



CERTSWARRIOR

SAS Institute

A00-231

**SAS Certified Specialist: Base Programming Using SAS®
9.4**

Questions&AnswersPDF

ForMoreInformation:

<https://www.certswarrior.com/>

Features:

- 90DaysFreeUpdates
- 30DaysMoneyBackGuarantee
- InstantDownloadOncePurchased
- 24/7OnlineChat Support
- ItsLatestVersion

Latest Version: 6.0

Question: 1

You have a SAS dataset called 'SALES' with variables 'REGION', 'PRODUCT', and 'SALES AMOUNT'. You need to create a new dataset 'SALES SUMMARY' that contains the total sales amount for each region. Which of the following DATA step code snippets will achieve this?

- A.

```
data SALES_SUMMARY; set SALES; by REGION; if last.REGION then sum_sales = sum(SALES_AMOUNT); run;
```
- B.

```
data SALES_SUMMARY; set SALES; sum_sales = sum(SALES_AMOUNT); by REGION; run;
```
- C.

```
data SALES_SUMMARY; set SALES; by REGION; sum_sales + SALES_AMOUNT; if last.REGION then output; run;
```
- D.

```
data SALES_SUMMARY; set SALES; sum_sales + SALES_AMOUNT; if last.REGION then output; run;
```
- E.

```
data SALES_SUMMARY; set SALES; by REGION; sum_sales = sum(SALES_AMOUNT); if last.REGION then output; run;
```

Answer: E

Explanation:

The 'by REGION' statement groups the data by region. The 'sum_sales = sum(SALES_AMOUNT)' statement calculates the cumulative sum of 'SALES_AMOUNT' for each region. The 'if last.REGION then output' statement outputs the sum of 'SALES_AMOUNT' only when the last observation for a region is encountered, ensuring that the total sales amount for each region is captured in the new dataset.

Question: 2

You have a SAS dataset 'CUSTOMERS' with variables 'CUSTOMER ID', 'NAME', and 'CITY'. You need to create a new dataset 'ACTIVE CUSTOMERS' containing only customers who have made a purchase in the last 6 months. The dataset 'SALES' contains customer information and purchase date ('PURCHASE DATE'). How can you create 'ACTIVE CUSTOMERS' dataset using the 'SALES' and 'CUSTOMERS' datasets?

- A.

```
data ACTIVE_CUSTOMERS; set SALES; if PURCHASE_DATE >= intnx('month', today(), -6) then output; run;
```
- B.

```
data ACTIVE_CUSTOMERS; merge CUSTOMERS SALES; by CUSTOMER_ID; if PURCHASE_DATE >= intnx('month', today(), -6) then output; run;
```
- C.

```
data ACTIVE_CUSTOMERS; merge CUSTOMERS SALES; by CUSTOMER_ID; if PURCHASE_DATE >= intnx('month', today(), -6) then output; run;
```
- D.

```
data ACTIVE_CUSTOMERS; set CUSTOMERS; if CUSTOMER_ID in (select CUSTOMER_ID from SALES where PURCHASE_DATE >= intnx('month', today(), -6)) then output; run;
```

E.

```
data ACTIVE_CUSTOMERS; set CUSTOMERS; if CUSTOMER_ID in (select CUSTOMER_ID from SALES where PURCHASE_DATE >= intnx('month', today(), -6)); run;
```

Answer: D

Explanation:

This code snippet first filters the 'SALES' dataset to include only purchases made in the last 6 months using the 'intnx' function to calculate the date 6 months ago from today. Then, the 'select CUSTOMER ID' statement retrieves the 'CUSTOMER_ID' values from the filtered 'SALES' dataset. Finally, the 'if CUSTOMER_ID in (0)' statement checks if the 'CUSTOMER_ID' from the 'CUSTOMERS' dataset exists in the retrieved list of 'CUSTOMER_ID' values. If it does, the observation is output to the new dataset 'ACTIVE CUSTOMERS'.

Question: 3

You have a SAS dataset called 'EMPLOYEES' with variables 'EMPLOYEE ID', 'DEPARTMENT', and 'SALARY'. You need to create a new dataset called 'HIGH EARNERS' that contains only employees whose salary is in the top 10% for their department. Which of the following SAS DATA step code snippets would achieve this? (Assume there are no missing values in the 'SALARY' variable.)

A.

```
data HIGH_EARNERS; set EMPLOYEES; by DEPARTMENT; if SALARY >= quantile('SALARY', 0.9, 'DEPARTMENT') then output; run;
```

B.

```
data HIGH_EARNERS; set EMPLOYEES; by DEPARTMENT; if SALARY >= quantile('SALARY', 0.1, 'DEPARTMENT') then output; run;
```

C.

```
data HIGH_EARNERS; set EMPLOYEES; by DEPARTMENT; if rank(SALARY) >= 0.1 * count(SALARY) then output; run;
```

D.

```
data HIGH_EARNERS; set EMPLOYEES; by DEPARTMENT; if rank(SALARY) >= 0.9 * count(SALARY) then output; run;
```

E.

```
data HIGH_EARNERS; set EMPLOYEES; by DEPARTMENT; if SALARY >= quantile('SALARY', 0.9, 'DEPARTMENT') then output; run;
```

Answer: A

Explanation:

The code uses the 'quantile' function to calculate the 90th percentile of 'SALARY' for each department. Then, the 'if SALARY >= quantile(...)' statement filters the dataset to include only employees whose salary is greater than or equal to the 90th percentile. The 'by DEPARTMENT' statement ensures that the percentile calculation is performed separately for each department. Option B uses the 10th percentile, which would identify the bottom 10% of earners. Options C and D use the 'rank' function to determine the rank of each employee's salary within their department, but they do not accurately identify the top 10%.

Question: 4

You have two SAS datasets: 'ORDERS' with variables 'ORDER ID', 'CUSTOMER D', and 'ORDER DATE', and 'CUSTOMERS' with variables 'CUSTOMER D', 'NAME', and 'CITY'. You want to create a new dataset 'ORDERS WITH CUSTOMER INFO' that contains all the information from 'ORDERS' and the 'NAME' and 'CITY' variables from 'CUSTOMERS' for each matching customer. Which code snippet will correctly achieve this?

A.

```
data ORDERS_WITH_CUSTOMER_INFO; merge ORDERS CUSTOMERS; by CUSTOMER_ID; run;
```

B.

```
data ORDERS_WITH_CUSTOMER_INFO; set ORDERS; if CUSTOMER_ID in (select CUSTOMER_ID from CUSTOMERS) then output; run;
```

C.

```
data ORDERS_WITH_CUSTOMER_INFO; set ORDERS CUSTOMERS; run;
```

D.

```
data ORDERS_WITH_CUSTOMER_INFO; set ORDERS; set CUSTOMERS; output; run;
```

E.

```
data ORDERS_WITH_CUSTOMER_INFO; set ORDERS; if CUSTOMER_ID in (select CUSTOMER_ID from CUSTOMERS) then output; run;
```

Answer: A

Explanation:

The 'merge' statement combines the 'ORDERS' and 'CUSTOMERS' datasets based on the matching 'CUSTOMER ID'. This ensures that all the variables from both datasets are included in the new dataset 'ORDERS_WITH CUSTOMER INFO'. Option B only filters 'ORDERS' based on the existence of a matching 'CUSTOMER ID' in 'CUSTOMERS' but doesn't include the 'NAME' and 'CITY' variables. Options C and D either combine the datasets without merging based on 'CUSTOMER ID' or create duplicates. Option E is similar to option B, not including the extra variables.

Question: 5

You have a SAS dataset named 'CUSTOMER DATA' located in the directory '/sasdata/marketing/'. You need to access this dataset in your SAS program. Which LIBNAME statement would correctly assign a library reference name and make the dataset accessible?

- A. LIBNAME mylib '/sasdata/marketing/CUSTOMER_DATA';
- B. LIBNAME mylib '/sasdata/marketingr; DATA mylib_CUSTOMER_DATA;
- C. LIBNAME mylib '/sasdata/marketingr;
- D. LIBNAME mylib 'CUSTOMER_DATA';
- E. LIBNAME mylib Ysasdata/marketing/CUSTOMER_DATAs7bdaf;

Answer: C

Explanation:

The correct LIBNAME statement is "LIBNAME mylib '/sasdata/marketingr;". This statement assigns the library reference name 'mylib' to the directory '/sasdata/marketing/'. You can then access the dataset

'CUSTOMER DATA' using the fully qualified name 'mylib.CUSTOMER DATA'. Option A is incorrect because it tries to assign the library reference name to the dataset name directly. Option B is incorrect because it is creating a new dataset instead of accessing the existing one. Option D is incorrect because it is missing the directory path. Option E is incorrect because it specifies the file extension '.sas7bdat', which is not necessary in the LIBNAME statement.

Question: 6

You want to create a SAS library named 'TEMP DATA' that points to a directory called '/user/temp/sasdata'. The library should be accessible only by your user account. How would you accomplish this using the LIBNAME statement?

- A. LIBNAME TEMP_DATA 'fuser/temp/sasdata' DISK USER;
- B. LIBNAME TEMP DATA '/user/temp/sasdata' DISK OWNER,
- C. LIBNAME TEMP DATA '/user/temp/sasdata' DISK PUBLIC;
- D. LIBNAME TEMP_DATA '/user/temp/sasdata' DISK PRIVATE;
- E. LIBNAME TEMP_DATA '/user/temp/sasdata' DISK SYSTEM;

Answer: B

Explanation:

The correct LIBNAME statement is "LIBNAME TEMP_DATA '/user/temp/sasdata' DISK OWNER;" The DISK OWNER option restricts access to the library to the user who created it. This ensures that only your user account can create, modify or delete data within the TEMP_DATA library. Option A (DISK USER) is incorrect because it makes the library accessible to all users in the user group. Option C (DISK PUBLIC) is incorrect because it makes the library accessible to all users on the system. Option D (DISK PRIVATE) does not exist in SAS. Option E (DISK SYSTEM) is incorrect because it creates a system-level library accessible to all users.

Question: 7

You are working on a SAS program where you need to use two datasets: 'SALES DATA' and 'CUSTOMER DATA'. Both datasets reside in a directory called '/sasdata/marketing/'. Instead of writing the full path each time, you want to create a library reference name. Which LIBNAME statement will allow you to access these datasets using a shorter name?

- A.
`LIBNAME mylib '/sasdata/marketing/'; DATA mylib.SALES_DATA; DATA mylib.CUSTOMER_DATA;`
- B.
`LIBNAME mylib '/sasdata/marketing/'; DATA mylib.SALES_DATA; RUN; DATA mylib.CUSTOMER_DATA; RUN;`
- C.
`LIBNAME mylib '/sasdata/marketing/'; DATA mylib.SALES_DATA; RUN; DATA CUSTOMER_DATA; RUN;`
- D.
`LIBNAME mylib '/sasdata/marketing/'; DATA SALES_DATA; RUN; DATA CUSTOMER_DATA; RUN;`

E.

```
LIBNAME mylib '/sasdata/marketing/SALES_DATA'; LIBNAME mylib '/sasdata/marketing/CUSTOMER_DATA'; DATA mylib.SALES_DATA; DATA mylib.CUSTOMER_DATA;
```

Answer: A

Explanation:

The correct LIBNAME statement is "LIBNAME mylib '/sasdata/marketing/';" followed by "DATA mylib.SALES_DATA;" and "DATA mylib.CUSTOMER_DATA;" This statement assigns the library reference name 'mylib' to the directory '/sasdata/marketing/' making both datasets accessible using the shorter names 'mylib.SALES_DATA' and 'mylib.CUSTOMER_DATA'. Option B is similar to Option A, but it uses the RUN statement after each DATA step, which is not necessary in this case. Option C is incorrect because it attempts to access 'CUSTOMER DATA' without the library reference name, which will not work. Option D is incorrect because it accesses 'SALES_DATA' and 'CUSTOMER DATA' without the library reference name, resulting in an error. Option E is incorrect because it assigns the library reference name to the datasets directly and attempts to access them with the same library reference name, which leads to confusion.

Question: 8

You are trying to access a SAS dataset named 'SALES HISTORY' located in a library named 'MY LIBRARY'. Which LIBNAME statement would correctly assign the library reference name if the library is stored in the '/sasdata/sales/' directory?

A.

```
LIBNAME MY_LIBRARY '/sasdata/sales/'; DATA MY_LIBRARY.SALES_HISTORY;
```

B.

```
LIBNAME MY_LIBRARY '/sasdata/sales/SALES_HISTORY'; DATA MY_LIBRARY.SALES_HISTORY;
```

C.

```
LIBNAME MY_LIBRARY '/sasdata/sales/SALES_HISTORY.sas7bdat'; DATA MY_LIBRARY.SALES_HISTORY;
```

D.

```
LIBNAME MY_LIBRARY '/sasdata/sales/'; DATA SALES_HISTORY;
```

E.

```
LIBNAME MY_LIBRARY '/sasdata/sales'; DATA SALES_HISTORY;
```

Answer: A

Explanation:

The correct LIBNAME statement is "LIBNAME MY_LIBRARY '/sasdata/sales/';" followed by "DATA MY_LIBRARY.SALES_HISTORY;" This statement assigns the library reference name 'MY_LIBRARY' to the directory 'sasdata/sales'. You can then access the dataset 'SALES_HISTORY' using the fully qualified name 'MY_LIBRARY.SALES_HISTORY'. Option B is incorrect because it tries to assign the library reference name to the dataset name directly, which is not valid. Option C is incorrect because it specifies the file

extension 'sas7bdat', which is not necessary in the LIBNAME statement Option D is incorrect because it attempts to access the dataset without the library reference name. Option E is incorrect because it specifies an incomplete directory path.

Question: 9

You have a SAS data library called 'WORK.CUSTOMERS' containing customer data

a. You want to identify the data types of each variable and store them in a separate dataset called 'WORK.VARIABLE TYPES'. Which code snippet would you use to achieve this?

A.

```
proc contents data=WORK.CUSTOMERS out=WORK.VARIABLE_TYPES; run;
```

B.

```
proc contents data=WORK.CUSTOMERS out=WORK.VARIABLE_TYPES (keep=name type); run;
```

C.

```
proc contents data=WORK.CUSTOMERS out=WORK.VARIABLE_TYPES (drop=name type); run;
```

D.

```
proc contents data=WORK.CUSTOMERS out=WORK.VARIABLE_TYPES (keep=name type length); run;
```

E.

```
proc contents data=WORK.CUSTOMERS out=WORK.VARIABLE_TYPES (keep=name label type); run;
```

Answer: B

Explanation:

The 'PROC CONTENTS' procedure can output a dataset with information about the variables in the input dataset. To obtain only the variable name and type, use the 'KEEP' option in the 'OUT' statement The 'OUT' statement specifies the output dataset The 'KEEP' option instructs the procedure to include only the specified variables in the output dataset Options A, C, D, and E are incorrect because they either include all variables or do not include the necessary variables for determining the data types.

Question: 10

You have a SAS data set named 'CUSTOMERS' with variables 'CUSTOMER ID', 'NAME', and 'CITY'. You want to create a new data set named 'ACTIVE CUSTOMERS' that contains only customers from the city 'New York'. Which of the following code snippets would achieve this correctly?

A.

```
data ACTIVE_CUSTOMERS;  
set CUSTOMERS;  
if CITY = 'New York';  
run;
```

B.

```
data ACTIVE_CUSTOMERS;  
set CUSTOMERS;  
where CITY = 'New York';  
run;
```

C.

```
data ACTIVE_CUSTOMERS;  
input CUSTOMER_ID NAME CITY;  
infile 'CUSTOMERS.dat';  
if CITY = 'New York';  
run;
```

D.

```
data ACTIVE_CUSTOMERS;  
set CUSTOMERS;  
output;  
run;
```

E.

```
data ACTIVE_CUSTOMERS;  
set CUSTOMERS;  
CITY = 'New York';  
run;
```

Answer: B

Explanation:

The WHERE statement is used to filter observations based on a condition, in this case, selecting only customers from 'New York'. Option A incorrectly uses an IF statement, which would not filter observations but instead create a new variable. Option C uses an INPUT statement which is not relevant in this scenario. Option D is incorrect as it would output all the observations from the original dataset. Option E is incorrect as it would assign all the observations to the city 'New York', which is not the intended action.



CERTSWARRIOR

FULL PRODUCT INCLUDES:

Money Back Guarantee



Instant Download after Purchase



90 Days Free Updates



PDF Format Digital Download



24/7 Live Chat Support



Latest Syllabus Updates



For More Information – Visit link below:

<https://www.certswarrior.com>

16 USD Discount Coupon Code: U89DY2AQ