

**Getting Started With SAS®**  
**SESUG**  
**Mobile, Alabama**

**Charu Shankar**  
**22 October 2022**

**sas**  
THE POWER TO KNOW.

1

**CONTACT**

Email Charu.Shankar@sas.com  
Linkedin www.linkedin.com/in/Charushankar/  
Twitter charuyogacan  
Blogs.sas.com/content/sastraining/author/charushankar

**sas**  
THE POWER TO KNOW.

2

Copyright © 2012, SAS Institute Inc. All rights reserved.

## Chapter 1 Introduction

- 1.1 Overview of SAS Foundation
- 1.2 Course Logistics
- 1.3 Course Data Files

3

3

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 1: Introduction

1.1 Overview of SAS Foundation

1.2 Course Logistics

1.3 Course Data Files

1

1

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 1: Introduction

1.1 Overview of SAS Foundation

1.2 Course Logistics

1.3 Course Data Files

2

2

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Objectives

- Characterize SAS software.
- Describe the functionality of Base SAS and SAS Foundation tools.

3

3

**sas** | THE POWER TO KNOW.

## What Is SAS?

SAS is a suite of business solutions and technologies to help organizations solve business problems.



The diagram consists of three concentric circles. The innermost circle is dark blue and contains the text "HIGH-PERFORMANCE ANALYTICS". The middle ring is light blue and contains the text "BUSINESS INTELLIGENCE". The outermost ring is medium blue and contains the text "INFORMATION MANAGEMENT" and "ANALYTICS". The entire diagram is set against a white background with a thin black border.

**sas**

4

4

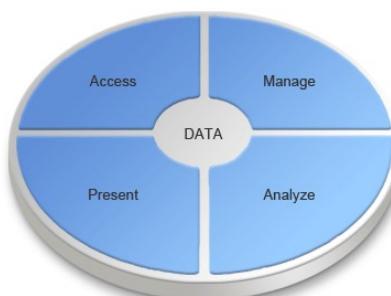
Copyright © 2012, SAS Institute Inc. All rights reserved.

sas THE POWER TO KNOW.

## What Can You Do with SAS?

SAS software enables you to do the following:

- access data across multiple sources
- manage data
- perform sophisticated analyses
- deliver information across your organization



5

5

sas THE POWER TO KNOW.

## About This Class

This class focuses on writing SAS programs to do the following:



- access data in various forms
- create SAS data sets
- use prewritten procedures to analyze data and write basic reports
- combine data sets
- generate detail and summary reports in various formats, including HTML, RTF, PDF

6

6

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 1: Introduction

1.1 Overview of SAS Foundation

1.2 Course Logistics

1.3 Course Data Files

7

7

sas | THE POWER TO KNOW.

## Objectives

- Describe the data used in the course.
- Designate the editors and processing mode available for workshops.
- Specify the naming convention used for course files.
- Define the three levels of exercises.
- Navigate the Help facility.

8

8

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Orion Star Sports & Outdoors**

This course focuses on a fictitious global sports and outdoors retailer that has traditional stores, an online store, and a catalog business.

The diagram illustrates the three components of Orion Star's business model: a traditional storefront with a blue facade and green awning, a computer monitor representing the online store displaying currency symbols (\$, €, ₽), and a catalog icon representing the catalog store with similar currency symbols.

9

9

**Orion Star Data**

Large amounts of data are stored in transactional systems in various formats.

The diagram shows the scale of data for different entities:

- 1,000 employees (represented by a hierarchical tree of green icons)
- 150,000 orders (represented by a clipboard icon)
- 90,000 customers (represented by a group of colorful people icons)
- 64 suppliers (represented by a stack of yellow boxes icon)

A large blue arrow points from the left side of the diagram towards a cluster of three database icons (yellow cylinder, red cylinder, blue cylinder with an Excel 'X' icon) on the right, symbolizing the storage and processing of this data.

10

10

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Orion Star Business Scenarios**

In this course, you **write** SAS programs that access Orion Star data and create reports using an editor.

SAS Enterprise Guide  
Program Editor

11

11

**What Is SAS Enterprise Guide?**

*SAS Enterprise Guide* is a powerful Windows client application that provides a GUI for transparently accessing the power of SAS.

It provides the following:

- a point-and-click interface with menus and wizards that enable the user to define tasks
- SAS code generation and execution based on user selections
- a full programming interface that can be used to write, edit, and submit SAS code

This class uses the programming interface.

12

12

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Running SAS Programs

In this course, you invoke SAS in interactive mode (SAS Enterprise Guide) to **process** programs.



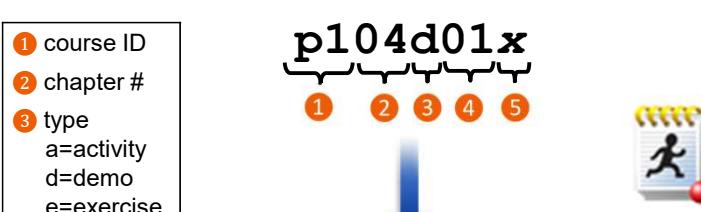
13

13

**sas** | THE POWER TO KNOW.

## Program Naming Conventions

In this course, you retrieve and save SAS programs using the structure below.



① course ID  
② chapter #  
③ type  
  a=activity  
  d=demo  
  e=exercise  
  s=solution  
④ item #  
⑤ placeholder

p104d01x

1    2    3    4    5

Programming 1, Chapter 4, Demo 1

14

14

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas THE POWER TO KNOW.**

## Locating Data Files

In this course, macro variable references are used to give a more flexible approach for locating files.

Examples:

```
%let path=s:\workshop;
```

```
infile "&path\sales.csv";
```

```
infile "&path\payroll.dat";
```

15

15

**Sas THE POWER TO KNOW.**



16

16

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 1: Introduction

1.1 Overview of SAS Foundation

1.2 Course Logistics

1.3 Course Data Files

17

17

sas | THE POWER TO KNOW.

## Objectives

- Execute a SAS program to create the course data files.
- Execute a SAS program to define the data location.

18

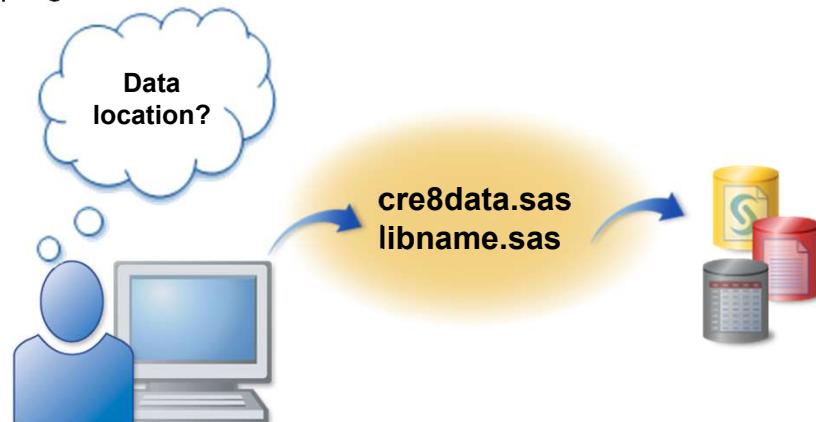
18

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas THE POWER TO KNOW.**

## Business Scenario

Identify a location for the course data files and execute programs to create the files and define the location.



19

19

**Sas THE POWER TO KNOW.**



## Creating Course Data Files

---

This demonstration illustrates how to create the course data files and define the data location.

20

20

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas THE POWER TO KNOW.

## Chapter 2: SAS® Programs

2.1 Introduction to SAS Programs

2.2 Submitting a SAS Program

2.3 SAS Program Syntax

1

1

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas THE POWER TO KNOW.

## Chapter 2: SAS® Programs

2.1 Introduction to SAS Programs

2.2 Submitting a SAS Program

2.3 SAS Program Syntax

2

2

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Objectives

- List the components of a SAS program.

3

3

**SAS Programs**

A *SAS program* is a sequence of one or more steps.

The diagram illustrates the sequential flow of a SAS program. It starts with three data sources on the left: a yellow cylinder labeled 'SAS', a red cylinder labeled 'DATA', and a grey cylinder labeled 'LIBRARY'. Arrows point from these sources to a blue rectangular box labeled 'DATA Step'. From the 'DATA Step' box, an arrow points to a grid icon representing a data set. Another arrow points from this grid icon to a yellow rectangular box labeled 'PROC Step'. Finally, an arrow points from the 'PROC Step' box to a blue document icon representing a report.

- *DATA steps* typically create SAS data sets.
- *PROC steps* typically process SAS data sets to generate reports and graphs, and to manage data.

4

4



## SAS Program Steps

A *step* is a sequence of SAS statements. This program has a DATA step and a PROC step.

```
data work.newemps;
  infile "&path\newemps.csv" dlm=',';
  input First $ Last $ Title $ Salary;
run;

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

5

5



## Step Boundaries

SAS steps begin with either of the following:

- a DATA statement
- a PROC statement

SAS detects the end of a step when it encounters one of the following:

- a RUN statement (for most steps)
- a QUIT statement (for some procedures)
- the beginning of another step (DATA statement or PROC statement)

6

6

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas THE POWER TO KNOW.**

## 2.01 Short Answer Poll

How many steps are in program **p102d01**?

```

data work.newsalesemps;
  length First_Name $ 12
    Last_Name $ 18 Job_Title $ 25;
  infile "&path\newemps.csv" dlm=',';
  input First_Name $ Last_Name $
    Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
  var Salary;
run;

```

7 p102d01

**Sas THE POWER TO KNOW.**

## 2.01 Short Answer Poll – Correct Answer

How many steps are in this program? **three**

```

data work.newsalesemps;
  length First_Name $ 12
    Last_Name $ 18 Job_Title $ 25;
  infile "&path\newemps.csv" dlm=',';
  input First_Name $ Last_Name $
    Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
  var Salary;
run;

```

The diagram illustrates the three steps of the program. A blue bracket on the right side groups the first two lines of code (the DATA step) under the label "DATA Step". Below this, another blue bracket groups the next two lines of code (the first PROC step) under the label "PROC Step". Finally, another blue bracket groups the last two lines of code (the second PROC step) under the label "PROC Step". Each bracket is accompanied by a small yellow square icon containing a blue staircase-like graphic.

8 p102d01



## SAS Program Example

This DATA step creates a temporary SAS data set named **work.newsalesemps** by reading four fields from a file.

```
data work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
var Salary;
run;
```

9

p102d01

9



## SAS Program Example

This PROC PRINT step lists the **work.newsalesemps** data set.

```
data work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
var Salary;
run;
```

10

p102d01

10

## SAS Program Example

This PROC MEANS step summarizes the **Salary** variable in the **work.newsalesemps** data set.

```
data work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
  var Salary;
run;
```

11

p102d01

11

## Idea Exchange

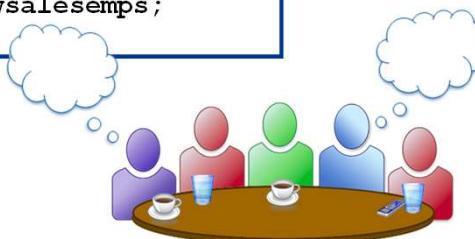
How does SAS detect the end of each step in this program?

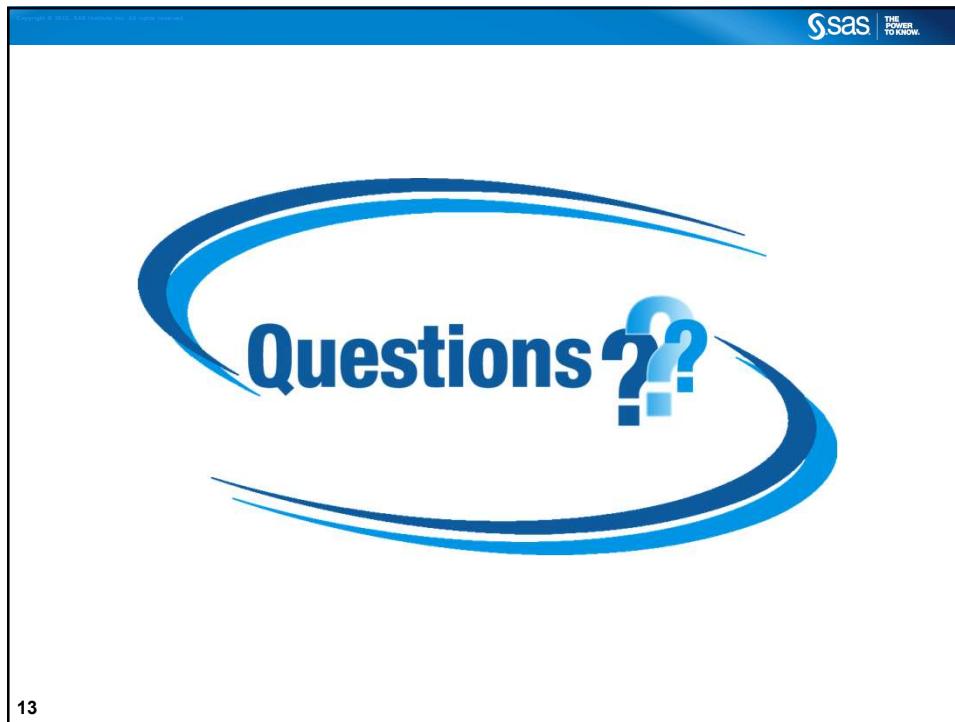
```
data work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
proc means data=work.newsalesemps;
  var Salary;
```

12

12





13

A presentation slide titled "Chapter 2: SAS® Programs" in blue text at the top. Below the title is a dark blue rectangular area containing three white text items: "2.1 Introduction to SAS Programs", "2.2 Submitting a SAS Program", and "2.3 SAS Program Syntax". The slide has the SAS logo in the top right corner and a small number "14" in the bottom left corner.

14

14

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas THE POWER TO KNOW.

## Chapter 2: SAS® Programs

2.1 Introduction to SAS Programs

2.2 Submitting a SAS Program

2.3 SAS Program Syntax

15

15

sas THE POWER TO KNOW.

## Objectives

- Use SAS Enterprise Guide to open and submit a SAS program and browse the results.

16

16

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas** THE POWER TO KNOW.

## Business Scenario

Orion Star programmers will create and execute SAS programs and view results in an interactive environment.

A blue silhouette of a person is positioned above a computer monitor. The monitor displays the SAS logo. A blue arrow points from the person down towards a blue rectangular box containing the text "SAS Enterprise Guide".

17

17

 **Submitting a SAS Program:  
SAS Enterprise Guide**

---

This demonstration illustrates how to open and submit a SAS program and view the results using SAS Enterprise Guide.

18

18

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 2: SAS® Programs

2.1 Introduction to SAS Programs

2.2 Submitting a SAS Program

2.3 SAS Program Syntax

19

19

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 2: SAS® Programs

2.1 Introduction to SAS Programs

2.2 Submitting a SAS Program

2.3 SAS Program Syntax

20

20

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Objectives

- Identify the characteristics of SAS statements.
- Define SAS syntax rules.
- Document a program using comments.
- Diagnose and correct a program with errors.
- Save the corrected program.

21

21

**sas** | THE POWER TO KNOW.

## Business Scenario

Well-formatted, clearly documented SAS programs are an industry best practice.



22

22

Copyright © 2012, SAS Institute Inc. All rights reserved.

Sas THE POWER TO KNOW.

## SAS Syntax Rules: Statements

*SAS statements*

- usually begin with an ***identifying keyword***
- always end with a ***semicolon***.

```
data work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
var Salary;
run;
```

23 p102d01

23

Sas THE POWER TO KNOW.

## 2.03 Short Answer Poll

How many statements make up this DATA step?

```
data work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
run;
```

24

24

## 2.03 Short Answer Poll – Correct Answer

How many statements make up this DATA step?

```
data work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
run;
```

**This DATA step has five statements.**

25

25

## SAS Program Structure

SAS code is free format.

```
data work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
proc print data=work.newsalesemps; run;
  proc means data =work.newsalesemps;
  var Salary;run;
```

This program is syntactically correct but difficult to read.

26

p102d02

26

Copyright © 2012, SAS Institute Inc. All rights reserved.

**SAS Program Structure**

Rules for SAS Statements

- Statements can begin and end in any column.
- A single statement can span multiple lines.
- Several statements can appear on the same line.
- Unquoted values can be lowercase, uppercase, or mixed case.

```

data work.newsalesEmps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;run;
proc print data=work.newsalesemps; run;
  proc means data =work.newsalesemps;
var Salary;run;

```

unconventional  
formatting

27

27

**Recommended Formatting**

- Begin each statement on a new line.
- Use white space to separate words and steps.
- Indent statements within a step.
- Indent continued lines in multi-line statements.

```

data work.newsalesemps;
  length First_Name $ 12
        Last_Name $ 18 Job_Title $ 25;
  infile "&path\newemps.csv" dlm=',';
  input First_Name $ Last_Name $
        Job_Title $ Salary;
run;

proc print data=work.newsalesemps;
run;

proc means data=work.newsalesemps;
  var Salary;
run;

```

conventional  
formatting

28

28

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**Sas | THE POWER TO KNOW.**

## Program Documentation

You can embed comments in a program as explanatory text.

```

/* create a temporary data set, newsalesemps */
/* from the text file newemps.csv */
```

```

data work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
*read a comma delimited file;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
      /* comment statement */
run;
```

SAS ignores comments during processing but writes them to the SAS log.

29

29

**Sas | THE POWER TO KNOW.**

## SAS Comments

This program contains four comments.

```

*-----*
| This program creates and uses the |
| data set called work.newsalesemps. |
*-----*;
```

```

data work.newsalesemps;
length First_Name $ 12 Last_Name $ 18
      Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary /*numeric*/;②
run;
/*③
proc print data=work.newsalesemps;
run;
*/
proc means data=work.newsalesemps;
  *var Salary; ④
run;
```

30 p102d03

30

30



## 2.04 Short Answer Poll

Open and examine **p102a01**. Based on the comments, which steps do you think will execute and what output will be generated?

Submit the program. Which steps were executed?

31

31



## 2.04 Short Answer Poll – Correct Answer

Open and examine **p102a01**. Based on the comments, which steps do you think will execute and what output will be generated?

Submit the program. Which steps were executed?

- **The DATA step executes and creates an output data set.**
- **The PROC PRINT step executes and produces a report.**
- **The PROC MEANS step is “commented out” and therefore does not execute.**

32

32



33

The slide features the SAS logo at the top right. The main title "Business Scenario" is centered in a large, bold, blue font. Below the title, a subtitle reads: "Orion Star programmers must be able to identify and correct syntax errors in a SAS program." An icon of a person looking at a computer monitor is positioned in the center. The monitor displays a document with several syntax errors, represented by blue lines and symbols. In the bottom left corner of the slide, the number "34" is visible.

34

34

## Syntax Errors

A *syntax error* is an error in the spelling or grammar of a SAS statement. SAS finds syntax errors as it compiles each SAS statement, before execution begins.

Examples of syntax errors:

- misspelled keywords
- unmatched quotation marks
- missing semicolons
- invalid options

35

35

## 2.05 Short Answer Poll

This program includes three syntax errors. One is an invalid option. What are the other two syntax errors?

```
daat work.newsalesemps;
  length First_Name $ 12
    Last_Name $ 18 Job_Title $ 25;
  infile "&path\newemps.csv" dlm=',';
  input First_Name $ Last_Name $
    Job_Title $ Salary;
run;

proc print data=work.newsalesemps
run;

proc means data=work.newsalesemps average min;
  var Salary;
run;
```



An arrow points from the word "average" in the third line of the SAS code to the green callout box.

36

36

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas THE POWER TO KNOW.**

## 2.05 Short Answer Poll – Correct Answer

This program includes three syntax errors. One is an invalid option. What are the other two syntax errors?

```

daat work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
run;

proc print data=work.newsalesemps
run;

proc means data=work.newsalesemps average min;
  var Salary;
run;

```

misspelled keyword

missing semicolon

invalid option

37

37

**Sas THE POWER TO KNOW.**

## Syntax Errors

The Program Editor in SAS Enterprise Guide uses the color red to indicate a potential error in your SAS code.

```

daat work.newsalesemps;
length First_Name $ 12
      Last_Name $ 18 Job_Title $ 25;
infile "&path\newemps.csv" dlm=',';
input First_Name $ Last_Name $
      Job_Title $ Salary;
run;

proc print data=work.newsalesemps
run;

proc means data=work.newsalesemps average min;
  var Salary;
run;

```

38

38

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas** THE POWER TO KNOW.

## Syntax Errors

When SAS encounters a syntax error, it writes a warning or error message to the log.

**ERROR 22-322: Syntax error, expecting one of the following:  
a name, a quoted string, (, /, ;, \_DATA\_, \_LAST\_,  
\_NULL\_.**

**WARNING: Data set WORK.TEST was not replaced because this step was  
stopped.**

 You should always check the log to make sure that the program ran successfully, even if output is generated.

39

39

**Questions ??**

40

40

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 3: Accessing Data

3.1 Examining SAS Data Sets

3.2 Accessing SAS Libraries

1

Chapter 3: Accessing Data

3.1 Examining SAS Data Sets

3.2 Accessing SAS Libraries

2

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Objectives

- Define the components of a SAS data set.
- Browse the descriptor portion of a SAS data set using the CONTENTS procedure.
- Browse the data portion of a SAS data set using the PRINT procedure.
- Define a SAS variable.
- Define a missing value.

3

3

**sas** | THE POWER TO KNOW.

## Business Scenario

Many SAS data sets related to the Orion Star project already exist. The programmers need to know how to display the structure and contents of the data sets.

The diagram illustrates a business scenario involving SAS data. In the center is a computer monitor displaying the SAS logo. A blue arrow points from the monitor to a icon representing a 'SAS Data Set', which is depicted as a grid of colored squares with a red ball at the bottom right corner. Another blue arrow points from the monitor to a icon representing a 'Report', which is shown as a document with a grid of text. A blue silhouette of a person is positioned below the monitor, indicating they are the user interacting with the system.

4

4

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## What Is a SAS Data Set?

A SAS *data set* is a specially structured data file that SAS creates and that only SAS can read. A SAS data set is a table that contains observations and variables.

5

5

sas | THE POWER TO KNOW.

## SAS Data Set Terminology

A SAS data set contains a descriptor portion and a data portion.

6

6

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Descriptor Portion**

The *descriptor portion* contains the following metadata:

- general properties (such as data set name and number of observations)
- variable properties (such as name, type, and length)

Partial `work.newsalesemps`

Data Set Name	WORK.NEWSALESEMPSP						
Engine	V9						
Created	Mon, Feb 27, 2012 01:28 PM						
Observations	71						
Variables	4						
...							
First_Name	\$ 12	Last_Name	\$ 18	Job_Title	\$ 25	Salary	N 8

7

**Browsing the Descriptor Portion**

Use `PROC CONTENTS` to display the descriptor portion of a SAS data set.

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

**PROC CONTENTS DATA=SAS-data-set;**
**RUN;**

8

p103d01

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Viewing the Output

### Partial PROC CONTENTS Output

The CONTENTS Procedure			
Data Set Name	WORK.NEWSALESEMPS	Observations	71
Member Type	DATA	Variables	4
Engine	V9	Indexes	0
Created	Mon, Feb 27, 2012 01:28:51 PM	Observation Length	64
Last Modified	Mon, Feb 27, 2012 01:28:51 PM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO

Engine/Host Dependent Information

...

Alphabetic List of Variables and Attributes

#	Variable	Type	Len
1	First_Name	Char	12
3	Job_Title	Char	25
2	Last_Name	Char	18
4	Salary	Num	8

9

9

**sas** | THE POWER TO KNOW.

## 3.01 Short Answer Poll

Open program **p103a01**. Add a PROC CONTENTS step after the DATA step to view **work.donations**. Submit the program and review the results. How many observations are in the data set **work.donations**?

10

10

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**3.01 Short Answer Poll – Correct Answer**

Open program **p103a01**. Add a PROC CONTENTS step after the DATA step to view **work.donations**. Submit the program and review the results. How many observations are in the data set **work.donations**? **124 observations**

```
data work.donations;
  infile "&path\donation.dat";
  input Employee_ID Qtr1 Qtr2 Qtr3 Qtr4;
  Total=sum(Qtr1,Qtr2,Qtr3,Qtr4);
run;

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

11 p103a01s

11

**Data Portion**

The *data portion* of a SAS data set contains the data values, which are either character or numeric.

**Partial work.newsalesemps**

First_Name	Last_Name	Job_Title	Salary
Satyakam	Denny	Sales Rep. II	26780
Monica	Kletschkus	Sales Rep. IV	30890
Kevin	Lyon	Sales Rep. I	26955
Petrea	Soltau	Sales Rep. II	27440

variable names

data values

character values

numeric values

12

12

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Browsing the Data Portion**

Use *PROC PRINT* to display the data portion of a SAS data set.

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

**PROC PRINT DATA=SAS-data-set;  
RUN;**

13 p103d02

13

**Viewing the Output**

Partial PROC PRINT Output

Obs	First_Name	Last_Name	Job_Title	Salary
1	Satyakam	Denny	Sales Rep. II	26780
2	Monica	Kletschkus	Sales Rep. IV	30890
3	Kevin	Lyon	Sales Rep. I	26955
4	Petrea	Soltau	Sales Rep. II	27440
5	Marina	Iyengar	Sales Rep. III	29715

14

14

Copyright © 2012, SAS Institute Inc. All rights reserved.

**SAS Variable Names**

SAS variable names

- can be 1 to 32 characters long.
- must start with a letter or underscore. Subsequent characters can be letters, underscores, or numerals.
- can be uppercase, lowercase, or mixed case.
- are not case sensitive.

Salary
 
\_score2\_
  
 
cust\_ID
 
month1
 
FirstName

15

15

**3.02 Multiple Answer Poll**

Which variable names are invalid?

- a. data5mon
- b. 5monthsdata
- c. data#5
- d. five months data
- e. five\_months\_data
- f. FiveMonthsData
- g. fivemonthsdata

16

16

## 3.02 Multiple Answer Poll – Correct Answer

Which variable names are invalid?

- a. data5mon
- b. 5monthsdata**
- c. data#5
- d. five months data**
- e. five\_months\_data
- f. FiveMonthsData
- g. fivemonthsdata

17

17

## Data Types

A SAS data set supports two types of variables.

### *Character variables*

- can contain any value: letters, numerals, special characters, and blanks
- range from 1 to 32,767 characters in length
- have 1 byte per character.

### *Numeric variables*

- store numeric values using floating point or binary representation
- have 8 bytes of storage by default
- can store 16 or 17 significant digits.

18

18

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** THE POWER TO KNOW.

## Missing Data Values

Missing values are valid values in a SAS data set.

Partial `work.newsalesemps`

First_Name	Last_Name	Job_Title	Salary
Monica	Kletschkus	Sales Rep. IV	.
Kevin	Lyon	Sales Rep. I	26955
Petrea	Soltau		27440

A blank represents a missing character value.

A period represents a missing numeric value.

A value must exist for every variable in every observation.

19

19

**sas** THE POWER TO KNOW.

## SAS Date Values

SAS stores calendar dates as numeric values.

A SAS *date value* is stored as the number of days between January 1, 1960, and a specific date.

20

20



21

This slide has a blue header bar with the SAS logo and tagline. The main title "Chapter 3: Accessing Data" is centered in a large, blue, sans-serif font. Below the title is a dark blue rectangular box with a thin gray border. Inside this box, there are two sections of text: "3.1 Examining SAS Data Sets" at the top and "3.2 Accessing SAS Libraries" at the bottom. The background of the slide features a subtle, abstract blue design with circular patterns.

22

22

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 3: Accessing Data

3.1 Examining SAS Data Sets

3.2 Accessing SAS Libraries

23

23

sas | THE POWER TO KNOW.

## Objectives

- Explain the concept of a SAS library.
- State the difference between a temporary library and a permanent library.
- Assign a library reference name to a SAS library using a LIBNAME statement.
- Investigate a SAS library programmatically and interactively.
- Access a data set in a user-created permanent library.

24

24

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Business Scenario**

Orion Star programmers need to access existing SAS data sets, so they need to understand how the data sets are stored in SAS.

The diagram illustrates a business scenario. On the left, a blue silhouette of a person is facing a computer monitor. A blue arrow points from the person to a server rack icon on the right. The server rack icon contains several smaller icons representing data sets, with one red sphere at the bottom right.

25

25

**SAS Libraries**

SAS data sets are stored in *SAS libraries*. A SAS library is a collection of SAS files that are referenced and stored as a unit.

The diagram shows three server rack icons with red spheres at their bases. Blue arrows point from each rack to a central yellow folder icon, which represents a library.

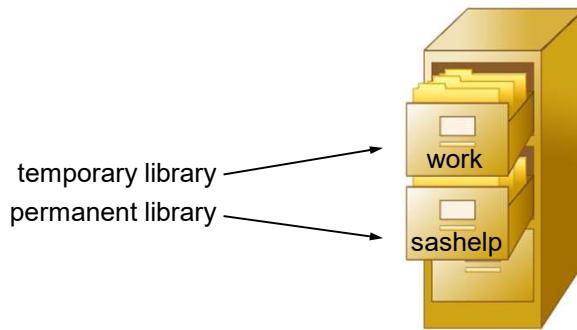
Each file is a member of the library.

26

26

## How SAS Libraries Are Defined

When a SAS session starts, SAS automatically creates one temporary and at least one permanent SAS library that you can access. These libraries are open and ready to be used.

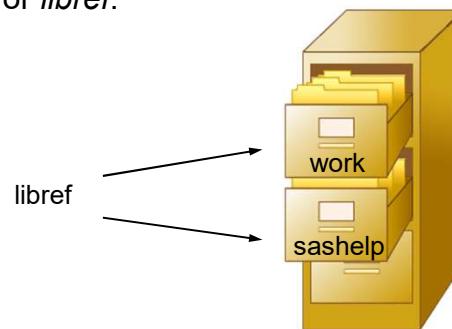


27

27

## Assigning a Libref

Regardless of the operating system that you use, you refer to a SAS library by a logical name called a library reference name, or *libref*.



A libref is a shortcut to the library.

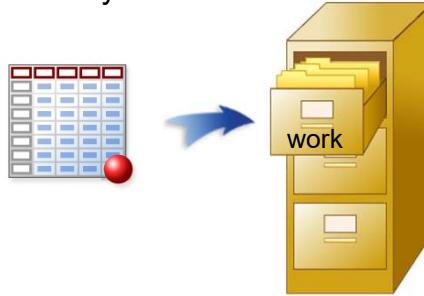
28

28

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Temporary Library**

**Work** is a temporary library where you can store and access SAS data sets for the duration of the SAS session. It is the default library.



⚠️ SAS deletes the **work** library and its contents when the session terminates.

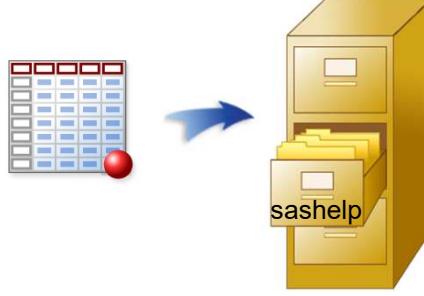
29

29

**sas** | THE POWER TO KNOW.

**Permanent Libraries**

**Sashelp** is a permanent library that contains sample SAS data sets you can access during your SAS session.



30

30

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Accessing SAS Data Sets**

All SAS data sets have a two-level name that consists of the libref and the data set name, separated by a period.

```

graph TD
    subgraph Top [ ]
        direction TB
        L1[libref] --> L2[work]
        L1[libref] --> L3[sashelp]
        L2[work] --> D1[newsalesemps]
        L3[sashelp] --> D2[class]
        D1[newsalesemps] --- D2[class]
    end
    subgraph Bottom [ ]
        direction TB
        L4[libref] --> L5[work]
        L4[libref] --> L6[sashelp]
        L5[work] --> D3[newsalesemps]
        L6[sashelp] --> D4[class]
        D3[newsalesemps] --- D4[class]
    end
    subgraph Template [libref.data-set-name]
        direction LR
        L7[libref] --- L8[data-set-name]
    end

```

When a data set is in the temporary **work** library, you can use a one-level name (for example, **newsalesemps**).

31

31

**Business Scenario**

Orion Star programmers need to access and view SAS data sets that are stored in a permanent user-defined library.

32

32

Copyright © 2012, SAS Institute Inc. All rights reserved.

**User-Defined Libraries**

Users can create their own SAS libraries. A user-defined library

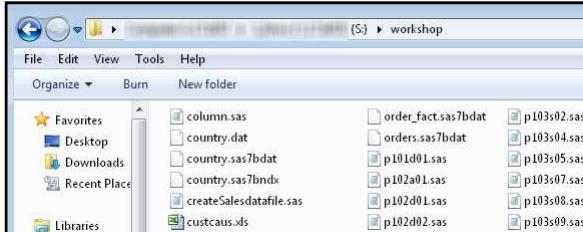
- is permanent. Data sets are stored until the user deletes them.
- is implemented within the operating environment's file system.
- is not automatically available in a SAS session.

33

33

**Accessing a Permanent Library**

**Step 1** Identify the location of the library.



In this example, **s:\workshop**, a Microsoft Windows folder, is used as the SAS library.

**!** Identify the location of **your** course data.

34

34

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** THE POWER TO KNOW.

## Accessing a Permanent Library

**Step 2** Use a SAS LIBNAME statement to associate the libref with the physical location of the library.

Associate the libref **orion** with the Windows folder so that it is available to your SAS session.

35

35

**LIBNAME Statement**

The SAS LIBNAME statement is a *global* SAS statement.

```
libname orion "s:\workshop";
LIBNAME libref "SAS-library" <options>;
```

- It is not required to be in a DATA step or PROC step.
- It does not require a RUN statement.
- It executes immediately.
- It remains in effect until changed or canceled, or until the session ends.

**!** Use the location of **your** course data in your LIBNAME statement.

36

36

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Viewing the Log

### Partial SAS Log

```
47 libname orion "s:\workshop";
NOTE: Libref ORION was successfully assigned as follows:
      Engine:      V9
      Physical Name: s:\workshop
```

37

37

sas | THE POWER TO KNOW.

## Browsing a Library

**Step 3** You can browse a library interactively in a SAS or SAS Enterprise Guide session, or programmatically using the CONTENTS procedure.

The illustration depicts a filing cabinet with a folder labeled 'orion' being magnified by a person. Another person is using a magnifying glass to look at a PROC CONTENTS output, which is shown as a yellow document with sun-like icons.

38

38

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Browsing a Library Programmatically**

Use PROC CONTENTS with the `_ALL_` keyword to generate a list of all SAS files in a library.

```
proc contents data=orion._all_ nods;
run;
```

```
PROC CONTENTS DATA=libref._ALL_ NODS;
RUN;
```

- `_ALL_` requests all files in the library.
- The NODS option suppresses the individual data set descriptor information.
- NODS can be used only with the keyword `_ALL_`.

39 p103d03

39

**Viewing the Output**

Partial PROC CONTENTS Output

The CONTENTS Procedure				
Directory				
Libref	ORION			
Engine	V9			
Physical Name	S:\workshop			
Filename	S:\workshop			
Member      File				
#	Name	Type	Size	Last Modified
1	CHARITIES	DATA	9216	23Aug12:15:58:39
2	CONSULTANTS	DATA	5120	23Aug12:15:58:39
3	COUNTRY	DATA	17408	13Oct10:19:04:39
	COUNTRY	INDEX	17408	13Oct10:19:04:39
4	CUSTOMER	DATA	33792	04Nov11:09:52:27
5	CUSTOMER_DIM	DATA	33792	04Nov11:09:52:27

40

40

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Accessing a Permanent Data Set

**Step 4** After the libref is assigned, you can access SAS files in the library.

```
proc print data=orion.country;
run;
```

PROC PRINT Output

Obs	Country	Country_Name	Population	Country_ID	Continent_ID	CountryFormerName
1	AU	Australia	20,000,000	160	96	
2	CA	Canada	.	260	91	
3	DE	Germany	80,000,000	394	93	East/West Germany
4	IL	Israel	5,000,000	475	95	
5	TR	Turkey	70,000,000	905	95	
6	US	United States	280,000,000	926	91	
7	ZA	South Africa	43,000,000	801	94	

41 p103d04

41

**sas** | THE POWER TO KNOW.

## Viewing the Log

Partial SAS Log

```
25 proc print data=orion.country;
26 run;
```

NOTE: There were 7 observations read from the data set  
ORION.COUNTRY.

The libref **orion** remains in effect until you change or cancel it, or until you end your SAS session.

42

42

Copyright © 2012, SAS Institute Inc. All rights reserved.

 **sas** | THE WIDEST  
VISION TO KNOW.



## Browsing SAS Libraries: SAS Enterprise Guide

---

This demonstration illustrates defining and accessing a SAS library using SAS Enterprise Guide.

43

libname  
p103d03

43

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 4: Producing Detail Reports

1

1

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 4: Producing Detail Reports

2

2

## Objectives

- Create a default PROC PRINT report.
- Select variables with a VAR statement.
- Select observations with a WHERE statement.

3

3

## Business Scenario

Orion Star management wants a report that displays the names, salaries, and a salary total for all sales employees.



PROC PRINT



Obs	Last_Name	First_Name	Salary
1	xxxxxxx	xxxxxxxxxx	99999
2	xxxxxxx	xxxxxxxxxx	99999
3	xxxxxxx	xxxxxxxxxx	99999
-----			99999



4

4

THE  
POWER  
TO KNOW.

## PRINT Procedure

By default, PROC PRINT displays all observations, all variables, and an Obs column on the left side.

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

### Partial PROC PRINT Output

Obs	Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title	Country	Birth_Date	Hire_Date
1	120102	Tom	Zhou	M	108255	Sales Manager	AU	3510	10744
2	120103	Wilson	Dawes	M	87975	Sales Manager	AU	-3996	5114
3	120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	-5630	5114
4	120122	Christina	Ngan	F	27475	Sales Rep. II	AU	-1984	6756
5	120123	Kimiko	Hotstone	F	26190	Sales Rep. I	AU	1732	9405

Statements and options can be added to the PRINT procedure to modify the default behavior.

5

p104d01

5

## VAR Statement

The VAR statement selects variables to include in the report and specifies their order.

```
proc print data=orion.sales;
  var Last_Name First_Name Salary;
run;
```

VAR variable(s);

### Partial PROC PRINT Output

Obs	Last_Name	First_Name	Salary
1	Zhou	Tom	108255
2	Dawes	Wilson	87975
3	Elvish	Irenie	26600
4	Ngan	Christina	27475
5	Hotstone	Kimiko	26190

6

p104d01

6

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas THE POWER TO KNOW.**

## Business Scenario

Orion Star management wants a report that displays the names and salaries of the sales employees earning less than \$25,500. Suppress the Obs column.

The diagram illustrates a business scenario. A blue user icon with a yellow glow points to a data table icon labeled "orion.sales". An arrow points from the "orion.sales" table to a "PROC PRINT" icon, which is represented by a yellow square containing two sun-like symbols. Below the icons is a table with the following data:

Last_Name	First_Name	Salary
xxxxxxx	xxxxxxx	25000
xxxxxxx	xxxxxxx	20000
xxxxxxx	xxxxxxx	23000

7

**Sas THE POWER TO KNOW.**

## WHERE Statement

The *WHERE statement* selects observations that meet the criteria specified in the WHERE expression.

```
proc print data=orion.sales;
  var Last_Name First_Name Salary;
  where Salary<25500;
run;
```

**WHERE WHERE-expression;**

8

p104d02

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Viewing the Log

Only 7 of the 165 observations from **orion.sales** were selected by the WHERE statement.

```

295 proc print data=orion.sales;
296   var Last_Name First_Name Salary;
297   where Salary<25500;
298 run;
```

**NOTE:** There were 7 observations read from the data set ORION.SALES.  
WHERE Salary<25500;

9

**sas** | THE POWER TO KNOW.

## Viewing the Output

PROC PRINT Output

Obs	Last_Name	First_Name	Salary
49	Tilley	Kimiko	25185
50	Barcoe	Selina	25275
85	Anstey	David	25285
104	Voron	Tachaun	25125
111	Polky	Asishana	25110
131	Ould	Tulsidas	22710
148	Buckner	Burnetta	25390

original observation numbers

10

10

THE POWER  
TO KNOW.

## Comparison Operators

*Comparison operators* compare a variable with a value or with another variable.

Symbol	Mnemonic	Definition
=	EQ	Equal to
^=    !=    ~=	NE	Not equal to
>	GT	Greater than
<	LT	Less than
>=	GE	Greater than or equal
<=	LE	Less than or equal
	IN	Equal to one of a list

11

11

THE POWER  
TO KNOW.

## Comparison Operators

### Examples

```
where Gender eq ' ' ;
where Salary ne . ;
where Salary>=50000 ;
where Hire_Date<'01Jan2000'd ;
where Country in ('AU','US') ;
where Country in ('AU' 'US') ;
where Order_Type in (1,2,3) ;
```

The value list in the IN operator must be enclosed in parentheses and separated by either commas or blanks. Character values must be enclosed in quotation marks.

12

12

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Logical Operators

*Logical operators combine or modify WHERE expressions.*

```
proc print data=orion.sales;
  where Country='AU' and
        Salary<30000;
run;
```

**WHERE WHERE-expression-1 AND | OR  
WHERE-expression-n;**

13 p104d03

13

**sas** | THE POWER TO KNOW.

## Business Scenario

Orion Star management wants a report that lists only the Australian sales representatives.

The diagram illustrates a business scenario. It starts with a grid icon labeled "orion.sales". An arrow points from this icon to a map of Australia. Another arrow points from the map to a blue user icon. A final arrow points from the user icon to a table.

Last_Name	First_Name	Country	Job_Title
xxxxxxxxxx	xxxxxx	xx	xxxxxxxxxxxxxx
xxxxxxxxxx	xxxxxx	xx	xxxxxxxxxxxxxx
xxxxxxxxxx	xxxxxx	xx	xxxxxxxxxxxxxx
xxxxxxxxxx	xxxxxx	xx	xxxxxxxxxxxxxx

14

14

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas | THE POWER TO KNOW.**

## Exploring the Data

```
proc print data=orion.sales noobs;
  var Last_Name First_Name Country
      Job_Title;
run;
```

Partial PROC PRINT Output

Plested	Billy	AU	Sales Rep. II
Wills	Matsuoka	AU	Sales Rep. III
George	Vino	AU	Sales Rep. II
Body	Meera	AU	Sales Rep. III
Highpoint	Harry	US	Chief Sales Officer
Magolan	Julienne	US	Sales Rep. II
Desanctis	Scott	US	Sales Rep. IV
Ridley	Cherda	US	Sales Rep. IV

15 p104d04

15

**sas | THE POWER TO KNOW.**

## Subsetting in a PROC PRINT Step

Include a WHERE statement to subset by **Country** and **Job\_Title**.

```
proc print data=orion.sales noobs;
  var Last_Name First_Name Country
      Job_Title;
  where Country='AU' and
        Job_Title contains 'Rep';
run;
```

CONTAINS is a special WHERE operator.

16 p104d04

16

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas | THE POWER TO KNOW.**

## CONTAINS Operator

The **CONTAINS** operator selects observations that include the specified substring.

Equivalent Statements
<code>where Job_Title contains 'Rep' ;</code>
<code>where Job_Title ? 'Rep' ;</code>

- ? can be used instead of the mnemonic.
- The position of the substring within the variable's values is not important.
- Comparisons made with the CONTAINS operator are case sensitive.

17

17

**sas | THE POWER TO KNOW.**

## Viewing the Output

Partial PROC PRINT Output

Last_Name	First_Name	Country	Job_Title
Elvish	Irenie	AU	Sales Rep. II
Ngan	Christina	AU	Sales Rep. II
Hotstone	Kimiko	AU	Sales Rep. I
Daymond	Lucian	AU	Sales Rep. I
Hofmeister	Fong	AU	Sales Rep. IV

18

18



19

The image shows a white rectangular card with a thin black border. At the top right is the SAS logo. Below it, the title "Chapter 4: Producing Detail Reports" is displayed in a large, blue, sans-serif font. A large, dark blue rectangular box is centered on the page. It contains two sections of text: "4.1 Subsetting Report Data" at the top and "4.2 Enhancing Reports" below it. The background of the slide features a subtle, abstract blue design with circular patterns. In the bottom left corner, the number "20" is printed.

20

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 4: Producing Detail Reports



4.1 Subsetting Report Data

4.2 Enhancing Reports

21

21

sas | THE POWER TO KNOW.

## Objectives

- Include titles and footnotes in a report.
- Define descriptive column headings using the LABEL statement.

22

22

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas THE POWER TO KNOW.**

## Business Scenario

Enhance the payroll report by adding titles, footnotes, and descriptive column headings.

Obs	Employee_ID	Last_Name	Salary
1	9999	xxxxxxxxxx	99999
2	9999	xxxxxxxxxx	99999
3	9999	xxxxxxxxxx	99999



Orion Star Sales Staff  
Salary Report

Obs	Employee ID	Last Name	Annual Salary
1	9999	xxxxxxxxxx	99999
2	9999	xxxxxxxxxx	99999
3	9999	xxxxxxxxxx	99999

Confidential

23

**Sas THE POWER TO KNOW.**

## Displaying Titles and Footnotes

Use TITLE and FOOTNOTE statements to enhance the report.

```

title1 'Orion Star Sales Staff';
title2 'Salary Report';

footnotel 'Confidential';

proc print data=orion.sales;
  var Employee_ID Last_Name Salary;
run;

title;
footnote;

```

24

p104d11

24

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Viewing the Output

Partial PROC PRINT Output

Orion Star Sales Staff Salary Report			
Obs	Employee_ID	Last_Name	Salary
1	120102	Zhou	108255
2	120103	Dawes	87975
3	120121	Elvish	26600
...			
164	121144	Capachietti	83505
165	121145	Lansberry	84260

**Confidential**

25

25

**sas** | THE POWER TO KNOW.

## TITLE Statement

The global **TITLE statement** specifies title lines for SAS output.

**TITLE***n* 'text';

- Titles appear at the top of the page.
- The default title is **The SAS System**.
- The value of *n* can be from 1 to 10.
- An unnumbered **TITLE** is equivalent to **TITLE1**.
- Titles remain in effect until they are changed or canceled, or you end your SAS session.

26

26

Copyright © 2012, SAS Institute Inc. All rights reserved.

**FOOTNOTE Statement**

The global **FOOTNOTE** statement specifies footnote lines for SAS output.

```
FOOTNOTE $n$  'text';
```

- Footnotes appear at the bottom of the page.
- No footnote is printed unless one is specified.
- The value of  $n$  can be from 1 to 10.
- An unnumbered **FOOTNOTE** is equivalent to **FOOTNOTE1**.
- Footnotes remain in effect until they are changed or canceled, or you end your SAS session.

27

27

**Changing Titles and Footnotes**

To change a title line, submit a **TITLE** statement with the same number but different text.

- replaces a previous title with the same number
- cancels all titles with higher numbers

```
title1 'ABC Company';
title2 'Sales Division';
title3 'Salary Report';
```

```
title1 'Salary Report';
```

This statement changes title 1 and cancels titles 2 and 3.

Footnotes are changed the same way.

28

28

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas THE POWER TO KNOW.**

## Cancelling All Titles and Footnotes

- The null TITLE statement cancels all titles.  
`title;`
- The null FOOTNOTE statement cancels all footnotes.  
`footnote;`

29

29

**Sas THE POWER TO KNOW.**



30

30

Copyright © 2012, SAS Institute Inc. All rights reserved.

**SAS** THE POWER TO KNOW.

## LABEL Statement and Option

Use a LABEL statement and the LABEL option to display descriptive column headings instead of variable names.

```

title1 'Orion Star Sales Staff';
title2 'Salary Report';
footnote1 'Confidential';

proc print data=orion.sales label;
  var Employee_ID Last_Name Salary;
  label Employee_ID='Sales ID'
        Last_Name='Last Name'
        Salary='Annual Salary';
run;

title;
footnote;

```

**LABEL** variable-1='label'  
 ...  
 variable-n='label';

31 p104d12

31

**SAS** THE POWER TO KNOW.

## LABEL Statement

The LABEL statement assigns descriptive labels to variables.

- A label can be up to 256 characters and include any characters, including blanks.
- Labels are used automatically by many procedures.
- The PRINT procedure uses labels when the LABEL = option is specified.

32

32

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** | THE POWER TO KNOW.

## Viewing the Output

Obs	Sales ID	Last Name	Annual Salary
1	120102	Zhou	108255
2	120103	Dawes	87975
3	120121	Elvish	26600
...			
164	121144	Capachietti	83505
165	121145	Lansberry	84260

Confidential

33

33



34

34

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 5: Reading Data

1

1

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 5: Reading Data

2

2

## Objectives

- Define the business scenario that will be used when reading from a data source to create a SAS data set.
- Use a DATA step to create a SAS data set from an existing SAS data set.
- Subset observations with a WHERE statement.
- Create a new variable with an assignment statement.

3

3

## Business Scenario

Information about Orion Star sales employees resides in several input sources.

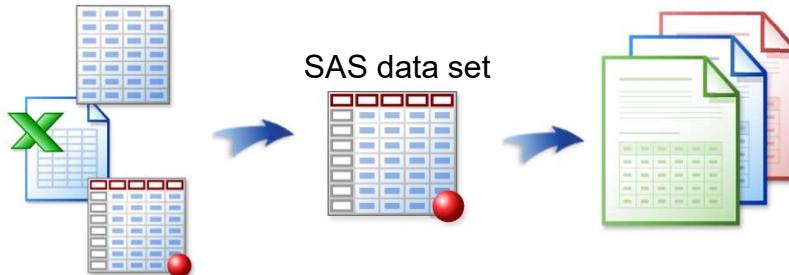


4

4

## Considerations

Management wants a series of reports for Australian sales employees. You will read data from various input sources to create a SAS data set that can be analyzed and presented.

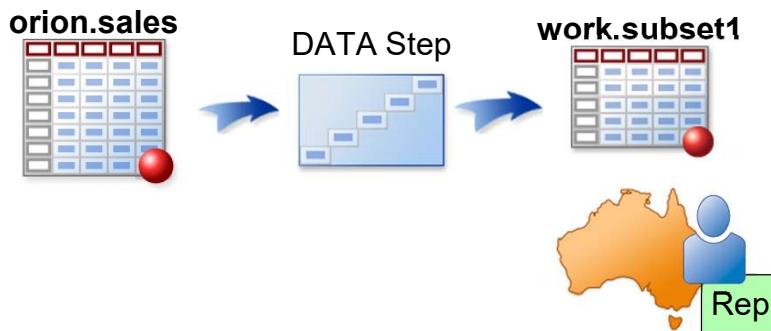


5

5

## Business Scenario: Part 1

Read an existing SAS data set to create a new data set. The new data set should include only the observations for the Australian sales representatives.



6

6

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** THE POWER TO KNOW.

## Using a SAS Data Set as Input

```
data work.subset1;
  set orion.sales;
  where Country='AU' and
        Job_Title contains 'Rep';
run;
```

**DATA** *output-SAS-data-set*;
**SET** *input-SAS-data-set*;
**WHERE** *WHERE-expression*;
**RUN**;

7 p106d01

**sas** THE POWER TO KNOW.

## DATA Statement

The *DATA statement* begins a DATA step and provides the name of the SAS data set to create.

```
data work.subset1;
  set orion.sales;
  where Country='AU' and
        Job_Title contains 'Rep';
run;
```

**DATA** *output-SAS-data-set*;

A DATA step can create temporary or permanent data sets.

- ✍ The rules for SAS variable names also apply to data set names.

8 p106d01

Copyright © 2012, SAS Institute Inc. All rights reserved.

**SAS | THE POWER TO KNOW.**

## SET Statement

The *SET statement* reads observations from an existing SAS data set for further processing in the DATA step.

```
data work.subset1;
  set orion.sales;
  where Country='AU' and
        Job_Title contains 'Rep';
run;
```

- The SET statement reads all observations and all variables from the input data set.
- Observations are read sequentially, one at a time.
- The SET statement can read temporary or permanent data sets.

9 p106d01

9

**WHERE Statement**

The *WHERE statement* selects observations from a SAS data set that meet a particular condition.

```
data work.subset1;
  set orion.sales;
  where Country='AU' and
        Job_Title contains 'Rep';
run;
```

The variables named in the WHERE expression must exist in the input SAS data set.

10 p106d01

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** THE POWER TO KNOW.

## Viewing the Log

### Partial SAS Log

```

42  data work.subset1;
43    set orion.sales;
44    where Country='AU' and
45      Job_Title contains 'Rep';
46  run;

```

NOTE: There were 61 observations read from the data set ORION.SALES.  
 WHERE (Country='AU') and Job\_Title contains 'Rep';  
 NOTE: The data set WORK.SUBSET1 has 61 observations and 9 variables.

SAS read 61 of the 165 observations.

11

11

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** THE POWER TO KNOW.

## Viewing the Output

```

proc print data=work.subset1 noobs;
run;

```

### Partial PROC PRINT Output

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title	Country	Birth_Date	Hire_Date
120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	-4169	6575
120122	Christina	Ngan	F	27475	Sales Rep. II	AU	-523	8217
120123	Kimiko	Hotstone	F	26190	Sales Rep. I	AU	3193	10866
120124	Lucian	Daymond	M	26480	Sales Rep. I	AU	1228	8460
120125	Fong	Hofmeister	M	32040	Sales Rep. IV	AU	-391	8460

p106d01

12

12



13

A presentation slide titled "Business Scenario: Part 2". The text states: "Orion Star management wants to give a 10% bonus to each Australian Sales representative". The slide features icons of three people, a green dollar bill labeled "Bonus" with a red ribbon, a map of Australia with a blue person icon labeled "Rep", and a calendar for November showing days 1 through 30. A blue arrow points from the people icon to the bonus icon. A small number "14" is visible in the bottom left corner.

14

14

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** THE POWER TO KNOW.

## Considerations

Create a data set that includes the new variable, **Bonus**, which represents a 10% bonus.

15

15

**sas** THE POWER TO KNOW.

## Assignment Statement

The *assignment statement* evaluates an expression and assigns the result to a new or existing variable.

```
data work.subset1;
  set orion.sales;
  where Country='AU' and
        Job_Title contains 'Rep';
  Bonus=Salary*.10;
run;
```

variable=expression;

16

p106d02a

16

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas** THE POWER TO KNOW.

## Sample Assignment Statements

Example	Type
<code>Salary=26960 ;</code>	Numeric constant
<code>Gender='F' ;</code>	Character constant
<code>Hire_Date='21JAN1995'd;</code>	Date constant
<code>BonusMonth=month(Hire_Date) ;</code>	SAS function
<code>Bonus=Salary*.10 ;</code>	Arithmetic expression

17

17

**Sas** THE POWER TO KNOW.

## Viewing the Log

### Partial SAS Log

```

214  data work.subset1;
215    set orion.sales;
216    where Country='AU' and
217      Job_Title contains 'Rep' and
218      Hire_Date<'01jan2000'd;
219    Bonus=Salary*.10;
220  run;

```

**NOTE:** There were 29 observations read from the data set ORION.SALES.  
**WHERE** (Country='AU') and Job\_Title contains 'Rep' and  
(Hire\_Date<'01JAN2000'D);  
**NOTE:** The data set WORK.SUBSET1 has 29 observations and 10 variables.

The input data set has 9 variables, and the new data set has 10 variables.

18

18

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** THE POWER TO KNOW.

## Viewing the Output

```
proc print data=work.subset1 noobs;
  var First_Name Last_Name Salary
      Job_Title Bonus Hire_Date;
  format _Hire_Date date9.;
run;
```

Partial PROC PRINT Output

First_Name	Last_Name	Salary	Job_Title	Bonus	Hire_Date
Irenie	Elvish	26600	Sales Rep. II	2660.0	01JAN1978
Christina	Ngan	27475	Sales Rep. II	2747.5	01JUL1982
Kimiko	Hotstone	26190	Sales Rep. I	2619.0	01OCT1989
Lucian	Daymond	26480	Sales Rep. I	2648.0	01MAR1983
Fong	Hofmeister	32040	Sales Rep. IV	3204.0	01MAR1983

19 p106d02a

19

20

20

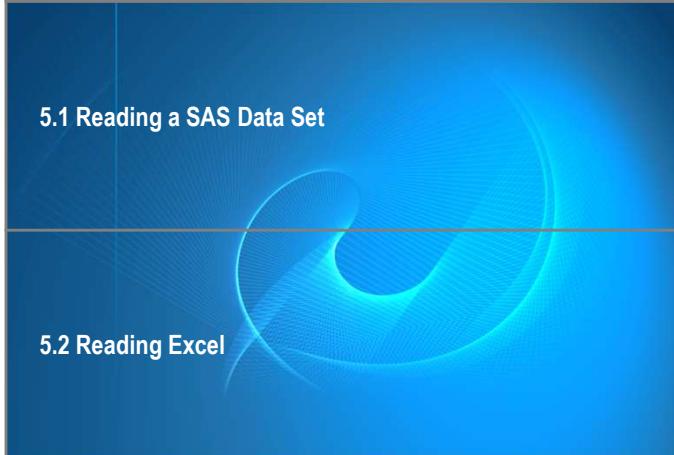
10



Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 5: Reading Data



5.1 Reading a SAS Data Set

5.2 Reading Excel

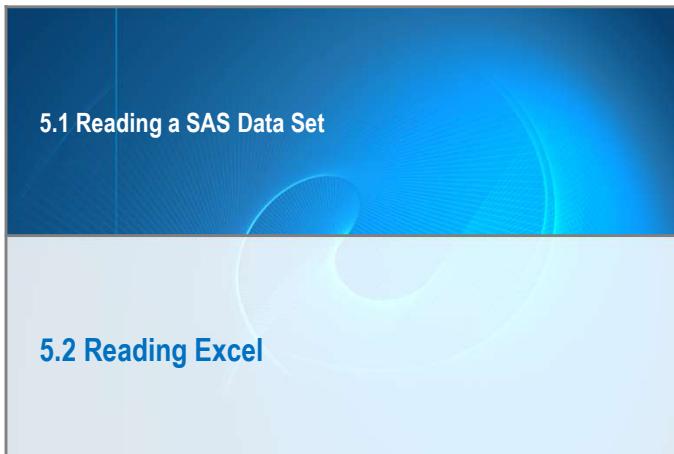
21

21

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas | THE POWER TO KNOW.

## Chapter 5: Reading SAS® Data Sets



5.1 Reading a SAS Data Set

5.2 Reading Excel

22

22

## Objectives

- Assign a libref to a Microsoft Excel workbook using a SAS/ACCESS LIBNAME statement.
- Access an Excel worksheet using a SAS two-level name.
- Create a SAS data set using a subset of worksheet data.

23

23

## Business Scenario

The Sales Manager has requested a report about Orion Star sales employees from Australia and the United States.

The input data is in an Excel workbook.



24

24

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Business Scenario**

Use SAS/ACCESS Interface to PC Files to read the worksheets within the **sales.xls** workbook as if they were SAS data sets.

25

25

**Examine the Workbook**

**Partial sales.xls**

**two worksheets**

**cells formatted as dates**

	A	B	C	D	E	F	G	H	I
1	Employee First Name	Last Name	Gender	Salary	Job Title	Country	Birth Date	Hire Date	
2	120102 Tom	Zhou	M	108255	Sales Manager	AU	11-Aug-73	6/1/1993	
3	120103 Wilson	Dawes	M	87975	Sales Manager	AU	22-Jan-53	1/1/1978	
4	120121 Irene	Elvish	F	26600	Sales Rep. II	AU	2-Aug-48	1/1/1978	
5	120122 Christina	Ngan	F	27475	Sales Rep. II	AU	27-Jul-58	7/1/1982	
6	120123 Kimiko	Hotstone	F	26190	Sales Rep. I	AU	28-Sep-68	10/1/1989	
7	120124 Lucien	Daymond	M	26480	Sales Rep. I	AU	13-May-63	3/1/1983	
8	120125 Fong	Hofmeister	M	32040	Sales Rep. IV	AU	6-Dec-58	3/1/1983	
9	120126 Satyakam	Denny	M	26780	Sales Rep. II	AU	20-Sep-92	8/1/2010	

26

26

**SAS Explorer Window**

SAS treats the workbook as a library, and each worksheet as a SAS data set.

The screenshot shows two windows. The left window is titled 'Explorer' and shows 'Active Libraries' with icons for Maps, Mapsgfk, Mapssas, Orion, Orionx (which is highlighted with a red box), and Sashelp. The right window is titled 'Explorer' and shows the 'Contents of 'Orionx'' with icons for Australia, Australia\$, UnitedStates, and UnitedStat...'. A line connects the 'Orionx' icon in the first window to the second window.

- A named range might exist for each worksheet.
- Worksheet names end with a dollar sign.
- Named ranges do **not** end with a dollar sign.

27

27

**CONTENTS Procedure**

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

**The CONTENTS Procedure**

**Directory**

Libref	ORIONX
Engine	PCFILES
Physical Name	s:\workshop\sales.xls
Schema/Owner	.

**Member DBMS Member**

#	Name	Type	Type
1	Australia	DATA	TABLE
2	Australia\$	DATA	SYSTEM TABLE
3	UnitedStates	DATA	TABLE
4	UnitedStates\$	DATA	SYSTEM TABLE

p107d01

28

28

The CONTENTS Procedure

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

Alphabetic List of Variables and Attributes

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

29

The CONTENTS Procedure

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

Alphabetic List of Variables and Attributes

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

30

30

Copyright © 2012, SAS Institute Inc. All rights reserved.

**SAS Name Literals**

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

SAS name literals permit special characters in data set names.

31

31

**Printing an Excel Worksheet**

```
libname orionx pcfiles path="&path\sales.xls";
proc print data=orionx.'Australia$n';
run;
```

Partial PROC PRINT Output

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title	Country	Birth_Date	Hire_Date	
1	120102	Tom	Zhou	M	108255	Sales Manager	AU	11AUG1973	01JUN1993
2	120103	Wilson	Dawes	M	87975	Sales Manager	AU	22JAN1953	01JAN1978
3	120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	02AUG1948	01JAN1978
4	120122	Christina	Ngan	F	27475	Sales Rep. II	AU	27JUL1958	01JUL1982
5	120123	Kimiko	Hotstone	F	26190	Sales Rep. I	AU	28SEP1968	01OCT1989

32

p107d01

32

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas** THE POWER TO KNOW.

## Subsetting Worksheet Data

You can select a subset of the worksheet data.

```
libname orionx pcfiles path="&path\sales.xls";
proc print data=orionx.'Australia'$ n noobs;
  where Job_Title ? 'IV';
  var Employee_ID Last_Name Job_Title Salary;
run;
```

33 p107d01

33

**Sas** THE POWER TO KNOW.

## Viewing the Output

PROC PRINT Output

Employee_ID	Last_Name	Job_Title	Salary
120125	Hofmeister	Sales Rep. IV	32040
120128	Kletschkus	Sales Rep. IV	30890
120135	Platts	Sales Rep. IV	32490
120159	Phoumirath	Sales Rep. IV	30765
120166	Nowd	Sales Rep. IV	30660

34

34

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas THE POWER TO KNOW.**

## Disassociating a Libref

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

```
libname orionx pcfiles path="&path\sales.xls";
/* program to access the worksheets */
libname orionx clear;
```

SAS disconnects from the data source and closes any resources associated with the connection.

35 p107d01

35

**Sas THE POWER TO KNOW.**

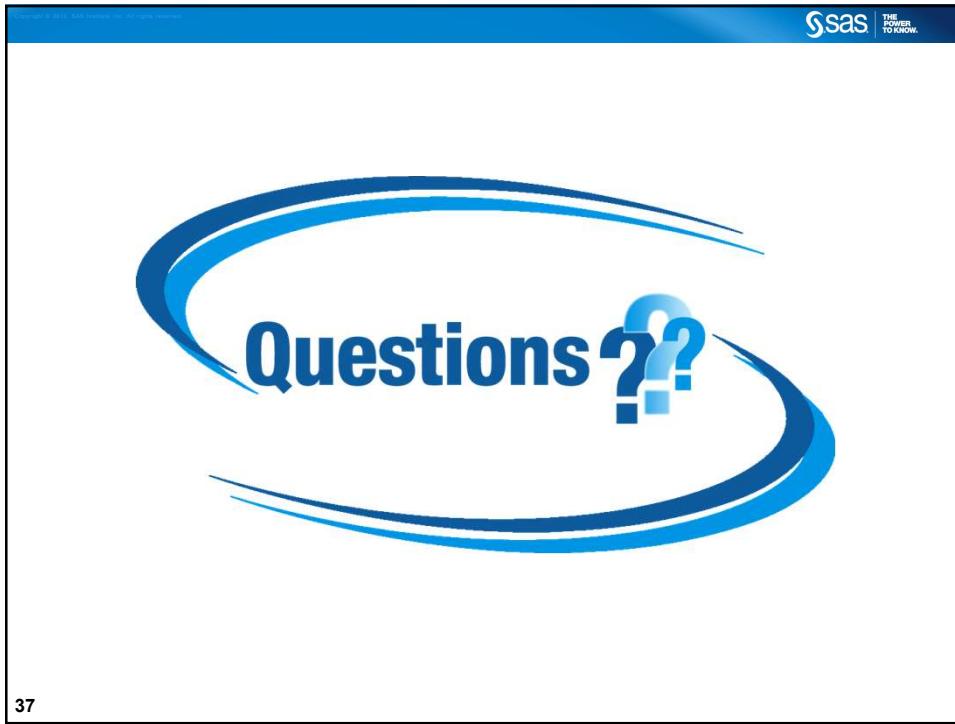


## Reading Excel Data

This demonstration illustrates reading from an Excel workbook using SAS Enterprise Guide and the SAS windowing environment.

36 p107d01

36



37

37

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas THE POWER TO KNOW.

## Chapter 6: Combining Data Sets



1

1

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas THE POWER TO KNOW.

## Chapter 6: Combining Data Sets



2

2

## Objectives

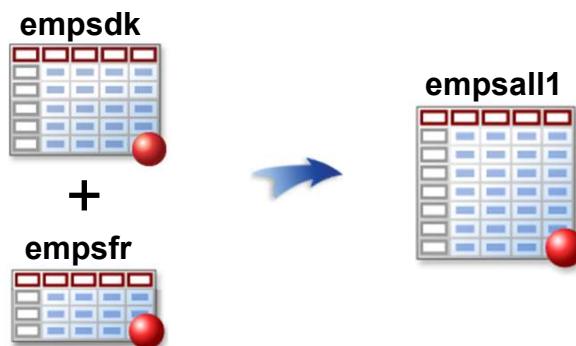
- Concatenate two or more SAS data sets using the SET statement in a DATA step.

3

3

## Business Scenario

You have been asked to combine the data sets containing information about Orion Star employees from Denmark and France into a new data set.



4

4

Copyright © 2012, SAS Institute Inc. All rights reserved.

**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.

5

**Using a DATA Step**

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.

6

p110d01

Copyright © 2012, SAS Institute Inc. All rights reserved.

Sas THE POWER TO KNOW.

# 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

7

p110d01

3

# 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;
```

Initialize PDV

PDV		
First	Gender	Country

## empsall1

8

—

8

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Lars	M	Denmark			

9 ...

9

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

Implicit OUTPUT;  
Implicit RETURN;

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Lars	M	Denmark	Lars	M	Denmark

10 ...

10

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Lars	M	Denmark	Lars	M	Denmark

Data set variables are not reinitialized.

11 ...

11

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Kari	F	Denmark	Lars	M	Denmark

12 ...

12

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

**Implicit OUTPUT;  
Implicit RETURN;**

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Kari	F	Denmark	Lars	M	Denmark
			Kari	F	Denmark

13 ...

13

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Kari	F	Denmark	Lars	M	Denmark
			Kari	F	Denmark

**PDV is not reinitialized.**

14 ...

14

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Jonas	M	Denmark	Lars	M	Denmark
			Kari	F	Denmark

15 ...

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

Implicit OUTPUT;  
Implicit RETURN;

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Jonas	M	Denmark	Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark

16 ...

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Jonas	M	Denmark	Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark

PDV is not reinitialized.

17 ...

17

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Jonas	M	Denmark	Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark

18 ...

18

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
			Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark

PDV is reinitialized before processing the next data set.

19 ...

19

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Pierre	M	France	Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark

20 ...

20

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

**Implicit OUTPUT;  
Implicit RETURN;**

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Pierre	M	France	Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark
			Pierre	M	France

21 ...

21

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Pierre	M	France	Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark
			Pierre	M	France

**PDV is not reinitialized.**

22 ...

22

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Sophie	F	France	Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark
			Pierre	M	France

23 ...

23

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

Implicit OUTPUT;  
Implicit RETURN;

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Sophie	F	France	Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark
			Pierre	M	France
			Sophie	F	France

24 ...

24

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Sophie	F	France	Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark
			Pierre	M	France
			Sophie	F	France
					...

25

PDV is not reinitialized.

25

Copyright © 2012, SAS Institute Inc., All Rights Reserved.

**sas** THE POWER TO KNOW.

## Execution

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

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

PDV			empsall1		
First	Gender	Country	First	Gender	Country
Sophie	F	France	Lars	M	Denmark
			Kari	F	Denmark
			Jonas	M	Denmark
			Pierre	M	France
			Sophie	F	France

26

26

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** THE POWER TO KNOW.

## 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.EMPSSL1 has 5 observations and 3 variables.

27

27

**sas** THE POWER TO KNOW.

## Unlike-Structured Data Sets

Concatenate **empscn** and **empsjp** to create a new data set named **empsall2**.

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

The data sets do not contain the same variables.

```
data empsall2;
  set empscn empsjp;
run;
```

28

p110d02

28

Copyright © 2012, SAS Institute Inc. All rights reserved.

**10.01 Short Answer Poll**

How many variables will be in **empsall2** after concatenating **empscn** and **empsjp**?

<b>empscn</b>			<b>empsjp</b>		
<b>First</b>	<b>Gender</b>	<b>Country</b>	<b>First</b>	<b>Gender</b>	<b>Region</b>
Chang	M	China	Cho	F	Japan
Li	M	China	Tomi	M	Japan
Ming	F	China			

```
data empsall2;
  set empscn empsjp;
run;
```

29

29

**10.01 Short Answer Poll – Correct Answer**

How many variables will be in **empsall2** after concatenating **empscn** and **empsjp**?

<b>empscn</b>			<b>empsjp</b>		
<b>First</b>	<b>Gender</b>	<b>Country</b>	<b>First</b>	<b>Gender</b>	<b>Region</b>
Chang	M	China	Cho	F	Japan
Li	M	China	Tomi	M	Japan
Ming	F	China			

**Four variables: First, Gender, Country, and Region**

30

30

Copyright © 2012, SAS Institute Inc. All rights reserved.

 THE POWER TO KNOW.

# 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 empscн empsjp;
run;
```

**PDV**

First	Gender	Country

31 p110d02

31

p110d02

Copyright © 2012, SAS Institute Inc. All rights reserved.

# Compilation

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

empsjp		
First	Gender	Region
Cho	F	Japan
Tomi	M	Japan

THE POWER  
TO KNOW.

```
data empsall2;
    set empscn empsjp;
run;
```

32

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas THE POWER TO KNOW.

## 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.

33

33

sas THE POWER TO KNOW.



Questions ??

34

34

Copyright © 2012, SAS Institute Inc. All rights reserved.

sas THE POWER TO KNOW.

## Chapter 6: Combining Data Sets

**6.1 Concatenating Data Sets**

**6.2 Merging Data Sets One-to-One**

35

35

sas THE POWER TO KNOW.

## Objectives

- Prepare data sets for merging using the SORT procedure.
- Merge SAS data sets one-to-one based on a common variable.

36

36

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** THE POWER TO KNOW.

## Match-Merging

**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**  
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.

**Non-matches**  
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.

37

**sas** THE POWER TO KNOW.

## Match-Merging: Sorting the Data Sets

The data sets in a match-merge must be sorted by the common variable or variables being matched.

```
PROC SORT DATA=input-SAS-data-set
    <OUT=output-SAS-data-set>;
    BY <DESCENDING> by-variable(s);
RUN;
```

38

Copyright © 2012, SAS Institute Inc. All rights reserved.

**sas** THE POWER TO KNOW.

## Business Scenario

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

empsau			phoneh	
First	Gender	EmpID	EmpID	Phone

+

empsauh			
First	Gender	EmpID	Phone

39

39

**sas** THE POWER TO KNOW.

## 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 Emp ID;
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.

40

p110d04

40

## 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.

41

41

## One-to-One Merge

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

**empsau**

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

**phoneh**

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

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

42

42

Copyright © 2012, SAS Institute Inc. All rights reserved.

**Sas THE POWER TO KNOW.**

## Final Results

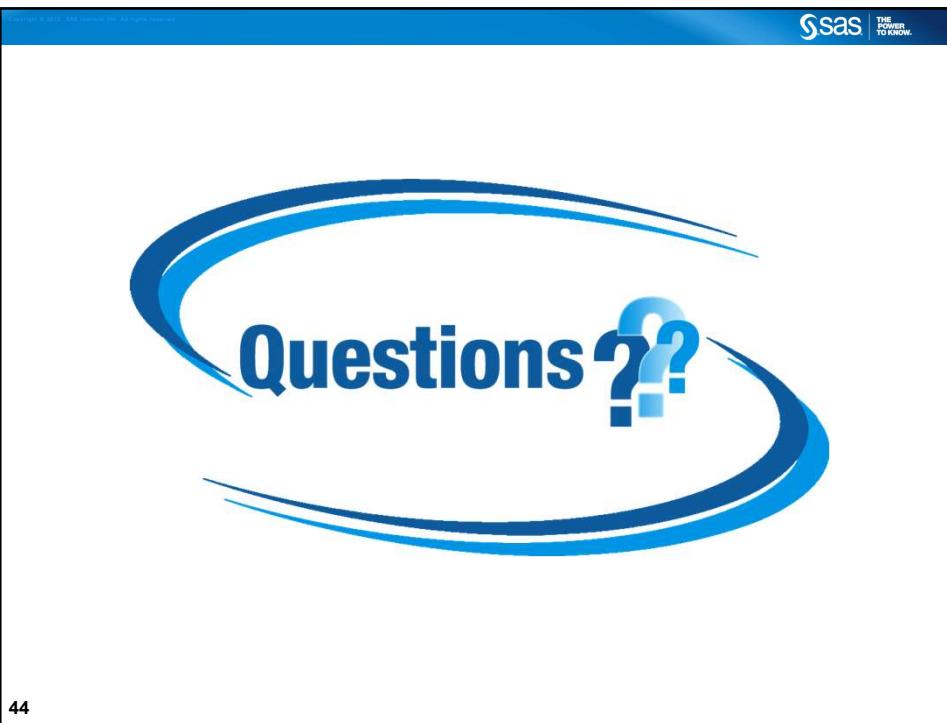
empsau			phoneh	
First	Gender	EmpID	EmpID	Phone
Togar	M	121150	121150	+61(2)5555-1793
Kylie	F	121151	121151	+61(2)5555-1849
Birin	M	121152	121152	+61(2)5555-1665

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

43

43



44



45

45