STAT604 SAS Lesson 13

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;
```

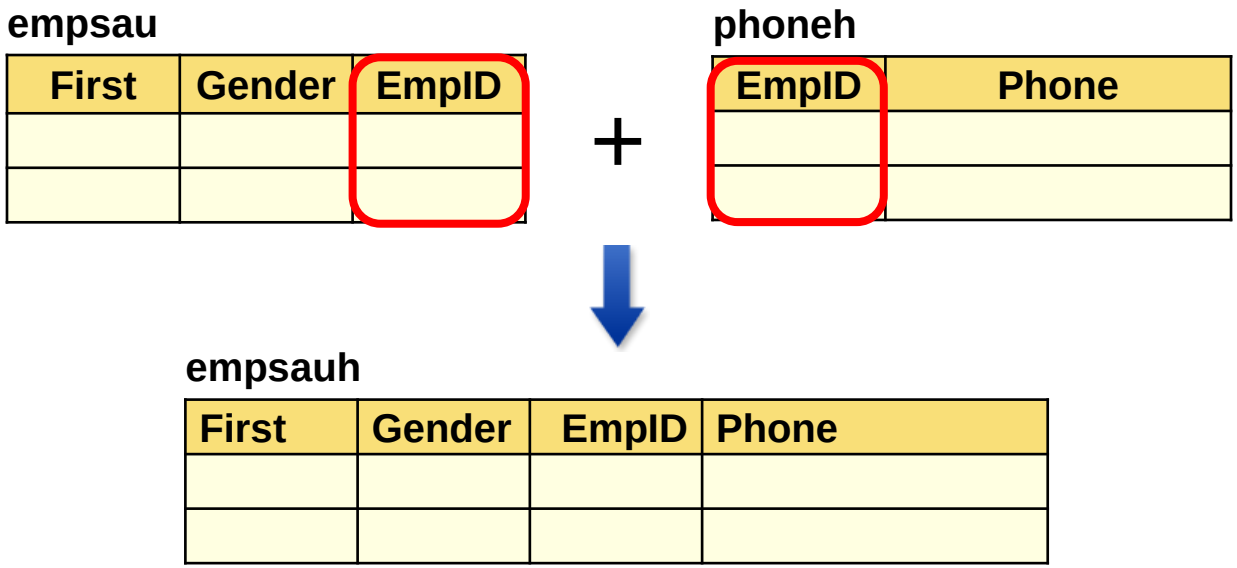
```
DATA combined-data-table;
      MERGE sorted-output-table1 sorted-output-table2;
      BY <DESCENDING> col-name(s);
RUN;
```

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:

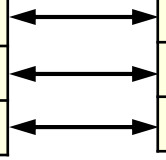
- Two or more data sets are listed in the MERGE statement.
- The variables in the BY statement must be common to all data sets.
- The data sets must be sorted by the variables listed in the BY statement.

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

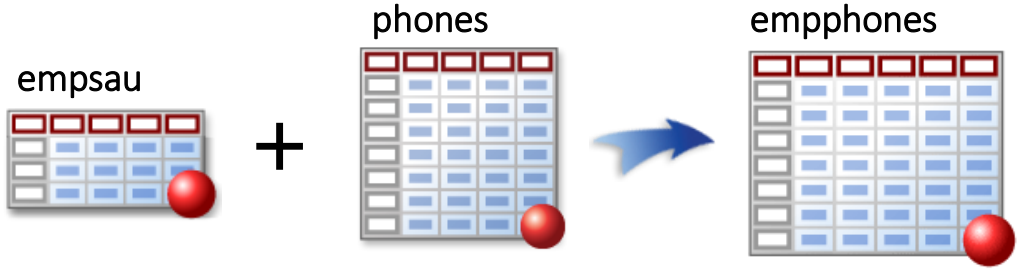
EmpID	Phone
121150	+61(2)5555-1793
121151	+61(2)5555-1849
121152	+61(2)5555-1665

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

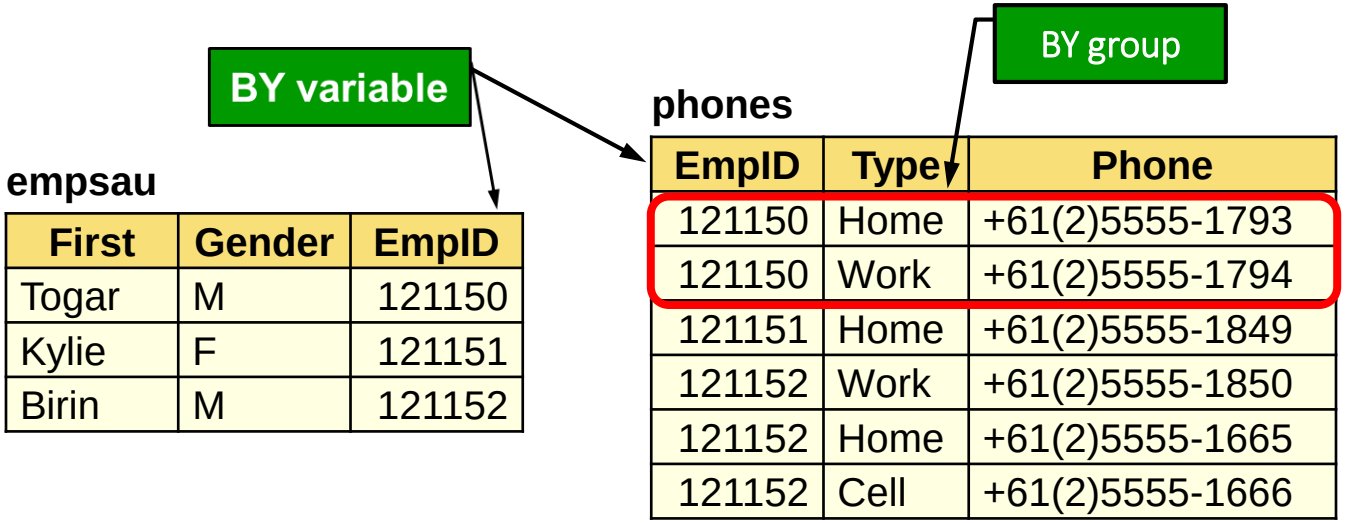
The data sets are sorted by **EmpID**.

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?

Only changes the order of the columns.

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

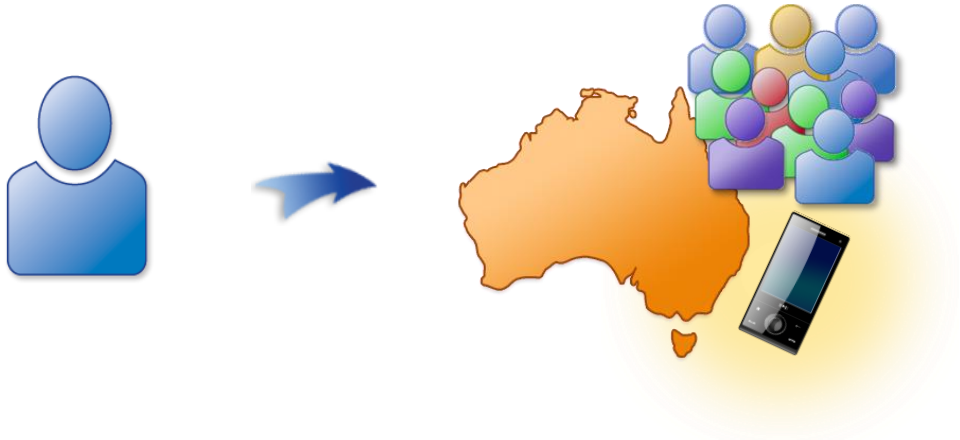
- The results are the same as the one-to-many merge.
- The order of variables is different.

Combining Data Sets

Merging Data Sets with Non-Matches

Business Scenario

A manager in Australia requested
an inventory of company phone numbers.



Merge with Nonmatches

There are observations in **empsau** that do not have a match in **phonec**, and some in **phonec** that do not match any observations in **empsau**.

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

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

Match-Merging

Merge **empsau** and **phonec** by **EmpID** to create a new data set named **empsauc**.

```
data empsauc;
   merge empsau phonec;
   by EmpID;
run;
```

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

Initialize PDV

```
data empsauc;
   merge empsau phonec;
   by EmpID;
run;
```

PDV

First	Gender	EmpID	Phone
		.	

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Do the EmpID values match?

Yes

PDV

First	Gender	EmpID	Phone
		.	

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Reads both observations into the PDV.

PDV

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

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
   merge empsau phonec;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

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

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Do the EmpID values match?

No

PDV

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

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Does either **EmpID** match PDV?

No

PDV

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

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

Reinitialize PDV

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

reinitialize b/c flag don't match

PDV

First	Gender	EmpID	Phone
		.	

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Which **EmpID** value sequentially comes first?

121151

PDV

First	Gender	EmpID	Phone
		.	

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Reads that observation into the PDV.

PDV

First	Gender	EmpID	Phone
Kylie	F	121151	

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
   merge empsau phonec;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

First	Gender	EmpID	Phone
Kylie	F	121151	

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Do the EmpID values match?

Yes

PDV

First	Gender	EmpID	Phone
Kylie	F	121151	

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Does **EmpID** match PDV?

No

Not an issue b/c val already matched

PDV

First	Gender	EmpID	Phone
Kylie	F	121151	

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

Reinitialize PDV

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

PDV

First	Gender	EmpID	Phone
		.	

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Reads both observations into the PDV.

PDV

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

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
   merge empsau phonec;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

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

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

EOF

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Does **EmpID** match PDV?

No

PDV

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

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

EOF

Reinitialize PDV

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

PDV

First	Gender	EmpID	Phone
		.	

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

EOF

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

Reads the observation into the PDV.

PDV

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

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

EOF

```
data empsauc;
   merge empsau phonec;
   by EmpID;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

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

Execution

empsau

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

EOF

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

EOF

```
data empsauc;
    merge empsau phonec;
    by EmpID;
run;
```

PDV

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

Final Results

empsauc

First	Gender	EmpID	Phone
Togar	M	121150	+61(2)5555-1795
Kylie	F	121151	
Birin	M	121152	+61(2)5555-1667
		121153	+61(2)5555-1348

The final results include both matches and nonmatches.

Short Answer Poll

Consider the data set **empsauc** created by the program in the previous example. Which input data sets contributed information to the last observation?

a. empsau
b. phonec
c. both empsau and phonec
d. There is insufficient information.

empsauc

First	Gender	EmpID	Phone
Togar	M	121150	+61(2)5555-1795
Kylie	F	121151	
Birin	M	121152	+61(2)5555-1667
		121153	+61(2)5555-1348

Short Answer Poll – Correct Answer

Consider the data set **empsauc** created by the program in the previous example. Which input data sets contributed information to the last observation?

a. empsau
b. phonec
c. both empsau and phonec
d. There is insufficient information.

empsauc

First	Gender	EmpID	Phone
Togar	M	121150	+61(2)5555-1795
Kylie	F	121151	
Birin	M	121152	+61(2)5555-1667
		121153	+61(2)5555-1348

Business Scenario

The manager has now requested three phone inventory reports.

- employees with company phones
- employees without company phones
- phones with an invalid employee ID

IN= Data Set Option

The *IN= data set option* creates a variable that indicates whether the data set contributed to building the current observation.

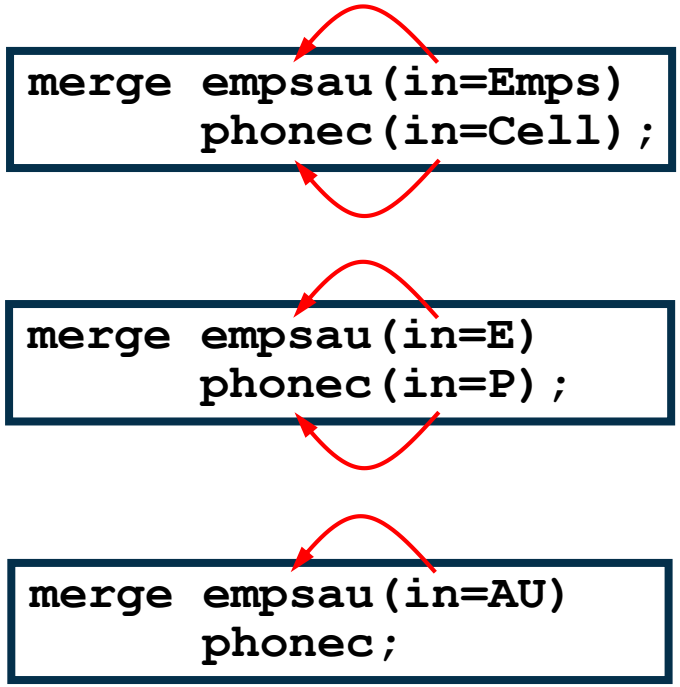
MERGE *SAS-data-set* **(IN=***variable***) …**

variable is a temporary numeric variable that has two possible values:

0	Indicates that the data set did ***not*** contribute to the current observation.
1	Indicates that the data set ***did*** contribute to the current observation.

IN= Data Set Option

MERGE statement examples:



Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
   merge empsau(in=Emps)
         phonec(in=Cell);
   by EmpID;
run;
```

match

PDV

First	Gender	EmpID	D Emps	Phone	D Cell
Togar	M	121150	1	+61(2)5555-1795	1

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
   merge empsau(in=Emps)
         phonec(in=Cell);
   by EmpID;
run;
```

nonmatch

PDV

First	Gender	EmpID	D Emps	Phone	D Cell
Kylie	F	121151	1		0

Execution

empsau

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

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
   merge empsau(in=Emps)
         phonec(in=Cell);
   by EmpID;
run;
```

match

PDV

First	Gender	EmpID	D Emps	Phone	D Cell
Birin	M	121152	1	+61(2)5555-1667	1

Short Answer Poll

What are the values of **Emps** and **Cell**?

empsau

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

EOF

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
   merge empsau(in=Emps)
         phonec(in=Cell);
   by EmpID;
run;
```

PDV

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

Short Answer Poll – Correct Answer

What are the values of **Emps** and **Cell**?

empsau

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

EOF

phonec

EmpID	Phone
121150	+61(2)5555-1795
121152	+61(2)5555-1667
121153	+61(2)5555-1348

```
data empsauc;
   merge empsau(in=Emps)
         phonec(in=Cell);
   by EmpID;
run;
```

nonmatch

PDV

First	Gender	EmpID	Emps	Phone	Cell
		121153	0	+61(2)5555-1348	1

PDV Results

PDV

First	Gender	EmpID	D Emps	Phone	D Cell
Togar	M	121150	1	+61(2)5555-1795	1
Kylie	F	121151	1		0
Birin	M	121152	1	+61(2)5555-1667	1
		121153	0	+61(2)5555-1348	1

The variables created with the IN= data set option are available only during DATA step execution.

- They are not written to the SAS data set.
- Their value can be tested using conditional logic.

Matches Only

Add a subsetting IF statement to select the employees that have company phones.

```
data empsauc;
   merge empsau(in=Emps)
         phonec(in=Cell);
   by EmpID;
   if Emps=1 and Cell=1;
run;
```

empsauc

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

Nonmatches from empsau

Select the employees that do not have company phones.

```
data empsauc;
   merge empsau(in=Emps)
         phonec(in=Cell);
   by EmpID;
   if Emps=1 and Cell=0;
run;
```

empsauc

First	Gender	EmpID	Phone
Kylie	F	121151	

Nonmatches from phonec

Select the phones associated with an invalid employee ID.

```
data empsauc;
   merge empsau(in=Emps)
         phonec(in=Cell);
   by EmpID;
   if Emps=0 and Cell=1;
run;
```

empsauc

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

All Nonmatches

```
data empsauc;
   merge empsau(in=Emps)
         phonec(in=Cell);
   by EmpID;
   if Emps=0 or Cell=0;
run;
```

empsauc

First	Gender	EmpID	Phone
Kylie	F	121151	
		121153	+61(2)5555-1348

Use the OR operator, not the AND operator.

Discussion

How could we make the creation of the three data sets more efficient?

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;
```

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;
```

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

Alternate Syntax

When checking a variable for a value of *1* or *0* as in the previous scenario, you can use the following syntax:

Instead of	You can use
if Emps=1 and Cell=1;	if Emps and Cell;
if Emps=1 and Cell=0;	if Emps and not Cell;
if Emps=0 and Cell=1;	if not Emps and Cell;
if Emps=0 or Cell=0;	If not Emps or not Cell;

Alternate Syntax

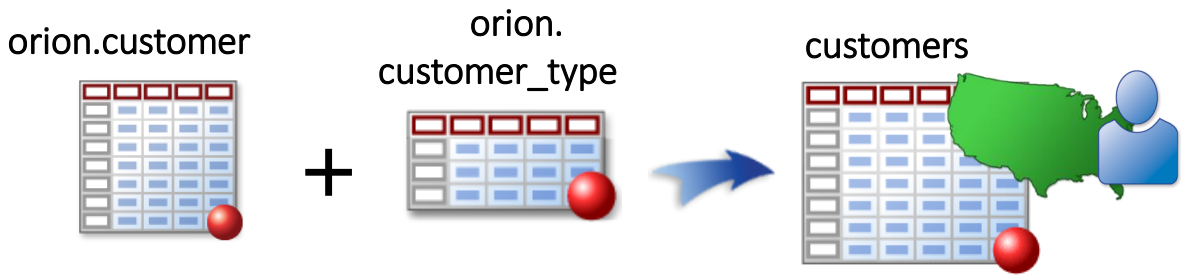
Both programs create a report of employees without cell phones.

```
data empsphone;
   merge empsact(in=inEmps)
         phoneact(in=inCell);
   by EmpID;
   if inEmps=1 and inCell=0;
run;
```

```
data empsphone;
   merge empsact(in=inEmps)
         phoneact(in=inCell);
   by EmpID;
   if inEmps and not inCell;
run;
```

Business Scenario

Merge customer information with customer type data to obtain a customer description. The new data set should include only US customers.



Considerations

The **orion.customer** data set is not sorted by **Customer_Type_ID**, the common variable. The subsetting variable, **Country**, is defined in only one data set.

orion.customer

Customer_ ID	Country	Customer_ Name	…	Birth_ Date	Customer_Type_ ID

orion.customer_type

Customer_ Group	Customer_Group_ ID	Customer_ Type	Customer_Type_ ID

Short Answer Poll

What change is needed to correct the error?

```
proc sort data=orion.customer
          out=cust_by_type;
   by Customer_Type_ID;
run;

data customers;
   merge cust_by_type orion.customer_type;
   by Customer_Type_ID;
   where Country='US';
run;
```

Short Answer Poll – Correct Answer

What change is needed to correct the error?

```
397  proc sort data=orion.customer
398            out=cust_by_type;
399     by Customer_Type_ID;
400  run;
NOTE: There were 77 observations read from the data set ORION.CUSTOMER.
NOTE: The data set WORK.CUST_BY_TYPE has 77 observations and 12 variables.

401
402  data customers;
403     merge cust_by_type orion.customer_type;
404     by Customer_Type_ID;
405     where Country='US';
ERROR: Variable Country is not on file ORION.CUSTOMER_TYPE.
406  run;

NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.CUSTOMERS may be incomplete.  When this step
was stopped there were 0 observations and 15 variables.
```

Country is not defined in both data sets. Replace the WHERE statement with a subsetting IF statement.

Subsetting IF

Use a subsetting IF statement when the subsetting variable is not in all data sets that are named in the MERGE statement.

```
proc sort data=orion.customer
          out=cust_by_type;
   by Customer_Type_ID;
run;

data customers;
   merge cust_by_type  orion.customer_type;
   by Customer_Type_ID;
   if Country='US';
run;
```

Viewing the Output

Partial SAS Log

```
407  proc sort data=orion.customer
408            out=cust_by_type;
409     by Customer_Type_ID;
410  run;
NOTE: There were 77 observations read from the data set ORION.CUSTOMER.
NOTE: The data set WORK.CUST_BY_TYPE has 77 observations and 12 variables.

411
412  data customers;
413     merge cust_by_type orion.customer_type;
414     by Customer_Type_ID;
415     if Country='US';
416  run;

NOTE: There were 77 observations read from the data set WORK.CUST_BY_TYPE.
NOTE: There were 8 observations read from the data set ORION.CUSTOMER_TYPE.
NOTE: The data set WORK.CUSTOMERS has 28 observations and 15 variables.
```

WHERE versus Subsetting IF Statement

Step and Usage	WHERE	IF
PROC step	Yes	No
DATA step (source of variable)		
SET statement	Yes	Yes
assignment statement	No	Yes
INPUT statement	No	Yes
SET/MERGE statement (multiple data sets)		
Variable in ALL data sets	Yes	Yes
Variable not in ALL data sets	No	Yes

Combining Data Sets

Merging Data Sets with Many-to-Many Matches

Many-to-Many Merge

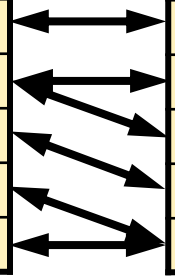
Merge **EmpsAUUS** and **PhoneO** by **Country** to create a new data set named **EmpsOfc**.

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**.

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

Merging Many-to-Many

This demonstration compares a many to many merge with a SQL join.

Many-to-Many Merge

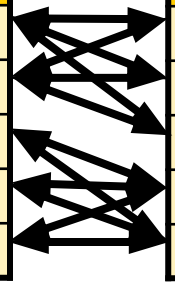
The SQL procedure creates different results than the DATA step for a many-to-many merge.

EmpsAUUS

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

PhoneO

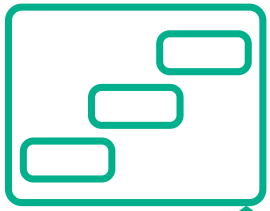
Country	Phone
AU	+61(2)5555-1500
AU	+61(2)5555-1600
AU	+61(2)5555-1700
US	+1(305)555-1500
US	+1(305)555-1600



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

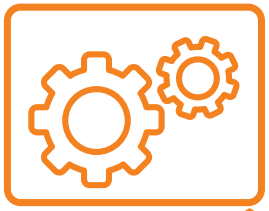
DATA Step Merge and PROC SQL Join

DATA step merge



- requires sorted input data
- efficient, sequential processing
- can create multiple tables for matches and nonmatches in one step
- provides additional complex data processing syntax

PROC SQL join



- does not require sorted data
- matching columns do not need the same name
- easy to define complex matching criteria between multiple tables in a single query
- can be used to create a Cartesian product for many-to-many joins

Combining Data Sets

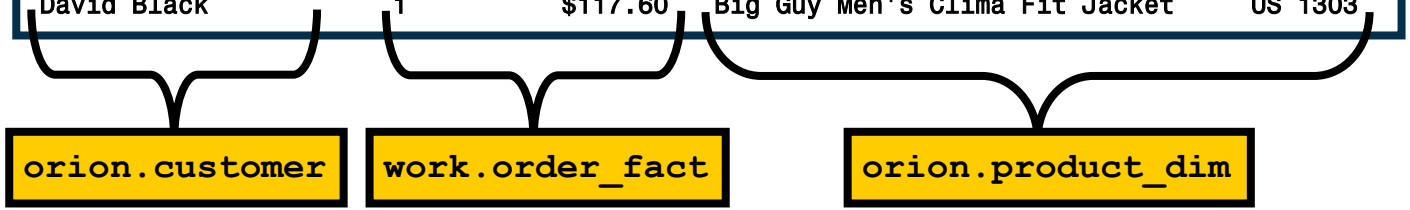
Using Data Manipulation Techniques with Match-Merging

Multiple Data Sets without a Common Variable

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

Partial PROC PRINT Output

Customer_Name	Quantity	Total_Retail_ 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



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
...

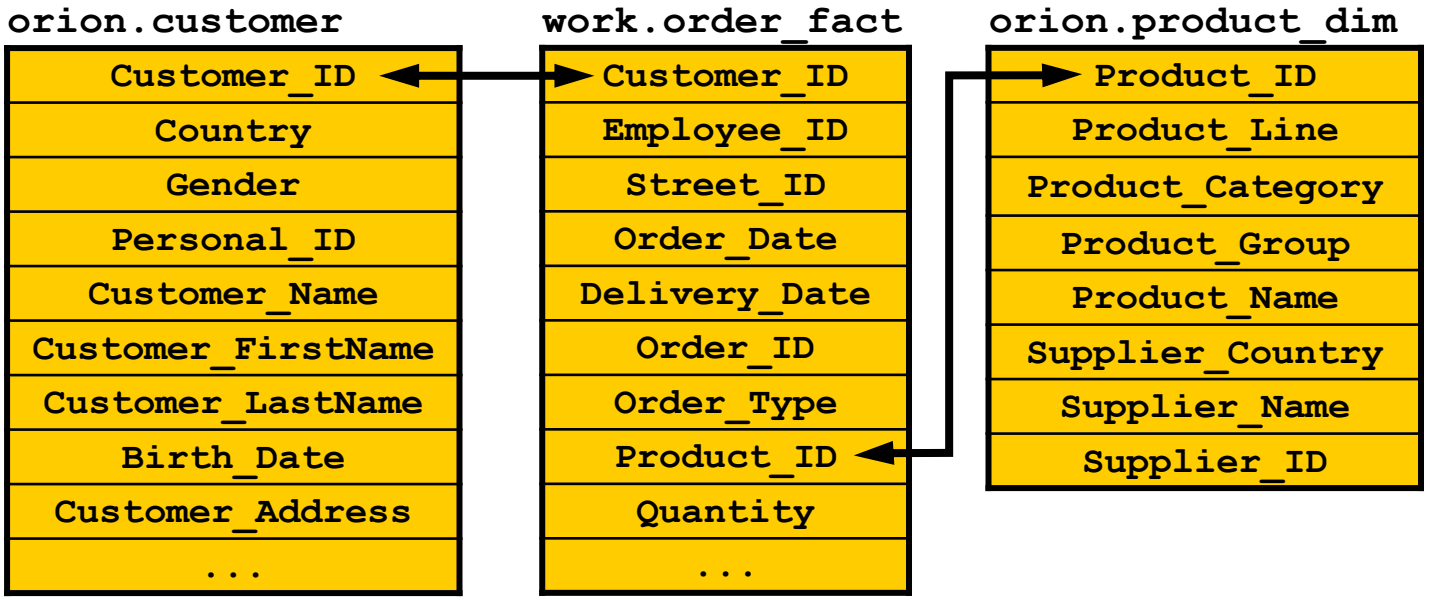
orion.product_dim

Product_ID
Product_Line
Product_Category
Product_Group
Product_Name
Supplier_Country
Supplier_Name
Supplier_ID

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.



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

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

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.xlsx** 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.xlsx** to determine the customers that will be sent gifts.

work.CustOrdProd

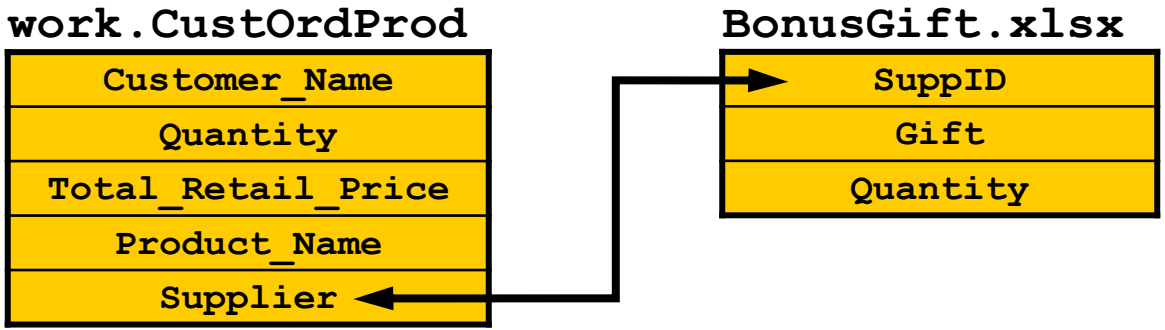
Customer_Name
Quantity
Total_Retail_Price
Product_Name
Supplier

BonusGift.xlsx

SuppID
Gift
Quantity

Business Scenario – Details

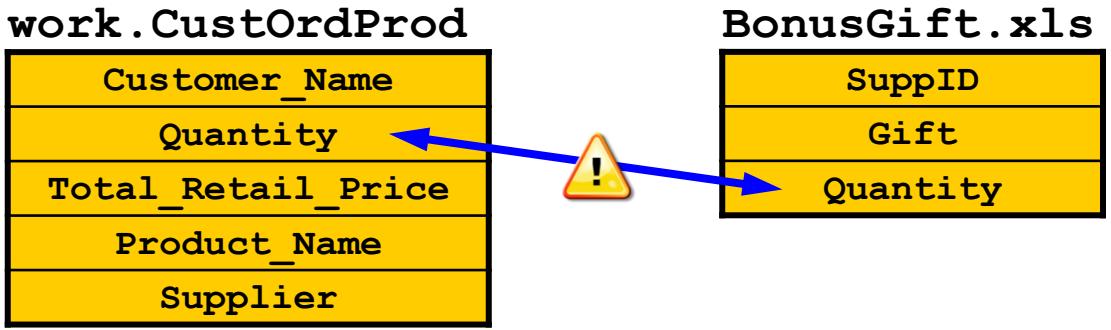
The data sets **work.CustOrdProd** and **BonusGift.xlsx** must be merged on values that are in two differently named variables.



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**.



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

Create Gift List – Solution

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

```
libname bonus xlsx 'C:\Sample Data\BonusGift.xlsx';

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

libname bonus clear;
```

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 xlsx 'C:\Sample Data\BonusGift.xlsx';

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

libname bonus clear;
```

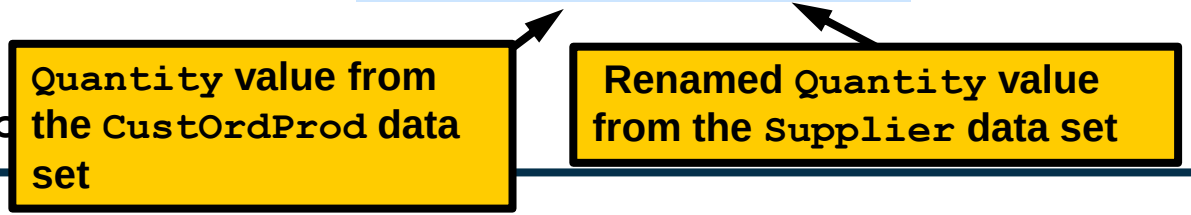
Create Gift List – Solution

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

```
libname bonus xlsx 'C:\Sample Data\BonusGift.xlsx';

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

libname bo
```



Create Gift List – Solution

Fifty-two gifts will be sent to customers.

Partial SAS log

```
208
209  data CustOrdProdGift;
210     merge CustOrdProd(in=c)
211           bonus.Supplier(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.
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;
```

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

Merging Multiple Data Sets

The DATA statement can merge multiple input data sets as long as they all have a common variable.

payroll06

Obs	Employee_ID	Employee_Gender	Salary	Birth_Date	Employee_Hire_Date	Employee_Term_Date	Marital_Status	Dependents
1	120101	M	163040	6074	01JUL2003	.	S	0
2	120102	M	108255	3510	01JUN1989	.	O	2
3	120103	M	87975	-3996	01JAN1974	.	M	1
4	120104	F	46230	-2061	01JAN1981	.	M	1
5	120105	F	27110	5468	01MAY1999	.	S	0

payroll07

Obs	Employee_ID	Employee_Gender	Salary	Birth_Date	Employee_Hire_Date	Employee_Term_Date	Marital_Status	Dependents
1	120101	M	167931	6074	01JUL2003	.	S	0
2	120102	M	111503	3510	01JUN1989	.	O	2
3	120103	M	90614	-3996	01JAN1974	.	M	1
4	120104	F	47617	-2061	01JAN1981	.	M	1
5	120105	F	27923	5468	01MAY1999	.	S	0

payroll08

Obs	Employee_ID	Employee_Gender	Salary	Birth_Date	Employee_Hire_Date	Employee_Term_Date	Marital_Status	Dependents
1	120101	M	172969	6074	01JUL2003	.		.
2	120102	M	114848	3510	01JUN1989	.		.
3	120103	M	93332	-3996	01JAN1974	.		.
4	120104	F	49046	-2061	01JAN1981	.		.
5	120105	F	28761	5468	01MAY1999	.	S	0

Merging Multiple Data Sets

Note: The SAS log is extremely important in troubleshooting merges.

```
data payroll_hist;
    merge mylib.payroll06
          mylib.payroll07
          mylib.payroll08(drop=marital_status);
    by employee_id;
run;
```

Common variables are overwritten from the right. Use UPDATE instead of MERGE to prevent missing data from overwriting existing data.

Merging Multiple Data Sets

This demonstration shows how data can be overwritten in a merge and how the log can provide useful troubleshooting information.

Lesson Quiz

1. Which of the following statements is true about merging SAS data sets by using the DATA step?
    a. Merging combines observations from two or more data sets into a single observation in a new data set.
    b. SAS can merge data sets based on the position of observations in the original data set or by the values of one or more common variables.
    c. Match-merging is merging by values of one or more common variables.
    d. To match-merge data sets, all input data sets must be sorted or indexed on the BY variable or variables.
    e. all of the above

1. Which of the following statements is true about merging SAS data sets by using the DATA step?
   a. Merging combines observations from two or more data sets into a single observation in a new data set.
   b. SAS can merge data sets based on the position of observations in the original data set or by the values of one or more common variables.
   c. Match-merging is merging by values of one or more common variables.
   d. To match-merge data sets, all input data sets must be sorted or indexed on the BY variable or variables.
   e. all of the above

2. Which of the following programs concatenates the data sets **sales** and **products**, in that order?

a.
```
data newsales;
   set products sales;
run;
```

b.

c.
```
data newsales;
   set sales products;
run;
```

```
data newsales;
   set sales;
   set products;
run;
```

2. Which of the following programs concatenates the data sets **sales** and **products**, in that order?

a.
```
data newsales;
   set products sales;
run;
```

b.
```
data newsales;
   set sales products;
run;
```

c.
```
data newsales;
   set sales;
   set products;
run;
```

3. If you run this DATA step, what observations does the data set **bonuses** contain?

```
data bonuses;
   merge managers staff;
   by EmpID;
run;
```

a. all of the observations from **managers**, and only those observations from **staff** with matching values for **EmpID**

b. all of the observations from **staff**, and only those observations from **managers** with matching values for **EmpID**

c. all observations from **staff** and all observations from **managers**, whether or not they have matching values

d. only those observations from **staff** and **managers** with matching values for **EmpID**

3. If you run this DATA step, what observations does the data set **bonuses** contain?

```
data bonuses;
   merge managers staff;
   by EmpID;
run;
```

a. all of the observations from **managers**, and only those observations from **staff** with matching values for **EmpID**

b. all of the observations from **staff**, and only those observations from **managers** with matching values for **EmpID**

c. all observations from **staff** and all observations from **managers**, whether or not they have matching values

d. only those observations from **staff** and **managers** with matching values for **EmpID**

4. If you concatenate the data sets below in the order shown, what is the value of **Sale** in observation 2 of the new data set?

Reps

ID	Name
1	Nay Rong
2	Kelly Windsor
3	Julio Meraz
4	Richard Krabill

Close

ID	Sale
1	$28,000
2	$30,000
2	$40,000
3	$15,000
3	$20,000
3	$25,000
4	$35,000

a. missing
b. $30,000
c. $40,000
d. You cannot concatenate these data sets.

4. If you concatenate the data sets below in the order shown, what is the value of **Sale** in observation 2 of the new data set?

Reps

ID	Name
1	Nay Rong
2	Kelly Windsor
3	Julio Meraz
4	Richard Krabill

Close

ID	Sale
1	$28,000
2	$30,000
2	$40,000
3	$15,000
3	$20,000
3	$25,000
4	$35,000

a. missing
b. $30,000
c. $40,000
d. You cannot concatenate these data sets.

5. What happens if you submit the following program to merge **donors1** and **donors2**, shown below?

```
data merged;
   merge donors1 donors2;
   by ID;
run;
```

donors1

ID	Type	Units
2304	O	16
1129	A	48
1129	A	50
1129	A	57
2486	B	63

donors2

ID	Code	Units
6488	65	27
1129	63	32
5438	62	39
2304	61	45
1387	64	67

a. The **merged** data set contains some missing values because not all observations have matching observations in the other data set.

b. The **merged** data set contains eight observations.

c. The DATA step produces errors.

5. What happens if you submit the following program to merge **donors1** and **donors2**, shown below?

```
data merged;
   merge donors1 donors2;
   by ID;
run;
```

donors1

ID	Type	Units
2304	O	16
1129	A	48
1129	A	50
1129	A	57
2486	B	63

donors2

ID	Code	Units
6488	65	27
1129	63	32
5438	62	39
2304	61	45
1387	64	67

a. The **merged** data set contains some missing values because not all observations have matching observations in the other data set.

b. The **merged** data set contains eight observations.

c. The DATA step produces errors.

6. Suppose you want to concatenate these data sets. Which DATA step creates an output data set that combines the values of **Color** and **Hue** in the single variable **Color**?

a.
```
data widgets_all;
   set widget1(rename=(Color=Hue))
       widget2;
run;
```

b.
```
data widgets_all;
   set widget1
       widget2(rename=(Hue=Color));
run;
```

c.
```
data widgets_all;
   set widget1
       widget2(Hue=Color);
run;
```

Widget1

Tag	Color	Model
77904	blue	AB42
56012	red	BA25
35499	orange	FC36

Widget2

Tag	Hue	Model
89325	red	SP17
65888	yellow	BA12
00167	green	PG20

6. Suppose you want to concatenate these data sets. Which DATA step creates an output data set that combines the values of **Color** and **Hue** in the single variable **Color**?

a.
```
data widgets_all;
   set widget1(rename=(Color=Hue))
       widget2;
run;
```

b.
```
data widgets_all;
   set widget1
       widget2(rename=(Hue=Color));
run;
```

c.
```
data widgets_all;
   set widget1
       widget2(Hue=Color);
run;
```

Widget1

Tag	Color	Model
77904	blue	AB42
56012	red	BA25
35499	orange	FC36

Widget2

Tag	Hue	Model
89325	red	SP17
65888	yellow	BA12
00167	green	PG20

7. What is the syntax error in this DATA step?

```
data returns_qtr1;
   set returns_jan(rename=(ID=CustID)
                          (Return=Item))
       returns_feb(rename=(Dt=Date))
       returns_mar;
run;
```

a. You cannot specify more than two data sets in the SET statement.

b. There are too many sets of parentheses in the RENAME= option.

c. You cannot specify multiple variables in the RENAME= option.

d. The BY statement is missing.

7. What is the syntax error in this DATA step?

```
data returns_qtr1;
   set returns_jan(rename=(ID=CustID)
                   (Return=Item))
       returns_feb(rename=(Dt=Date))
       returns_mar;
run;
```

a. You cannot specify more than two data sets in the SET statement.

b. There are too many sets of parentheses in the RENAME= option.

c. You cannot specify multiple variables in the RENAME= option.

d. The BY statement is missing.

8. In the second iteration of this DATA step, after the data is merged, what are the values of **C** and **A**?

```
data client_amount;
   merge clients(in=C)
         amounts(in=A);
   by Name;
run;
```

Clients

Name	EmpID
Ankerton	11123
Davis	22298
Masters	33351
Wolmer	44483

Amounts

Name	Date	Amt
Ankerton	08OCT96	92
Ankerton	15OCT96	43
Davis	04OCT96	16
Masters	.	27
Thomas	21OCT96	15

a. C=1, A=0
b. C=0, A=1
c. C=1, A=1
d. missing
e. unknown

8. In the second iteration of this DATA step, after the data is merged, what are the values of **C** and **A**?

```
data client_amount;
   merge clients(in=C)
         amounts(in=A);
   by Name;
run;
```

Clients

Name	EmpID
Ankerton	11123
Davis	22298
Masters	33351
Wolmer	44483

Amounts

Name	Date	Amt
Ankerton	08OCT96	92
Ankerton	15OCT96	43
Davis	04OCT96	16
Masters	.	27
Thomas	21OCT96	15

a. C=1, A=0
b. C=0, A=1
c. C=1, A=1
d. missing
e. unknown

9. If you run this DATA step, what observations does the data set **bonuses** contain?

```
data bonuses;
   merge managers (in=M)
         staff (in=S);
   by EmpID;
   if M=0 and S=1;
run;
```

a. only the observations from **staff** that have no match in **managers**

b. only the observations from **managers** that have no match in **staff**

c. all observations from both **managers** and **staff**, whether or not they match

d. no observations

9. If you run this DATA step, what observations does the data set **bonuses** contain?

```
data bonuses;
   merge managers (in=M)
         staff (in=S);
   by EmpID;
   if M=0 and S=1;
run;
```

a. only the observations from **staff** that have no match in **managers**

b. only the observations from **managers** that have no match in **staff**

c. all observations from both **managers** and **staff**, whether or not they match

d. no observations

10. What is the relationship of the data set **first** to the data set **second** when merged by the variable **ID**?

a. one-to-one
b. one-to-many
c. many-to-one
d. many-to-many
e. nonmatching

first

Name	ID	Age
Togar	121150	39
Kylie	121152	34
Birin	121153	32
Gloria	121154	12
James	121155	36
Gene	121156	28
Tom	121157	27

second

ID	Date
121150	02/15/05
121152	05/22/07
121153	03/04/06
121154	11/22/05
121155	07/08/06
121156	12/15/06
121157	04/30/07

10. What is the relationship of the data set **first** to the data set **second** when merged by the variable **ID**?

a. one-to-one
b. one-to-many
c. many-to-one
d. many-to-many
e. nonmatching

first

Name	ID	Age
Togar	121150	39
Kylie	121152	34
Birin	121153	32
Gloria	121154	12
James	121155	36
Gene	121156	28
Tom	121157	27

second

ID	Date
121150	02/15/05
121152	05/22/07
121153	03/04/06
121154	11/22/05
121155	07/08/06
121156	12/15/06
121157	04/30/07