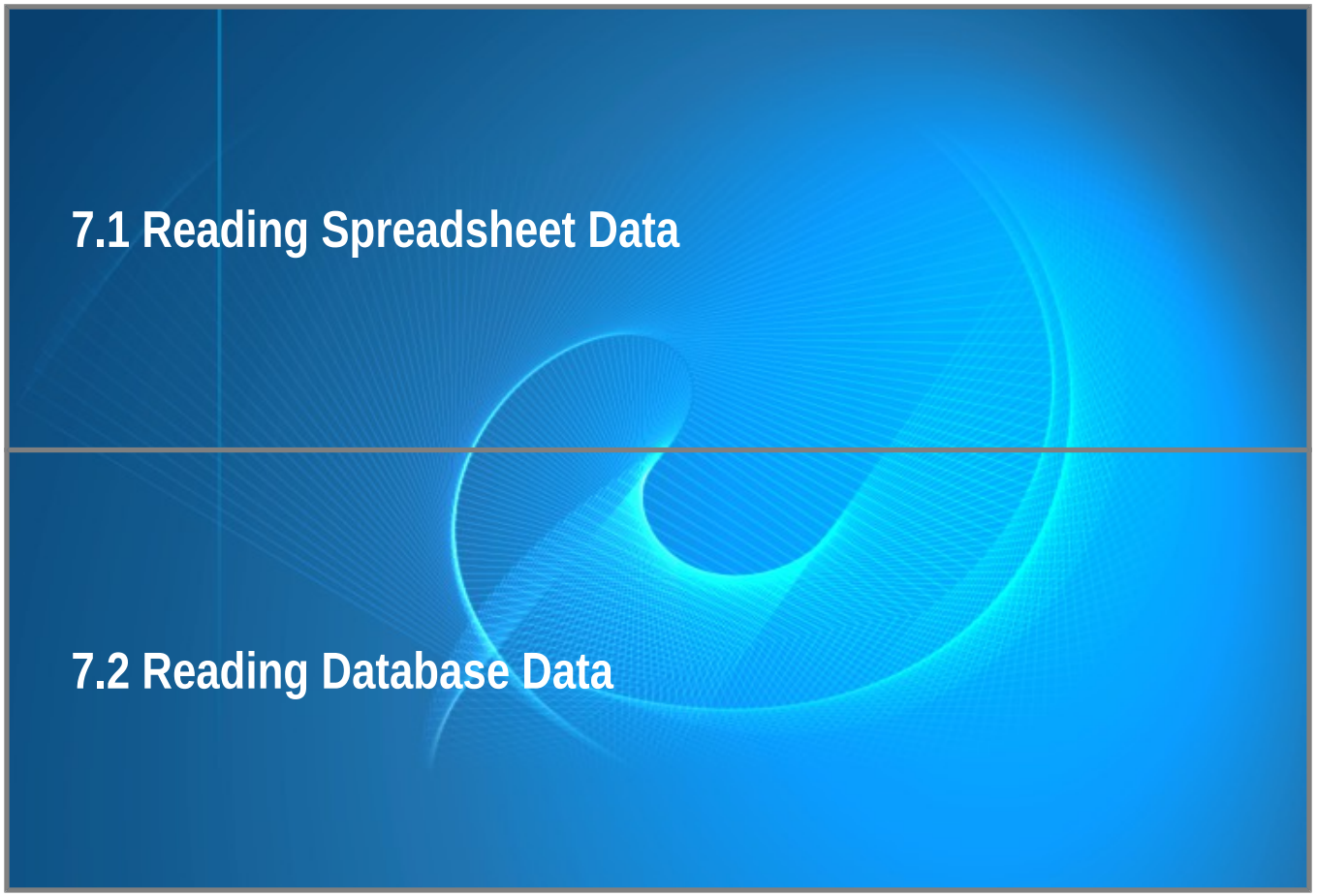
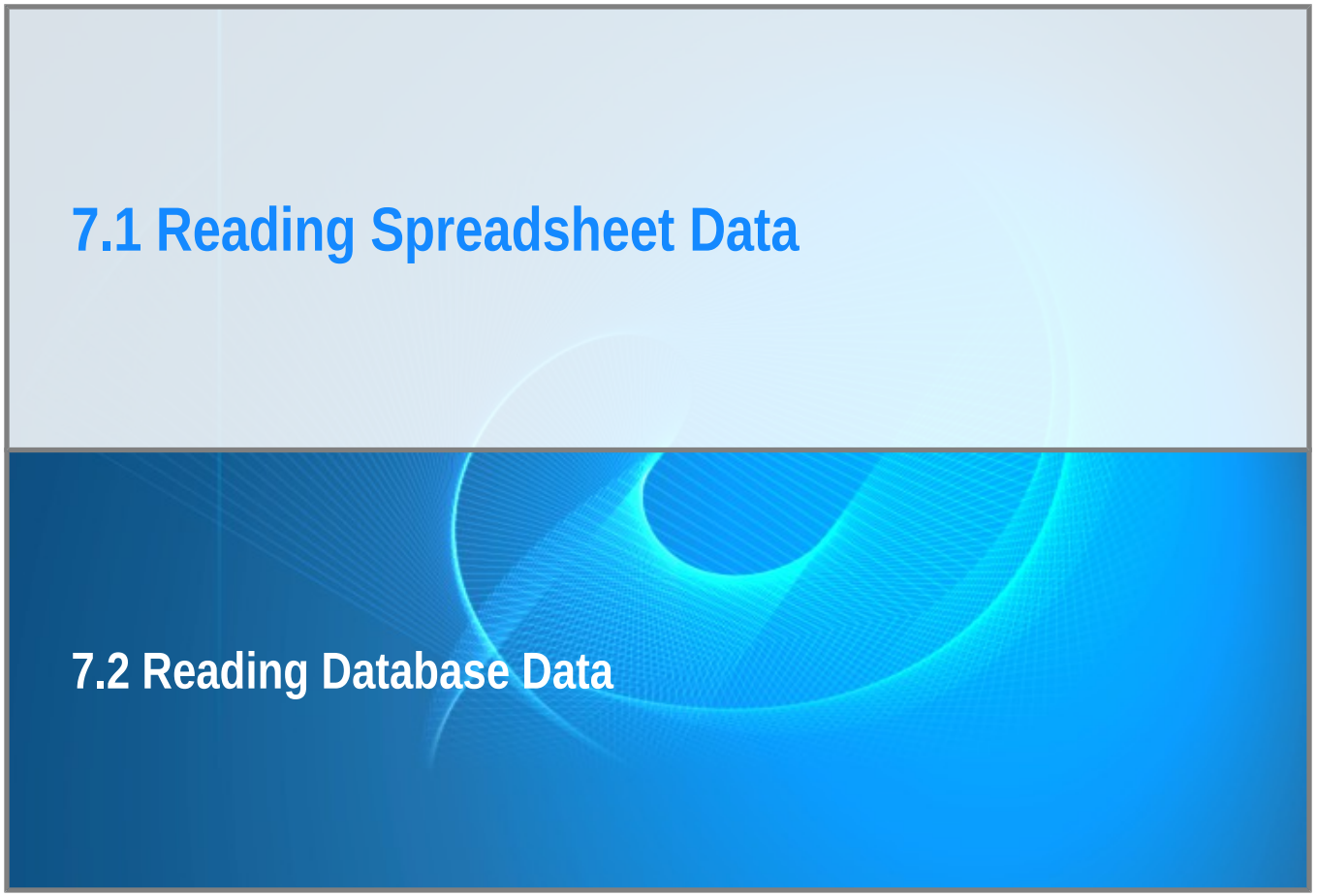
Chapter 7: Reading Spreadsheet and Database Data



Chapter 7: Reading Spreadsheet and Database Data



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.

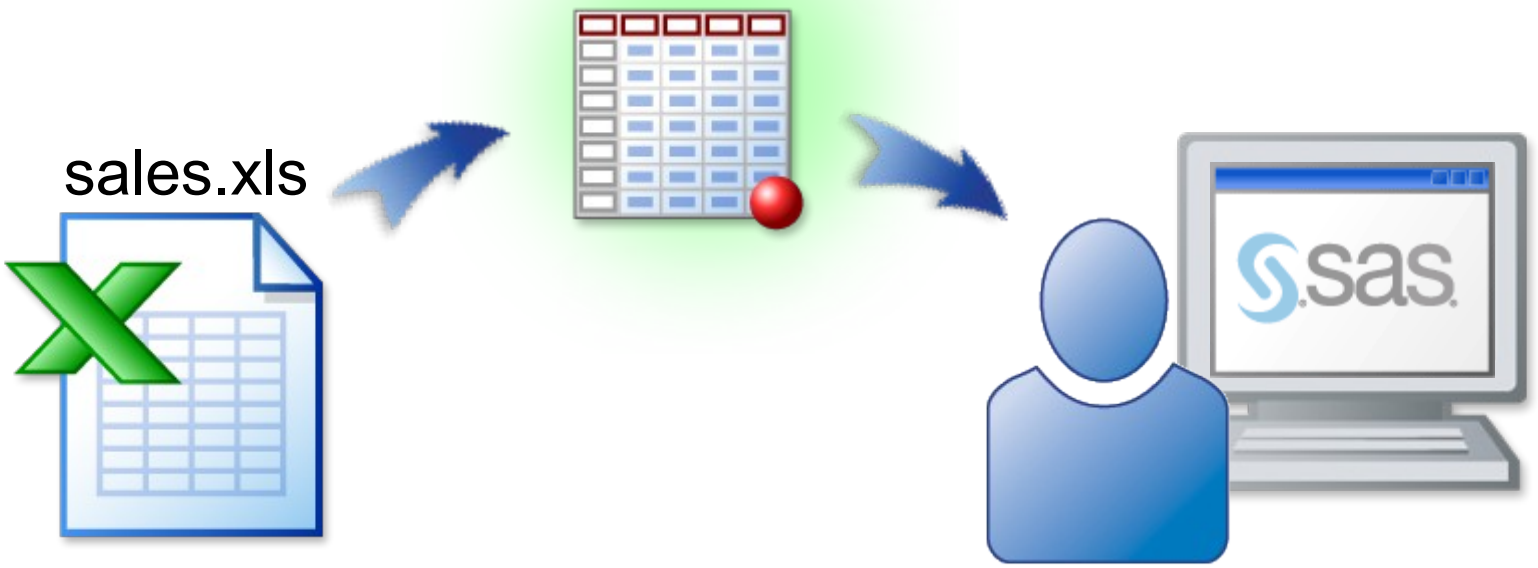
Business Scenario

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.

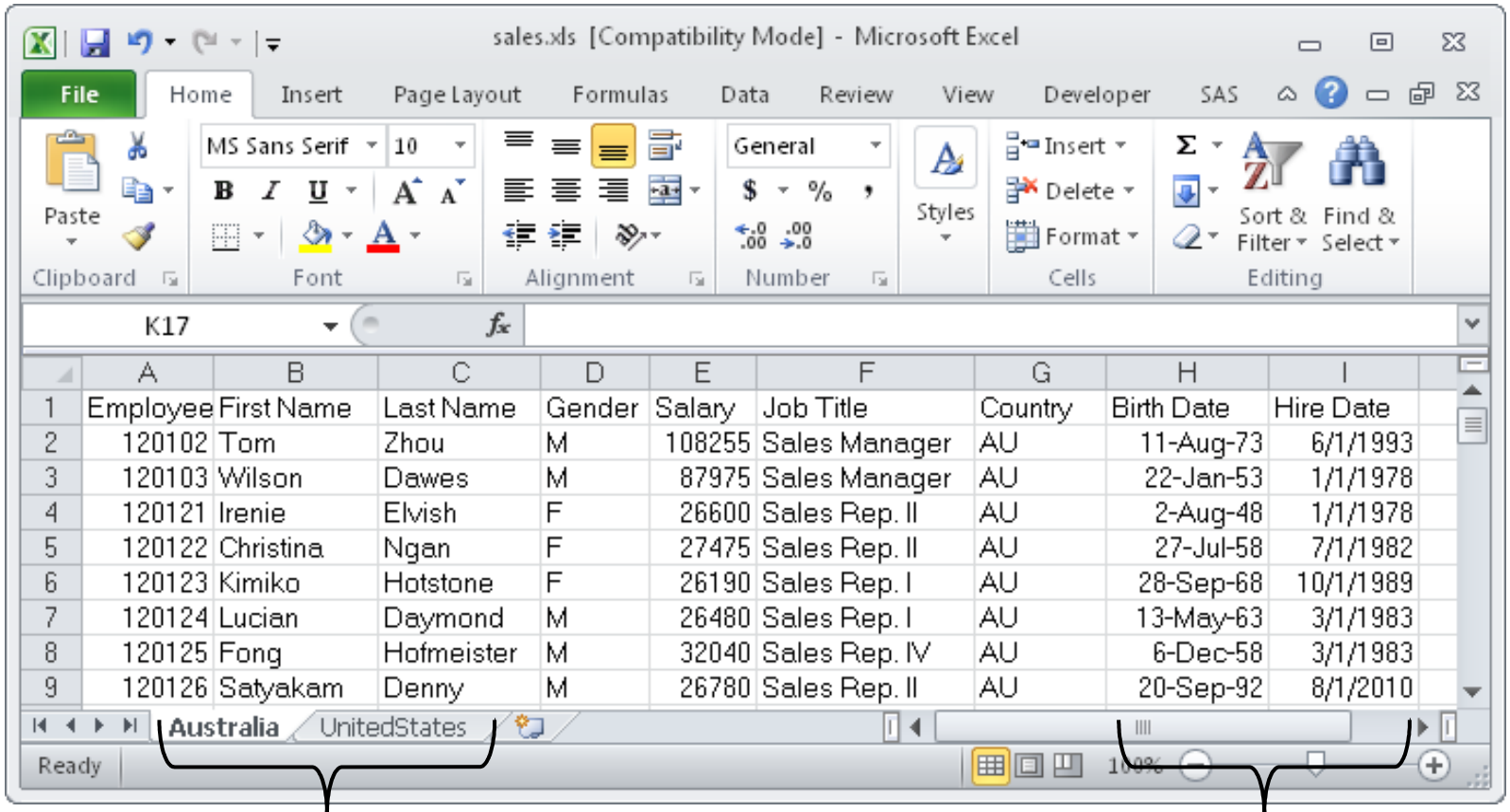
Business Scenario

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



Examine the Workbook

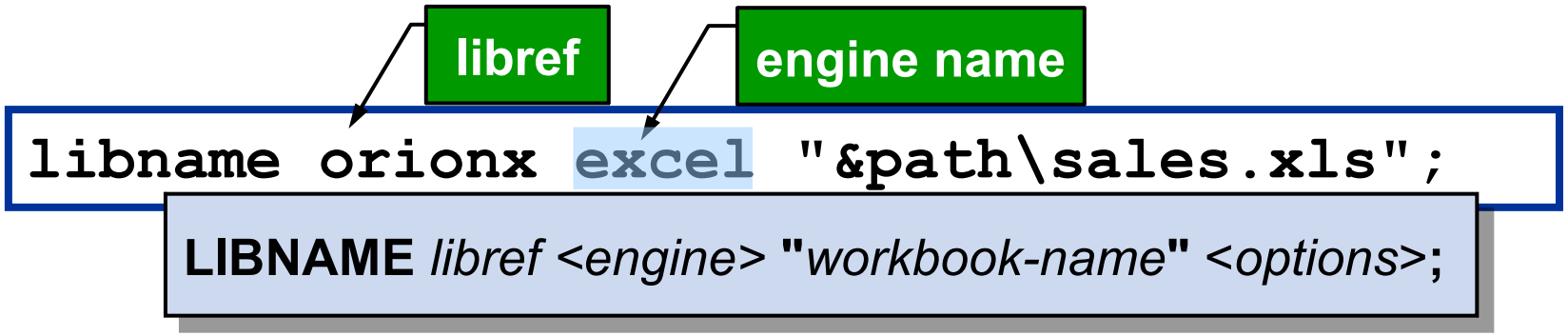
Partial **sales.xls**



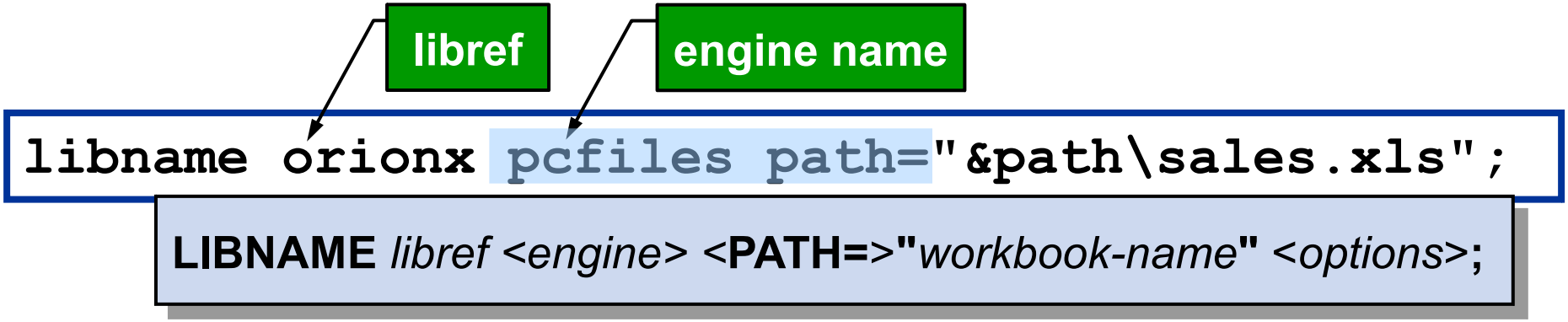
	A	B	C	D	E	F	G	H	I
1	Employee	First Name	Last Name	Gender	Salary	Job Title	Country	Birth Date	Hire Date
2	120102	Tom	Zhou	M	108255	Sales Manager	AU	11-Aug-73	6/1/1993
3	120103	Wilson	Dawes	M	87975	Sales Manager	AU	22-Jan-53	1/1/1978
4	120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	2-Aug-48	1/1/1978
5	120122	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	Lucian	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

SAS/ACCESS LIBNAME Statement

When the bit count of SAS and Microsoft Office are the same (both are 32-bit or both are 64-bit), you can use the default SAS/ACCESS Excel engine.

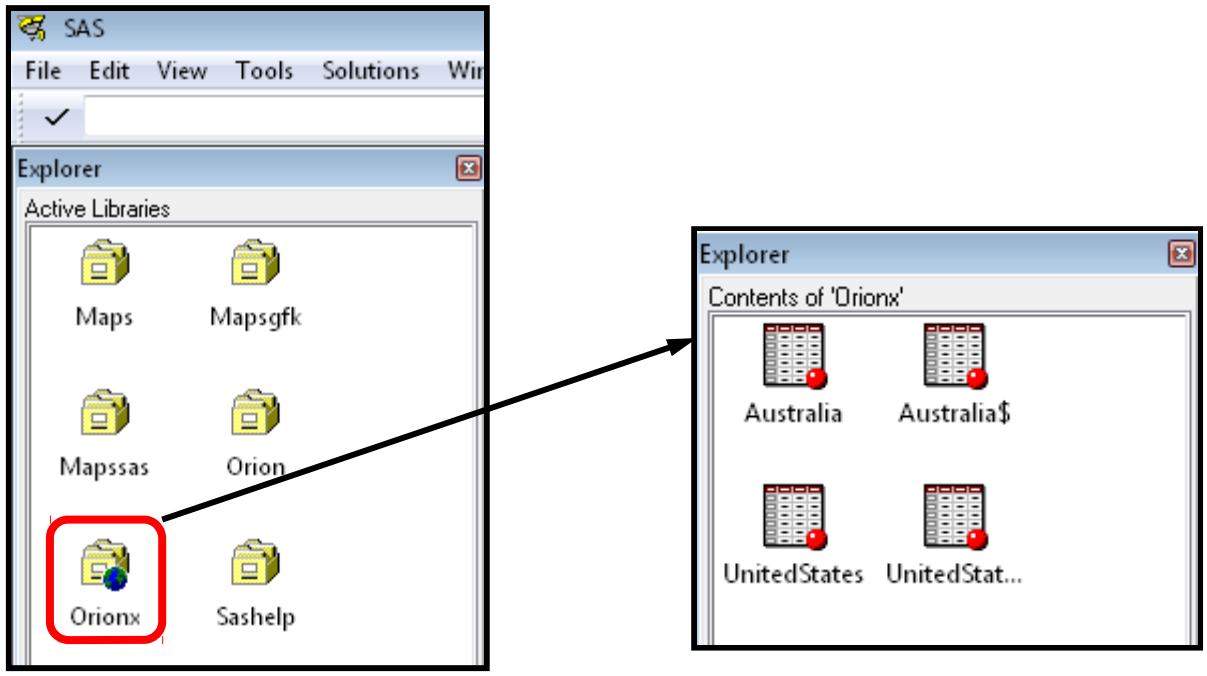


When the bit counts differ, use the PC Files Server.



SAS Explorer Window

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



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

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

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

The CONTENTS Procedure

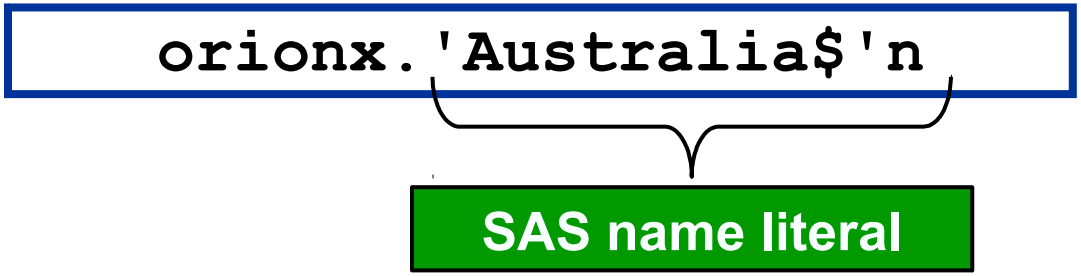
Data Set Name	ORIONX.'UnitedStates$'n	Observations	.
Member Type	DATA	Variables	9
Engine	PCFILES	Indexes	0
Created	.	Observation Length	0
Last Modified	.	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	Default		
Encoding	Default		

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

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.

Printing an Excel Worksheet

```
libname orionx pcfiles path="&path\sales.xls";

proc print data=orionx.'Australia$'n;
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	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

p107d01

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;
```


p107d01

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

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.

7.01 Quiz

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

a.
```
proc print data=orionx.'UnitedStates';
run;
```

b.
```
proc print data=orionx.'UnitedStates$';
run;
```

c.
```
proc print data=orionx.'UnitedStates'n;
run;
```

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

7.01 Quiz – Correct Answer

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

a.
```
proc print data=orionx.'UnitedStates';
run;
```

b.
```
proc print data=orionx.'UnitedStates$';
run;
```

c.
```
proc print data=orionx.'UnitedStates'n;
run;
```

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

Reading Excel Data

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

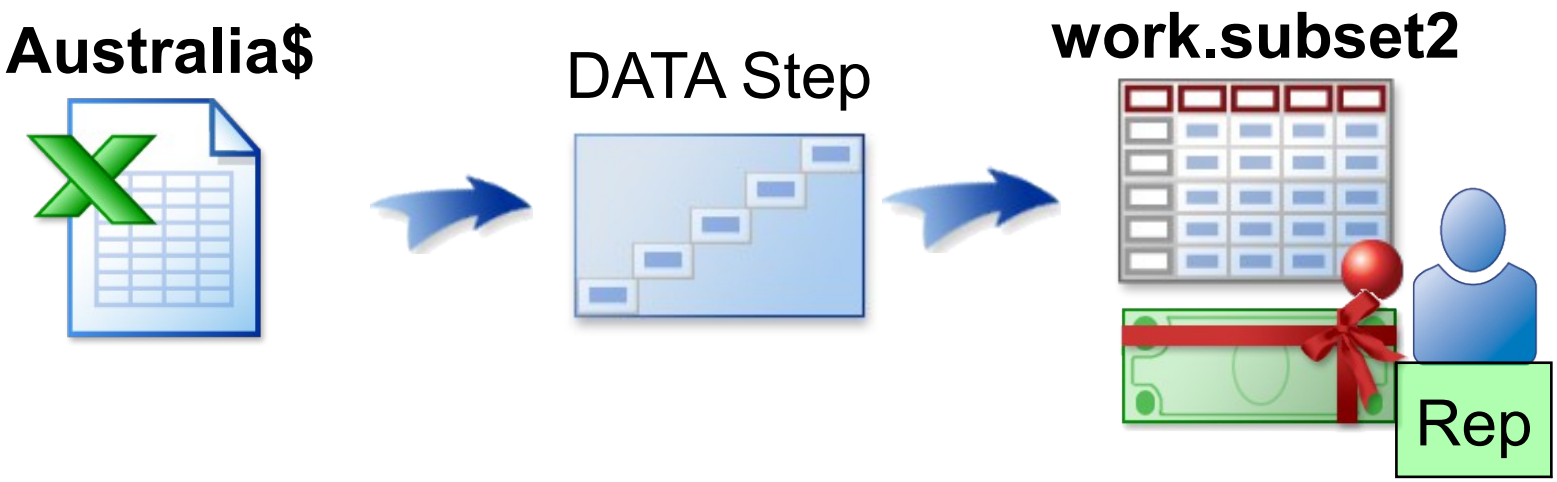
p107d01

Business Scenario

Create a SAS data set using a Microsoft Excel workbook as input.

Considerations

Use a SAS/ACCESS LIBNAME statement to read the **Australia$** worksheet and create a temporary data set.



The new data set should include the following:

- only the employees with **Rep** in their job title
- a **Bonus** variable that is 10% of **Salary**
- permanent labels and formats

7.02 Poll

A DROP or KEEP statement can be used to control which worksheet columns are written to the new data set.

- True
- False

7.02 Poll – Correct Answer

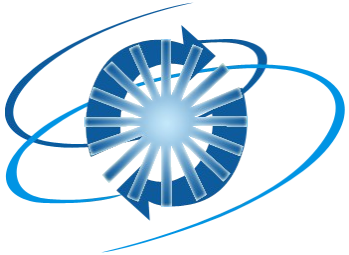
A DROP or KEEP statement can be used to control which worksheet columns are written to the new data set.

- True
- False

Creating a SAS Data Set

This demonstration illustrates reading from a Microsoft Excel workbook to generate a report and create a SAS data set.

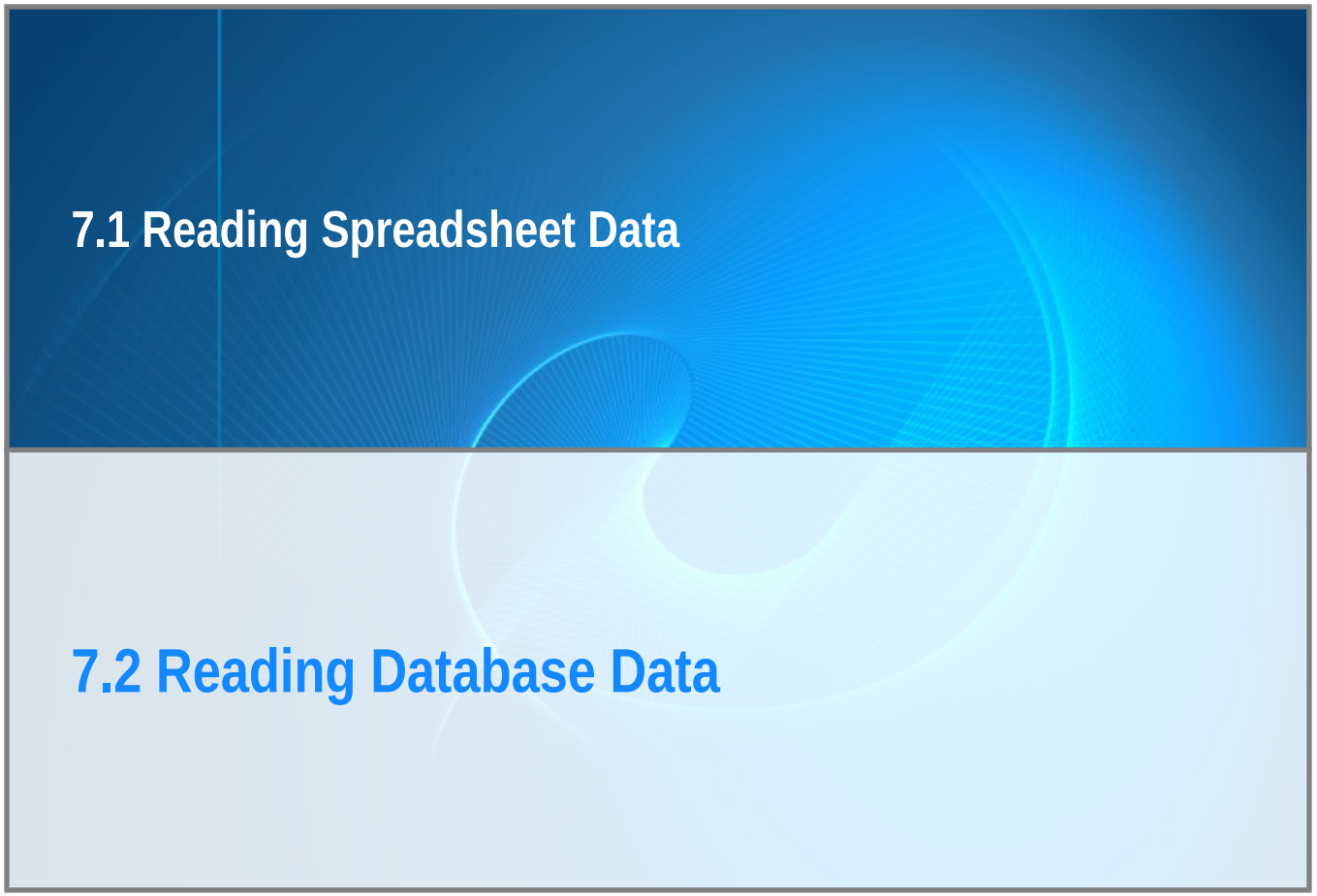
p107d02



Exercise

This exercise reinforces the concepts discussed previously.

Chapter 7: Reading Spreadsheet and Database Data



Objectives

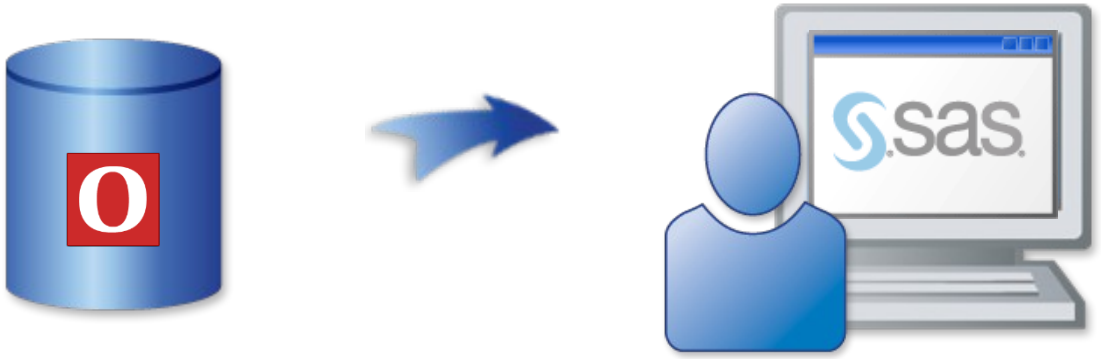
- Assign a libref to an Oracle database using a SAS/ACCESS LIBNAME statement.
- Access an Oracle table using a SAS two-level name.
- Create a SAS data set that contains a subset of an Oracle table.

Business Scenario

The Northeast Sales Manager requested a report listing supervisors from New York and New Jersey. The input data is in an Oracle database.

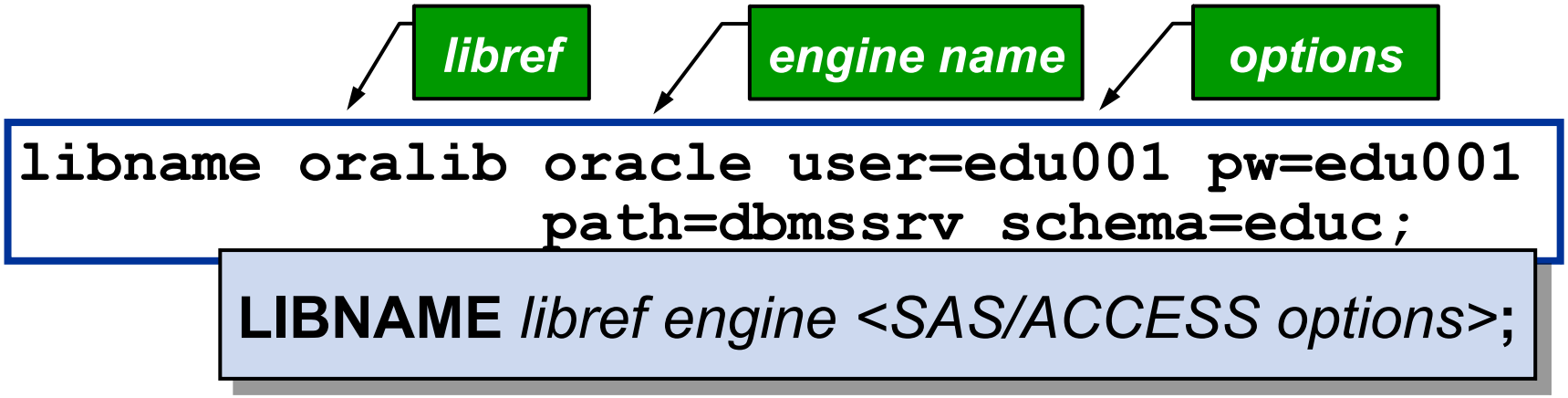
Business Scenario

Use SAS/ACCESS to read the tables within the database as if they were SAS data sets.



SAS/ACCESS LIBNAME Statement

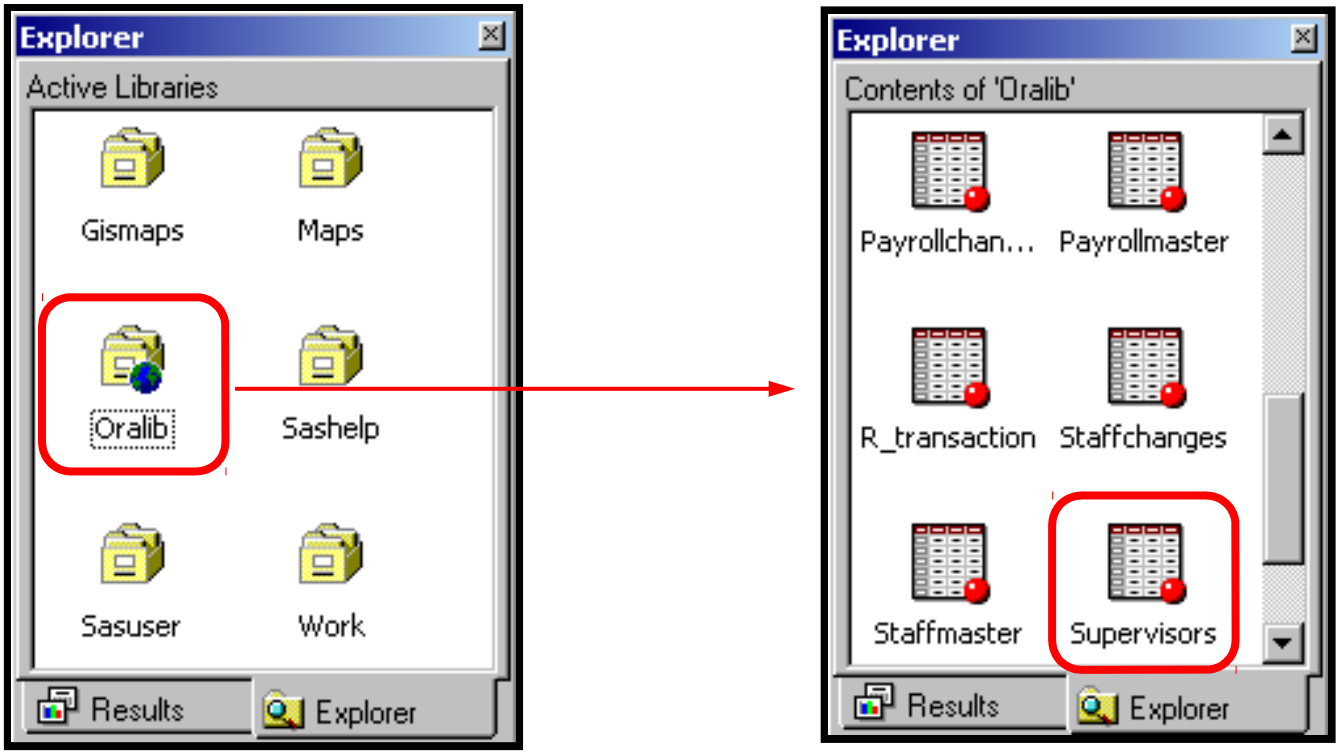
The SAS/ACCESS LIBNAME statement assigns a libref to a relational database.



This example uses the LIBNAME statement supported by the SAS/ACCESS Interface to Oracle.

Accessing an Oracle Database

The Oracle database is treated like a SAS library.



Any table in this Oracle database can be referenced using a SAS two-level name.

7.03 Multiple Choice Poll

Which of the following statements will read the Oracle table **supervisors** as if it were a SAS data set?

- input oralib.supervisors;
- input 'oralib.supervisors$'n;
- set oralib.supervisors;
- set 'oralib.supervisors$'n;

7.03 Multiple Choice Poll – Correct Answer

Which of the following statements will read the Oracle table **supervisors** as if it were a SAS data set?

- input oralib.supervisors;
- input 'oralib.supervisors$'n;
- set oralib.supervisors;
- set 'oralib.supervisors$'n;

Printing an Oracle Table

```
libname oralib oracle
        user=edu001 pw=edu001
        path=dbmssrv schema=educ;

proc print data=oralib.supervisors;
   where state in ('NY' 'NJ');
run;

libname oralib clear;
```

Remember to release the database and associated resources by submitting a LIBNAME statement that contains the CLEAR option.

Viewing the Output

Partial PROC PRINT Output

Obs	EMPID	STATE	JOBCATEGORY
1	1834	NY	BC
2	1433	NJ	FA
3	1983	NY	FA
4	1420	NJ	ME
5	1882	NY	ME

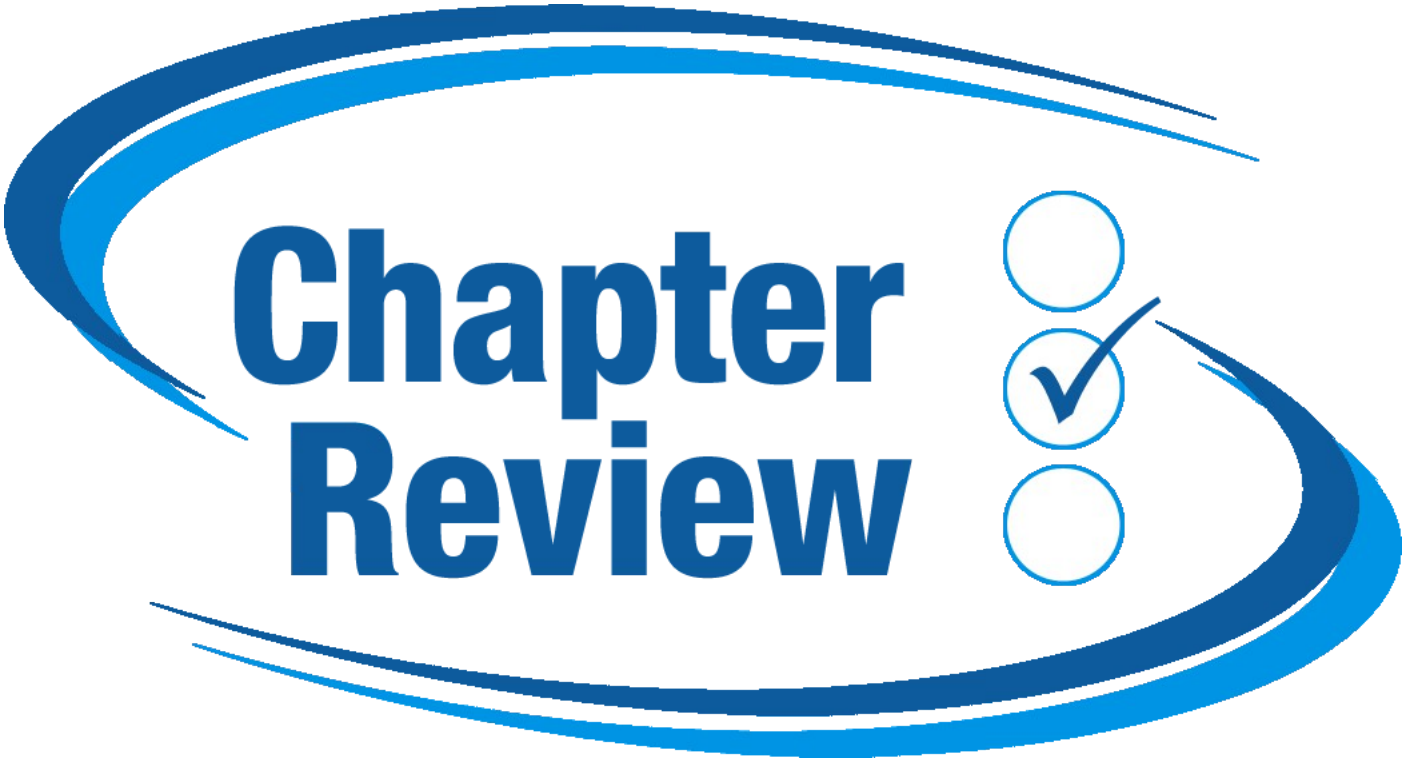
Creating a SAS Data Set

```
libname oralib oracle user=edu001
            pw=edu001 path=dbmssrv
schema=educ;

data nynjsup;
   set oralib.supervisors;
   where state in ('NY' 'NJ');
run;

proc print data=nynjsup;
run;

libname oralib clear;
```

Viewing the Output

PROC PRINT Output

Obs	EMPID	STATE	JOBCATEGORY
1	1834	NY	BC
2	1433	NJ	FA
3	1983	NY	FA
4	1420	NJ	ME
5	1882	NY	ME



1. In this SAS/ACCESS LIBNAME statement, what does **oracle** represent?

```
libname sports oracle
        user=edu001 pw=edu001
        path=dbmssrv schema=educ;
```

- libref
- option
- table name
- engine name

1. In this SAS/ACCESS LIBNAME statement, what does **oracle** represent?

```
libname sports oracle
        user=edu001 pw=edu001
        path=dbmssrv schema=educ;
```

- libref
- option
- table name
- engine name

2. Which PROC step successfully prints a list of all data sets in the **orionx** library without printing descriptor portions for the individual data sets?

   - proc contents data=orionx.nods _all_;
     run;
   - proc contents data=orionx._all_ nods;
     run;
   - proc print data=orionx._all_ noobs;
     run;
   - proc print data=orionx._all_ nods;
     run;

2. Which PROC step successfully prints a list of all data sets in the **orionx** library without printing descriptor portions for the individual data sets?

   - proc contents data=orionx.nods _all_;
     run;
   - proc contents data=orionx._all_ nods;
     run;
   - proc print data=orionx._all_ noobs;
     run;
   - proc print data=orionx._all_ nods;
     run;

3. When you use the SAS/ACCESS LIBNAME statement to assign a libref to a Microsoft Excel workbook, you can view the workbook using SAS Explorer or Microsoft Excel.

- True
- False

3. When you use the SAS/ACCESS LIBNAME statement to assign a libref to a Microsoft Excel workbook, you can view the workbook using SAS Explorer or Microsoft Excel.

- True
- False

4. When you use the SAS/ACCESS LIBNAME statement to assign a libref to a Microsoft Excel workbook, SAS treats each worksheet within the workbook as a library.

   - True
   - False

4. When you use the SAS/ACCESS LIBNAME statement to assign a libref to a Microsoft Excel workbook, SAS treats each worksheet within the workbook as a library.

- True
- False

5. What does the program shown here create?

```
libname sales excel 'c:\mydata\annual.xls';

data sales.qtr1_2011;
   set sasdata.qtr1_2011;
run;

data sales.qtr2_2011;
   set sasdata.qtr2_2011;
run;
```

- a data set named **sales.qtr1_2011** and a data set named **sales.qtr2_2011**
- an Excel workbook named **sales.qtr1_2011** and an Excel workbook named **sales.qtr2_2011**
- an Excel workbook named **annual** that contains two worksheets, **qtr1_2011** and **qtr2_2011**
- an Excel workbook named **sales** that contains two worksheets, **qtr1_2011** and **qtr2_2011**

5. What does the program shown here create?

```
libname sales excel 'c:\mydata\annual.xls';

data sales.qtr1_2011;
   set sasdata.qtr1_2011;
run;

data sales.qtr2_2011;
   set sasdata.qtr2_2011;
run;
```

- a data set named **sales.qtr1_2011** and a data set named **sales.qtr2_2011**
- an Excel workbook named **sales.qtr1_2011** and an Excel workbook named **sales.qtr2_2011**
- an Excel workbook named **annual** that contains two worksheets, **qtr1_2011** and **qtr2_2011**
- an Excel workbook named **sales** that contains two worksheets, **qtr1_2011** and **qtr2_2011**

6. What statement is used to read an Oracle table in a DATA step?

   a. DATA statement
   b. WHERE statement
   c. SET statement
   d. assignment statement

6. What statement is used to read an Oracle table in a DATA step?

a. DATA statement
b. WHERE statement
c. SET statement
d. assignment statement

7. A SAS name literal is a string that contains one or more special characters, enclosed in quotation marks, followed by the letter n.

   - True
   - False

7. A SAS name literal is a string that contains one or more special characters, enclosed in quotation marks, followed by the letter n.

- True
- False

8. When a date value is read from a worksheet, it is converted automatically to a SAS date. Which SAS format is used to display the value?

   - MMDDYY8.
   - MMDDYY10.
   - DATE7.
   - DATE9.

8. When a date value is read from a spreadsheet, it is converted automatically to a SAS date. Which SAS format is used to display the value?

   - MMDDYY8.
   - MMDDYY10.
   - DATE7.
   - DATE9.

9. When a SAS data set is created from a spreadsheet, the spreadsheet column headings are always stored as which of the following?

   - variable names
   - labels
   - formats
   - descriptor

9. When a SAS data set is created from a spreadsheet, the spreadsheet column headings are always stored as which of the following?

- variable names
- labels
- formats
- descriptor

10. A WHERE statement or a subsetting IF can be used to subset a Microsoft Excel worksheet.

   - True
   - False

10. A WHERE statement or a subsetting IF can be used to subset a Microsoft Excel worksheet.

- True
- False