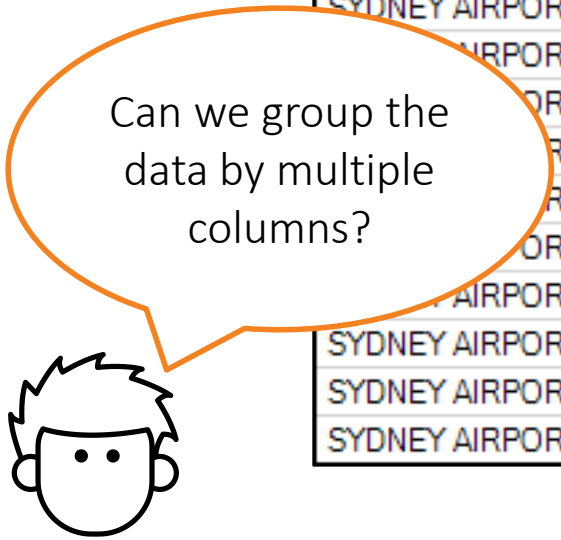
# STAT604 SAS Lesson 12

# Multiple BY Columns

| Name | Date | Rain_ mm | Year | Qtr |
|---|---|---|---|---|
| SYDNEY AIRPORT, AS | 01-2017 | 48.4 | 2017 | 1 |
| SYDNEY AIRPORT, AS | 02-2017 | 158 | 2017 | 1 |
| SYDNEY AIRPORT, AS | 03-2017 | 189.2 | 2017 | 1 |
| SYDNEY AIRPORT, AS | 04-2017 | 94.4 | 2017 | 2 |
| SYDNEY AIRPORT, AS | 05-2017 | 32.4 | 2017 | 2 |
| IRPORT, AS | 06-2017 | 113.6 | 2017 | 2 |
| ORT, AS | 07-2017 | 18 | 2017 | 3 |
| RT, AS | 08-2017 | 27.2 | 2017 | 3 |
| RT, AS | 09-2017 | 0.2 | 2017 | 3 |
| ORT, AS | 10-2017 | 59.6 | 2017 | 4 |
| AIRPORT, AS | 11-2017 | 37.8 | 2017 | 4 |
| SYDNEY AIRPORT, AS | 12-2017 | 44.4 | 2017 | 4 |
| SYDNEY AIRPORT, AS | 01-2018 | 27.6 | 2018 | 1 |
| SYDNEY AIRPORT, AS | 02-2018 | 88.4 | 2018 | 1 |



# Multiple BY Columns

```
data sydney_summary;
     set pg2.weather_sydney;
     by Year Qtr;
run;
```

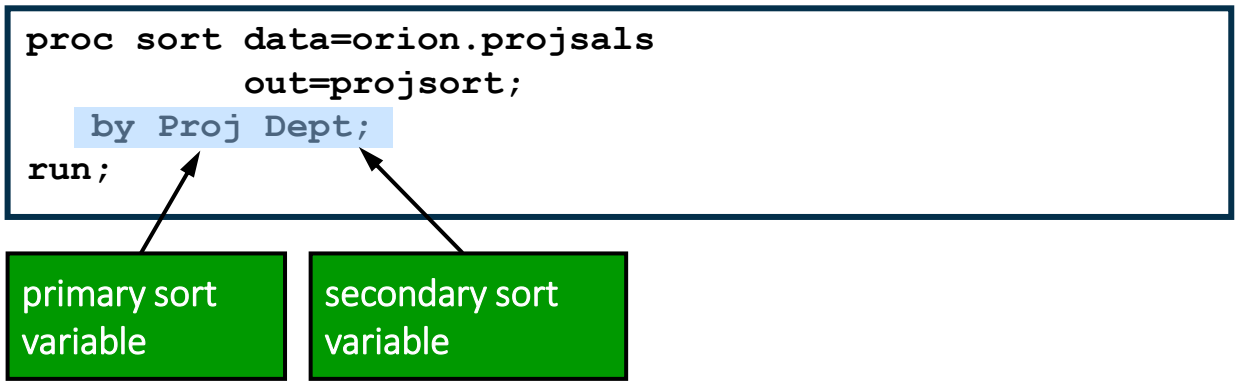
**PDV**

| *other columns* | Year | Qtr | First.Year | Last.Year | First.Qtr | Last.Qtr |
|---|---|---|---|---|---|---|
| | | | D | D | D | D |

**First./Last.** variables are created for each column in the BY statement.

# Sort by Multiple Variables First

In the upcoming illustration, the data must be sorted by multiple BY variables: **Proj** and **Dept**.



# First. / Last. Values: First DATA Step Iteration

| Proj | Dept |
|---|---|
| A | ADMIN |
| A | ADMIN |
| A | ADMIN |
| B | ADMIN |
| B | ENGINR |
| B | ENGINR |
| C | ENGINR |
| C | SALES |
| C | SALES |
| C | SALES |
| C | SALES |

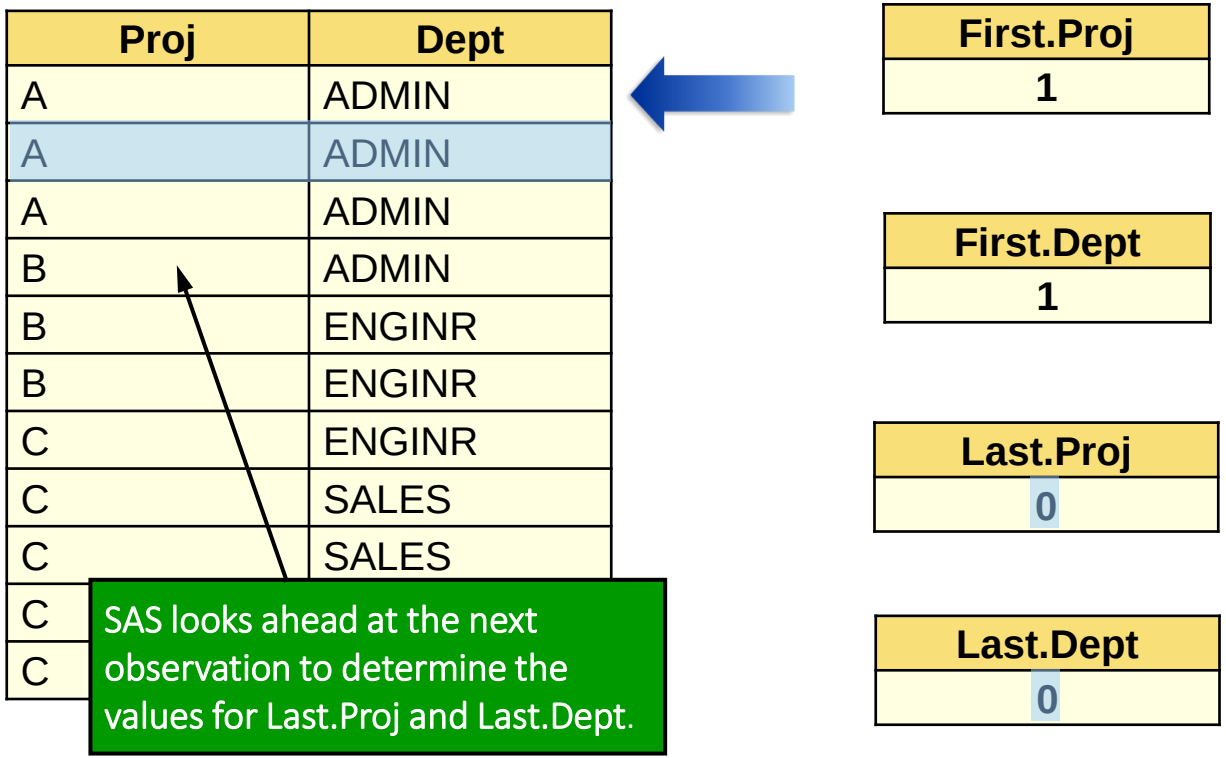
| First.Proj |
|---|
| 1 |

| First.Dept |
|---|
| 1 |

| Last.Proj |
|---|
| ? |

| Last.Dept |
|---|
| ? |

# First. / Last. Values: First DATA Step Iteration



# First. / Last. Values: Second DATA Step Iteration

| Proj | Dept |
|---|---|
| A | ADMIN |
| A | ADMIN |
| A | ADMIN |
| B | ADMIN |
| B | ENGINR |
| B | ENGINR |
| C | ENGINR |
| C | SALES |
| C | SALES |
| C | SALES |
| C | SALES |

| First.Proj |
|---|
| 0 |

| First.Dept |
|---|
| 0 |

| Last.Proj |
|---|
| 0 |

| Last.Dept |
|---|
| 0 |

# First. / Last. Values: Third DATA Step Iteration

| Proj | Dept |
|---|---|
| A | ADMIN |
| A | ADMIN |
| A | ADMIN |
| B | ADMIN |
| B | ENGINR |
| B | ENGINR |
| C | ENGINR |
| C | SALES |
| C | SALES |
| C | SALES |
| C | SALES |

| First.Proj |
|---|
| 0 |

| First.Dept |
|---|
| 0 |

| Last.Proj |
|---|
| 1 |

| Last.Dept |
|---|
| 1 |

# First. / Last. Values: Fourth DATA Step Iteration

| Proj | Dept |
|---|---|
| A | ADMIN |
| A | ADMIN |
| A | ADMIN |
| B | ADMIN |
| B | ENGINR |
| B | ENGINR |
| C | ENGINR |
| C | SALES |
| C | SALES |
| C | SALES |
| C | SALES |

| First.Proj |
|---|
| 1 |

| First.Dept |
|---|
| 1 |

| Last.Proj |
|---|
| 0 |

| Last.Dept |
|---|
| 1 |

# First. / Last. Values: Fifth DATA Step Iteration

| Proj | Dept |
|---|---|
| A | ADMIN |
| A | ADMIN |
| A | ADMIN |
| B | ADMIN |
| B | ENGINR |
| B | ENGINR |
| C | ENGINR |
| C | SALES |
| C | SALES |
| C | SALES |
| C | SALES |

| First.Proj |
|---|
| 0 |

| First.Dept |
|---|
| 1 |

| Last.Proj |
|---|
| 0 |

| Last.Dept |
|---|
| 0 |

# Short Answer Poll

What are the values for First. and Last. variables when the DATA step is processing the observation indicated by the arrow?

| Proj | Dept |
|---|---|
| A | ADMIN |
| A | ADMIN |
| A | ADMIN |
| B | ADMIN |
| B | ENGINR |
| B | ENGINR |
| C | ENGINR |
| C | SALES |
| C | SALES |

| First.Proj |
|---|
| ? |

| First.Dept |
|---|
| ? |

| Last.Proj |
|---|
| ? |

| Last.Dept |
|---|
| ? |

# Short Answer Poll – Correct Answer

What are the values for First. and Last. variables when the DATA step is processing the observation indicated by the arrow?

| Proj | Dept |
|---|---|
| A | ADMIN |
| A | ADMIN |
| A | ADMIN |
| B | ADMIN |
| B | ENGINR |
| B | ENGINR |
| C | ENGINR |
| C | SALES |
| C | SALES |

| First.Proj |
|---|
| 0 |

| First.Dept |
|---|
| 0 |

| Last.Proj |
|---|
| 1 |

| Last.Dept |
|---|
| 1 |

# First. and Last. for Multiple BY Variables

When you use more than one variable in the BY statement, **Last.***BY-variable*=1 for the ***primary variable*** forces **Last.***BY-variable*=1 for the ***secondary variable(s)***.

| Proj | Dept | First. Proj | Last. Proj | First. Dept | Last.Dept |
|---|---|---|---|---|---|
| A | ADMIN | 1 | 0 | 1 | 0 |
| A | ADMIN | 0 | 0 | 0 | 0 |
| A | ADMIN | 0 | 1 | 0 | 1 |
| B | ADMIN | 1 | 0 | 1 | 1 |
| B | ENGINR | 0 | 0 | 1 | 0 |

goes over nodup stuff. need to review.

# Using Multiple By Groups

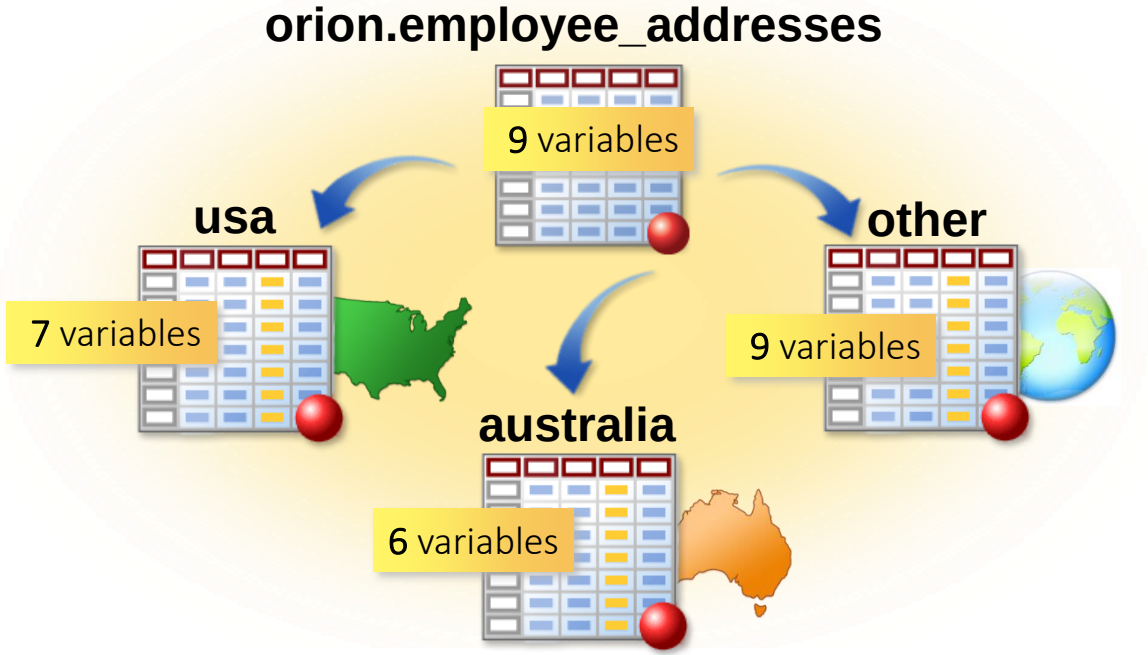
This demonstration compares PROC SORT with the DATA step from removing duplicates and illustrates using multiple by groups to create multiple accumulating variables.

# Controlling Column Usage

Drop and Keep Data Set Options – Prep Guide Page 156

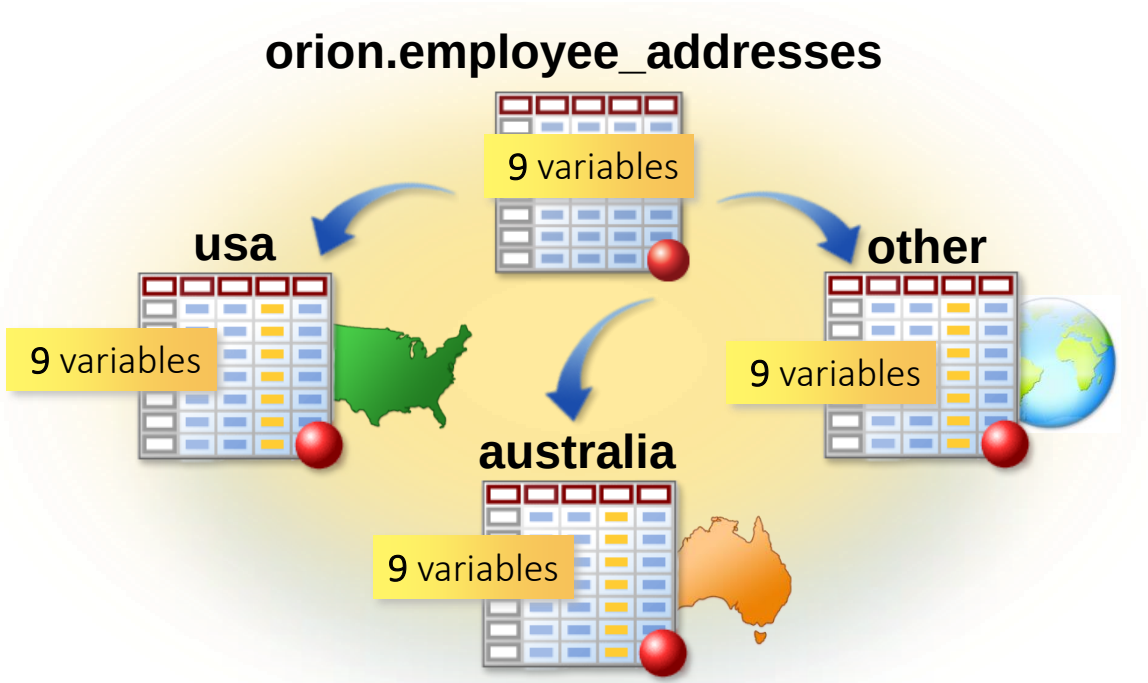
# Business Scenario

You have controlled the observations written to the three new data sets, and now you want to control the variables that are written to each.



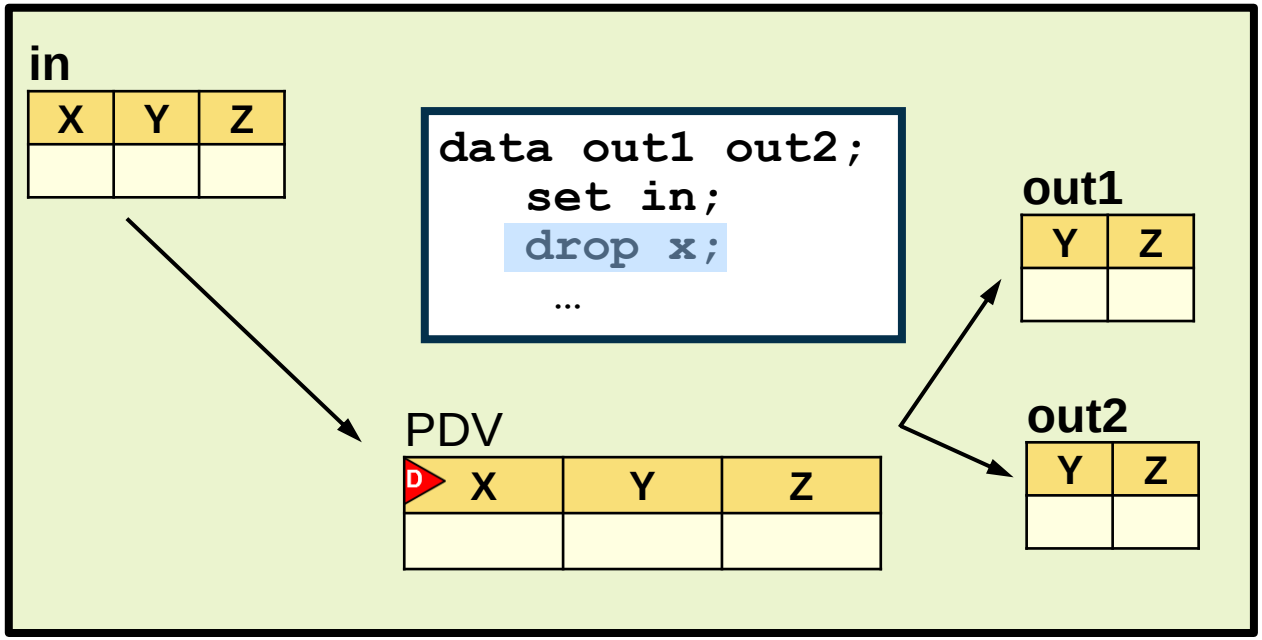
# Controlling Variable Output (Review)

By default, SAS writes all variables from the input data set to every output data set.



# Controlling Variable Output (Review)

In the DATA step, the DROP and KEEP statements can be used to control which variables are written to output data sets.



# DROP Statement

The *DROP statement* drops variables from every output data set.

```
data usa australia other;
   drop Street_ID;
   set orion.employee_addresses;
   if Country='US' then output usa;
   else if Country='AU' then output australia;
   else output other;
run;
```

Partial SAS Log

```
NOTE: There were 424 observations read from the data set
      ORION.EMPLOYEE_ADDRESSES.
NOTE: The data set WORK.USA has 311 observations and 8 variables.
NOTE: The data set WORK.AUSTRALIA has 105 observations and 8
      variables.
NOTE: The data set WORK.OTHER has 8 observations and 8 variables.
```

# Controlling Variable Output

The task is to drop **Street_ID** and **Country** from **usa**, drop **Street_ID**, **Country**, and **State** from **australia**, and keep all variables in **other**.

USA

| Employee_ ID | Employee_Name | Street_ Number | Street_Name | City | State | Postal_ Code |
|---|---|---|---|---|---|---|
| 121044 | Abbott, Ray | 2267 | Edwards Mill Rd | Miami-Dade | FL | 33135 |
| 120761 | Akinfolarin, Tameaka | 5 | Donnybrook Rd | Philadelphia | PA | 19145 |
| 120656 | Amos, Salley | 3524 | Calico Ct | San Diego | CA | 92116 |

Australia

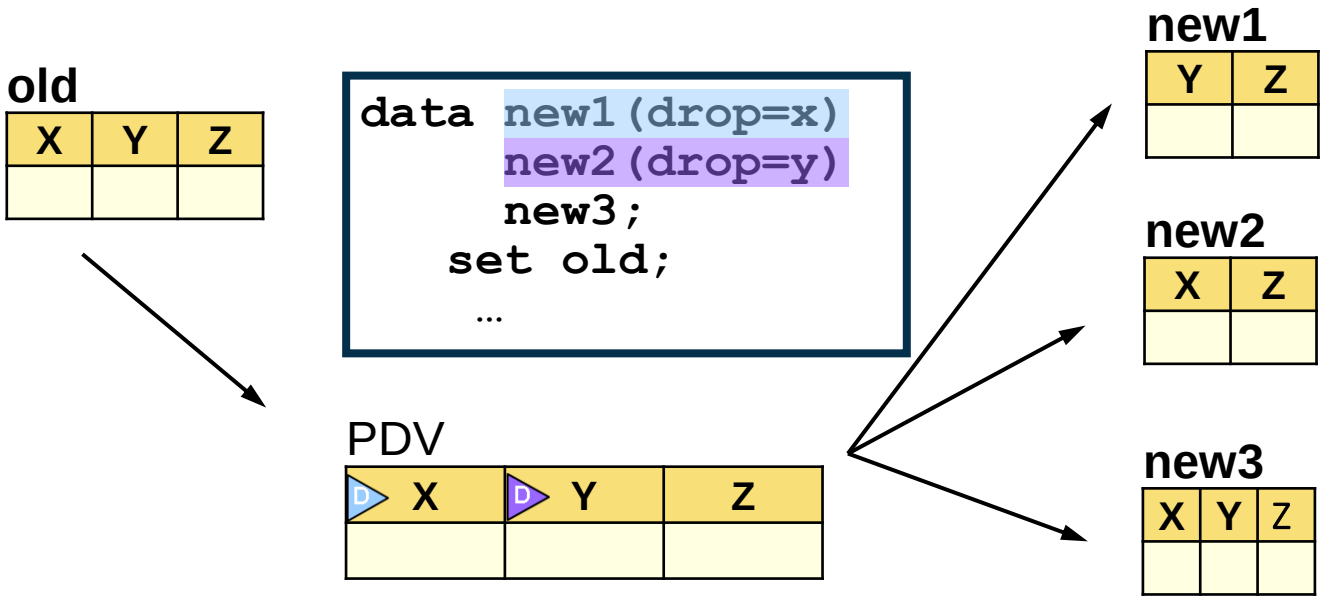
| Employee_ ID | Employee_Name | Street_ Number | Street_Name | City | Postal_ Code |
|---|---|---|---|---|---|
| 120145 | Aisbitt, Sandy | 30 | Bingera Street | Melbourne | 2001 |
| 120185 | Bahlman, Sharon | 24 | LaTrobe Street | Sydney | 2165 |
| 120109 | Baker, Gabriele | 166 | Toorak Road | Sydney | 2119 |

Other

| Employee_ ID | Employee_Name | Street_ID | Street_ Number | Street_Name | City | State | Postal_ Code | Country |
|---|---|---|---|---|---|---|---|---|
| 121019 | Desanctis, Scott | 9260121087 | 765 | Greenhaven Ln | Philadelphia | PA | 19102 | us |
| 120997 | Donathan, Mary | 9260121069 | 4923 | Gateridge Dr | Philadelphia | PA | 19152 | us |
| 120747 | Farthing, Zashia | 9260123756 | 763 | Chatterson Dr | San Diego | CA | 92116 | us |

# DROP= Option on an Output Data Set

*SAS-data-set*(**DROP=***variable-1 <variable-2 …variable-n>*)



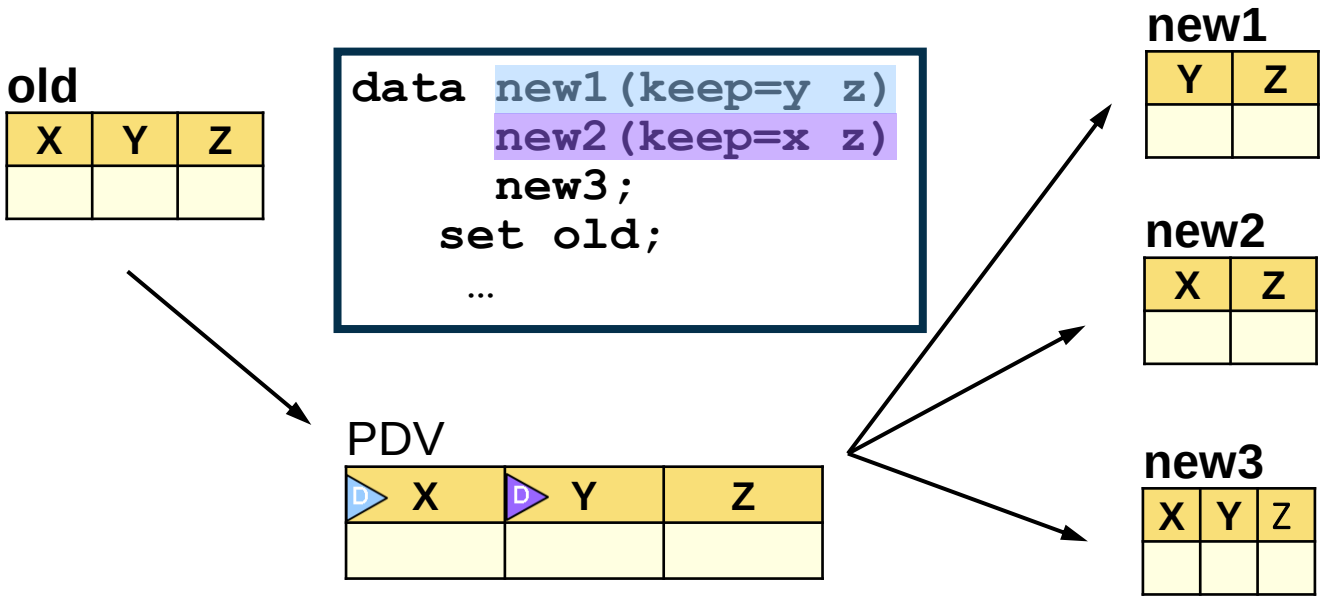
# Using the DROP= Data Set Option

```
data usa(drop=Street_ID Country)
     australia(drop=Street_ID State Country)
     other;
   set orion.employee_addresses;
   if Country='US' then output usa;
   else if Country='AU' then
            output australia;
   else output other;
run;
```

```
NOTE: There were 424 observations read from the data set
      ORION.EMPLOYEE_ADDRESSES.
NOTE: The data set WORK.USA has 311 observations
      and 7 variables.
NOTE: The data set WORK.AUSTRALIA has 105 observations
      and 6 variables.
NOTE: The data set WORK.OTHER has 8 observations
      and 9 variables.
```

# KEEP= Option on an Output Data Set

*SAS-data-set*(**KEEP=***variable-1 <variable-2 …variable-n>*)



# Using the DROP= and KEEP= Options

The DROP= and KEEP= options can both be used in a SAS program.

```
data usa(keep=Employee_Name City State)
     australia(drop=Street_ID State)
     other;
   set orion.employee_addresses;
   if Country='US' then output usa;
   else if Country='AU' then output australia;
   else output other;
run;
```

# Short Answer Poll

The data set **orion.employee_addresses** contains nine variables. How many variables are in the **usa**, **australia**, and **other** data sets?

```
data usa(keep=Employee_Name City State Country)
     australia(drop=Street_ID State Country)
     other;
   set orion.employee_addresses;
   if Country='US' then output usa;
   else if Country='AU' then output australia;
   else output other;
run;
```

# Short Answer Poll – Correct Answer

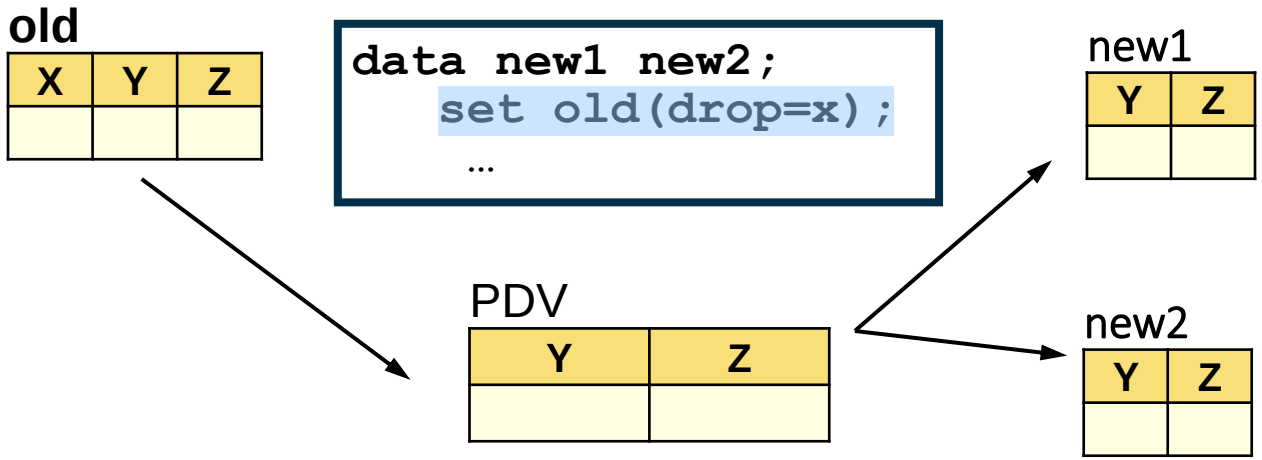
The data set **orion.employee_addresses** contains nine variables. How many variables are in the **usa**, **australia**, and **other** data sets? **4, 6, 9**

```
data usa(keep=Employee_Name City State Country)
     australia(drop=Street_ID State Country)
     other;
   set orion.employee_addresses;
   if Country='US' then output usa;
   else if Country='AU' then output australia;
   else output other;
run;
```

Four variables are kept in usa, three are dropped from australia, and there is no DROP or KEEP for other.

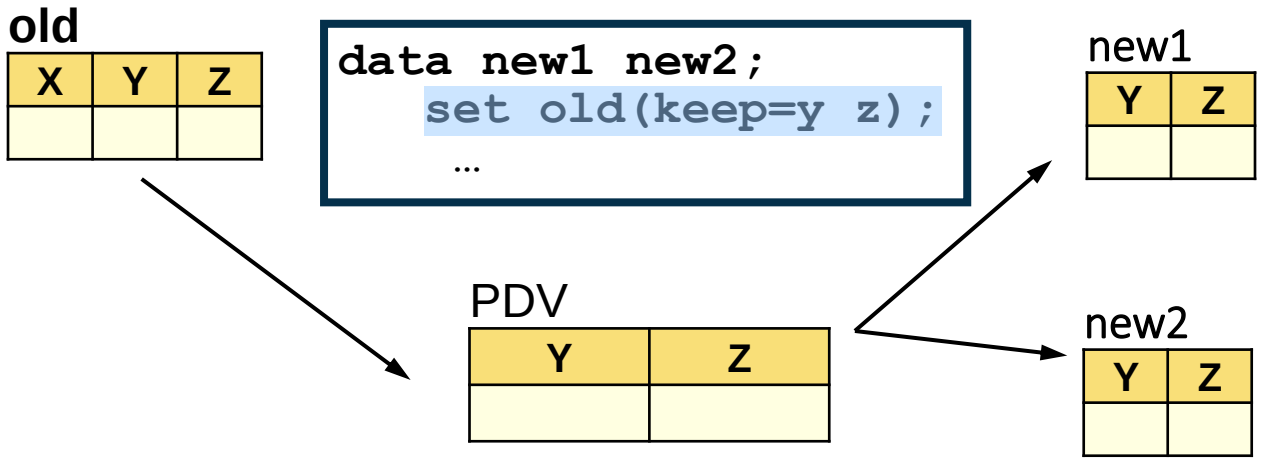
# Using DROP= Option on an Input Data Set

When a **DROP=** data set option is used on an input data set, the specified variables are not read into the PDV and therefore are ***not*** available for processing.



# Using KEEP= Option on an Input Data Set

When a **KEEP=** data set option is used on an input data set, only the specified variables are read into the PDV and therefore are available for processing.



# Short Answer Poll

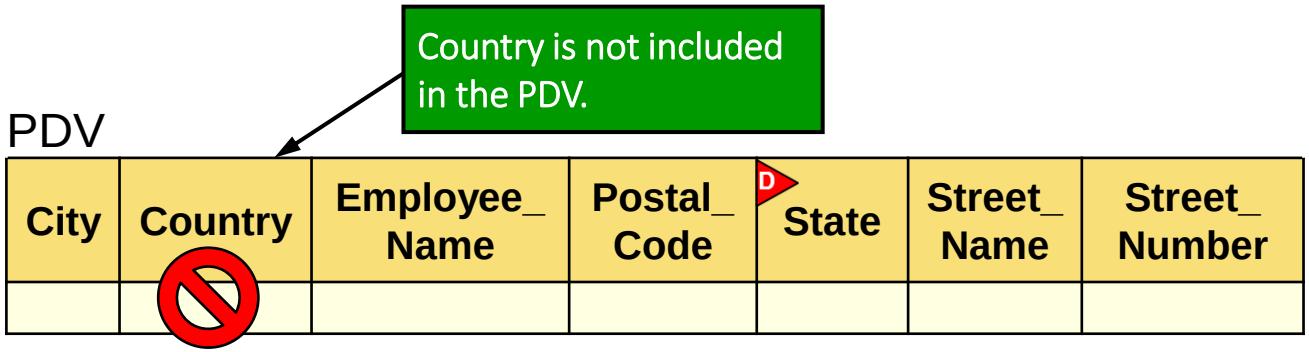
Evaluate the program shown below. The intent is to drop **Country**, **Street_ID**, and **Employee_ID** from every data set, and to drop **State** from **australia**. What is wrong with the program?

```
data usa australia(drop=State) other;
   set orion.employee_addresses
       (drop=Country Street_ID Employee_ID);
   if Country='US' then output usa;
   else if Country='AU' then output australia;
   else output other;
run;
```

# 2.07 Short Answer Poll – Correct Answer

Country is dropped on input, and therefore it is not available for processing. Every observation is written to other.

```
data usa australia(drop=State) other;
   set orion.employee_addresses
       (drop=Country Street_ID Employee_ID);
   if Country='US' then output usa;
   else if Country='AU' then output australia;
   else output other;
run;
```



# Improved Solution

Use a combination of the DROP= option and the DROP statement to achieve the desired results.

```
data usa australia(drop=State) other;
  set orion.employee_addresses
      (drop=Street_ID Employee_ID);
  drop Country;
  if Country='US' then output usa;
  else if Country='AU' then output australia;
  else output other;
run;
```

State is dropped only from australia.

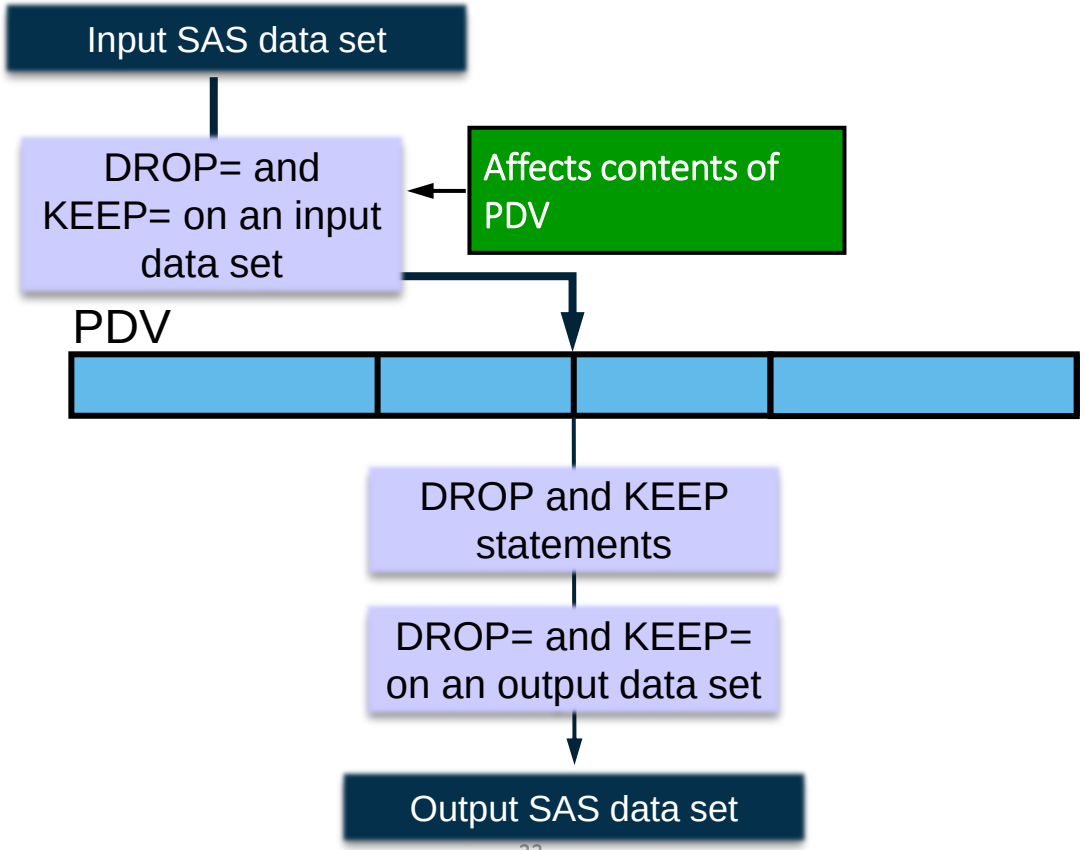
PDV

| City | D Country | Employee_ Name | Postal_ Code | D State | Street_ Name | Street_ Number |
|---|---|---|---|---|---|---|
| | | | | | | |

# Check the SAS Log

```
NOTE: There were 424 observations read from the data set
      ORION.EMPLOYEE_ADDRESSES.
NOTE: The data set WORK.USA has 311 observations
      and 6 variables.
NOTE: The data set WORK.AUSTRALIA has 105 observations
      and 5 variables.
NOTE: The data set WORK.OTHER has 8 observations
      and 6 variables.
```

# Controlling Variable Input



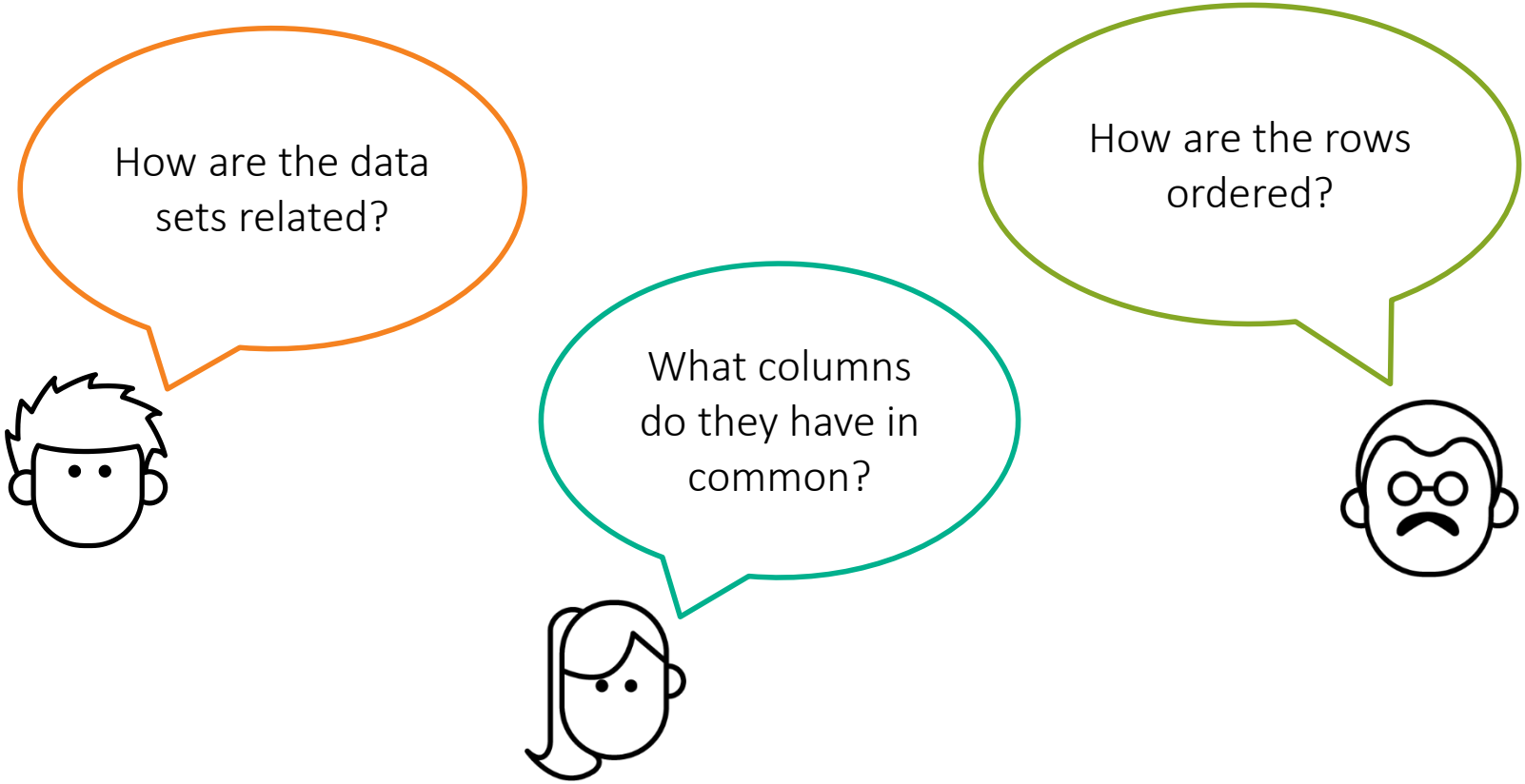
# Controlling Columns

This demonstration illustrates the effects of using a DROP option on the input data set.

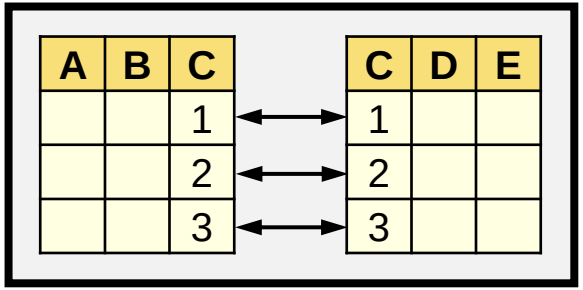
# Combining Data

Combining SAS Data Sets – Chapter 10

# Before Combining – Know Thy Data

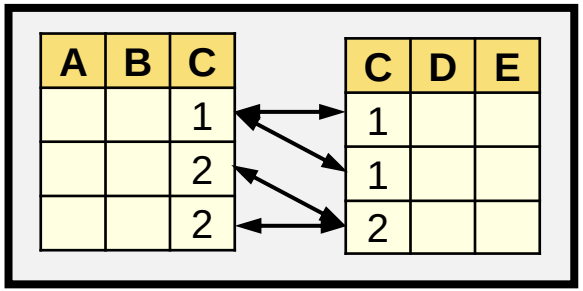


# Relationships in Data




## One-to-One

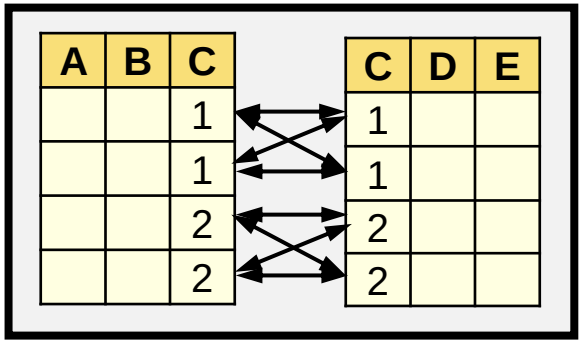
A single observation in one data set is related to exactly one observation in another data set based on the values of one or more selected variables.




## One-to-Many or Many-to-One

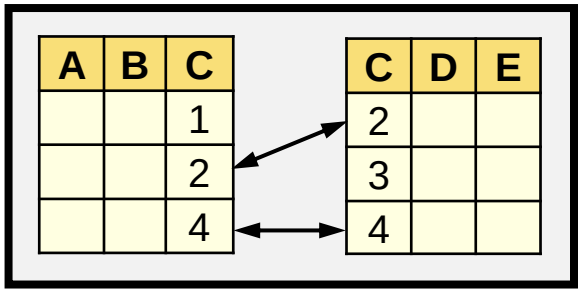
A single observation in one data set is related to more than one observation in another data set based on the values of one or more selected variables.

# Relationships in Data




## Many-to-Many

Multiple observations in one data set are related to more than one observation in another data set based on the values of one or more selected variables.

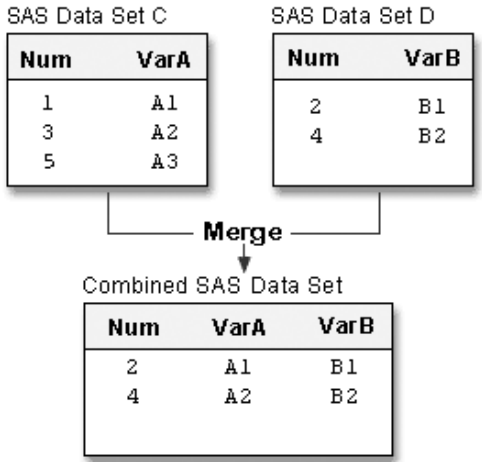



## Nonmatches

At least one observation in one data set is unrelated to any observation in another data set based on the values of one or more selected variables.

# One-to-One Reading

- Creates observations that contain all of the variables from each contributing data set.
- Variables with the same name overwrite with values of last data set.
- Combines observations based on their relative position in each data set.
- Using multiple SET statements - DATA step stops after it has read the last row from the smallest data set.
- Similar to MERGE with no BY, which reads all observations and populates extra rows with missing values.




| Num | VarA | VarB |
|---|---|---|
| 2 | A1 | B1 |
| 4 | A2 | B2 |
| 5 | A3 | |

# One-to-One Reading

In the following example, you have basic patient data in Cert.Patients that you want to combine with other patient data that is stored in Cert.Measure. The height and weight data is stored in the data set Cert.Measure. Both data sets are sorted by the variable ID.

Notice that Cert.Patients contains nine out of eleven observations in which the patient age is less than 60.

***Figure 10.2 Example: One-to-One Reading***

**SAS Data Set Cert.Patients**

| | ID | Sex | Age |
|---|---|---|---|
| 1 | 1129 | F | 48 |
| 2 | 1387 | F | 57 |
| 3 | 2304 | F | 16 |
| 4 | 2488 | F | 69 |
| 5 | 4753 | F | 60 |
| 6 | 5438 | F | 42 |
| 7 | 6488 | F | 59 |
| 8 | 9012 | F | 39 |
| 9 | 9125 | F | 56 |
| 10 | 8045 | M | 40 |
| 11 | 8125 | M | 39 |

**SAS Data Set Cert.Measure**

| | ID | Height | Weight |
|---|---|---|---|
| 1 | 1129 | 61 | 137 |
| 2 | 1387 | 64 | 142 |
| 3 | 2304 | 61 | 102 |
| 4 | 5438 | 62 | 168 |
| 5 | 6488 | 64 | 154 |
| 6 | 8045 | 72 | 200 |
| 7 | 8125 | 70 | 176 |
| 8 | 9012 | 63 | 157 |
| 9 | 9125 | 65 | 148 |

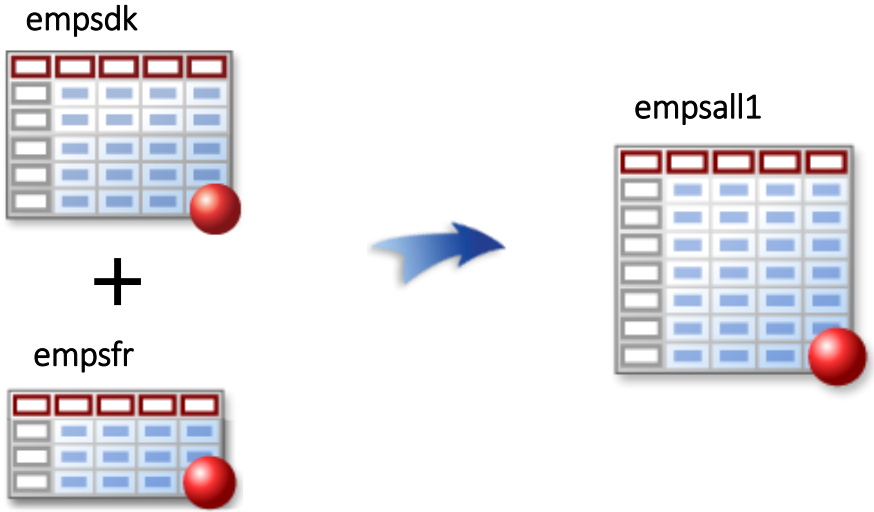
```
data one2one;
    set cert.patients;
    if age < 60 ;
    set cert.measure;
run;
```

# One-to-One Reading

The demonstration illustrates what happens when data sets that are not sorted are combined using one-to-one reading.

# Business Scenario

You were asked to combine the data sets that contain information about Orion Star employees from Denmark and France into a new data set.



# Considerations

Concatenate like-structured data sets, **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 |

**empsall1**

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

Both data sets contain the same variables.

# Using a DATA Step to Concatenate

Use a DATA step to concatenate the data sets.
List the data sets in the SET statement.

```
data empsall1;
   set empsdk empsfr;
run;
```

SET *SAS-data-set1 SAS-data-set2 . . .*;

- The SET statement reads observations from each data set in the order in which they are listed.
- Any number of data sets can be included in the SET statement.

# 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 |

Initialize PDV

```
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 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 |

Implicit OUTPUT;
Implicit RETURN;

```
data empsall1;
   set empsdk empsfr;
run;
```

**PDV**

| First | Gender | Country |
|---|---|---|
| Lars | M | 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;
```

**PDV**

| First | Gender | Country |
|---|---|---|
| Lars | M | Denmark |

**empsall1**

| First | Gender | Country |
|---|---|---|
| Lars | M | Denmark |

Data set variables are not reinitialized.

# 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 |

**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 empsall1;
    set empsdk empsfr;
run;
```

**PDV**

| First | Gender | Country |
|---|---|---|
| Kari | F | Denmark |

**empsall1**

| First | Gender | Country |
|---|---|---|
| Lars | M | Denmark |
| Kari | F | Denmark |

PDV is not reinitialized.

# 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 empsall1;
   set empsdk empsfr;
run;
```

Implicit OUTPUT;
Implicit RETURN;

**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 empsdk empsfr;
run;
```

**PDV**

| First | Gender | Country |
|---|---|---|
| Jonas | M | Denmark |

PDV is not reinitialized.

**empsall1**

| 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 empsdk empsfr;
run;
```

Reinitialize PDV

**PDV**

| First | Gender | Country |
|---|---|---|
| | | |

PDV is reinitialized before processing the next data set.

**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 empsall1;
   set empsdk empsfr;
run;
```

Implicit OUTPUT;
Implicit RETURN;

**PDV**

| First | Gender | Country |
|---|---|---|
| Pierre | M | 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;
```

**PDV**

| First | Gender | Country |
|---|---|---|
| Pierre | M | France |

PDV is not reinitialized.

**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;
```

**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 |

**empsall1**

| 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**

| First | Gender | Country |
|---|---|---|
| Pierre | M | France |
| Sophie | F | France |

```
data empsall1;
    set empsdk empsfr;
run;
```

**PDV**

| First | Gender | Country |
|---|---|---|
| Sophie | F | France |

PDV is not reinitialized.

**empsall1**

| 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**

| First | Gender | Country |
|---|---|---|
| Pierre | M | France |
| Sophie | F | France |

EOF

```
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 |

# Viewing the Log

Partial SAS Log

```
145  data empsall1;
146     set empsdk empsfr;
147  run;

NOTE: There were 3 observations read from the data set WORK.EMPSDK.
NOTE: There were 2 observations read from the data set WORK.EMPSFR.
NOTE: The data set WORK.EMPSALL1 has 5 observations and 3
variables.
```

# 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;
```

# Short Answer Poll

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;
```

# Short Answer Poll – 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 |

- **Region** has missing values due to PDV initialization.
- **Country** has missing values due to PDV reinitialization before processing the second data set.

# Business Scenario

Rename variables in one or more data sets to align columns.

**empscn**

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

**empsjp**

| First | Gender | Region |
|---|---|---|
| Cho | F | Japan |
| Tomi | M | Japan |

Rename **Region** to **Country**.

# RENAME= Data Set Option(Review)

The *RENAME= data set option* changes the name of a variable.

```
data empsall2;
   set empscn empsjp(rename=(Region=Country));
run;
```

*SAS-data-set* (RENAME=(*old-name-1*=*new-name-1*
*old-name-2*=*new-name-2*
...
*old-name-n*=*new-name-n*))

- The RENAME= option must be specified in parentheses immediately after the appropriate SAS data set name.
- The name change affects the PDV and the output data set. It has no effect on the input data set.

# RENAME= Data Set Option

Multiples variables can be renamed in one or more data sets.

```
set empscn(rename=(Country=Region))
    empsjp;
```

```
set empscn(rename=(First=Fname
                   Country=Region))
    empsjp(rename=(First=Fname));
```

```
set empscn
    empsjp(rename=(Region=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 |
|---|---|---|
| | | |

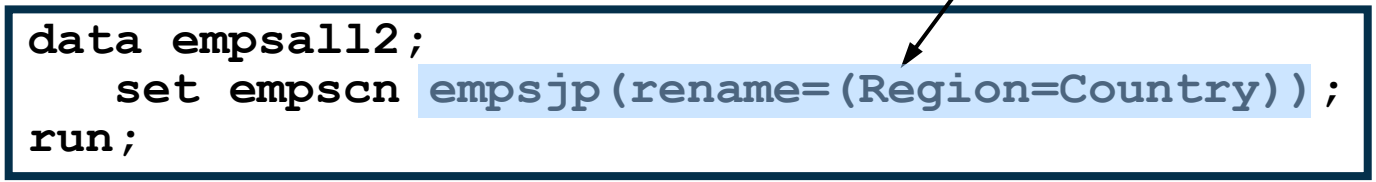
# Compilation

**empscn**

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

**empsjp**

| First | Gender | Region |
|---|---|---|
| Cho | F | Japan |
| Tomi | M | Japan |




**PDV**

| First | Gender | Country |
|---|---|---|
| | | |

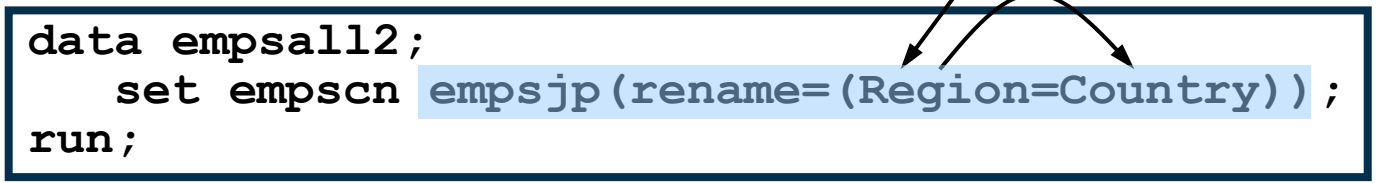
# Compilation

**empscn**

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

**empsjp**

| First | Gender | Region |
|---|---|---|
| Cho | F | Japan |
| Tomi | M | Japan |




**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

The **Region** values are stored in **Country**.

**empsall2**

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

# Combining Data Sets

Basic Match Merging

# Match-Merging: Sorting the Data Sets

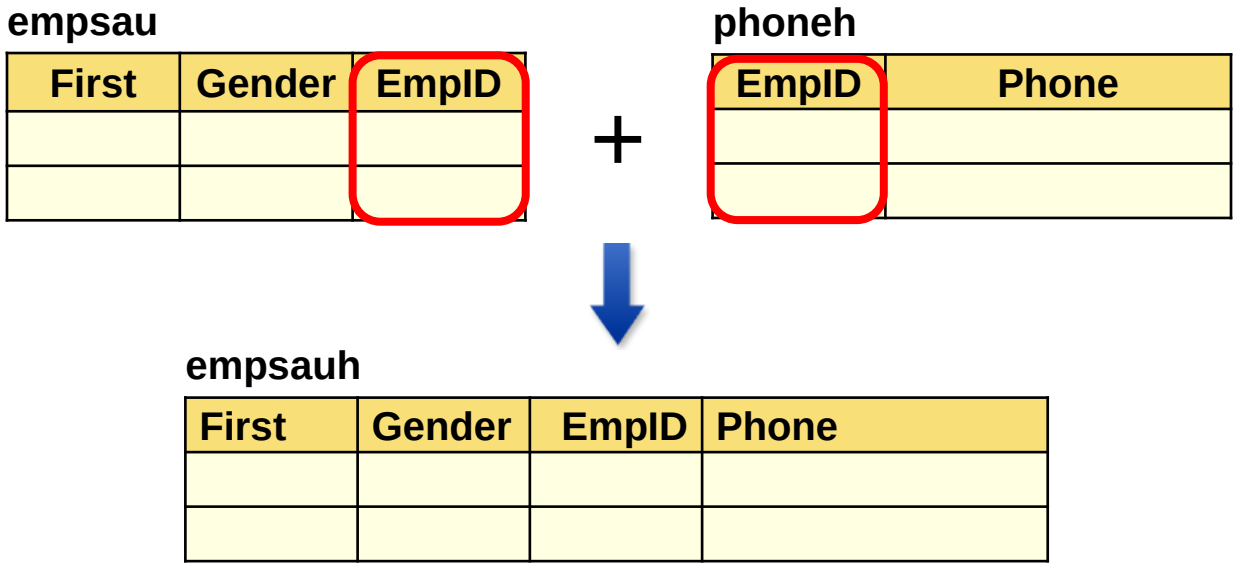
```
PROC SORT DATA=input-table1 <OUT=sorted-output-table1>;
      BY <DESCENDING> col-name(s);
RUN;

PROC SORT DATA=input-table2 <OUT=sorted-output-table2>;
      BY <DESCENDING> col-name(s);
RUN;

DATA combined-data-table;
      MERGE sorted-output-table1 sorted-output-table2;
      BY <DESCENDING> col-name(s);
RUN;
```

Tables must have 1 or more variables with same name, type, and order

Uses MERGE instead of SET to combine tables

# Business Scenario

Merge the Australian employee data set with a phone data set to obtain each employee's home phone number. Store the results in a new data set.



# Match-Merging

The *MERGE statement* in a DATA step joins observations from two or more SAS data sets into single observations.

```
data empsauh;
    merge empsau phoneh;
    by EmpID;
run;
```

**MERGE** *SAS-data-set1 SAS-data-set2 . . .*;
**BY** <DESCENDING> *BY-variable(s)*;

A *BY statement* indicates a match-merge and lists the variable or variables to match.

# MERGE and BY Statements

Requirements for match-merging:

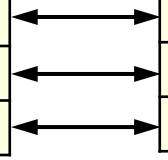
- Two or more data sets are listed in the MERGE statement.
- The variables in the BY statement must be common to all data sets.
- The data sets must be sorted by the variables listed in the BY statement.

# One-to-One Merge

One observation in **empsau** matches exactly one observation in **phoneh**.

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |



**phoneh**

| EmpID | Phone |
|---|---|
| 121150 | +61(2)5555-1793 |
| 121151 | +61(2)5555-1849 |
| 121152 | +61(2)5555-1665 |

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

# Final Results

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phoneh**

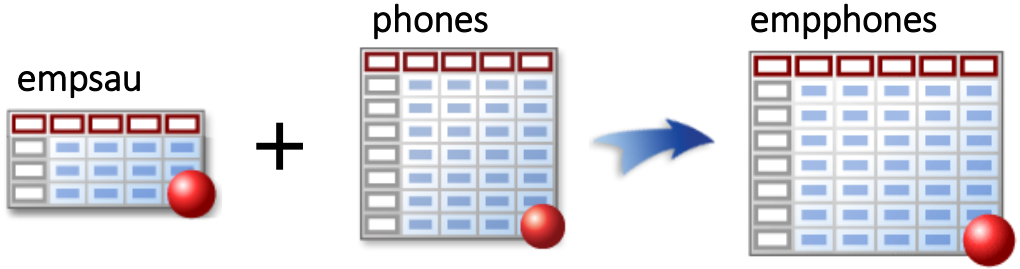
| EmpID | Phone |
|---|---|
| 121150 | +61(2)5555-1793 |
| 121151 | +61(2)5555-1849 |
| 121152 | +61(2)5555-1665 |

**empsauh**

| First | Gender | EmpID | Phone |
|---|---|---|---|
| Togar | M | 121150 | +61(2)5555-1793 |
| Kylie | F | 121151 | +61(2)5555-1849 |
| Birin | M | 121152 | +61(2)5555-1665 |

# Business Scenario

Merge the Australian employee information data set with the **phones** data set to obtain the phone numbers for each employee.



# Considerations

In this one-to-many merge, one observation in **empsau** matches one or more observations in **phones**.

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

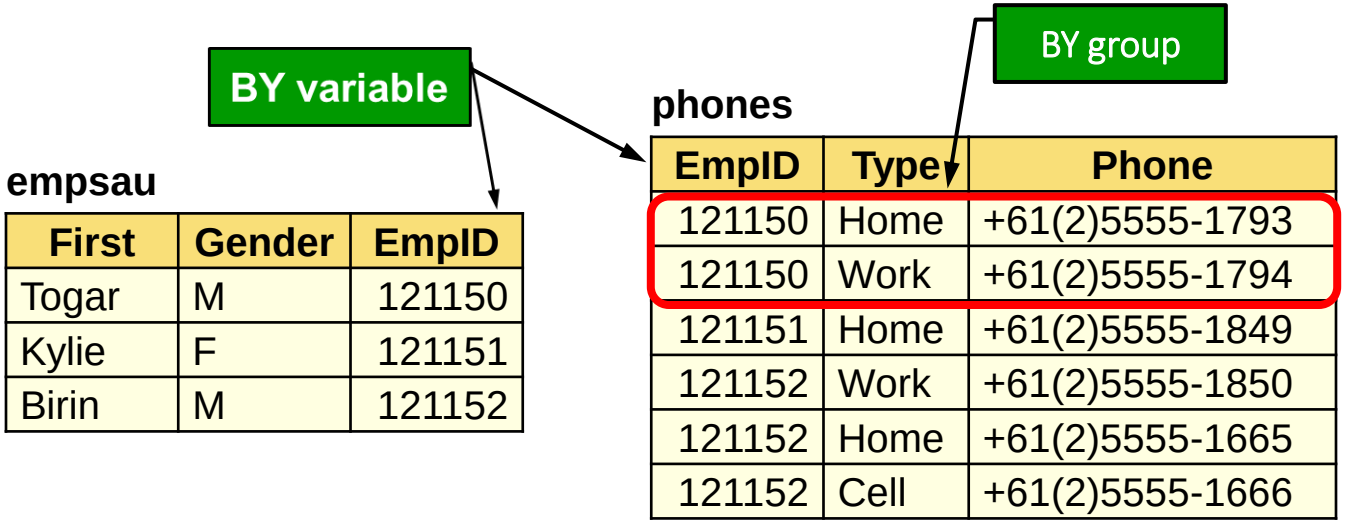
The data sets are sorted by **EmpID**.

# Match-Merging

Merge the two data sets by **EmpID** and create a new data set named **empphones**.

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

# Match-Merging




- The common variable is called the *BY variable*.
- The value of the BY variable is the *BY value*.
- A group of observations with the same BY value is a *BY group*.

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

Initialize PDV

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| | | . | | |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

Do the **EmpID** values match?

Yes

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| | | . | | |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

Reads both observations into the PDV.

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Togar | M | 121150 | Home | +61(2)5555-1793 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Togar | M | 121150 | Home | +61(2)5555-1793 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

Data set variables are not reinitialized.

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Togar | M | 121150 | Home | +61(2)5555-1793 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

Do the **EmpID** values match?

No

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Togar | M | 121150 | Home | +61(2)5555-1793 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

Does either **EmpID** match PDV?

Yes

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Togar | M | 121150 | Home | +61(2)5555-1793 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

Read the matching observation into the PDV.

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Togar | M | 121150 | Work | +61(2)5555-1794 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Togar | M | 121150 | Work | +61(2)5555-1794 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

Do the **EmpID** values match?

Yes

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Togar | M | 121150 | Work | +61(2)5555-1794 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

Did **EmpID** change?

Yes

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Togar | M | 121150 | Work | +61(2)5555-1794 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

Read both observations into the PDV.

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Kylie | F | 121151 | Home | +61(2)5555-1849 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Kylie | F | 121151 | Home | +61(2)5555-1849 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

Do the **EmpID** values match?

Yes

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Kylie | F | 121151 | Home | +61(2)5555-1849 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

Read both observations into the PDV.

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Birin | M | 121152 | Work | +61(2)5555-1850 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Birin | M | 121152 | Work | +61(2)5555-1850 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

EOF

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

Did **EmpID** change?

No

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Birin | M | 121152 | Work | +61(2)5555-1850 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

EOF

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

Read the matching observation into the PDV.

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Birin | M | 121152 | Home | +61(2)5555-1665 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

EOF

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Birin | M | 121152 | Home | +61(2)5555-1665 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

EOF

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

Did **EmpID** change?

No

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Birin | M | 121152 | Home | +61(2)5555-1665 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

EOF

```
data empphones;
    merge empsau phones;
    by EmpID;
run;
```

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

Reads the matching observation into the PDV.

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Birin | M | 121152 | Cell | +61(2)5555-1666 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

EOF

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Birin | M | 121152 | Cell | +61(2)5555-1666 |

# Execution

**empsau**

| First | Gender | EmpID |
|---|---|---|
| Togar | M | 121150 |
| Kylie | F | 121151 |
| Birin | M | 121152 |

EOF

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |

EOF

```
data empphones;
   merge empsau phones;
   by EmpID;
run;
```

**PDV**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Birin | M | 121152 | Cell | +61(2)5555-1666 |

# Final Results

**empphones**

| First | Gender | EmpID | Type | Phone |
|---|---|---|---|---|
| Togar | M | 121150 | Home | +61(2)5555-1793 |
| Togar | M | 121150 | Work | +61(2)5555-1794 |
| Kylie | F | 121151 | Home | +61(2)5555-1849 |
| Birin | M | 121152 | Work | +61(2)5555-1850 |
| Birin | M | 121152 | Home | +61(2)5555-1665 |
| Birin | M | 121152 | Cell | +61(2)5555-1666 |

```
NOTE: There were 3 observations read from the data set WORK.EMPSAU.
NOTE: There were 6 observations read from the data set WORK.PHONES.
NOTE: The data set WORK.EMPPHONES has 6 observations and 5
variables.
```

# Discussion

In a one-to-many merge, does it matter which data set is listed first in the MERGE statement?

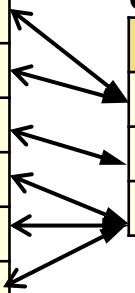
- Reverse the order of the data sets on the MERGE statement and submit the data step.
- Observe the results. How are they different?

# Many-to-One Merge

One or more rows in one data set match exactly one row in the other data set.

**phones**

| EmpID | Type | Phone |
|---|---|---|
| 121150 | Home | +61(2)5555-1793 |
| 121150 | Work | +61(2)5555-1794 |
| 121151 | Home | +61(2)5555-1849 |
| 121152 | Work | +61(2)5555-1850 |
| 121152 | Home | +61(2)5555-1665 |
| 121152 | Cell | +61(2)5555-1666 |



**empsau**

| EmpID | First | Gender |
|---|---|---|
| 121150 | Togar | M |
| 121151 | Kylie | F |
| 121512 | Birin | M |

```
data phones;
   merge phones empsau;
   by EmpID;
run;
```

# Viewing the Output

PROC PRINT Output

| Obs | EmpID | Type | Phone | First | Gender |
|---|---|---|---|---|---|
| 1 | 121150 | Home | +61(2)5555-1793 | Togar | M |
| 2 | 121150 | Work | +61(2)5555-1794 | Togar | M |
| 3 | 121151 | Home | +61(2)5555-1849 | Kylie | F |
| 4 | 121152 | Work | +61(2)5555-1850 | Birin | M |
| 5 | 121152 | Home | +61(2)5555-1665 | Birin | M |
| 6 | 121152 | Cell | +61(2)5555-1666 | Birin | M |

- The results are the same as the one-to-many merge.
- The order of variables is different.

# Lesson Quiz

10. What are the values for **First.City** and **Last.City** for the third row of the input table given the following information?

```
data StatePopulation;
    set Population;
    by State City;
run;
```

| State | City | Population |
|---|---|---|
| NC | Cary | 162320 |
| NC | Durham | 263016 |
| NC | Greenville | 91495 |
| SC | Greenville | 67453 |
| SC | Sumter | 40723 |

a. **First.City**=0 and **Last.City**=0

b. **First.City**=1 and **Last.City**=0

c. **First.City**=0 and **Last.City**=1

d. **First.City**=1 and **Last.City**=1

10. What are the values for **First.City** and **Last.City** for the third row of the input table given the following information?

```
data StatePopulation;
    set Population;
    by State City;
run;
```

| State | City | Population |
|---|---|---|
| NC | Cary | 162320 |
| NC | Durham | 263016 |
| NC | Greenville | 91495 |
| SC | Greenville | 67453 |
| SC | Sumter | 40723 |

a. **First.City**=0 and **Last.City**=0

b. **First.City**=1 and **Last.City**=0

c. **First.City**=0 and **Last.City**=1

d. **First.City**=1 and **Last.City**=1