Advanced SAS Test

***Note: Well-formatted and easy-to-read answers will be given extra credit**, as it reflects your attention to detail and your overall professionalism.*

The first three questions are based on these two data sets:

Data Set >	Customer		Transaction	
Variables >	**ID**	**Product**	**ID**	**Amount**
	A	Checking	B	100
	B	Credit	C	200
	D	Saving	C	300

Q1

1. Write a Proc SQL to perform an **inner join** of the two data sets, and also fill in the Result table below.

Answer:

```
/* Creating Table1 named "CUSTOMER" */
/*create empty table*/
proc sql;
  create table Customer
      (ID char(10),
       Product char(10));

/*insert values into table*/
  insert into Customer
     values('A', 'Checking')
     values('B', 'Credit')
     values('D', 'Saving');

/*display table*/
  select * from Customer;
run;


/* Creating Table2 named "Transaction" */
/*create empty table*/
proc sql;
  create table Transaction
      (ID char(10),
       Amount num);

/*insert values into table*/
  insert into Transaction
     values('B', 100)
     values('C', 200)
     values('C', 300);

/*display table*/
  select * from Transaction;
run;


/*
---------------------------------------------
-- */
/* 1.     Write a Proc SQL to perform an inner
join of the two data sets, and also fill in
```

Result:

ID	Product	Amount
B	Credit	100

```
the Result table below. */
proc sql;
   create table inner_join_table as
   select * from Customer as x join
Transaction as y
   on x.ID = y.ID;
quit;

/* The result shows only the data in which the
IDs are present in both the tables */
```

Q2

2. Write a Proc SQL to perform a **left join** of the two data sets, and also fill in the Result table below.

Answer:

```
/* 2.     Write a Proc SQL to perform
a left join of the two data sets, and
also fill in the Result table below.
*/


proc sql;
   create table left_join_table as
   select * from Customer as x left
join Transaction as y
   on x.ID = y.ID;
quit;
/* The result shows all items from
Customer table (which is considered
left table here) and matches/left
joins records
from the Transaction table  */
```

Result:

ID	**Product**	**Amount**
A	Checking	.
B	Credit	100
D	Saving	.

Q3

3. Write a Proc SQL to perform a **full outer join** of the two data sets, and also fill in the Result table below.

Answer:

```
/* 3.  Write a Proc SQL to perform
a full outer join of the two data
sets, and also fill in the Result
table below. */
proc sql;
   create table
full_outer_join_table as
   select coalesce(x.ID,
y.ID),x.Product,y.Amount from
Customer as x full join Transaction
as y
   on x.ID = y.ID;
quit;


/* The result shows all row from
each individual dataset.
We have to use coalesce function to
display the IDs from both tables
are returned in final output. If
not used, it would show
only the IDs from first table i.e
Customer table */
```

Result:

ID	Product	Amount
A	Checking	.
B	Credit	100
C		300
C		200
D	Saving	.

Q4

4. Please write a SAS program that will create this result:

ID	Num_Of_Months	Average_Checking	Average_Saving
A	3	150	20
B	2	250	25
C	1	.	.

Answer:

Code:

```
/* 4.      Please write a SAS program that will create this
result: */
/*    ID    Num_Of_Months  Average_Checking    Average_Saving */
/*    A     3     150   20 */
/*    B     2     250   25 */
/*    C     1     .     . */
/*   */


/* Creating Table1 named "table_create_q4" */
/*create empty table*/
proc sql;
  create table table_create_q4
      (ID char(10),
       Num_Of_Months num,
       Average_Checking num,
       Average_Saving num);

/*insert values into table*/
  insert into table_create_q4
     values('A', 3, 150, 20)
     values('B', 2 , 250, 25)
     values('C', 1 , null ,null);

/*display table*/
  create table a as select * from table_create_q4;
run;
```

Output:

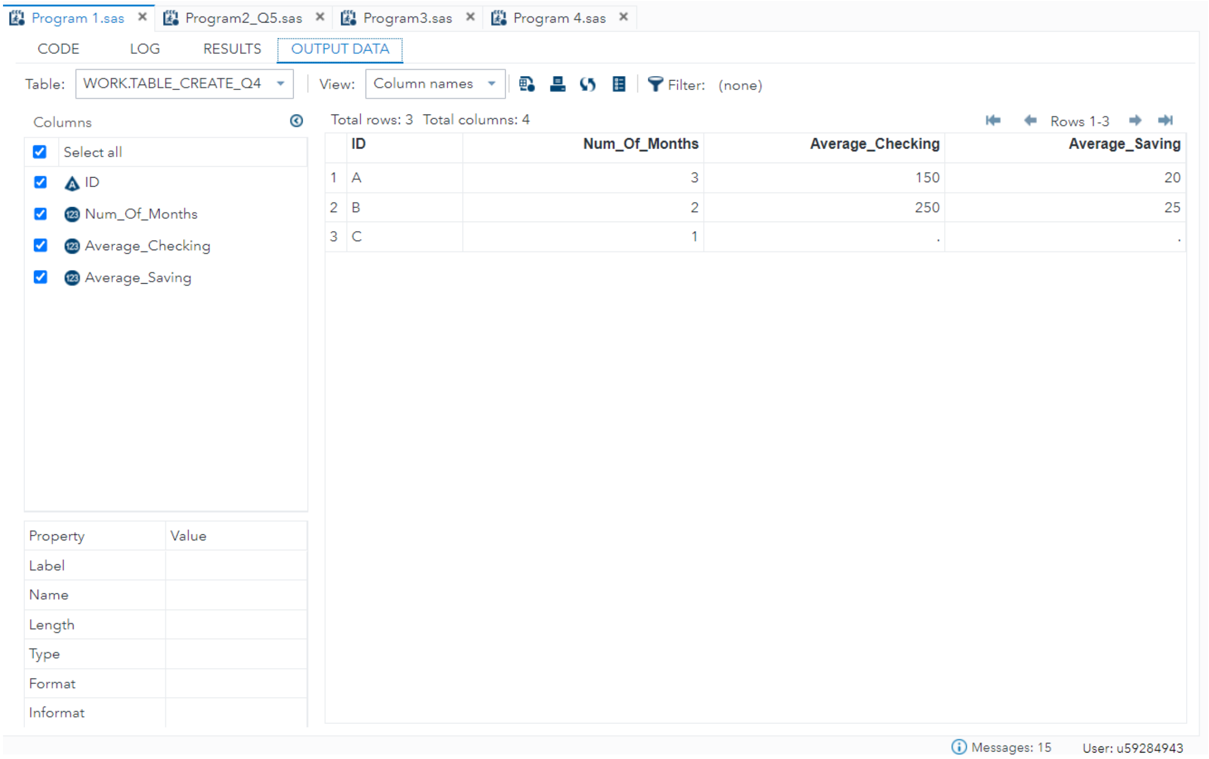


Table: WORK.TABLE_CREATE_Q4 View: Column names Filter: (none)

Total rows: 3 Total columns: 4

Columns: ID, Num_Of_Months, Average_Checking, Average_Saving

	ID	Num_Of_Months	Average_Checking	Average_Saving
1	A	3	150	20
2	B	2	250	25
3	C	1	.	.

Property	Value
Label	
Name	
Length	
Type	
Format	
Informat	

Messages: 15 User: u59284943

For the remaining questions, suppose you are given the following data set and a macro code:

```
%macro subset( cust=); proc print data= feedback;
  where customer = "&cust";
  run;
%mend;
```

Feedback		
CUSTOMER	SCORE	COMMENT
A	3	The is no parking
A	5	The food is expensive
B	.	I like the food
C	5	It tastes good
C	.	
C	3	I like the drink
D	4	The dessert is tasty
D	2	I don't like the services
[additional rows not shown]		

Q5

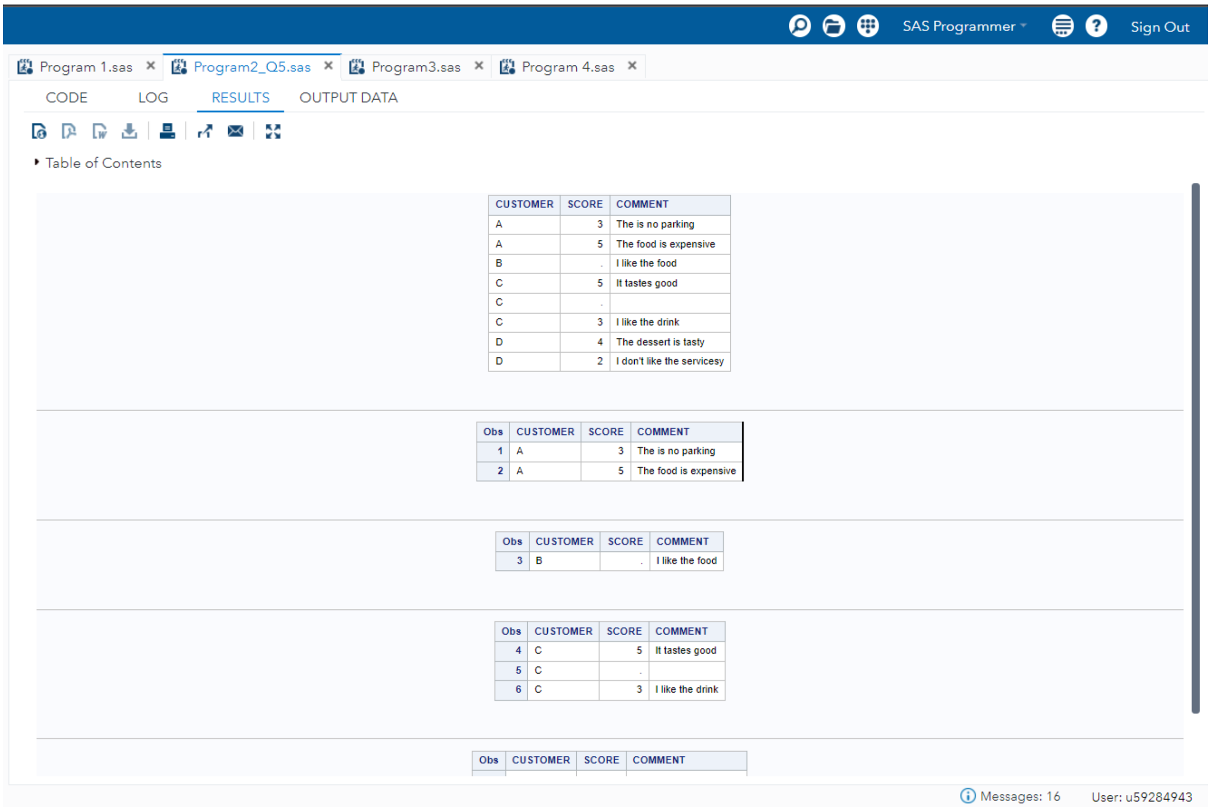
5. Write a program that will programmatically call **%subset** for each **customer** value in **Feedback**. Note that we do not know how many unique values of **customer** there are in the data set. Also, you cannot modify the macro **%subset** itself.

   Answer:

Code:

```
/* Creating Table1 named "CUSTOMER" */
/*create empty table*/
proc sql;
  create table feedback
      (CUSTOMER char(10),
       SCORE num,
       COMMENT char(100));

/*insert values into table*/
  insert into feedback
     values('A', 3,'The is no parking')
     values('A',5,'The food is expensive')
     values('B', null, 'I like the food')
     values('C', 5, 'It tastes good')
     values('C', null, '')
     values('C', 3, 'I like the drink')
     values('D', 4, 'The dessert is tasty')
     values('D', 2, "I don't like the services");

/*display table*/
  select * from feedback;
run;

%macro subset( cust=); proc print data= feedback;
 where customer = "&cust";
 run;
%mend;
/* ------------------------------------------------- */
/* -------------------------------------------------- */
/* Q5 */
/* Let's Create a table/dataset containing distinct customer
values */
proc sql;
 create table distinct_customers as
 select distinct customer
 from feedback;
quit;

/* looping through that list to call the %subset macro for
each unique customer value. */
/*macro named %call_subset is defined to make the
iteration through the distinct customer values and to invoke
%subset for each of them. */

%macro call_subset;
 data _null_;
   set distinct_customers;
/*     Below calls %subset, passing the current customer value
as a parameter/argument. */
   call execute('%nrstr(%subset(cust='||customer||'))');
 run;
%mend;

/* Call the macro for every customer */
%call_subset;

/* -------------------------------------------------- */
```

<u>Output:</u>



CUSTOMER	SCORE	COMMENT
A	3	The is no parking
A	5	The food is expensive
B	.	I like the food
C	5	It tastes good
C	.	
C	3	I like the drink
D	4	The dessert is tasty
D	2	I don't like the servicesy

Obs	CUSTOMER	SCORE	COMMENT
1	A	3	The is no parking
2	A	5	The food is expensive

Obs	CUSTOMER	SCORE	COMMENT
3	B	.	I like the food

Obs	CUSTOMER	SCORE	COMMENT
4	C	5	It tastes good
5	C	.	
6	C	3	I like the drink

Obs	CUSTOMER	SCORE	COMMENT

Q6

6. Write a program to find out which CUSTOMER has given a "5" score immediately followed by a missing score (among the data shown above, it is C).

   Answer:

Code:

```
/* Q6 */
/* We will loop through every customer till the last one */
data subset1;
 do until (last.customer);
   set feedback ;
   by customer ;
/* Group the data by customers*/
/*  Checking if the current customer's score is null and if
it's previous score is 5. If true, we set a boolean of
five_null =1    */
   if score=. and prev_score=5 then five_null=1;
   prev_score=score;
 end;
 if five_null;
/* If we have found=1 for the customer, we keep it, else we
ignore the customer*/
 keep customer;
run;
```

Output:

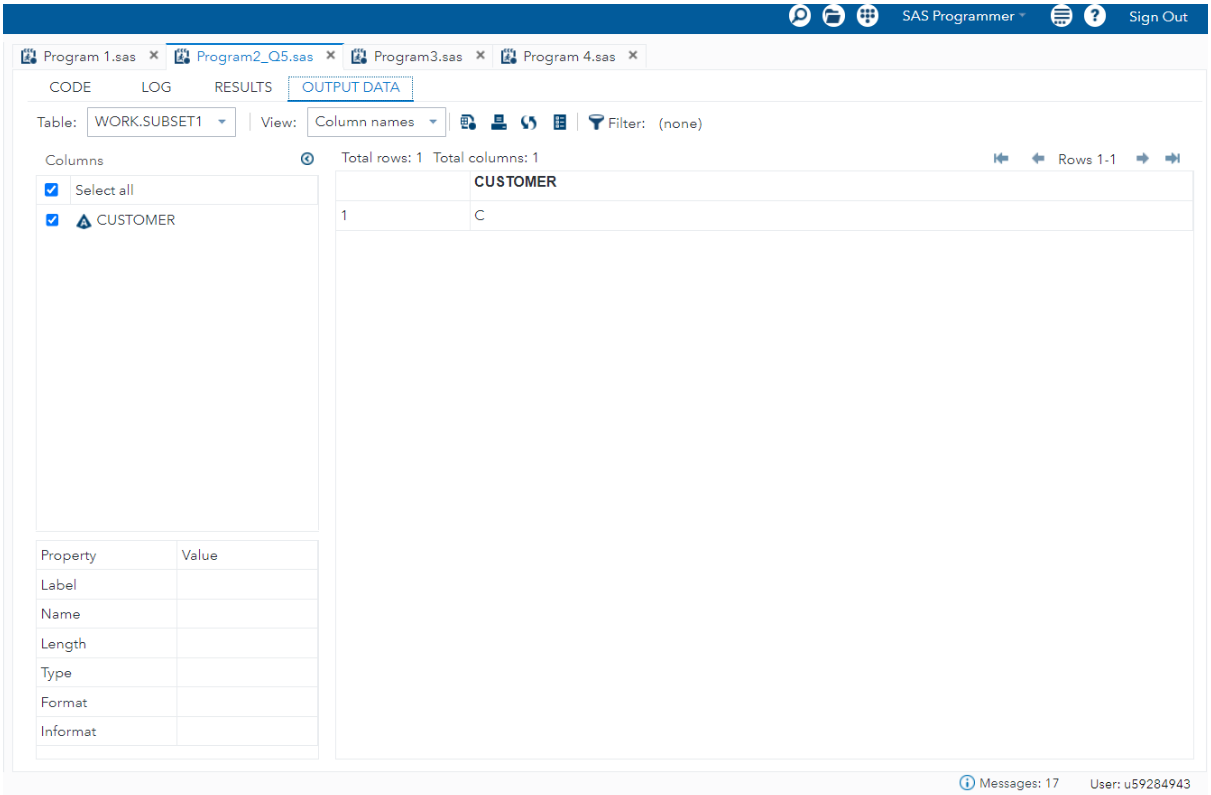


Table: WORK.SUBSET1 View: Column names Filter: (none)

Total rows: 1 Total columns: 1 Rows 1-1

☑ Select all
☑ CUSTOMER

	CUSTOMER
1	C

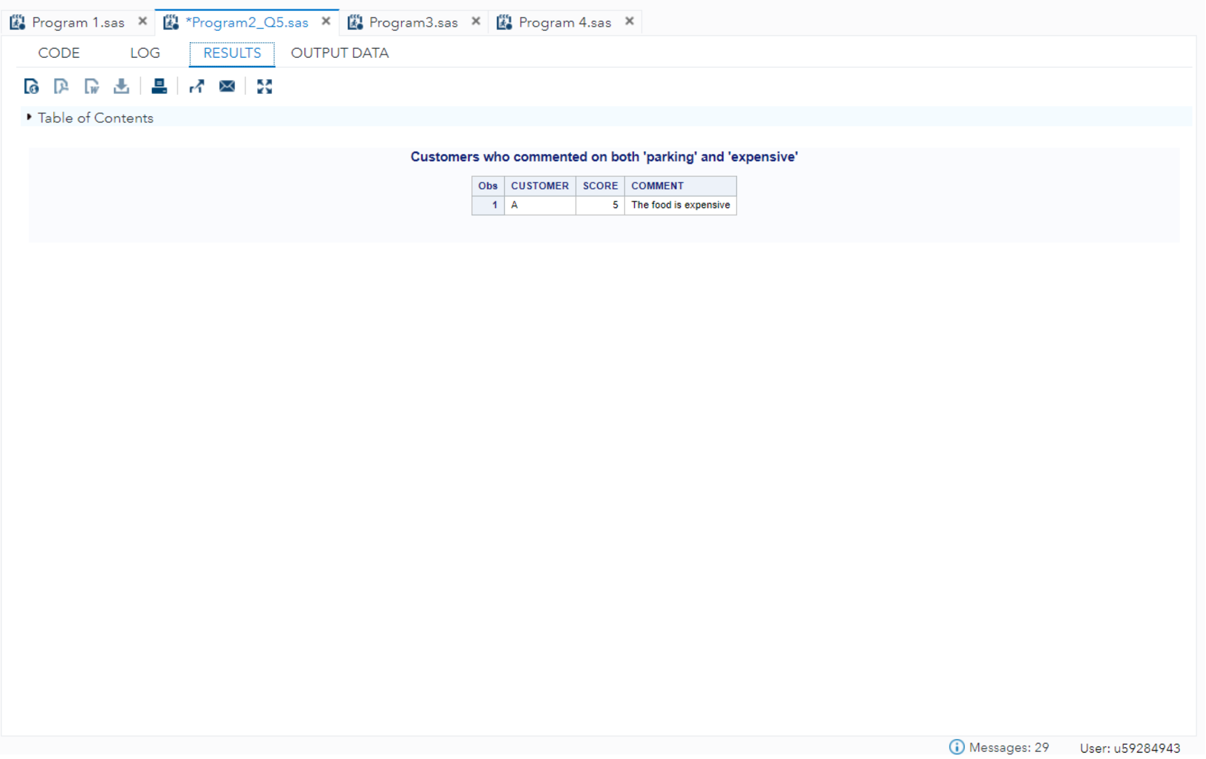
Q7

7. Write a program to find out which CUSTOMER has commented on “parking” and “expensive” (among the data shown above, it is A).

   Answer: <u>Code:</u>

```
/* ----------------------------------- */
/* Q7 */
/* Filter the dataset to select comments containing
"parking" */
data parking_comments;
 set feedback;
 if index(COMMENT, "parking") > 0;
run;

/* Filter the dataset to select comments containing
"expensive" */
data expensive_comments;
 set feedback;
 if index(COMMENT, "expensive") > 0;
run;

/* Merge the two filtered datasets to find customers who
commented on both */
data customers_with_both;
 merge parking_comments(in=commented_parking)
expensive_comments(in=commented_expensive);
 by CUSTOMER;
 if commented_parking and commented_expensive;
run;

/* Display the customers who commented on both "parking" and
"expensive" */
proc print data=customers_with_both;
 title "Customers who commented on both 'parking' and
'expensive'";
run;
/* ----------------------------------- */
```

<u>Output:</u>



Customers who commented on both 'parking' and 'expensive'

Obs	CUSTOMER	SCORE	COMMENT
1	A	5	The food is expensive

Q8

8. Using Proc SQL create the below dataset and instruct SAS to show the Landline Number of the customer, if the Mobile number column is missing otherwise show the Mobile number.

Obs	custid	Mobile	Landline
1	1	9800131356	9800231356
2	2	8900230356	.
3	3	.	9743231250

Result Output

custid	Phone
1	9800131356
2	8900230356
3	9743231250

<u>Code:</u>

```
/* Q8 */
/* Creating the given table in the question */
data landline;
length custid $ 5 Mobile $ 15 Landline $ 15;
input custid $ Mobile $ Landline;
datalines;
1 9800131356 9800231356
2 8900230356 .
3 .  9743231250
;
proc print data=landline;
run;

/*We will use the if/else clause to show the Landline Number
of the customer
if the Mobile number column is missing otherwise show the
Mobile number. */
data landline_output1;
  set landline;
/*    input Phone $ */
     if Mobile =. then
        Phone =Landline;  /*If mobile number is missing, set
output variable Phone as the value in Landline*/
     else Phone =Mobile; /* if above condition is false, set
output variable Phone as the value in Mobile*/
     drop landline Mobile;
run;
```

Output:

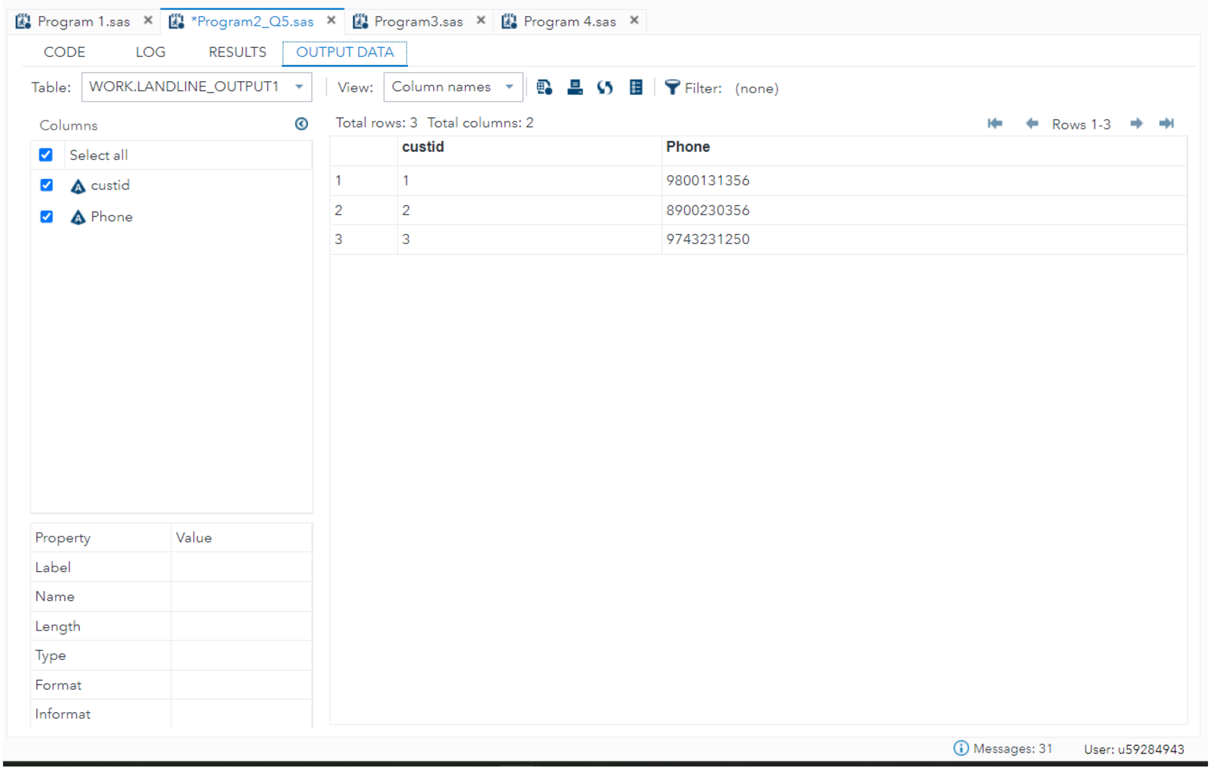


Table: WORK.LANDLINE_OUTPUT1

View: Column names

Columns: ☑ Select all, ☑ custid, ☑ Phone

Total rows: 3 Total columns: 2

Rows 1-3

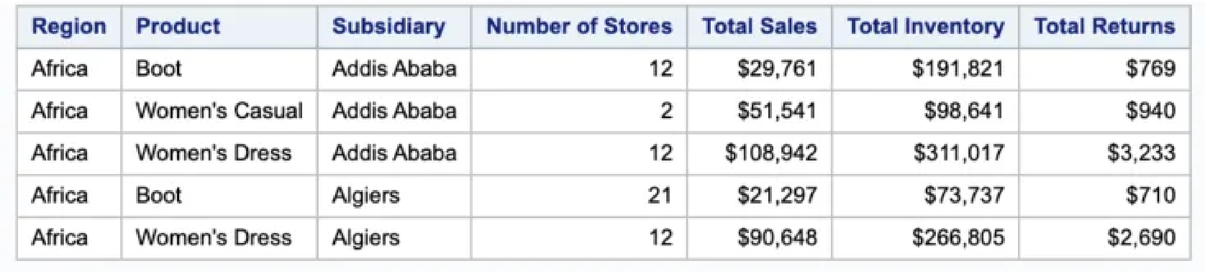
	custid	Phone
1	1	9800131356
2	2	8900230356
3	3	9743231250

Property	Value
Label	
Name	
Length	
Type	
Format	
Informat	

User: u59284943

Q9

9.Use SASHELP.SHOES and create the below output using like clause for the variable Product



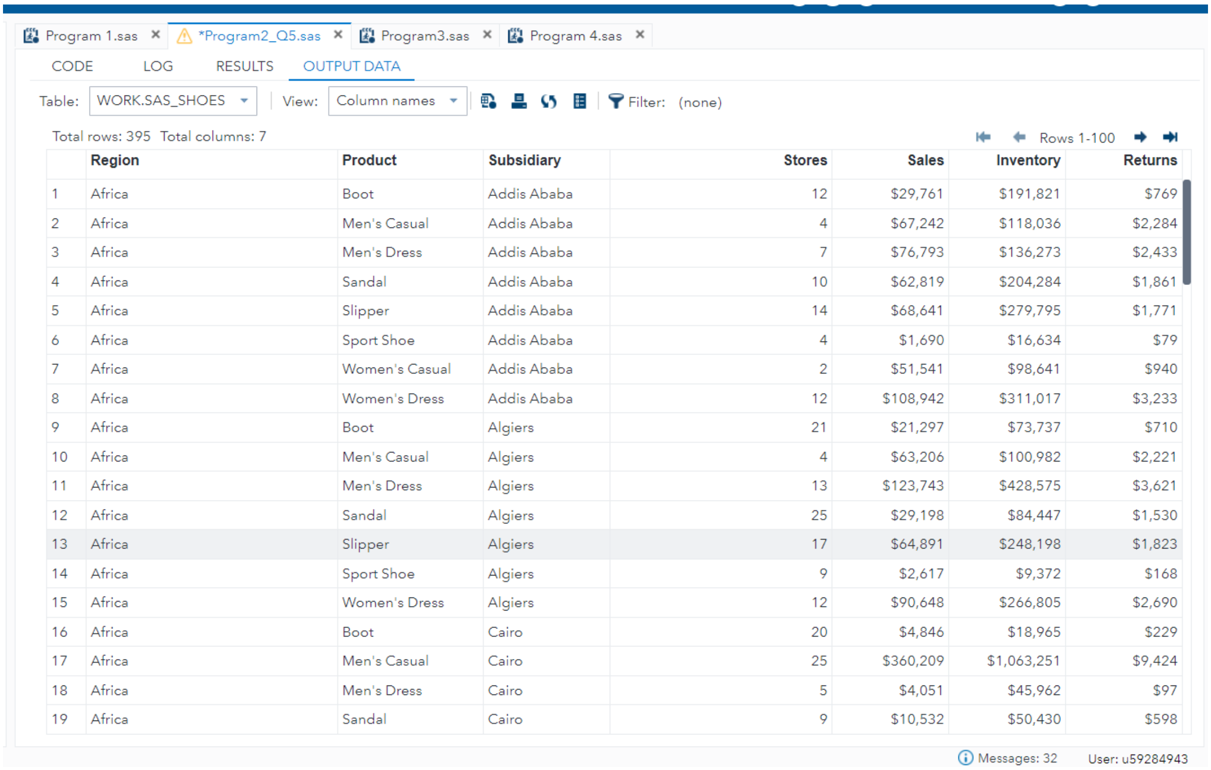
Region	Product	Subsidiary	Number of Stores	Total Sales	Total Inventory	Total Returns
Africa	Boot	Addis Ababa	12	$29,761	$191,821	$769
Africa	Women's Casual	Addis Ababa	2	$51,541	$98,641	$940
Africa	Women's Dress	Addis Ababa	12	$108,942	$311,017	$3,233
Africa	Boot	Algiers	21	$21,297	$73,737	$710
Africa	Women's Dress	Algiers	12	$90,648	$266,805	$2,690

Code:

```
/* Q9 */
/* Getting data from sashelp.shoes */
proc sql;
create table sas_shoes as
select * from sashelp.shoes;
quit;
```

```
proc sql outobs=5;
create table sas_shoes2 as
select * from sas_shoes
where Product like "%Boot%" or Product like "Women's%" ;
/*using like clause for the variable Product and then using
the outobs=5 because the question mentions
only those top 5 values after filtering condition */
quit;
```

<u>Output:</u>



Total rows: 395 Total columns: 7

	Region	Product	Subsidiary	Stores	Sales	Inventory	Returns
1	Africa	Boot	Addis Ababa	12	$29,761	$191,821	$769
2	Africa	Men's Casual	Addis Ababa	4	$67,242	$118,036	$2,284
3	Africa	Men's Dress	Addis Ababa	7	$76,793	$136,273	$2,433
4	Africa	Sandal	Addis Ababa	10	$62,819	$204,284	$1,861
5	Africa	Slipper	Addis Ababa	14	$68,641	$279,795	$1,771
6	Africa	Sport Shoe	Addis Ababa	4	$1,690	$16,634	$79
7	Africa	Women's Casual	Addis Ababa	2	$51,541	$98,641	$940
8	Africa	Women's Dress	Addis Ababa	12	$108,942	$311,017	$3,233
9	Africa	Boot	Algiers	21	$21,297	$73,737	$710
10	Africa	Men's Casual	Algiers	4	$63,206	$100,982	$2,221
11	Africa	Men's Dress	Algiers	13	$123,743	$428,575	$3,621
12	Africa	Sandal	Algiers	25	$29,198	$84,447	$1,530
13	Africa	Slipper	Algiers	17	$64,891	$248,198	$1,823
14	Africa	Sport Shoe	Algiers	9	$2,617	$9,372	$168
15	Africa	Women's Dress	Algiers	12	$90,648	$266,805	$2,690
16	Africa	Boot	Cairo	20	$4,846	$18,965	$229
17	Africa	Men's Casual	Cairo	25	$360,209	$1,063,251	$9,424
18	Africa	Men's Dress	Cairo	5	$4,051	$45,962	$97
19	Africa	Sandal	Cairo	9	$10,532	$50,430	$598

Messages: 32 User: u59284943

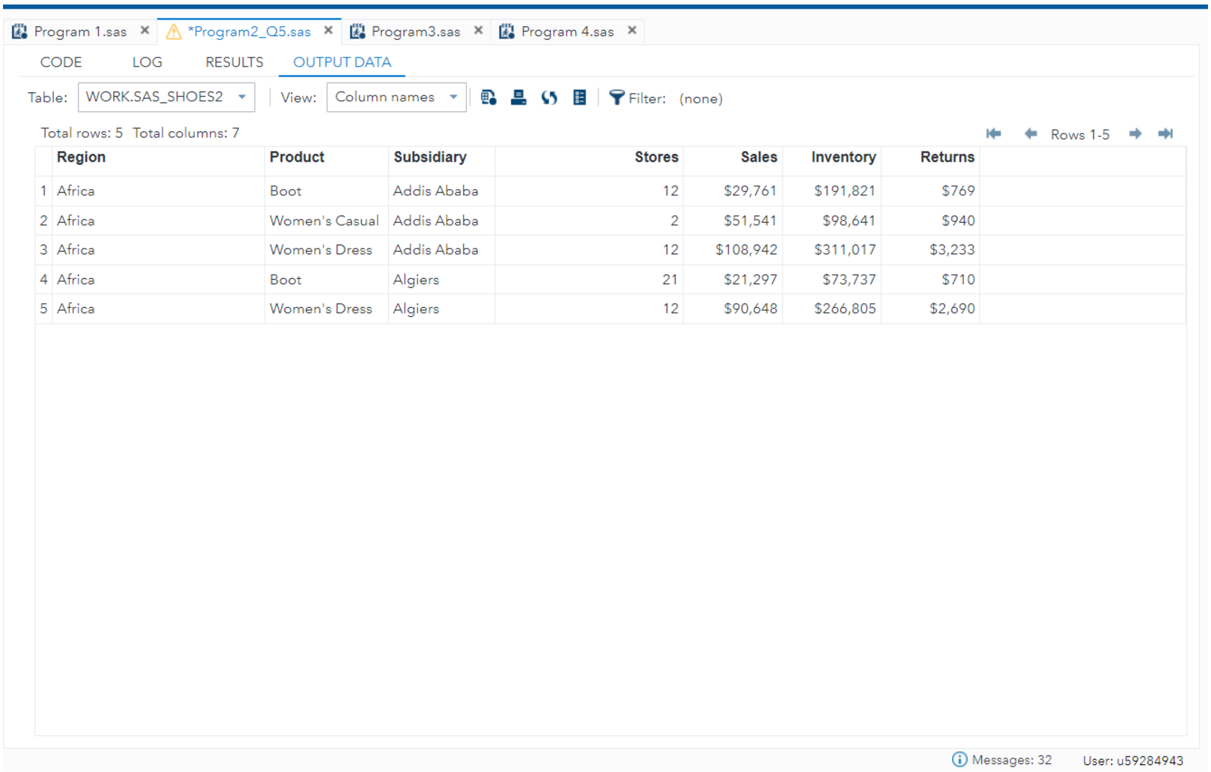


Table: WORK.SAS_SHOES2 View: Column names Filter: (none)

Total rows: 5 Total columns: 7 Rows 1-5

	Region	Product	Subsidiary	Stores	Sales	Inventory	Returns
1	Africa	Boot	Addis Ababa	12	$29,761	$191,821	$769
2	Africa	Women's Casual	Addis Ababa	2	$51,541	$98,641	$940
3	Africa	Women's Dress	Addis Ababa	12	$108,942	$311,017	$3,233
4	Africa	Boot	Algiers	21	$21,297	$73,737	$710
5	Africa	Women's Dress	Algiers	12	$90,648	$266,805	$2,690

Messages: 32 User: u59284943

END