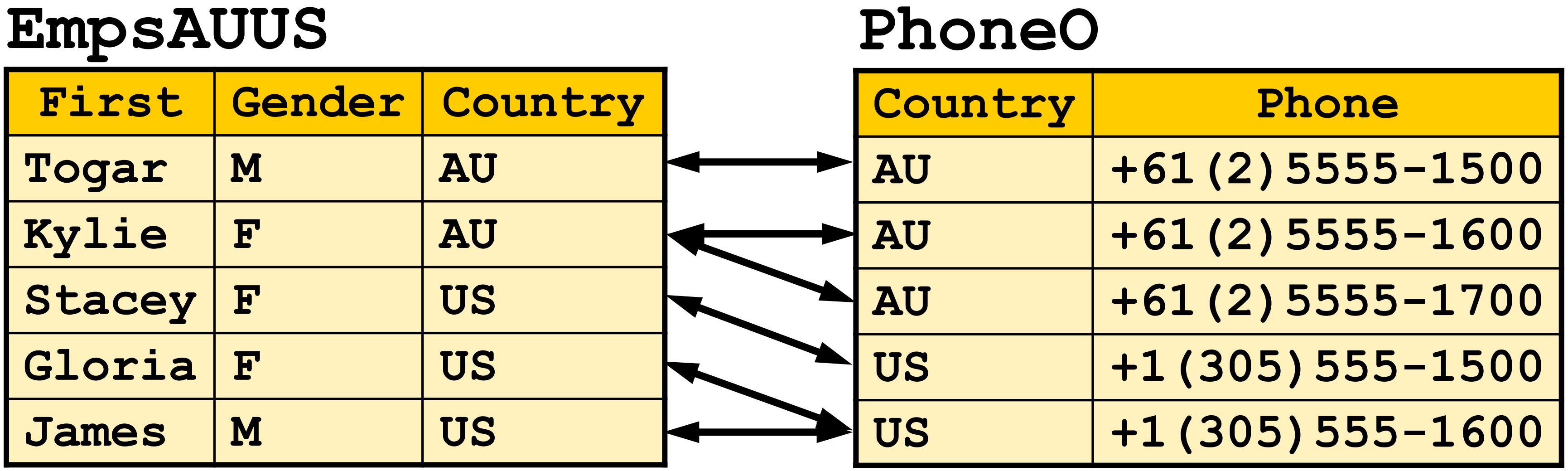
First	Gender	EmpID	Phone
Kylie	F	121151	

PhoneOnly

First	Gender	EmpID	Phone
		121153	+61(2)5555-1348

Many-to-Many Merge

Merge **EmpsAUUS** and **PhoneO** by **Country** to create a new data set named **EmpsOfc**.



EmpsAUUS

First	Gender	Country
Togar	M	AU
Kylie	F	AU
Stacey	F	US
Gloria	F	US
James	M	US

PhoneO

Country	Phone
AU	+61(2)5555-1500
AU	+61(2)5555-1600
AU	+61(2)5555-1700
US	+1(305)555-1500
US	+1(305)555-1600

```
data EmpsOfc;
   merge EmpsAUUS PhoneO;
   by Country;
run;
```

The data sets are sorted by **Country**.

p110d08

Many-to-Many Merge

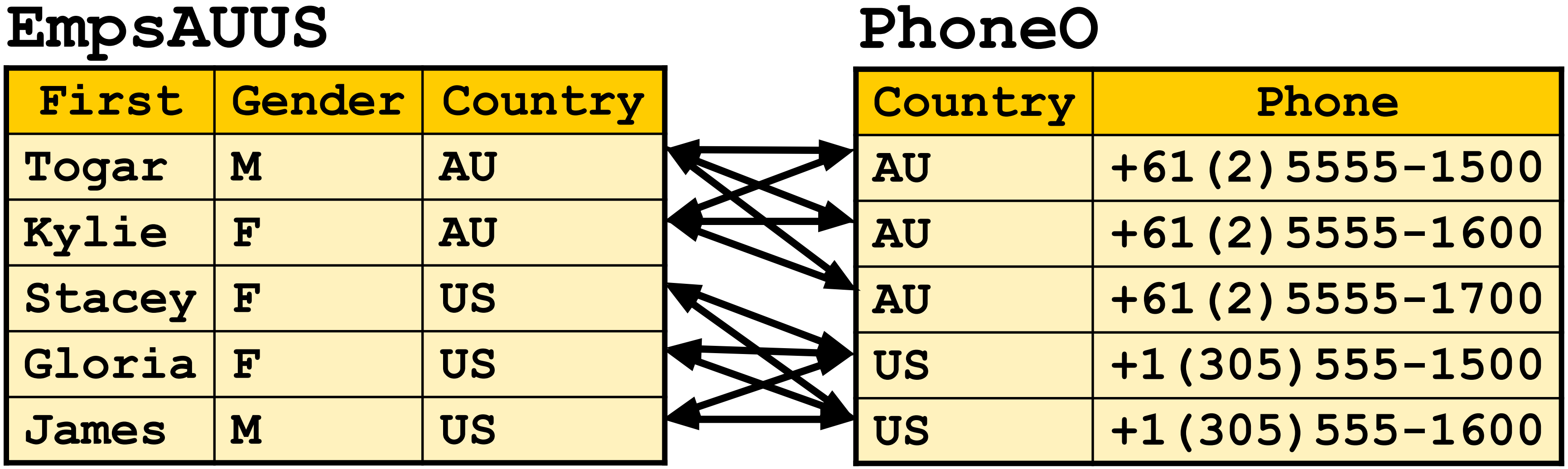
DATA Step Results:

EmpsOfc

First	Gender	Country	Phone
Togar	M	AU	+61(2)5555-1500
Kylie	F	AU	+61(2)5555-1600
Kylie	F	AU	+61(2)5555-1700
Stacey	F	US	+1(305)555-1500
Gloria	F	US	+1(305)555-1600
James	M	US	+1(305)555-1600

Many-to-Many Merge

The SQL procedure creates different results than the DATA step for a many-to-many merge.



EmpsAUUS

First	Gender	Country
Togar	M	AU
Kylie	F	AU
Stacey	F	US
Gloria	F	US
James	M	US

PhoneO

Country	Phone
AU	+61(2)5555-1500
AU	+61(2)5555-1600
AU	+61(2)5555-1700
US	+1(305)555-1500
US	+1(305)555-1600

```
proc sql;
   create table EmpsOfc as
   select First, Gender, PhoneO.Country, Phone
   from EmpsAUUS, PhoneO
   where EmpsAUUS.Country=PhoneO.Country;
```

p110d08

Many-to-Many Merge

PROC SQL Results:

EmpsOfc

First	Gender	Country	Phone
Togar	M	AU	+61(2)5555-1500
Togar	M	AU	+61(2)5555-1600
Togar	M	AU	+61(2)5555-1700
Kylie	F	AU	+61(2)5555-1500
Kylie	F	AU	+61(2)5555-1600
Kylie	F	AU	+61(2)5555-1700
Stacey	F	US	+1(305)555-1500
Stacey	F	US	+1(305)555-1600
Gloria	F	US	+1(305)555-1500
Gloria	F	US	+1(305)555-1600
James	M	US	+1(305)555-1500
James	M	US	+1(305)555-1600

P2-Chapter 9: Combining SAS Data Sets

9.1 Using Data Manipulation Techniques with Match-Merging

Objectives

- Show techniques to perform a match-merge for these special cases:
  - three or more SAS data sets that lack a single common variable
  - variable names that need to be altered to obtain the correct merge results.

Special Cases

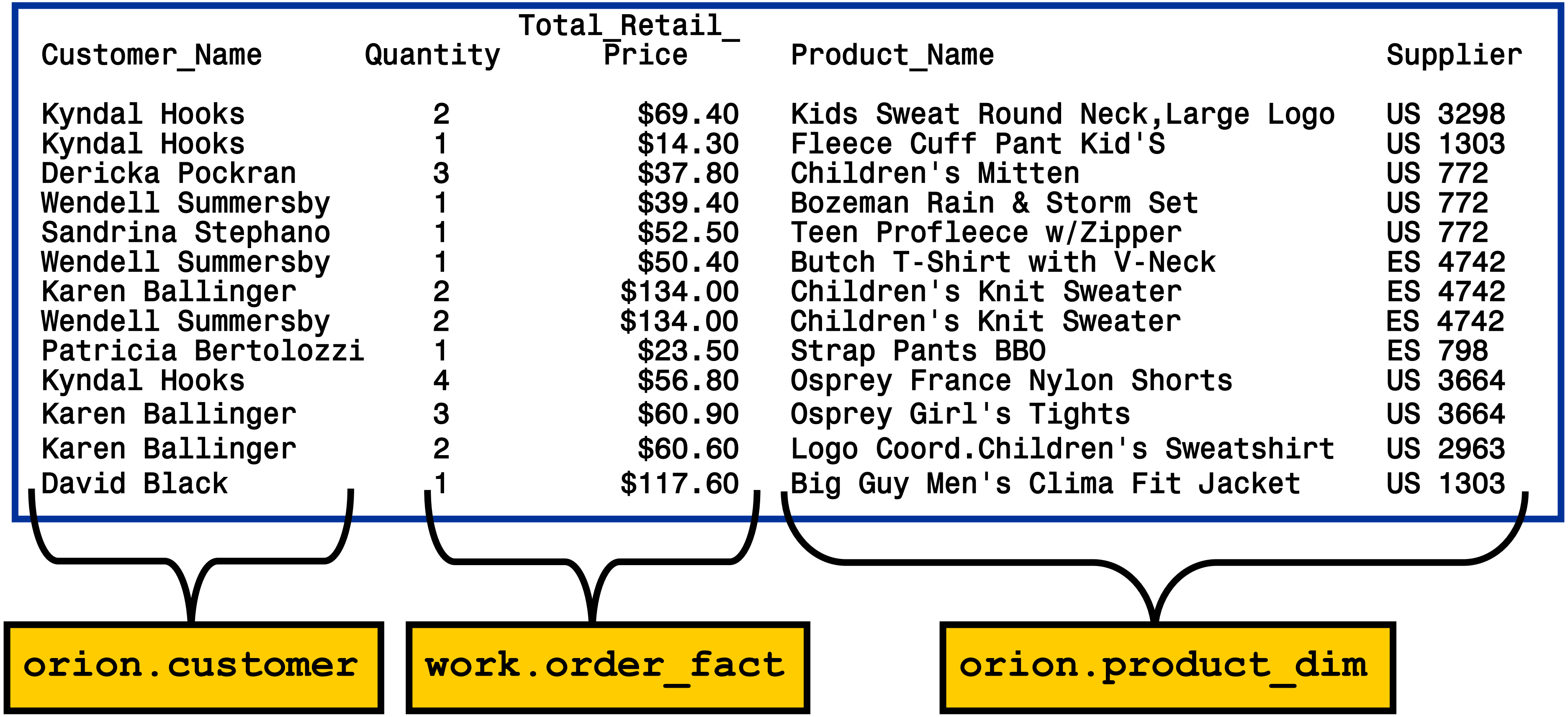
Show techniques to perform a match-merge for these special cases:

- Three or more SAS data sets that lack a common variable
- Variable names that need to be altered to get the correct merge results

Multiple Data Sets without a Common Variable

The following report needs to be created using data from three data sets.

Partial PROC PRINT Output



```
                                      Total_Retail_
Customer_Name         Quantity            Price       Product_Name                        Supplier

Kyndal Hooks              2              $69.40       Kids Sweat Round Neck,Large Logo    US 3298
Kyndal Hooks              1              $14.30       Fleece Cuff Pant Kid'S              US 1303
Dericka Pockran           3              $37.80       Children's Mitten                   US 772
Wendell Summersby         1              $39.40       Bozeman Rain & Storm Set            US 772
Sandrina Stephano         1              $52.50       Teen Profleece w/Zipper             US 772
Wendell Summersby         1              $50.40       Butch T-Shirt with V-Neck           ES 4742
Karen Ballinger           2             $134.00       Children's Knit Sweater             ES 4742
Wendell Summersby         2             $134.00       Children's Knit Sweater             ES 4742
Patricia Bertolozzi       1              $23.50       Strap Pants BBO                     ES 798
Kyndal Hooks              4              $56.80       Osprey France Nylon Shorts          US 3664
Karen Ballinger           3              $60.90       Osprey Girl's Tights                US 3664
Karen Ballinger           2              $60.60       Logo Coord.Children's Sweatshirt    US 2963
David Black               1             $117.60       Big Guy Men's Clima Fit Jacket      US 1303
```

Poll
Quiz

9.04 Quiz

Any number of data sets can be merged in a single DATA step. However, the data sets must have a common variable and be sorted by that variable.

What is the common variable in the following data sets?

orion.customer
Customer_ID
Country
Gender
Personal_ID
Customer_Name
Customer_FirstName
Customer_LastName
Birth_Date
Customer_Address
...

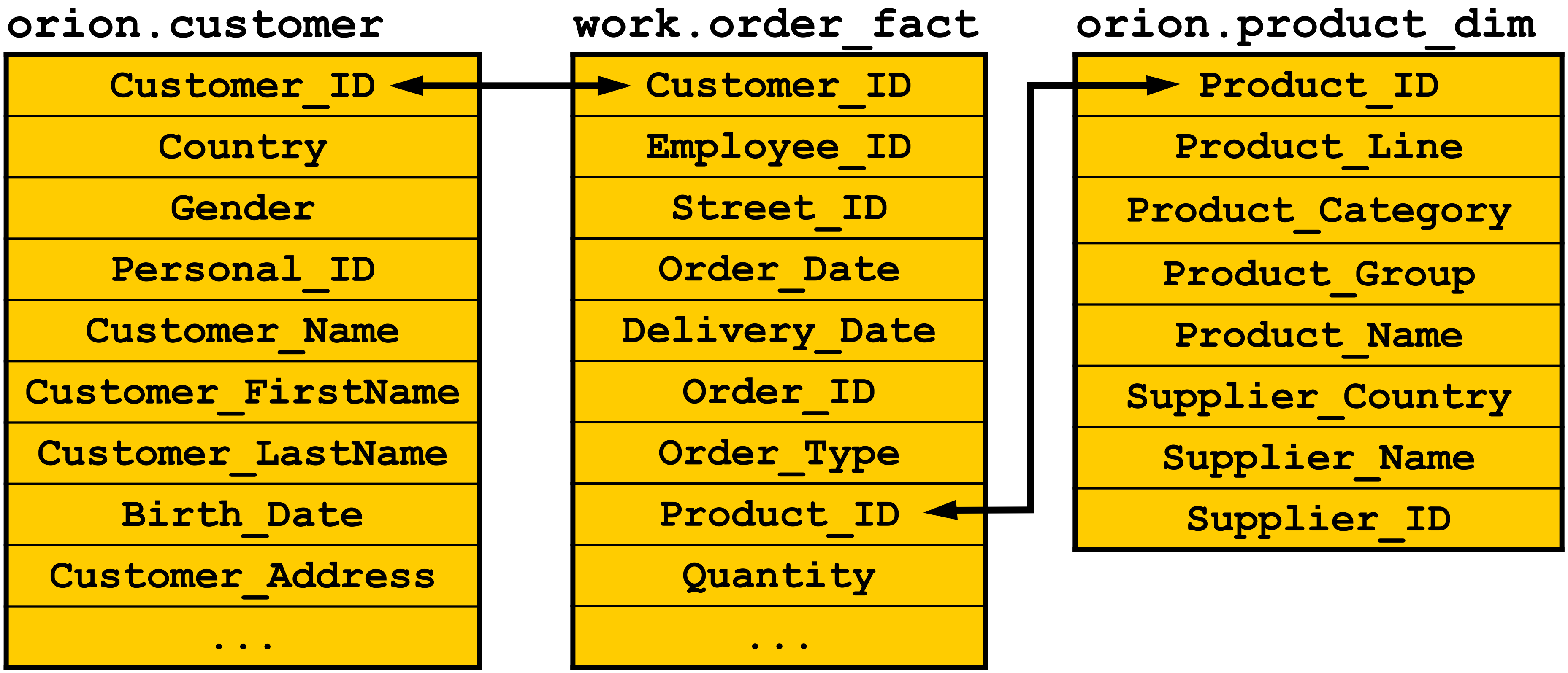
work.order_fact
Customer_ID
Employee_ID
Street_ID
Order_Date
Delivery_Date
Order_ID
Order_Type
Product_ID
Quantity
...

orion.product_dim
Product_ID
Product_Line
Product_Category
Product_Group
Product_Name
Supplier_Country
Supplier_Name
Supplier_ID

9.04 Quiz – Correct Answer

What is the common variable in the following data sets?

None. These data sets do not share one common variable. Therefore, they cannot be combined in a single DATA step.



orion.customer

Customer_ID
Country
Gender
Personal_ID
Customer_Name
Customer_FirstName
Customer_LastName
Birth_Date
Customer_Address
...

work.order_fact

Customer_ID
Employee_ID
Street_ID
Order_Date
Delivery_Date
Order_ID
Order_Type
Product_ID
Quantity
...

orion.product_dim

Product_ID
Product_Line
Product_Category
Product_Group
Product_Name
Supplier_Country
Supplier_Name
Supplier_ID

Match-Merge without a Common Variable

If data sets do not share a common variable, combine them by using a series of merges in separate DATA steps. As usual, the data sets must be sorted by the appropriate BY variable.

Step 1: Merge **`orion.customer`** and **`work.order_fact`** by **`Customer_ID`**.

Step 2: Merge the results of Step1 and **`orion.product_dim`** by **`Product_ID`**.

Without a Common Variable – Step 1

Merge **orion.customer** and **work.order_fact** by **Customer_ID**.

```
proc sort data=orion.order_fact
          out=work.order_fact;
   by Customer_ID;
   where year(Order_Date)=2007;
run;

data CustOrd;
   merge orion.customer(in=cust)
         work.order_fact(in=order);
   by Customer_ID;
   if cust=1 and order=1;
   keep Customer_ID Customer_Name Quantity
        Total_Retail_Price Product_ID;
run;
```

orion.customer is in order by **Customer_ID**

p209d03

Without a Common Variable – Step 2

Merge the results of Step 1, **CustOrd**, with **orion.product_dim** by **Product_ID**.

```
proc sort data=CustOrd;
   by Product_ID;
run;

data CustOrdProd;
   merge CustOrd(in=ord)
         orion.product_dim(in=prod);
   by Product_ID;
   if ord=1 and prod=1;
   Supplier=catx(' ',Supplier_Country,Supplier_ID);
   keep Customer_Name Quantity
        Total_Retail_Price Product_Name Supplier;
run;
```

Product_dim is in order by **Product_ID**

p209d03

Altering Variable Names

With match-merging, two situations might require altering variable names:

- The BY variables have different names in the input data sets being merged.
- The data sets being merged have identically named variables that must both be kept in the merged output.

In both cases, the RENAME= data set option can be used to alter the variable names to get the desired results.

Business Scenario – Create Gift List

The Excel workbook **BonusGift.xls** contains a list of suppliers that want to send gifts to customers who purchased more than a specified minimum quantity of a product.

Use **work.CustOrdProd** and **BonusGift.xls** to determine the customers that will be sent gifts.

work.CustOrdProd

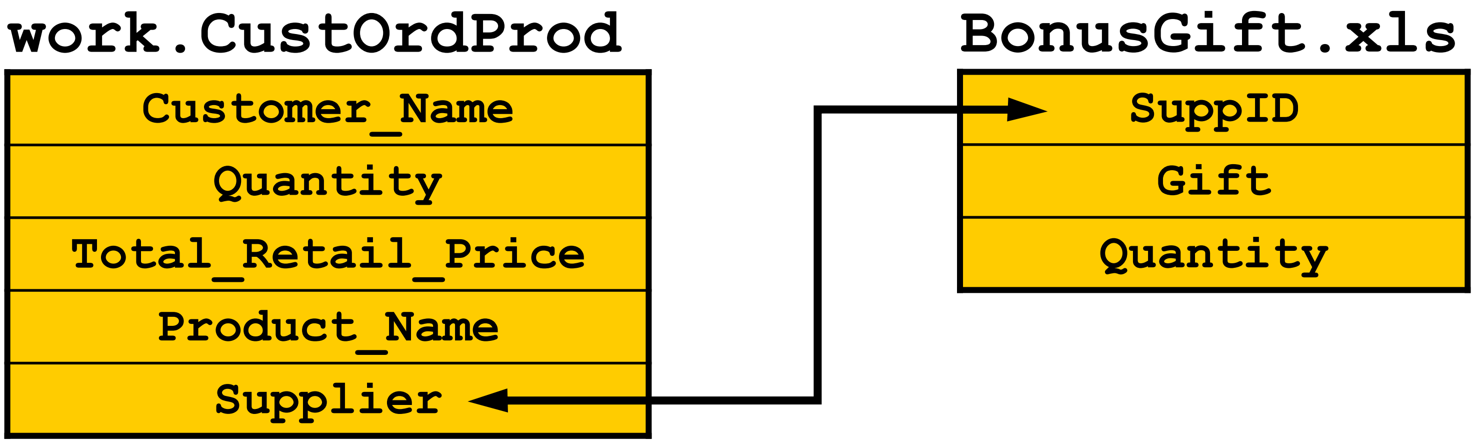
Customer_Name
Quantity
Total_Retail_Price
Product_Name
Supplier

BonusGift.xls

SuppID
Gift
Quantity

Business Scenario – Details

The data sets **work.CustOrdProd** and **BonusGift.xls** must be merged on values that are in two differently named variables.



work.CustOrdProd

Customer_Name

Quantity

Total_Retail_Price

Product_Name

Supplier

BonusGift.xls

SuppID

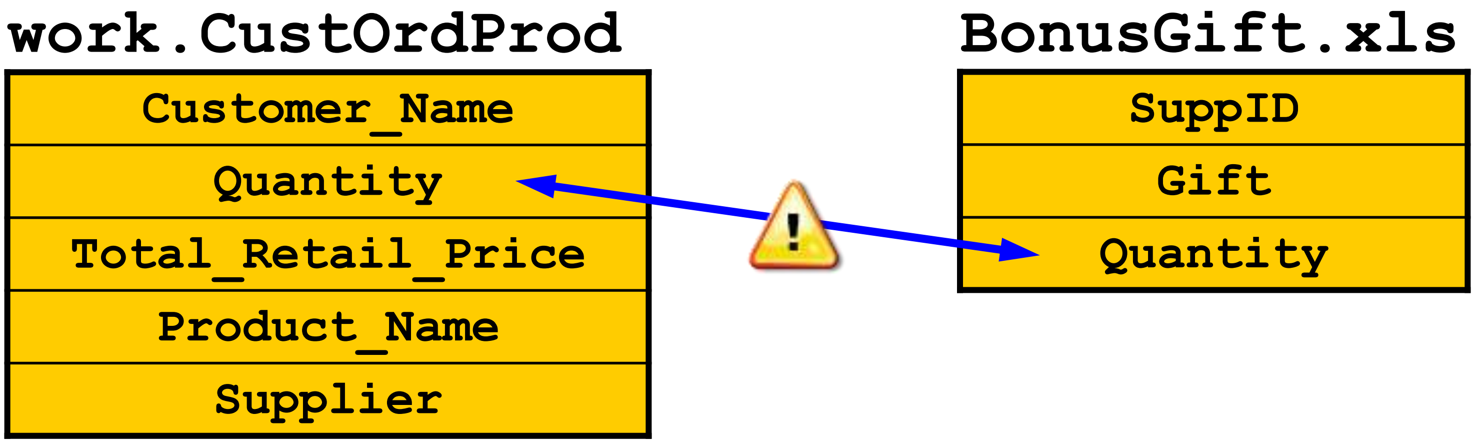
Gift

Quantity

The variables must have the same name for the match-merge to work correctly.

Business Scenario – Details

You want to keep merged observations where the value of **`Quantity`** in **`work.CustOrdProd`** is more than the value of **`Quantity`** in **`BonusGift.xls`**.



`work.CustOrdProd`

`Customer_Name`
`Quantity`
`Total_Retail_Price`
`Product_Name`
`Supplier`

`BonusGift.xls`

`SuppID`
`Gift`
`Quantity`

The variables must have different names so that you can use a subsetting IF statement to compare them.

Create Gift List – Solution

Access the Excel workbook, specify the worksheet to use, and release the workbook after the DATA step.

```
libname bonus pcfiles path='C:\SAS Sample
Data\prg2\BonusGift.xls'; *Rev. for 64 bit;

data CustOrdProdGift;
   merge CustOrdProd(in=c)
         bonus.'Supplier$'n(in=s
            rename=(SuppID=Supplier
                    Quantity=Minimum));
   by Supplier;
   if c=1 and s=1 and Quantity > Minimum;
run;

libname bonus clear;
```

p209d04

Create Gift List – Solution

Use the RENAME= data set option to ensure that the BY variable has the same name to use for merging.

```
libname bonus pcfiles path='C:\SAS Sample
Data\prg2\BonusGift.xls';

data CustOrdProdGift;
   merge CustOrdProd(in=c)
         bonus.'Supplier$'n(in=s
            rename=(SuppID=Supplier
                    Quantity=Minimum));
   by Supplier;
   if c=1 and s=1 and Quantity > Minimum;
run;

libname bonus clear;
```

p209d04

Create Gift List – Solution

Change the name of the **`Quantity`** variable from the Excel worksheet so that it can be use in a subsetting IF.

```
libname bonus pcfiles path='C:\SAS Sample
Data\prg2\BonusGift.xls';

data CustOrdProdGift;
   merge CustOrdProd(in=c)
         bonus.'Supplier$'n(in=s
            rename=(SuppID=Supplier
                    Quantity=Minimum));
   by Supplier;
   if c=1 and s=1 and Quantity > Minimum;
run;

libna
```

`Quantity` value from the `CustOrdProd` data set

Renamed `Quantity` value from the `'Supplier$'n` data set

p209d04

Create Gift List – Solution

Use the IN= option and a condition in the subsetting IF statement to keep only the matches.

```
libname bonus pcfiles path='C:\SAS Sample
Data\prg2\BonusGift.xls';

data CustOrdProdGift;
   merge CustOrdProd(in=c)
         bonus.'Supplier$'n(in=s
            rename=(SuppID=Supplier
                    Quantity=Minimum));
   by Supplier;
   if c=1 and s=1 and Quantity > Minimum;
run;

libname bonus clear;
```

p209d04

Create Gift List – Solution

Fifty-two gifts will be sent to customers.

Partial SAS log

```
207  libname bonus 'BonusGift.xls';
NOTE: Libref BONUS was successfully assigned as follows:
      Engine:        EXCEL
      Physical Name: BonusGift.xls
208
209  data CustOrdProdGift;
210     merge CustOrdProd(in=c)
211           bonus.'Supplier$'n(in=s
212              rename=(SuppID=Supplier
213                      Quantity=Minimum));
214     by Supplier;
215     if c=1 and s=1 and Quantity > Minimum;
216  run;

NOTE: There were 148 observations read from the data set WORK.CUSTORDPROD.
NOTE: There were 18 observations read from the data set BONUS.'Supplier$'n.
NOTE: The data set WORK.CUSTORDPRODGIFT has 52 observations and 7 variables.
NOTE: DATA statement used (Total process time):
      real time           0.04 seconds
      cpu time            0.03 seconds
```

Create Gift List – Output

Sort the data set by customer name prior to printing the list of customers and the gifts that they should receive.

```
proc sort data=CustOrdProdGift;
   by Customer_Name;
run;

proc print data=CustOrdProdGift;
   var Customer_Name Gift;
run;
```

p209d04

Create Gift List – Output

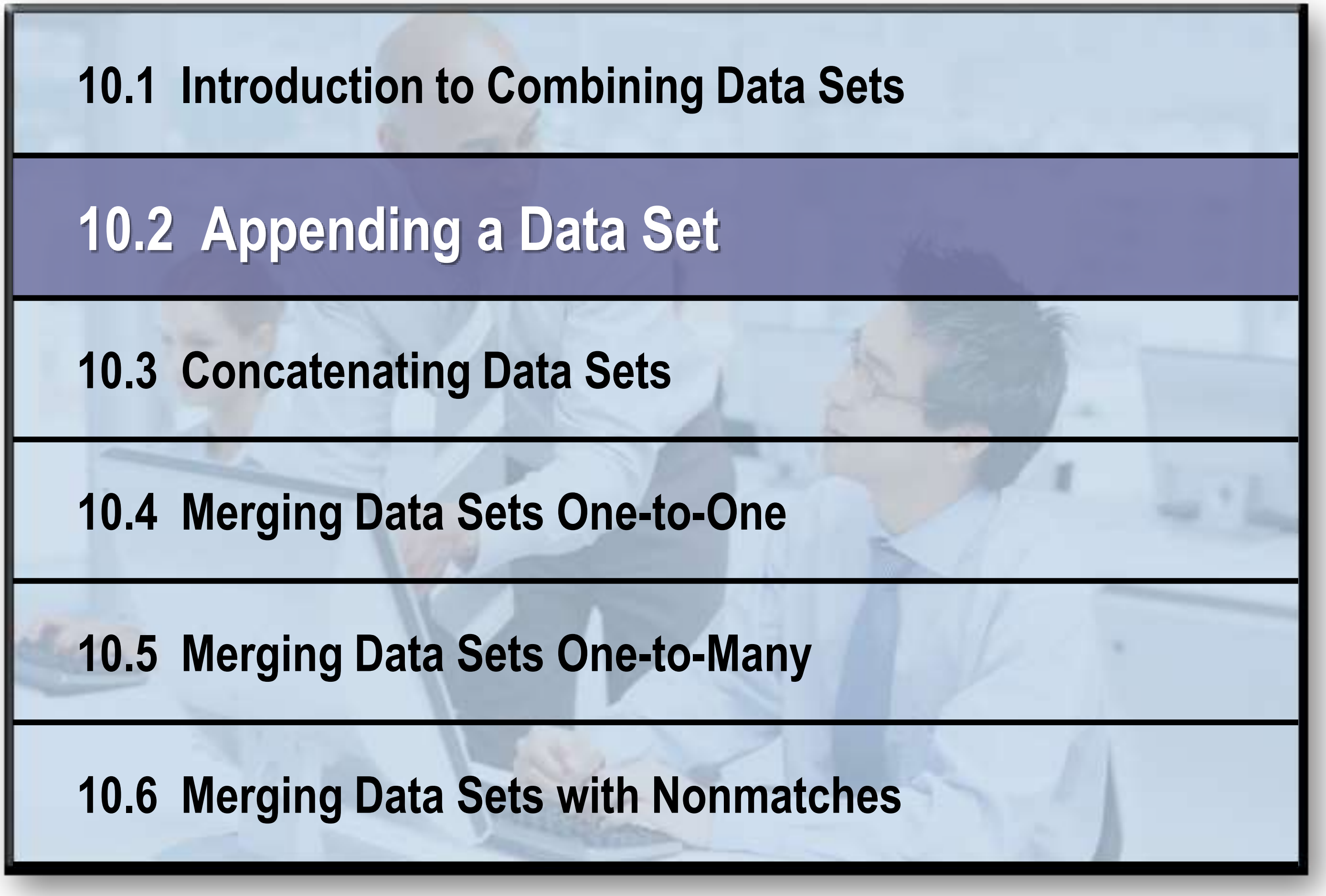
The output below shows the list of customers and gifts.

Partial PROC PRINT output

```
     Customer_Name               Gift

     Alvan Goheen                Travel Mug
     Angel Borwick               Belt Pouch
     Cynthia Martinez            Travel Set
     Cynthia Martinez            Gift Card
     Cynthia Martinez            Travel Mug
     Cynthia Mccluney            Tote Bag
     Cynthia Mccluney            Tote Bag
     Cynthia Mccluney            Gift Card
     David Black                 Backpack
     Dericka Pockran             Coupon
     Dericka Pockran             Travel Mug
     Dericka Pockran             Travel Mug
```

p209d04

Chapter 10: Combining SAS Data Sets



10.1 Introduction to Combining Data Sets

10.2 Appending a Data Set

10.3 Concatenating Data Sets

10.4 Merging Data Sets One-to-One

10.5 Merging Data Sets One-to-Many

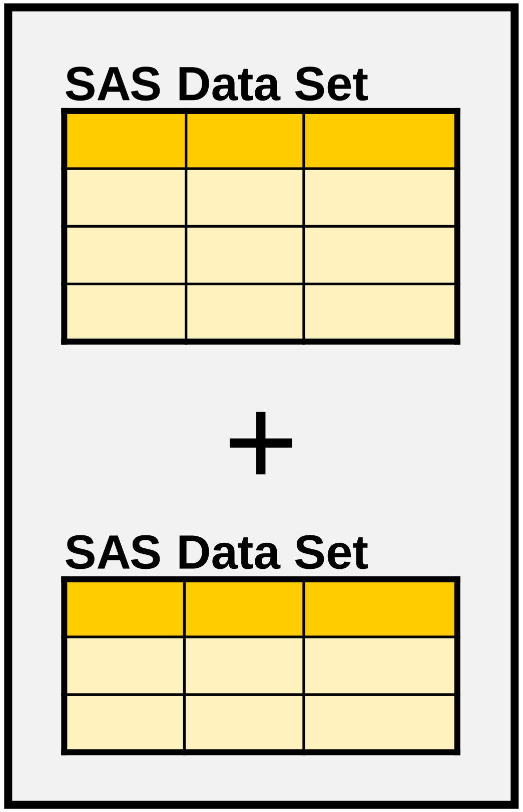
10.6 Merging Data Sets with Nonmatches

Objectives

- Append one SAS data set to another SAS data set by using the APPEND procedure.
- Append a SAS data set containing additional variables to another SAS data set by using the FORCE option with the APPEND procedure.

Appending and Concatenating

Appending and concatenating involves combining SAS data sets, one after the other, into a single SAS data set.



- Appending adds the observations in the second data set directly to the end of the original data set.
- Concatenating copies all observations from the first data set and then copies all observations from one or more successive data sets into a new data set.

The APPEND Procedure

The *APPEND procedure* adds the observations from one SAS data set to the end of another SAS data set.

General form of the APPEND procedure:

```
PROC APPEND  BASE = SAS-data-set
             DATA = SAS-data-set;
RUN;
```

BASE= names the data set to which observations are added.

DATA= names the data set containing observations that are added to the base data set.

The APPEND Procedure

Requirements:

- Only two data sets can be used at a time in one step.
- The observations in the base data set are not read.
- The variable information in the descriptor portion of the base data set cannot change.

Business Scenario

Emps is a master data set that contains employees hired in 2006 and 2007.

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007

Business Scenario

Emps is a master data set that contains employees hired in 2006 and 2007.

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007

The employees hired in 2008, 2009, and 2010 need to be appended.

Emps2008

First	Gender	HireYear
Brett	M	2008
Renee	F	2008

Emps2009

First	HireYear
Sara	2009
Dennis	2009

Emps2010

First	HireYear	Country
Rose	2010	Spain
Eric	2010	Spain

Poll
Quiz

10.02 Quiz

How many observations will be in **Emps** after appending the three data sets?

Emps2008

First	Gender	HireYear
Brett	M	2008
Renee	F	2008

Emps2009

First	HireYear
Sara	2009
Dennis	2009

Emps2010

First	HireYear	Country
Rose	2010	Spain
Eric	2010	Spain

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007

10.02 Quiz – Correct Answer

How many observations will be in **Emps** after appending the three data sets?

9 observations

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007

Emps2008

First	Gender	HireYear
Brett	M	2008
Renee	F	2008

Emps2009

First	HireYear
Sara	2009
Dennis	2009

Emps2010

First	HireYear	Country
Rose	2010	Spain
Eric	2010	Spain

Poll
Quiz

10.03 Quiz

How many variables will be in **Emps** after appending the three data sets?

Emps2008

First	Gender	HireYear
Brett	M	2008
Renee	F	2008

Emps2009

First	HireYear
Sara	2009
Dennis	2009

Emps2010

First	HireYear	Country
Rose	2010	Spain
Eric	2010	Spain

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007

10.03 Quiz – Correct Answer

How many variables will be in **Emps** after appending the three data sets?

3 variables

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007

The base data set variable information cannot change.

Emps2008

First	Gender	HireYear
Brett	M	2008
Renee	F	2008

Emps2009

First	HireYear
Sara	2009
Dennis	2009

Emps2010

First	HireYear	Country
Rose	2010	Spain
Eric	2010	Spain

Like-Structured Data Sets

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007

Emps2008

First	Gender	HireYear
Brett	M	2008
Renee	F	2008

The data sets contain the same variables.

```
proc append base=Emps
            data=Emps2008;
run;
```

p110d01

Like-Structured Data Sets

```
84   proc append base=Emps
85               data=Emps2008;
86   run;

NOTE: Appending WORK.EMPS2008 to WORK.EMPS.
NOTE: There were 2 observations read from the data set
      WORK.EMPS2008.
NOTE: 2 observations added.
NOTE: The data set WORK.EMPS has 5 observations and 3 variables.
```

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007
Brett	M	2008
Renee	F	2008

Unlike-Structured Data Sets

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007
Brett	M	2008
Renee	F	2008

Emps2009

First	HireYear
Sara	2009
Dennis	2009

The BASE= data set has a variable that is not in the DATA= data set.

```
proc append base=Emps
            data=Emps2009;
run;
```

p110d01

Unlike-Structured Data Sets

```
90    proc append base=Emps
91                data=Emps2009;
92    run;

NOTE: Appending WORK.EMPS2009 to WORK.EMPS.
WARNING: Variable Gender was not found on DATA file.
NOTE: There were 2 observations read from the data set
      WORK.EMPS2009.
NOTE: 2 observations added.
NOTE: The data set WORK.EMPS has 7 observations and 3 variables.
```

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007
Brett	M	2008
Renee	F	2008
Sara		2009
Dennis		2009

Unlike-Structured Data Sets

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007
Brett	M	2008
Renee	F	2008
Sara		2009
Dennis		2009

Emps2010

First	HireYear	Country
Rose	2010	Spain
Eric	2010	Spain

The DATA= data set has a variable that is not in the BASE= data set.

```
proc append base=Emps
            data=Emps2010;
run;
```

p110d01

Unlike-Structured Data Sets

```
96   proc append base=Emps
97               data=Emps2010;
98   run;

NOTE: Appending WORK.EMPS2010 to WORK.EMPS.
WARNING: Variable Country was not found on BASE file. The
         variable will not be added to the BASE file.
WARNING: Variable Gender was not found on DATA file.
ERROR: No appending done because of anomalies listed above.
       Use FORCE option to append these files.
NOTE: 0 observations added.
NOTE: The data set WORK.EMPS has 7 observations and 3 variables.
NOTE: Statements not processed because of errors noted above.

NOTE: The SAS System stopped processing this step because of
      errors.
```

Unlike-Structured Data Sets

The *FORCE option* forces the observations to be appended when the DATA= data set contains variables that are not in the BASE= data set.

General form of the FORCE option:

PROC APPEND BASE = *SAS-data-set*
DATA = *SAS-data-set* FORCE;
RUN;

The FORCE option causes the extra variables to be dropped and issues a warning message.

```
proc append base=Emps
            data=Emps2010 force;
run;
```

p110d01

Unlike-Structured Data Sets

```
100  proc append base=Emps
101              data=Emps2010 force;
102  run;

NOTE: Appending WORK.EMPS2010 to WORK.EMPS.
WARNING: Variable Country was not found on BASE file. The
         variable will not be added to the BASE file.
WARNING: Variable Gender was not found on DATA file.
NOTE: FORCE is specified, so dropping/truncating will occur.
NOTE: There were 2 observations read from the data set
      WORK.EMPS2010.
NOTE: 2 observations added.
NOTE: The data set WORK.EMPS has 9 observations and 3 variables.
```

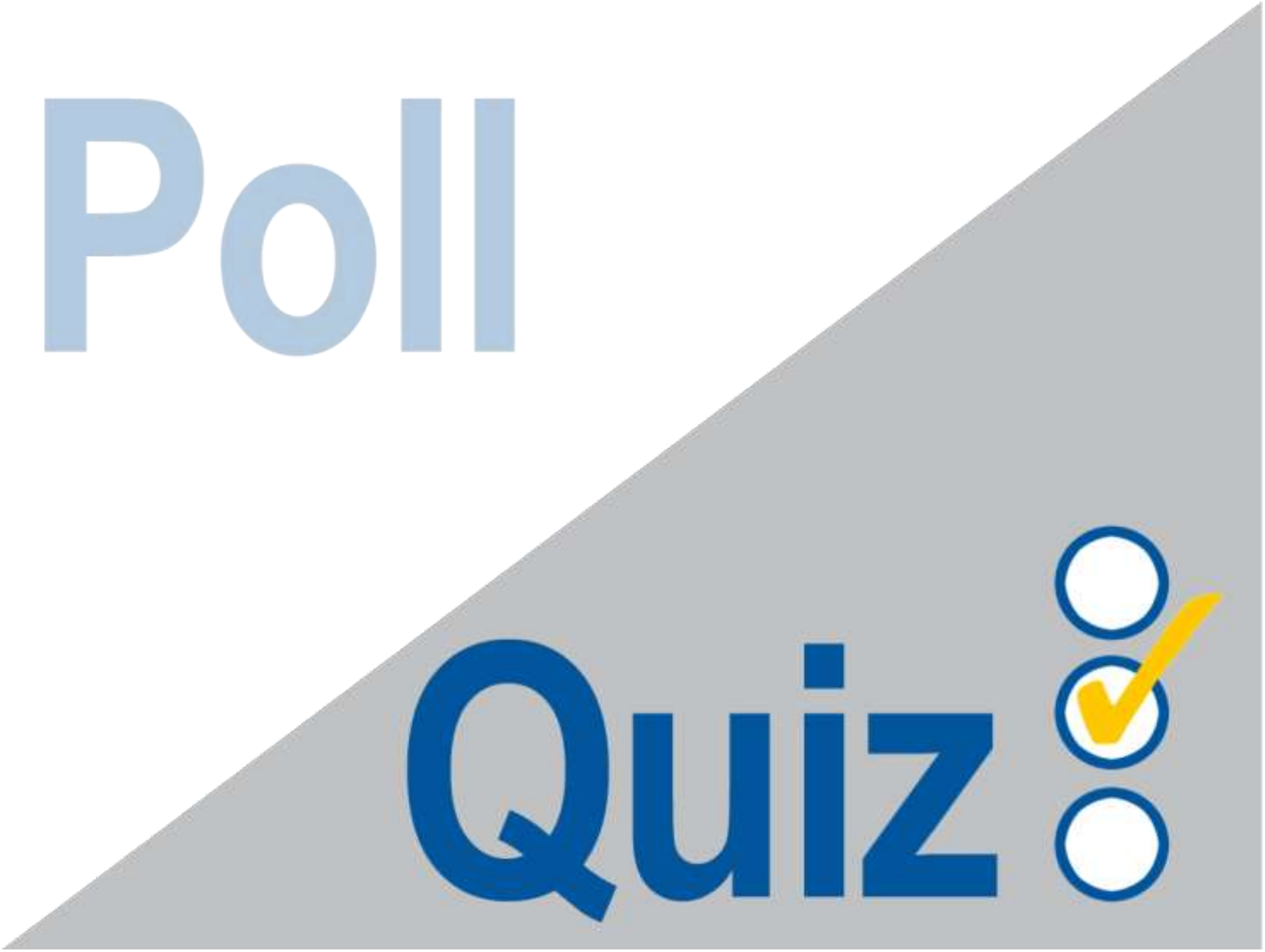
Unlike-Structured Data Sets

Emps

First	Gender	HireYear
Stacey	F	2006
Gloria	F	2007
James	M	2007
Brett	M	2008
Renee	F	2008
Sara		2009
Dennis		2009
Rose		2010
Eric		2010

Unlike-Structured Data Sets

Situation	Action
BASE= data set contains a variable that is not in the DATA= data set.	The observations are appended, but the observations from the DATA= data set have a missing value for the variable that was not present in the DATA= data set. The FORCE option is not necessary in this case.
DATA= data set contains a variable that is not in the BASE= data set.	Use the FORCE option in the PROC APPEND statement to force the concatenation of the two data sets. The statement drops the extra variable and issues a warning message.


Poll
Quiz

10.04 Quiz

How many observations will be in **Emps** if the program is submitted a second time?

Submitting this program once appends six observations to the **Emps** data set, which results in a total of nine observations.

```
proc append base=Emps
            data=Emps2008;
run;
proc append base=Emps
            data=Emps2009;
run;
proc append base=Emps
            data=Emps2010 force;
run;
```

3 obs + 2 obs = 5 obs

5 obs + 2 obs = 7 obs

7 obs + 2 obs = 9 obs

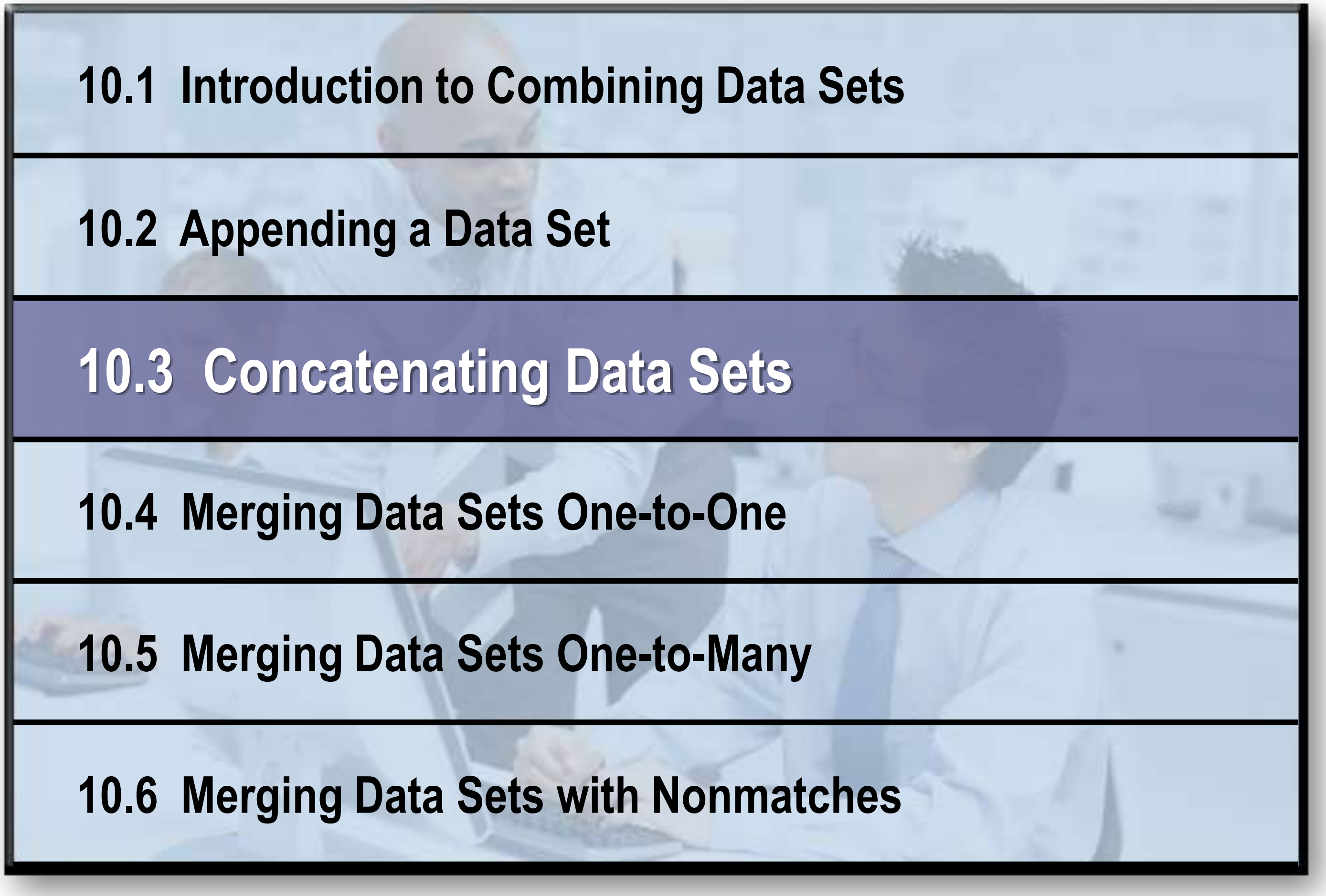
10.04 Quiz – Correct Answer

How many observations will be in **`Emps`** if the program is submitted a second time?

15 observations (9 + 2 + 2 + 2)

Be careful; observations are added to the BASE= data set every time that you submit the program.

Chapter 10: Combining SAS Data Sets



10.1 Introduction to Combining Data Sets

10.2 Appending a Data Set

10.3 Concatenating Data Sets

10.4 Merging Data Sets One-to-One

10.5 Merging Data Sets One-to-Many

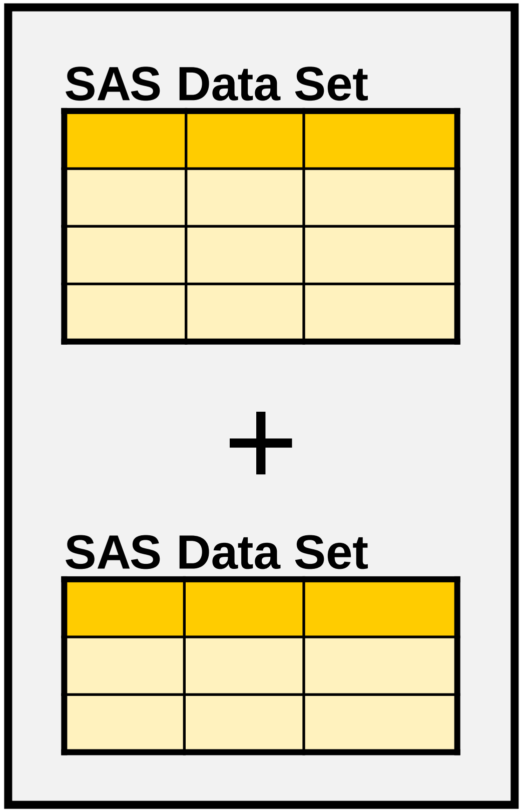
10.6 Merging Data Sets with Nonmatches

Objectives

- Concatenate two or more SAS data sets by using the SET statement in a DATA step.
- Change the names of variables by using the RENAME= data set option.
- Compare the APPEND procedure to the SET statement.
- Interleave two or more SAS data sets by using the SET and BY statements in a DATA step.

Appending and Concatenating

Appending and concatenating involves combining SAS data sets, one after the other, into a single SAS data set.



- Appending adds the observations in the second data set directly to the end of the original data set.

➡ Concatenating copies all observations from the first data set and then copies all observations from one or more successive data sets into a new data set.

The SET Statement

The *SET statement* in a DATA step reads observations from one or more SAS data sets.

```
DATA SAS-data-set;
     SET SAS-data-set1 SAS-data-set2 . . .;
     <additional SAS statements>
RUN;
```

- Any number of data sets can be in the SET statement.
- The observations from the first data set in the SET statement appear first in the new data set. The observations from the second data set follow those from the first data set, and so on.

Like-Structured Data Sets

Concatenate **EmpsDK** and **EmpsFR** to create a new data set named **EmpsAll1**.

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

The data sets contain the same variables.

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

p110d02

Compilation

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
    set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country

EmpsAll1

First	Gender	Country

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsD
run;
```

Initialize PDV

PDV

First	Gender	Country

EmpsAll1

First	Gender	Country

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Lars	M	Denmark

EmpsAll1

First	Gender	Country

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	Country
Lars	M	Denmark

First	Gender	Country
Lars	M	Denmark

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Kari	F	Denmark

EmpsAll1

First	Gender	Country
Lars	M	Denmark

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	Country
Kari	F	Denmark

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Jonas	M	Denmark

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll;
   set EmpsDK EmpsFR;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	Country
Jonas	M	Denmark

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
nas	M	Denmark

EOF

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Jonas	M	Denmark

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
    set EmpsD
run;
```

Reinitialize PDV

PDV

First	Gender	Country

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
    set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Pierre	M	France

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll;
   set EmpsDK EmpsFR;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	Country
Pierre	M	France

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark
Pierre	M	France

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Sophie	F	France

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark
Pierre	M	France

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

First	Gender	Country
Pierre	M	France
Sophie	F	France

```
data EmpsAll1;
   set EmpsDK EmpsFR;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	Country
Sophie	F	France

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark
Pierre	M	France
Sophie	F	France

...

Execution

EmpsDK

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark

EmpsFR

EOF

First	Gender	Country
Pierre	M	France
phie	F	France

```
data EmpsAll1;
    set EmpsDK EmpsFR;
run;
```

PDV

First	Gender	Country
Sophie	F	France

EmpsAll1

First	Gender	Country
Lars	M	Denmark
Kari	F	Denmark
Jonas	M	Denmark
Pierre	M	France
Sophie	F	France

Unlike-Structured Data Sets

Concatenate **EmpsCN** and **EmpsJP** to create a new data set named **EmpsAll2**.

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

The data sets do not contain the same variables.

```
data EmpsAll2;
   set EmpsCN EmpsJP;
run;
```

p110d03

Poll
Quiz

10.05 Quiz

How many variables will be in **EmpsAll2** after concatenating **EmpsCN** and **EmpsJP**?

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

```
data EmpsAll2;
   set EmpsCN EmpsJP;
run;
```

10.05 Quiz – Correct Answer

How many variables will be in **EmpsAll2** after concatenating **EmpsCN** and **EmpsJP**?

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

Four variables

First, Gender, Country, and Region

Compilation

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

```
data EmpsAll2;
   set EmpsCN EmpsJP;
run;
```

PDV

First	Gender	Country

...

Compilation

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

```
data EmpsAll2;
   set EmpsCN EmpsJP;
run;
```

PDV

First	Gender	Country	Region

Final Results

EmpsAll2

First	Gender	Country	Region
Chang	M	China	
Li	M	China	
Ming	F	China	
Cho	F		Japan
Tomi	M		Japan

Compilation Using the RENAME Option

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

```
data EmpsAll2;
   set EmpsCN EmpsJP(rename=(Region=Country));
run;
```

PDV

First	Gender	Country

p110d03

...

Compilation

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

```
data EmpsAll2;
   set EmpsCN EmpsJP(rename=(Region=Country));
run;
```

PDV

First	Gender	Country

Compilation

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

```
data EmpsAll2;
   set EmpsCN EmpsJP(rename=(Region=Country));
run;
```

PDV

First	Gender	Country

Compilation

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

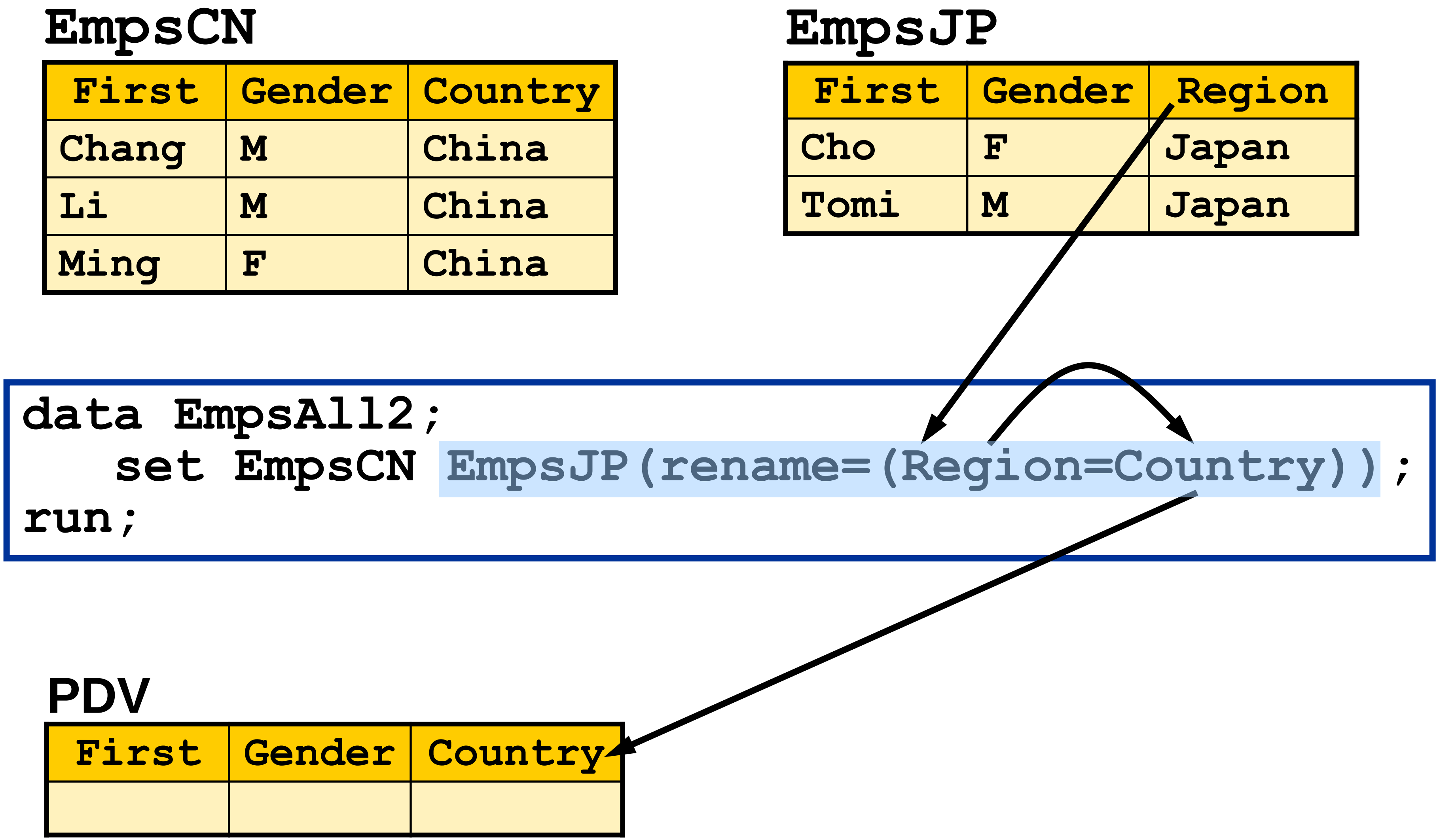
First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

```
data EmpsAll2;
   set EmpsCN EmpsJP(rename=(Region=Country));
run;
```

PDV

First	Gender	Country

Compilation



EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

```
data EmpsAll2;
   set EmpsCN EmpsJP(rename=(Region=Country));
run;
```

PDV

First	Gender	Country

Final Results

EmpsAll2

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China
Cho	F	Japan
Tomi	M	Japan

APPEND Procedure versus SET Statement

- The data set that results from concatenating two data sets with the SET statement is the same data set that results from concatenating them with the APPEND procedure if the two data sets contain the same variables.
- The APPEND procedure concatenates much faster than the SET statement because the APPEND procedure does not process the observations from the BASE= data set.
- The two methods are significantly different when the variables differ between data sets.

APPEND Procedure versus SET Statement

Criterion	APPEND Procedure	SET Statement
Number of data sets that you can concatenate	Uses two data sets.	Uses any number of data sets.
Handling of data sets that contain different variables	Uses all variables in the BASE= data set and assigns missing values to observations from the DATA= data set where appropriate; cannot include variables found only in the DATA= data set.	Uses all variables and assigns missing values where appropriate.

Poll
Quiz

10.07 Multiple Choice Poll

Which method would you use if you wanted to create a new variable at the time of concatenation?

a. APPEND procedure

b. SET statement

10.07 Multiple Choice Poll – Correct Answer

Which method would you use if you wanted to create a new variable at the time of concatenation?

a. APPEND procedure

[✓ b.] SET statement

```
data EmpsBonus;
   set EmpsDK EmpsFR;
   if Country='Denmark'
      then Bonus=300;
   else Bonus=500;
run;
```

Interleaving

Interleaving intersperses observations from two or more data sets, based on one or more common variables.

The SET statement with a BY statement in a DATA step interleaves SAS data sets.

```
DATA SAS-data-set;
     SET SAS-data-set1 SAS-data-set2 . . .;
     BY <DESCENDING> by-variable(s);
     <additional SAS statements>
RUN;
```

The data sets must be sorted by the BY *variable*.

Use the SORT procedure to sort the data sets by the BY variable.

Interleaving

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

Which value comes first?

Chang

```
data EmpsAll2;
   set EmpsCN EmpsJP(rename=(Region=Country));
   by First;
run;
```

PDV

First	Gender	Country
Chang	M	China

p110d03

...

Interleaving

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

Which value comes first?

Cho

```
data EmpsAll2;
   set EmpsCN                    (Region=Country));
   by First;
run;
```

Reinitialize PDV

PDV

First	Gender	Country

...

Interleaving

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

Which value comes first?

Cho

```
data EmpsAll2;
   set EmpsCN EmpsJP(rename=(Region=Country));
   by First;
run;
```

PDV

First	Gender	Country
Cho	F	Japan

...

Interleaving

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

Which value comes first?

Li

```
data EmpsAll2;
    set EmpsCN                      (Region=Country));
    by First;
run;
```

Reinitialize PDV

PDV

First	Gender	Country

...

Interleaving

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

Which value comes first?

Li

```
data EmpsAll2;
   set EmpsCN EmpsJP(rename=(Region=Country));
   by First;
run;
```

PDV

First	Gender	Country
Li	M	China

...

Interleaving

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
Ming	F	China

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

Which value comes first?

Ming

```
data EmpsAll2;
   set EmpsCN EmpsJP(rename=(Region=Country));
   by First;
run;
```

PDV

First	Gender	Country
Ming	F	China

...

Interleaving

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
ng	F	China

EOF

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

Which value comes first?

Tomi

```
data EmpsAll2;
   set EmpsCN                (Region=Country));
   by First;
run;
```

Reinitialize PDV

PDV

First	Gender	Country

...

Interleaving

EmpsCN

First	Gender	Country
Chang	M	China
Li	M	China
ng	F	China

EOF

EmpsJP

First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

Which value comes first?

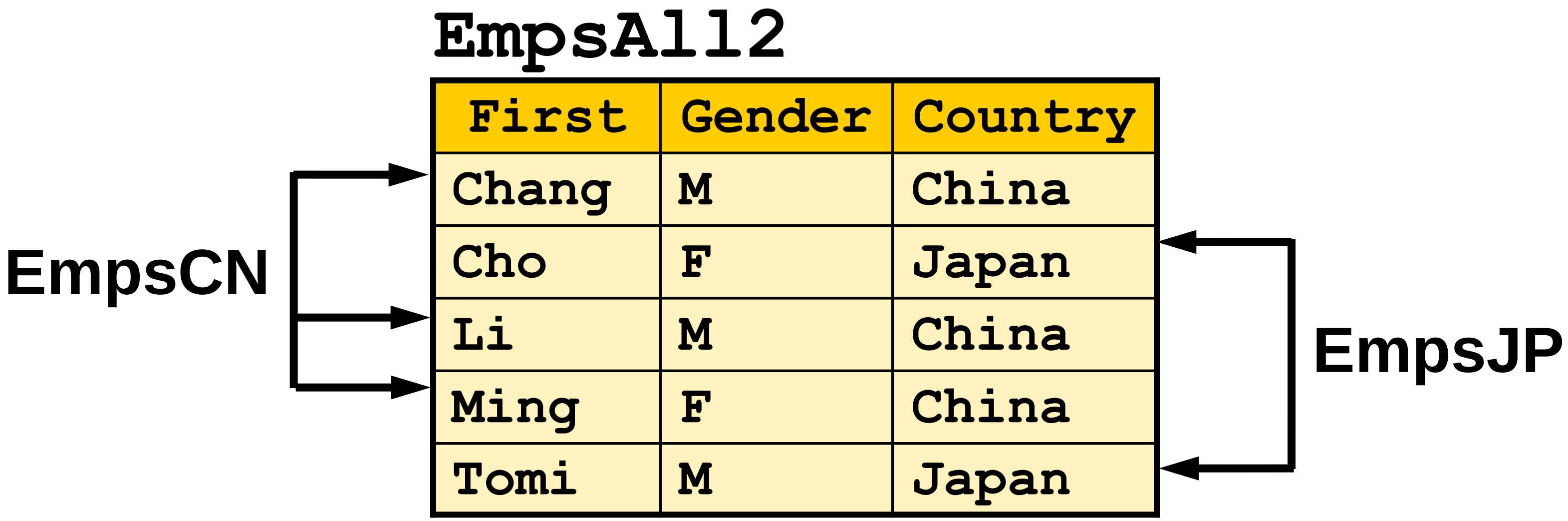
Tomi

```
data EmpsAll2;
   set EmpsCN EmpsJP(rename=(Region=Country));
   by First;
run;
```

PDV

First	Gender	Country
Tomi	M	Japan

Interleaving



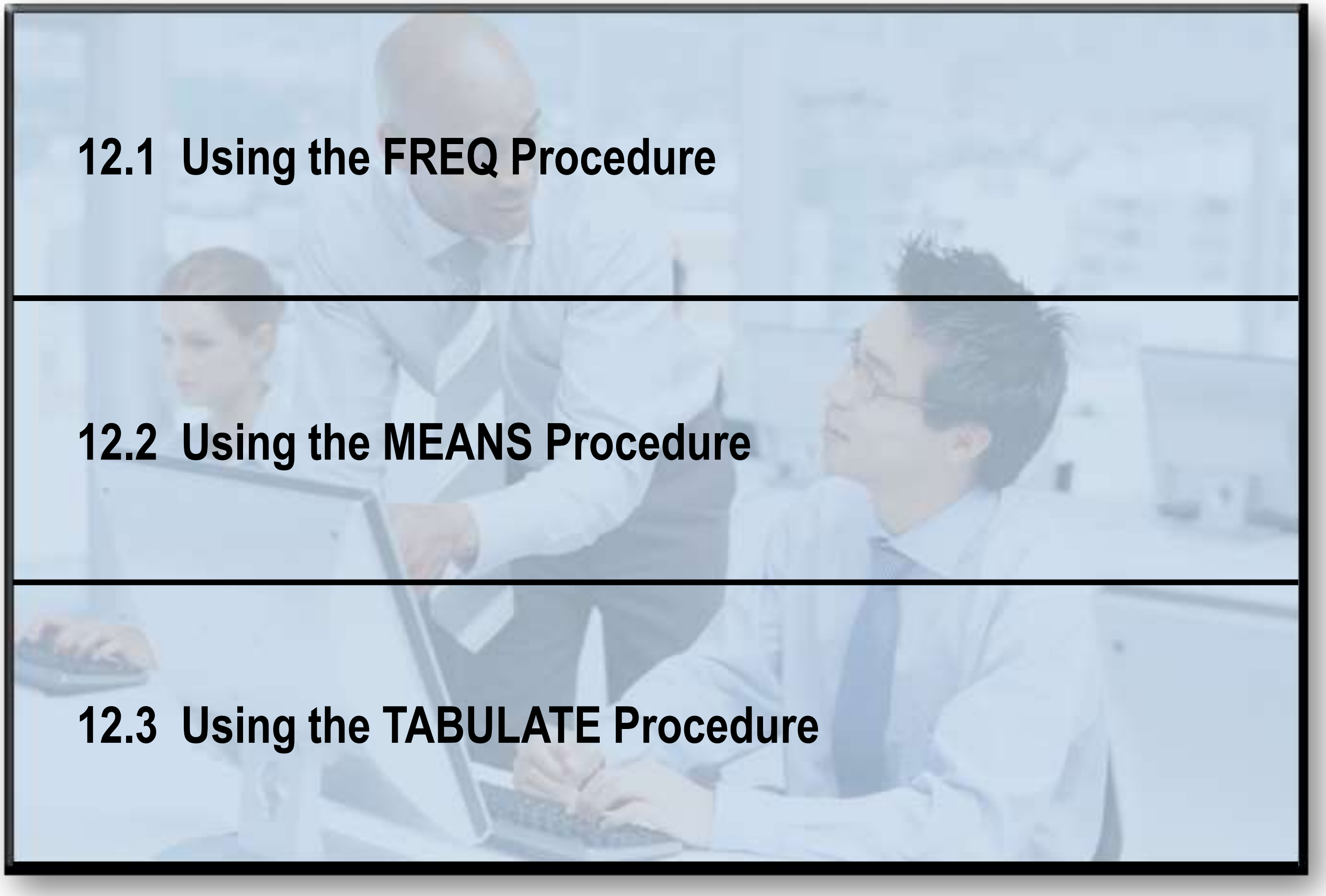
EmpsAll2

First	Gender	Country
Chang	M	China
Cho	F	Japan
Li	M	China
Ming	F	China
Tomi	M	Japan

EmpsCN

EmpsJP

Chapter 12: Producing Summary Reports

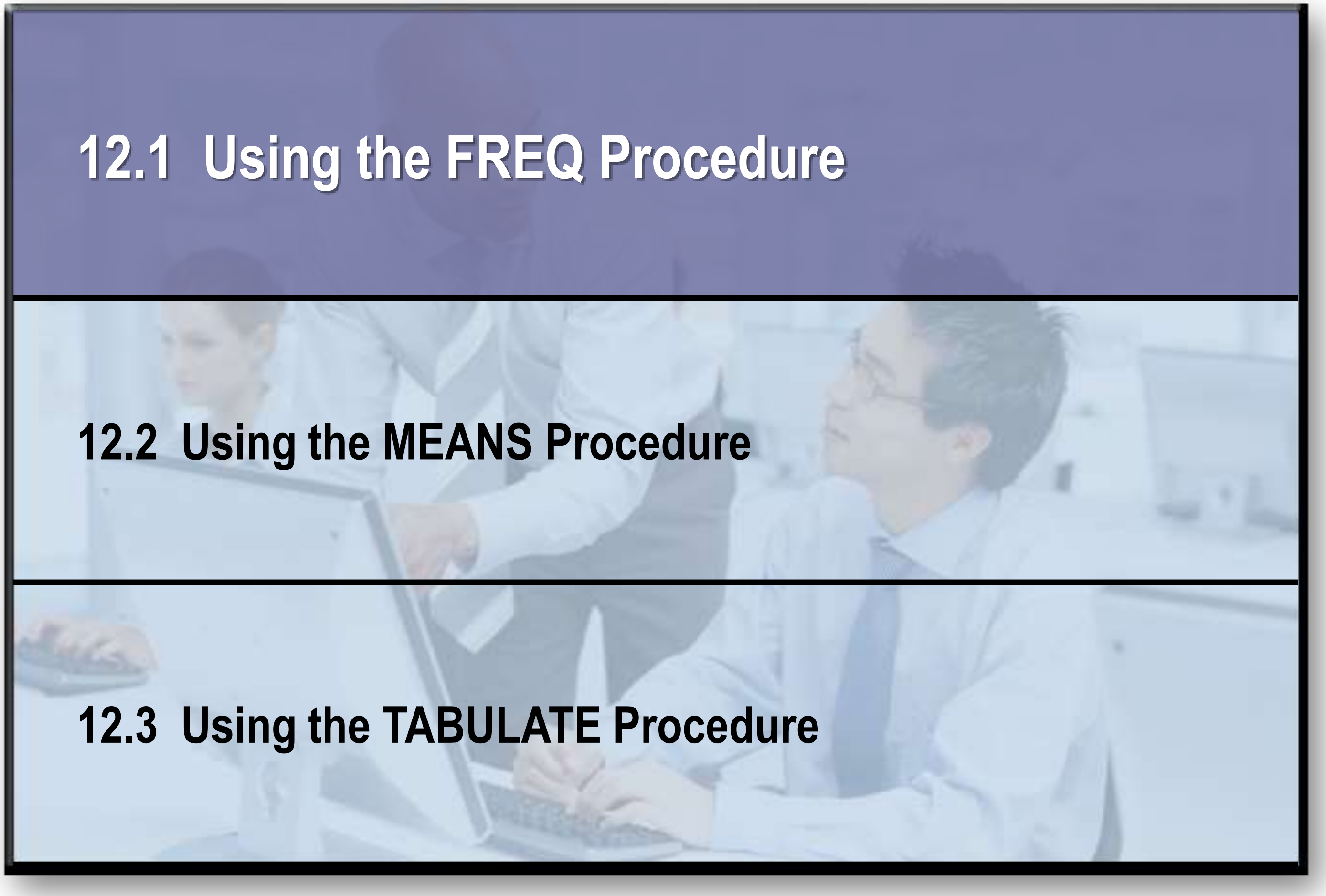


12.1 Using the FREQ Procedure

12.2 Using the MEANS Procedure

12.3 Using the TABULATE Procedure

Chapter 12: Producing Summary Reports



12.1 Using the FREQ Procedure

12.2 Using the MEANS Procedure

12.3 Using the TABULATE Procedure

Objectives

- Produce one-way and two-way frequency tables with the FREQ procedure.
- Enhance frequency tables with options.
- Produce output data sets by using the OUT= option in the TABLES and OUTPUT statements.

The FREQ Procedure

The FREQ procedure can do the following:

- produce one-way to *n*-way frequency and crosstabulation (contingency) tables
- compute chi-square tests for one-way to *n*-way tables and measures of association and agreement for contingency tables
- automatically display the output in a report and save the output in a SAS data set

General form of the FREQ procedure:

```
PROC FREQ DATA=SASdataset <option(s)>;
      TABLES variable(s) </ option(s)>;
RUN;
```

The FREQ Procedure

A FREQ procedure with no TABLES statement generates one-way frequency tables for all data set variables.

```
proc freq data=orion.sales;
run;
```

This PROC FREQ step creates a frequency table for the following nine variables:

- **Employee_ID**
- **First_Name**
- **Last_Name**
- **Gender**
- **Salary**
- **Job_Title**
- **Country**
- **Birth_Date**
- **Hire_Date**

p112d01

The TABLES Statement

The TABLES statement specifies the frequency and crosstabulation tables to produce.

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

one-way frequency tables

An asterisk between variables requests a *n*-way crosstabulation table.

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

two-way frequency table

p112d01

The TABLES Statement

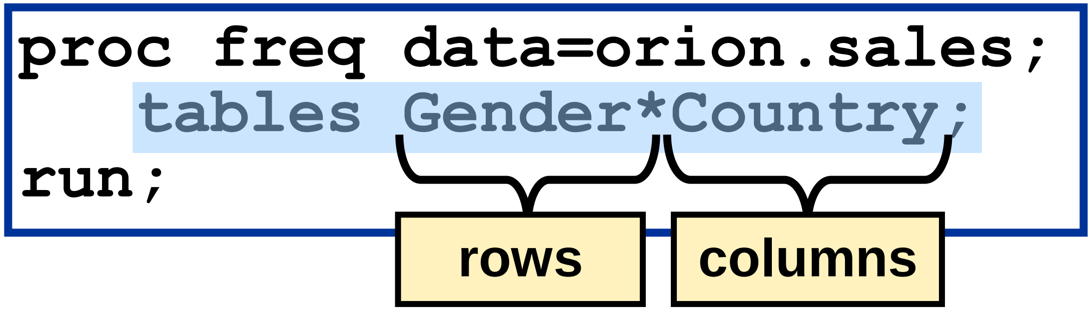
A one-way frequency table produces frequencies, cumulative frequencies, percentages, and cumulative percentages.

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

```
                    The FREQ Procedure

                                       Cumulative    Cumulative
Gender    Frequency     Percent        Frequency      Percent
----------------------------------------------------------------
F                68       41.21              68         41.21
M                97       58.79             165        100.00


                                        Cumulative    Cumulative
Country    Frequency     Percent        Frequency      Percent
-----------------------------------------------------------------
AU                63       38.18              63         38.18
US               102       61.82             165        100.00
```

The TABLES Statement

An *n*-way frequency table produces cell frequencies, cell percentages, cell percentages of row frequencies, and cell percentages of column frequencies, plus total frequency and percent.



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

rows

columns

The TABLES Statement

```
                The FREQ Procedure

          Table of Gender by Country

Gender     Country

Frequency|
Percent  |
Row Pct  |
Col Pct  |AU      |US      |  Total
---------+--------+--------+
F        |     27 |     41 |     68
         |  16.36 |  24.85 |  41.21
         |  39.71 |  60.29 |
         |  42.86 |  40.20 |
---------+--------+--------+
M        |     36 |     61 |     97
         |  21.82 |  36.97 |  58.79
         |  37.11 |  62.89 |
         |  57.14 |  59.80 |
---------+--------+--------+
Total          63      102      165
            38.18    61.82   100.00
```

Poll

Quiz

12.01 Multiple Choice Poll

Which of the following statements **cannot** be added to the PROC FREQ step to enhance the report?

a. FORMAT

b. SET

c. TITLE

d. WHERE

12.01 Multiple Choice Poll – Correct Answer

Which of the following statements **cannot** be added to the PROC FREQ step to enhance the report?

a. FORMAT

[✓ b.] SET

c. TITLE

d. WHERE

Additional SAS Statements

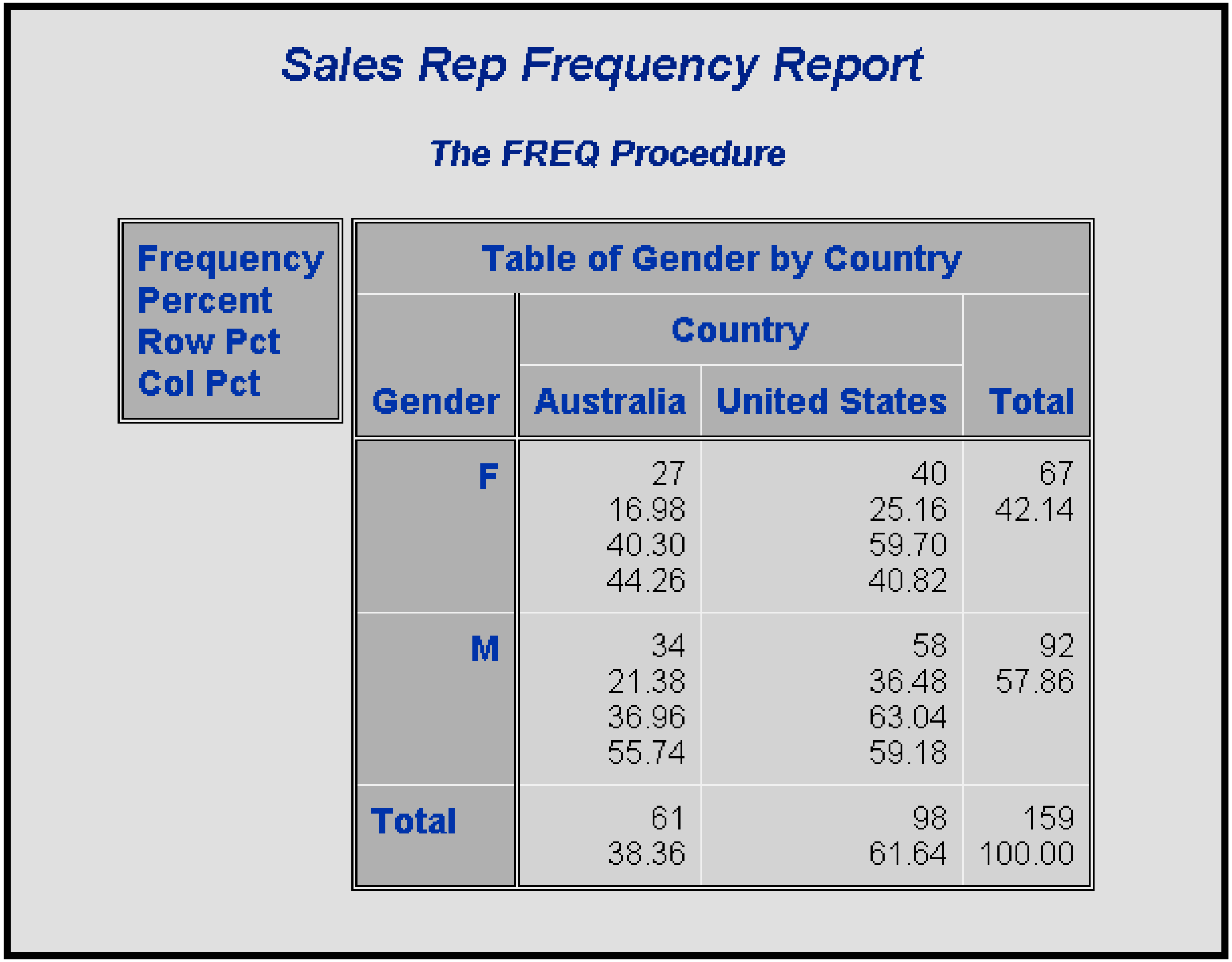
Additional statements can be added to enhance the report.

```
proc format;
   value $ctryfmt 'AU'='Australia'
                  'US'='United States';
run;

options nodate pageno=1;

ods html file='p112d01.html';
proc freq data=orion.sales;
   tables Gender*Country;
   where Job_Title contains 'Rep';
   format Country $ctryfmt.;
   title 'Sales Rep Frequency Report';
run;
ods html close;
```

p112d01

Additional SAS Statements

HTML Output



Sales Rep Frequency Report

The FREQ Procedure

Frequency Percent Row Pct Col Pct

Table of Gender by Country

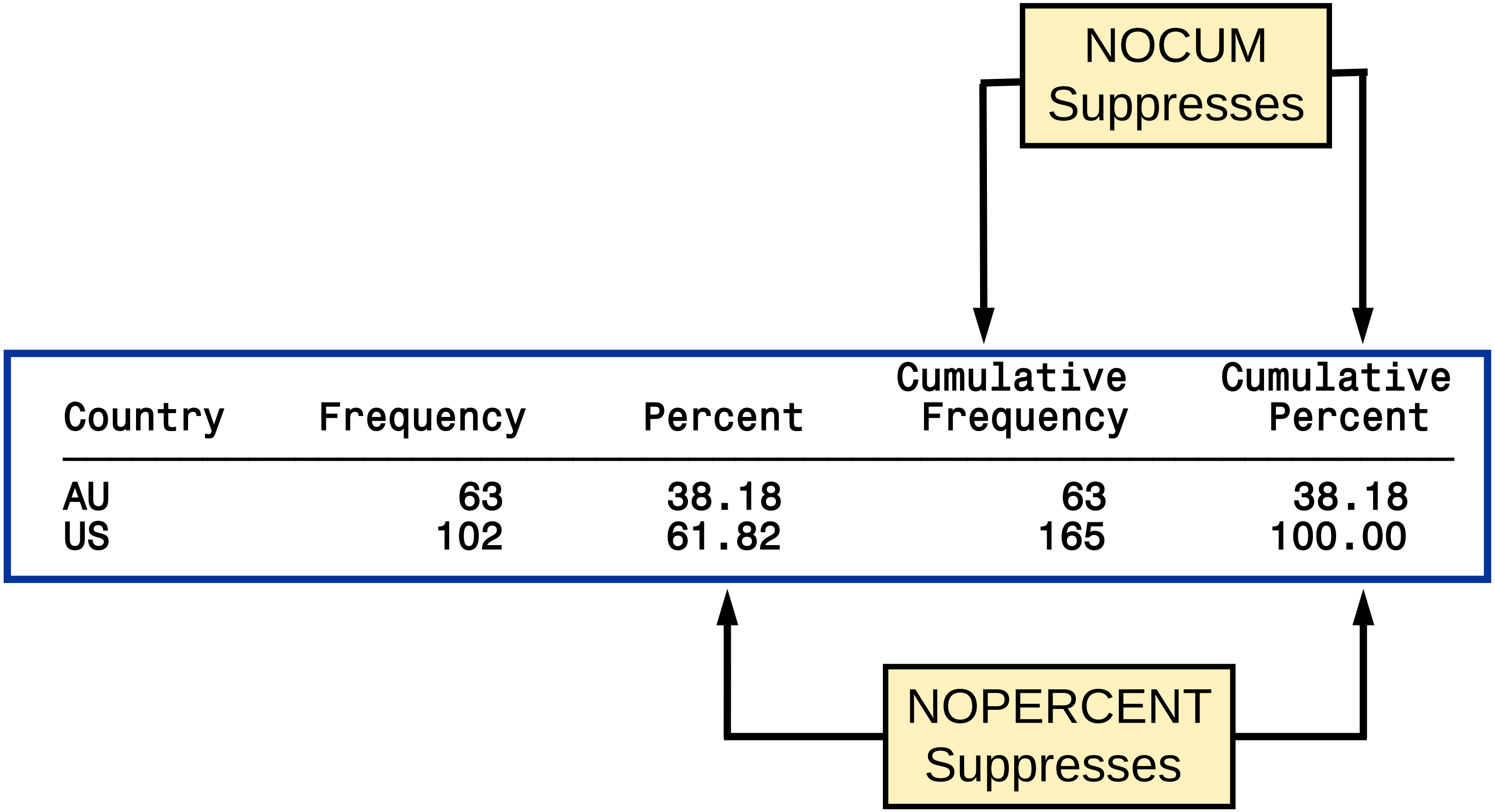
	Country		
Gender	**Australia**	**United States**	**Total**
F	27 16.98 40.30 44.26	40 25.16 59.70 40.82	67 42.14
M	34 21.38 36.96 55.74	58 36.48 63.04 59.18	92 57.86
Total	61 38.36	98 61.64	159 100.00

Options to Suppress Display of Statistics

Options can be placed in the TABLES statement after a forward slash to suppress the display of the default statistics.

Option	Description
NOCUM	suppresses the display of cumulative frequency and cumulative percentage.
NOPERCENT	suppresses the display of percentage, cumulative percentage, and total percentage.
NOFREQ	suppresses the display of the cell frequency and total frequency.
NOROW	suppresses the display of the row percentage.
NOCOL	suppresses the display of the column percentage.

Options to Suppress Display of Statistics

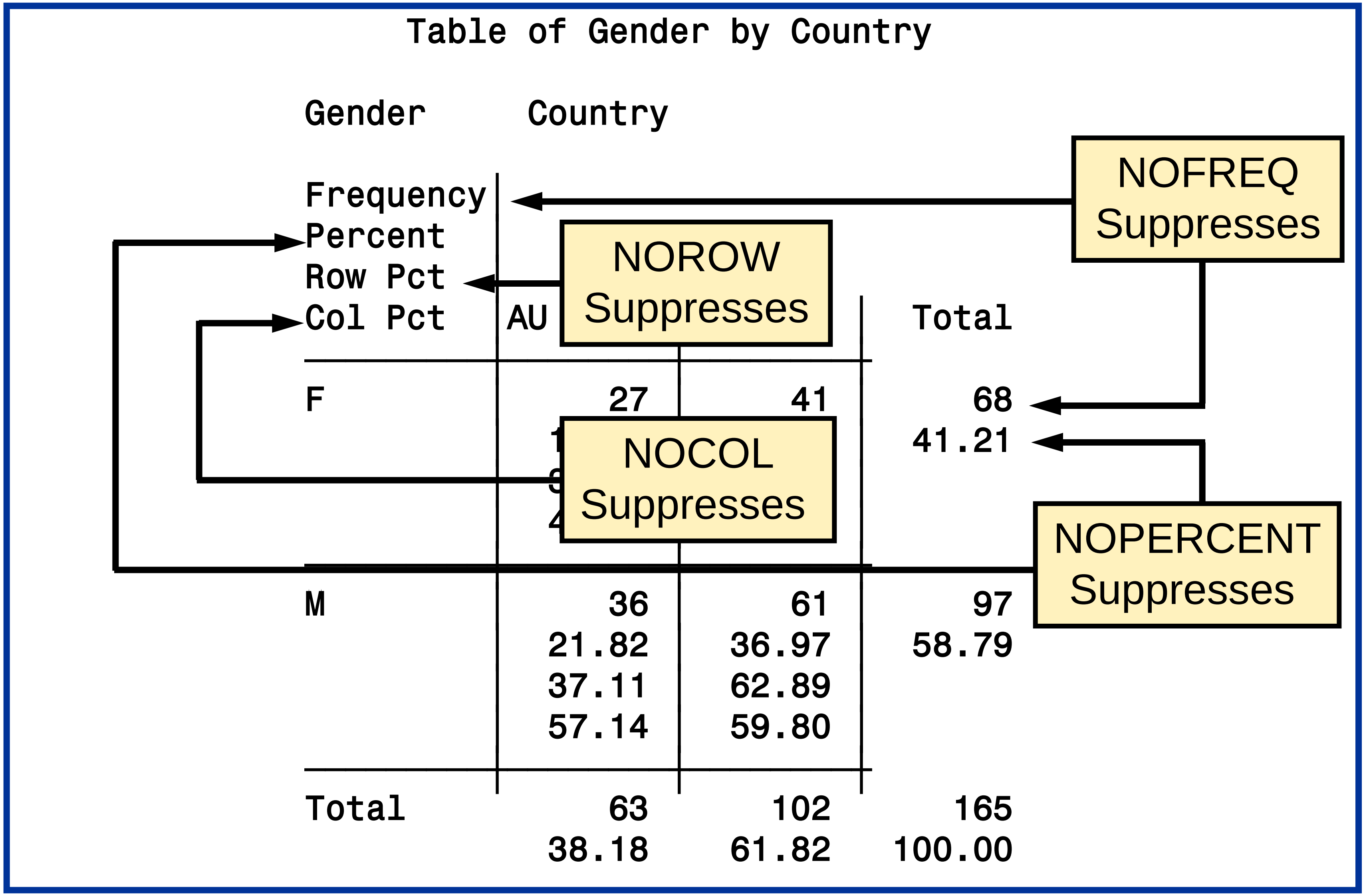


NOCUM Suppresses

```
                                                    Cumulative    Cumulative
Country     Frequency      Percent      Frequency       Percent
----------------------------------------------------------------------------
AU                63         38.18             63         38.18
US               102         61.82            165        100.00
```

NOPERCENT Suppresses

Options to Suppress Display of Statistics



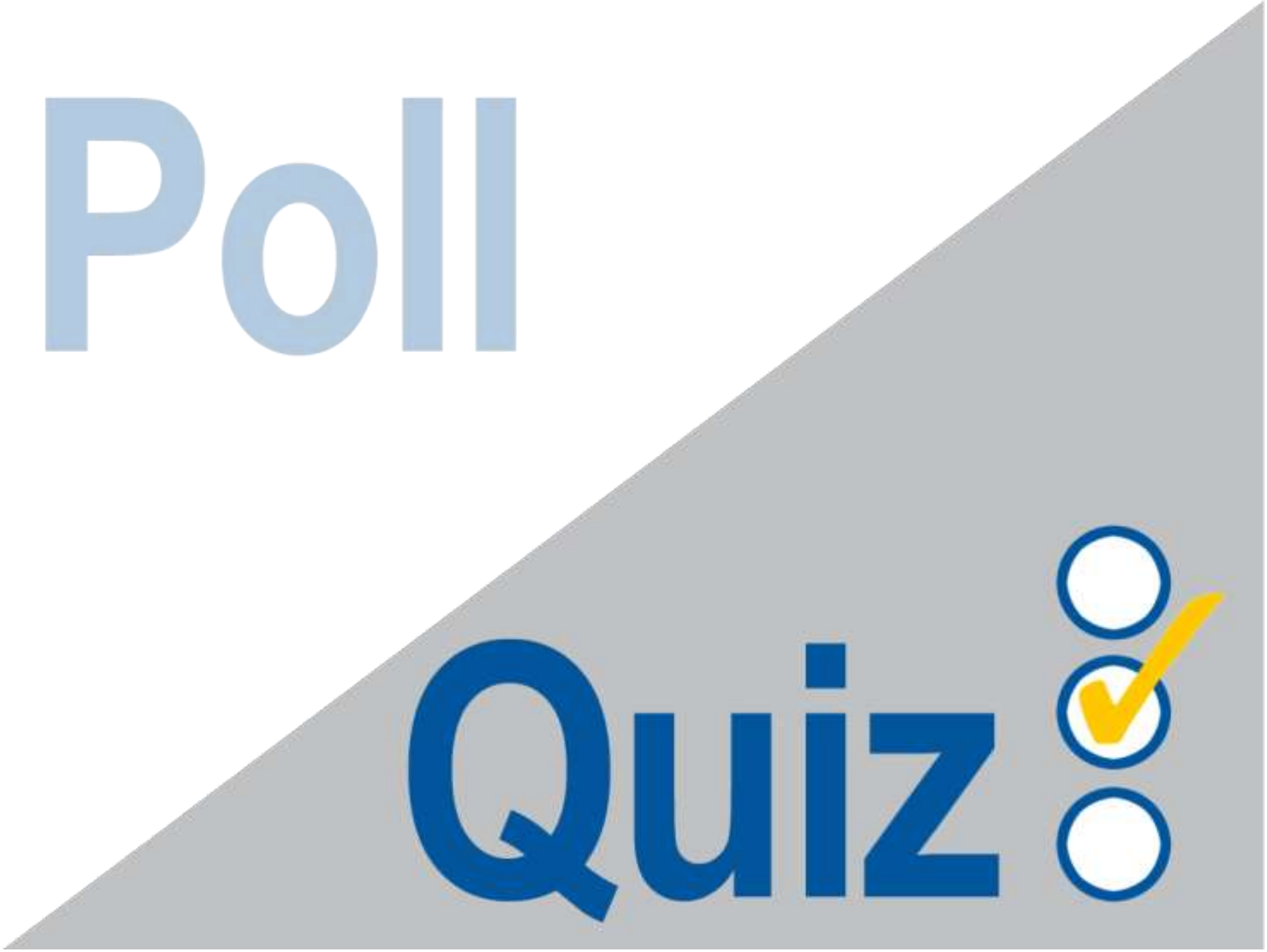
```
                 Table of Gender by Country

         Gender      Country

         Frequency|
         Percent  |
         Row Pct  |
         Col Pct  |AU     |         |  Total
         ---------+-------+---------+
         F        |     27|       41|     68
                  |1      |         |  41.21
                  |9      |         |
                  |4      |         |
         ---------+-------+---------+
         M        |     36|       61|     97
                  |  21.82|    36.97|  58.79
                  |  37.11|    62.89|
                  |  57.14|    59.80|
         ---------+-------+---------+
         Total          63       102     165
                     38.18     61.82  100.00
```

NOFREQ Suppresses

NOROW Suppresses

NOCOL Suppresses

NOPERCENT Suppresses


Poll
Quiz

12.02 Quiz

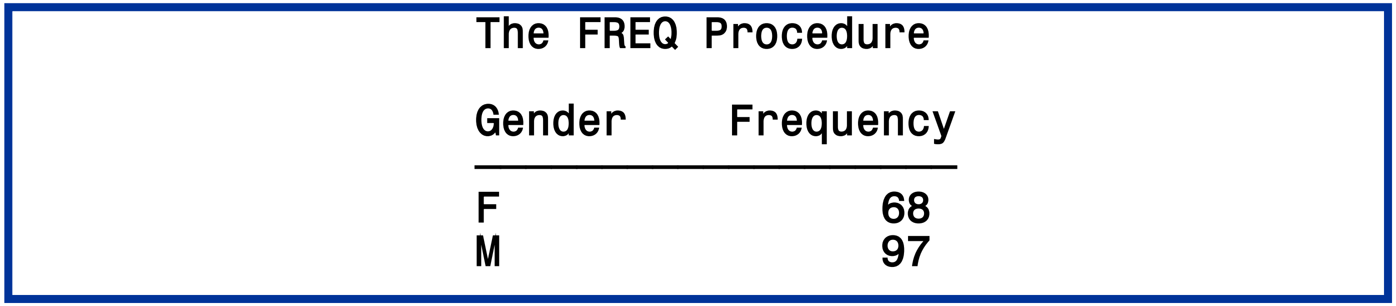
Which TABLES statement correctly creates the report?

a.
```
tables Gender nocum;
```

b.
```
tables Gender nocum nopercent;
```

c.
```
tables Gender / nopercent;
```

d.
```
tables Gender / nocum nopercent;
```



```
The FREQ Procedure

Gender     Frequency
--------------------
F                 68
M                 97
```

p112d01

12.02 Quiz – Correct Answer

Which TABLES statement correctly creates the report?

a. `tables Gender nocum;`

b. `tables Gender nocum nopercent;`

c. `tables Gender / nopercent;`

[✓ d.] `tables Gender / nocum nopercent;`

```
         The FREQ Procedure

         Gender     Frequency
         ____________________
         F                 68
         M                 97
```

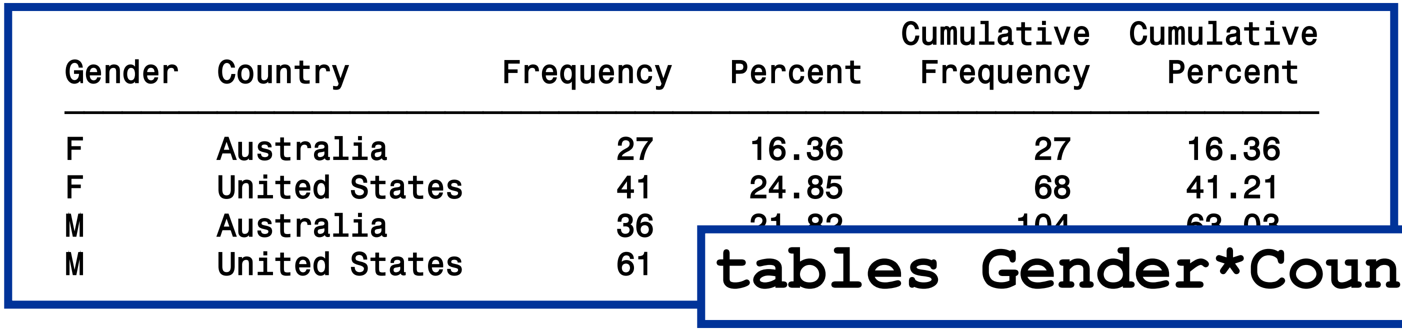
p112d01

Additional TABLES Statement Options

Additional options can be placed in the TABLES statement after a forward slash to control the displayed output.

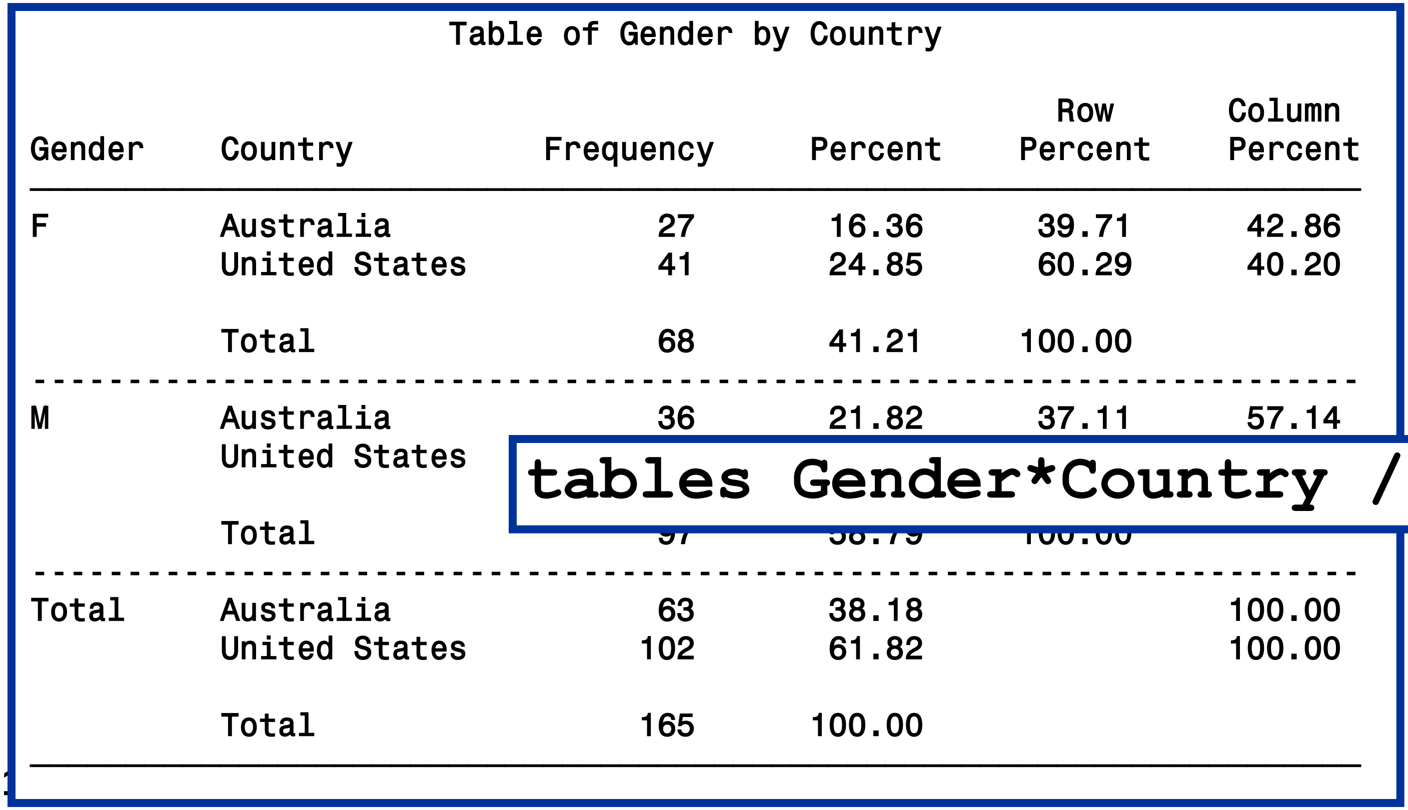
Option	Description
LIST	displays *n*-way tables in list format.
CROSSLIST	displays *n*-way tables in column format.
~~FORMAT=~~	formats the frequencies in *n*-way tables.

LIST and CROSSLIST Options



```
                                               Cumulative  Cumulative
Gender  Country         Frequency   Percent     Frequency     Percent
---------------------------------------------------------------------
F       Australia              27     16.36            27       16.36
F       United States          41     24.85            68       41.21
M       Australia              36     21.82           104       63.03
M       United States          61
```

```
tables Gender*Country / list;
```



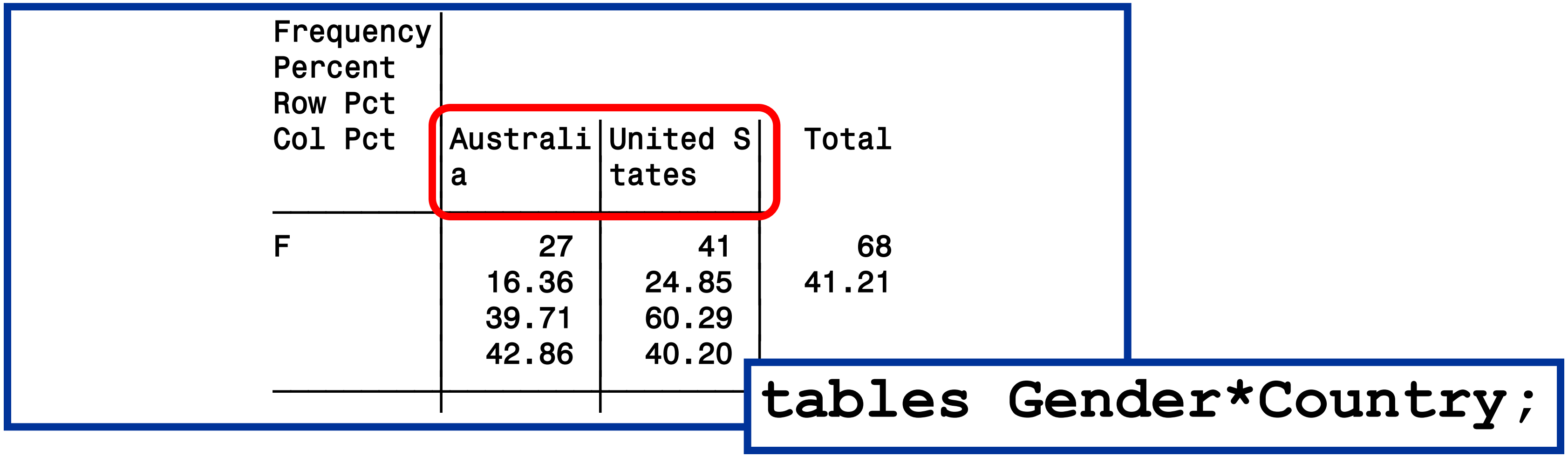
```
                      Table of Gender by Country

                                                       Row    Column
Gender    Country          Frequency   Percent     Percent   Percent
---------------------------------------------------------------------
F         Australia               27     16.36       39.71     42.86
          United States           41     24.85       60.29     40.20

          Total                   68     41.21      100.00
---------------------------------------------------------------------
M         Australia               36     21.82       37.11     57.14
          United States

          Total                   97     58.79      100.00
---------------------------------------------------------------------
Total     Australia               63     38.18                100.00
          United States          102     61.82                100.00

          Total                  165    100.00
---------------------------------------------------------------------
```

```
tables Gender*Country / crosslist;
```

p112d01

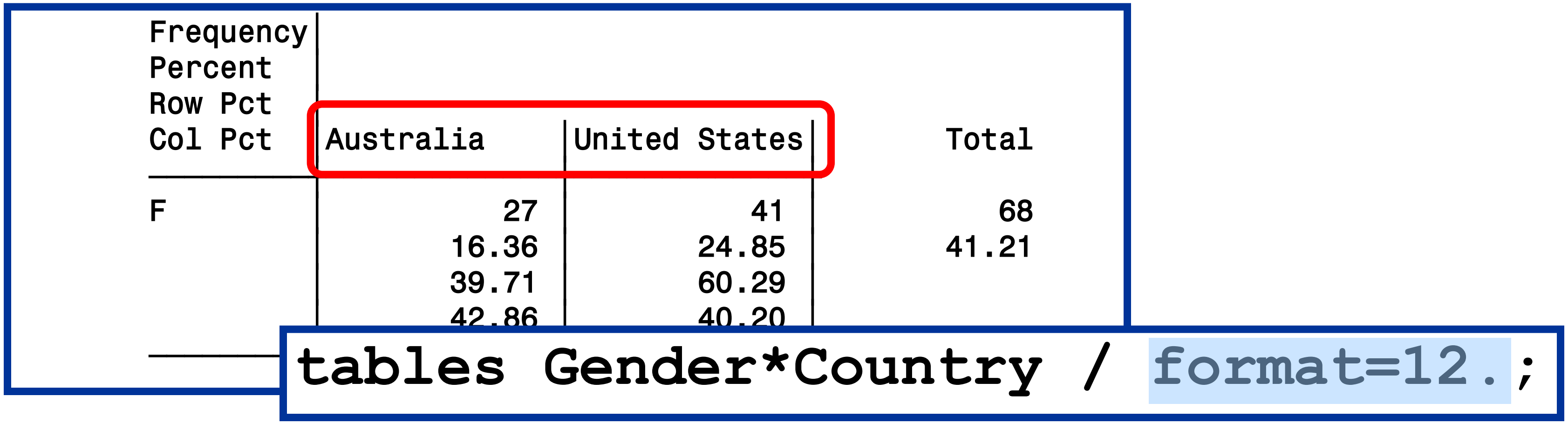
FORMAT= Option (Listing Output Only)

Partial PROC FREQ Outputs



```
Frequency |
Percent   |
Row Pct   |
Col Pct   |Australi|United S|  Total
          |a       |tates   |
----------+--------+--------+
F         |     27 |     41 |     68
          |  16.36 |  24.85 |  41.21
          |  39.71 |  60.29 |
          |  42.86 |  40.20 |
----------+--------+--------+
```

```
tables Gender*Country;
```



```
Frequency |
Percent   |
Row Pct   |
Col Pct   |Australia   |United States|       Total
----------+------------+-------------+
F         |         27 |          41 |          68
          |      16.36 |       24.85 |       41.21
          |      39.71 |       60.29 |
          |      42.86 |       40.20 |
----------
```

```
tables Gender*Country / format=12.;
```

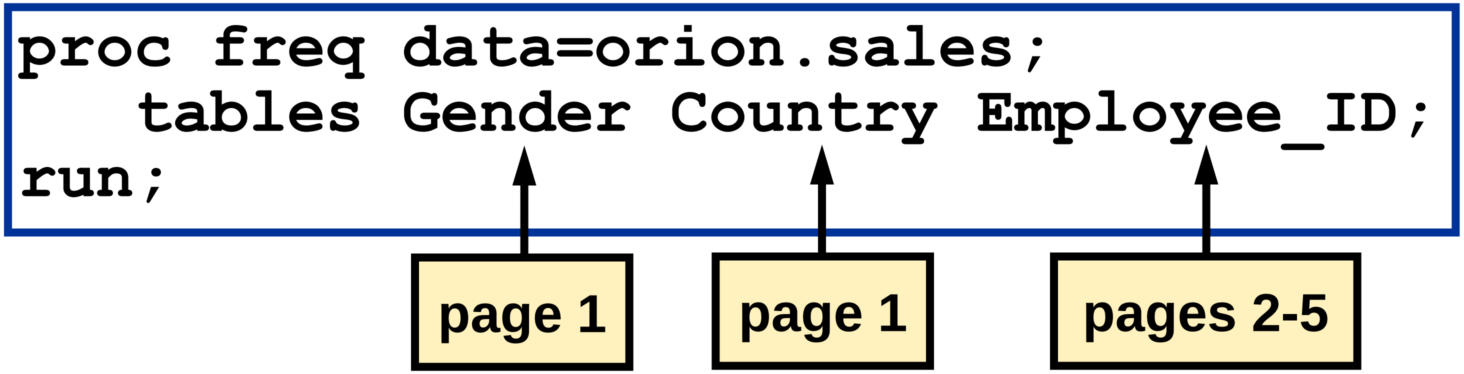
p112d01

PROC FREQ Statement Options

Options can also be placed in the PROC FREQ statement.

Option	Description
NLEVELS	displays a table that provides the number of levels for each variable named in the TABLES statement.
PAGE	displays only one table per page.
COMPRESS	begins the display of the next one-way frequency table on the same page as the preceding one-way table if there is enough space to begin the table.

COMPRESS Option

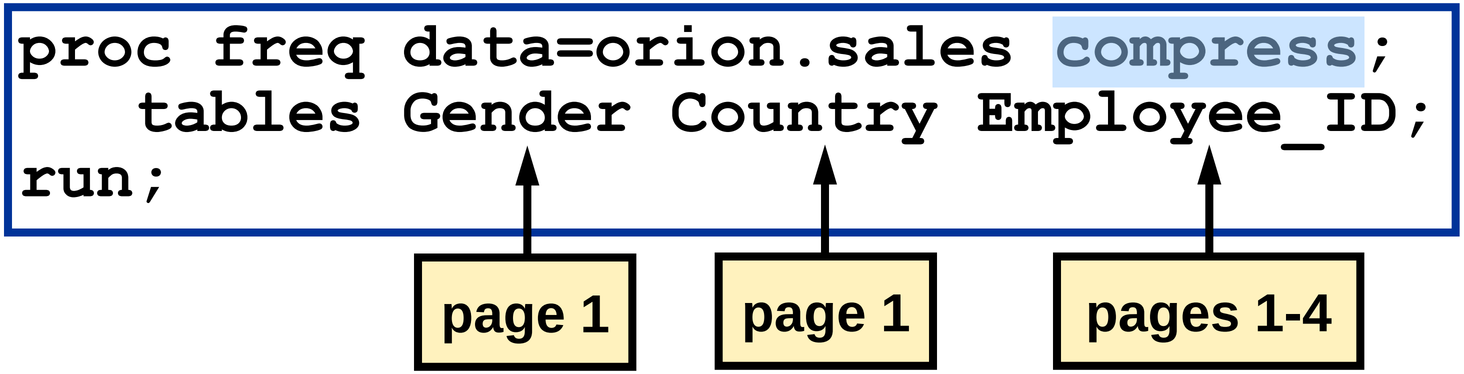


```
proc freq data=orion.sales;
   tables Gender Country Employee_ID;
run;
```

page 1

page 1

pages 2-5



```
proc freq data=orion.sales compress;
   tables Gender Country Employee_ID;
run;
```

page 1

page 1

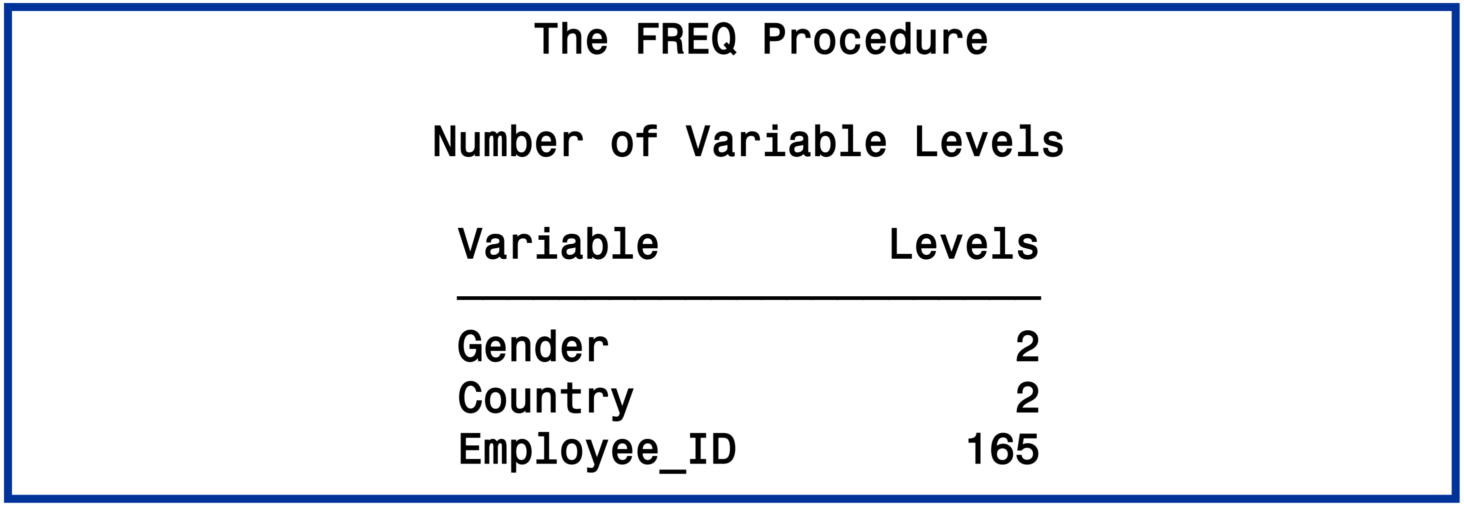
pages 1-4

p112d01

NLEVELS Option

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

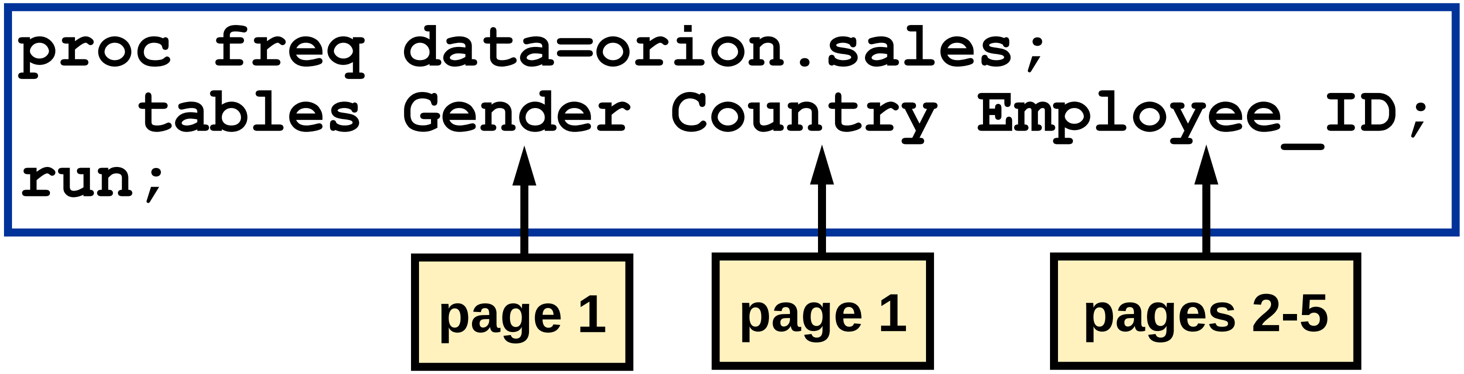
Partial PROC FREQ Output



```
                 The FREQ Procedure

            Number of Variable Levels

             Variable          Levels
             ------------------------
             Gender                 2
             Country                2
             Employee_ID          165
```

p112d01

PAGE Option

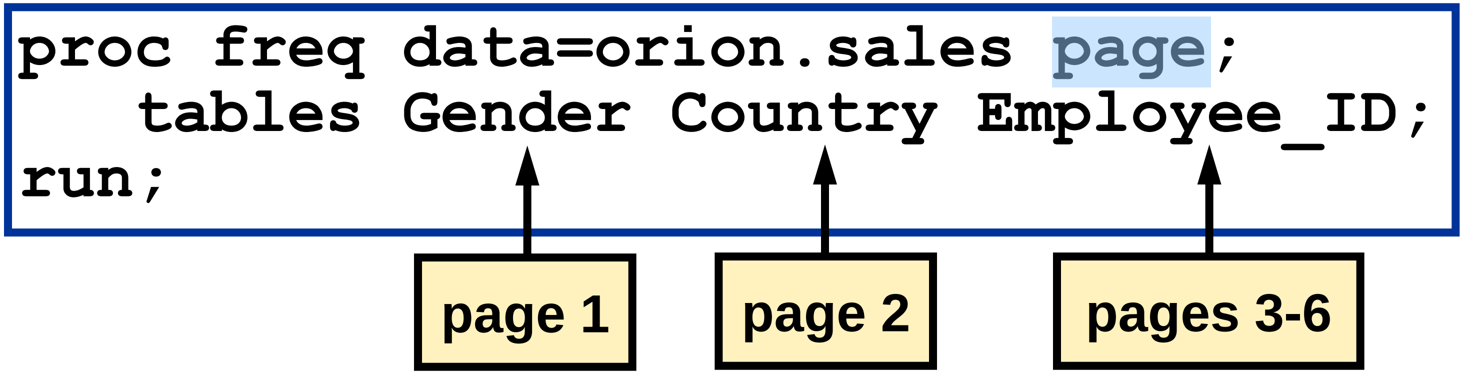


```
proc freq data=orion.sales;
   tables Gender Country Employee_ID;
run;
```

page 1

page 1

pages 2-5



```
proc freq data=orion.sales page;
   tables Gender Country Employee_ID;
run;
```

page 1

page 2

pages 3-6

p112d01