Advanced SAS Programmer

SAP103

103 JOINS, INNER JOIN, OUTER JOIN, Complex JOINS

```
/* Defines the path to your data and assigns the libref. */
%let path=~/ESQ1M6;

libname sq "&path/data";
************************************************************;
*  Activity 3.01                                           *;
*  1) Examine and run the two queries to explore the       *;
*     sq.smallcustomer and sq.smalltransaction tables.     *;
*     Confirm that the sq.smallcustomer contains 8 rows    *;
*     and the sq.smalltransaction contains 12 rows.        *;
*  2) In the next section, list the sq.smallcustomer and   *;
*     sq.smalltransaction table in the FROM clause and     *;
*     separate the tables by a comma. Run the query and    *;
*     view the log. What note do you see?                  *;
*  3) View the results. Name two issues with the report.   *;
************************************************************;

************************;
*EXPLORE THE TABLES    *;
************************;
proc sql number;
title "Table: smallcustomer";
select *
          from sq.smallcustomer;
title "Table: smalltransaction";
```

```
select *
        from sq.smalltransaction;
title;
quit;
```

Table: smallcustomer

Row	FirstName	LastName	State	BankID	Income	AccountID
1	Gary	Sienkiewicz	NY	101010101	67210.91	1010159565
2	Sergio	Lefeld	CA	101010101	86859.07	1010367330
3	John	Oliver	CA	202020202	43623.75	2020012887
4	Iva	Bower	NY	303030303	67949.96	3030085224
5	Janet	Sienkiewicz	NY	303030303	50111.59	3030101942
6	Olga	Comstock	NY	303030303	31896.96	3030165207
7	Ada	Vieyra	NY	404040404	29586.44	4040164206
8	Samantha	Carney	CA	.	25476.14	5540174271

Table: smalltransaction

Row	AccountID	DateTime	BankID	MerchantID	Amount	Services
1	.	07MAY18:15:35:02	.	542058	58.79	Bar
2	1010159565	16SEP18:14:57:08	101010101	568268	107.16	Lawn Care
3	1010183063	24FEB18:17:27:42	101010101	562326	370.53	Fancy Restaurant
4	1010367330	15MAY18:17:54:21	101010101	542058	23.39	Bar
5	1010367330	17OCT18:11:02:38	101010101	525576	21.02	Economy
6	1010367364	18OCT18:17:52:51	101010101	549940	37.24	Fast Food
7	2020012887	23FEB18:09:25:37	202020202	525576	108.22	Economy
8	3030085224	27JUL18:12:05:48	303030303	525576	26.1	Economy
9	3030101942	18SEP18:12:13:40	303030303	549940	37.38	Fast Food
10	3030165207	11MAR18:10:07:14	303030303	580881	319.95	Cable/Internet
11	3030231909	27JAN18:13:52:35	303030303	536123	1.76	Movie
12	3030231909	13FEB18:16:48:05	303030303	513178	115.48	Air

```
**********************************;
*CREATE THE DEFAULT JOIN         *;
**********************************;
title "Cartesian Product";
proc sql number;
select *
        from sq.smallcustomer, sq.smalltransaction;
```

quit;

title;

Cartesian Product

Row	FirstName	LastName	State	BankID	Income	AccountID	AccountID	DateTime	BankID	MerchantID	Amount	Services
1	Gary	Sienkiewicz	NY	101010101	67210.91	1010159565	.	07MAY18:15:35:02	.	542058	58.79	Bar
2	Gary	Sienkiewicz	NY	101010101	67210.91	1010159565	1010159565	16SEP18:14:57:08	101010101	568268	107.16	Lawn Care
3	Gary	Sienkiewicz	NY	101010101	67210.91	1010159565	1010183063	24FEB18:17:27:42	101010101	562326	370.53	Fancy Restaurant
4	Gary	Sienkiewicz	NY	101010101	67210.91	1010159565	1010367330	15MAY18:17:54:21	101010101	542058	23.39	Bar
5	Gary	Sienkiewicz	NY	101010101	67210.91	1010159565	1010367330	17OCT18:11:02:38	101010101	525576	21.02	Economy
6	Gary	Sienkiewicz	NY	101010101	67210.91	1010159565	1010367364	18OCT18:17:52:51	101010101	549940	37.24	Fast Food
7	Gary	Sienkiewicz	NY	101010101	67210.91	1010159565	2020012887	23FEB18:09:25:37	202020202	525576	108.22	Economy

```
************************************************************;
* Performing an Inner Join with PROC SQL                   *;
************************************************************;
* Syntax                                                   *;
*                                                          *;
* PROC SQL;                                                *;
*    SELECT col-name, col-name                             *;
*    FROM table1 INNER JOIN table2                         *;
*    ON table1.col-name=table2.col-name;                   *;
* QUIT;                                                    *;
************************************************************;


************************************************************;
* Demo                                                     *;
* 1) Open the s103d01.sas program in the demos folder      *;
*    and find the Demo section of the program. Run the     *;
*    queries in the Explore the Tables section to          *;
*    compare the columns of the sq.smallcustomer and       *;
*    sq.smalltransaction tables.                           *;
* 2) Find the Perform the INNER JOIN section and add       *;
*    sq.smallcustomer and sq.smalltransaction to the       *;
*    FROM clause to perform an inner join on AccountID.    *;
```

```
*    Qualify AccountID columns as table-name.col-name in *;
*    the ON expression only. Highlight and run the      *;
*    query.                                             *;
*  3) Add the AccountID column to the query after Amount. *;
*    Highlight and run the query. Examine the log. Why   *;
*    does the program fail?                             *;
*    Note: There is an ambiguous reference. The column   *;
*         AccountID is in more than one table.          *;
*  4) Modify the query to qualify the AccountID column in *;
*    the SELECT clause. Highlight the step and run the   *;
*    selected code.                                     *;
*    Note: Because AccountID occurs in both tables, you  *;
*         must qualify the column with the table name to*;
*         indicate which column you want to select.     *;
*  5) Modify the query to include a WHERE clause to      *;
*    subset for customers who have a State value of NY   *;
*    (New York) and an ORDER BY clause that sorts by     *;
*    descending Amount.                                 *;
************************************************************;


*******************************;
*EXPLORE THE TABLES          *;
*******************************;
proc sql;
select *
          from sq.smallcustomer;
select *
          from sq.smalltransaction;
quit;
```

FirstName	LastName	State	BankID	Income	AccountID
Gary	Sienkiewicz	NY	101010101	67210.91	1010159565
Sergio	Lefeld	CA	101010101	86859.07	1010367330
John	Oliver	CA	202020202	43623.75	2020012887
Iva	Bower	NY	303030303	67949.96	3030085224
Janet	Sienkiewicz	NY	303030303	50111.59	3030101942
Olga	Comstock	NY	303030303	31896.96	3030165207
Ada	Vieyra	NY	404040404	29586.44	4040164206
Samantha	Carney	CA	.	25476.14	5540174271

AccountID	DateTime	BankID	MerchantID	Amount	Services
.	07MAY18:15:35:02	.	542058	58.79	Bar
1010159565	16SEP18:14:57:08	101010101	568268	107.16	Lawn Care
1010183063	24FEB18:17:27:42	101010101	562326	370.53	Fancy Restaurant
1010367330	15MAY18:17:54:21	101010101	542058	23.39	Bar
1010367330	17OCT18:11:02:38	101010101	525576	21.02	Economy
1010367364	18OCT18:17:52:51	101010101	549940	37.24	Fast Food
2020012887	23FEB18:09:25:37	202020202	525576	108.22	Economy
3030085224	27JUL18:12:05:48	303030303	525576	26.1	Economy
3030101942	18SEP18:12:13:40	303030303	549940	37.38	Fast Food
3030165207	11MAR18:10:07:14	303030303	580881	319.95	Cable/Internet
3030231909	27JAN18:13:52:35	303030303	536123	1.76	Movie
3030231909	13FEB18:16:48:05	303030303	513178	115.48	Air

```
*********************************;
*Perform the INNER JOIN       *;
*********************************;
proc sql;
select FirstName, LastName, State, Income, DateTime, MerchantID, Amount
            from sq.smallcustomer inner join sq.smalltransaction
   on smallcustomer.AccountID = smalltransaction.AccountID;
quit;
```

FirstName	LastName	State	Income	DateTime	MerchantID	Amount
Gary	Sienkiewicz	NY	67210.91	16SEP18:14:57:08	568268	107.16
Sergio	Lefeld	CA	86859.07	15MAY18:17:54:21	542058	23.39
Sergio	Lefeld	CA	86859.07	17OCT18:11:02:38	525576	21.02
John	Oliver	CA	43623.75	23FEB18:09:25:37	525576	108.22
Iva	Bower	NY	67949.96	27JUL18:12:05:48	525576	26.1
Janet	Sienkiewicz	NY	50111.59	18SEP18:12:13:40	549940	37.38
Olga	Comstock	NY	31896.96	11MAR18:10:07:14	580881	319.95

```
proc sql;
select FirstName, LastName, State, Income, DateTime, MerchantID, Amount, smallcustomer.AccountID
        from sq.smallcustomer inner join sq.smalltransaction
   on smallcustomer.AccountID = smalltransaction.AccountID
   where State='NY'
   order by Amount desc;
quit;
```

FirstName	LastName	State	Income	DateTime	MerchantID	Amount	AccountID
Olga	Comstock	NY	31896.96	11MAR18:10:07:14	580881	319.95	3030165207
Gary	Sienkiewicz	NY	67210.91	16SEP18:14:57:08	568268	107.16	1010159565
Janet	Sienkiewicz	NY	50111.59	18SEP18:12:13:40	549940	37.38	3030101942
Iva	Bower	NY	67949.96	27JUL18:12:05:48	525576	26.1	3030085224

```
**********************************************************;
*  Activity 3.02                                 *;
*  1) Examine and run the two queries to explore the      *;
*     sq.statepopulation and sq.statecode tables. What    *;
*     columns can you use to join the tables?             *;
*  2) Specify the tables in the FROM clause and perform   *;
*     an inner join. Add the alias p for the              *;
*     sq.statepopulation table, and the alias s for the   *;
*     sq.statecode table.                                 *;
*  3) Complete the ON expression to match rows where      *;
*     p.Name = s.StateCode. Highlight and run the query.  *;
*     How many rows are in the new report?                *;
```

```
************************************************************;

**********************;
*EXPLORE THE TABLES    *;
**********************;
proc sql inobs=10;
title "Table: STATEPOPULATION";;
select Name, PopEstimate1, PopEstimate1, PopEstimate1
        from sq.statepopulation;
title "Table: STATECODE";
select *
        from sq.statecode;
quit;
```

Table: STATEPOPULATION

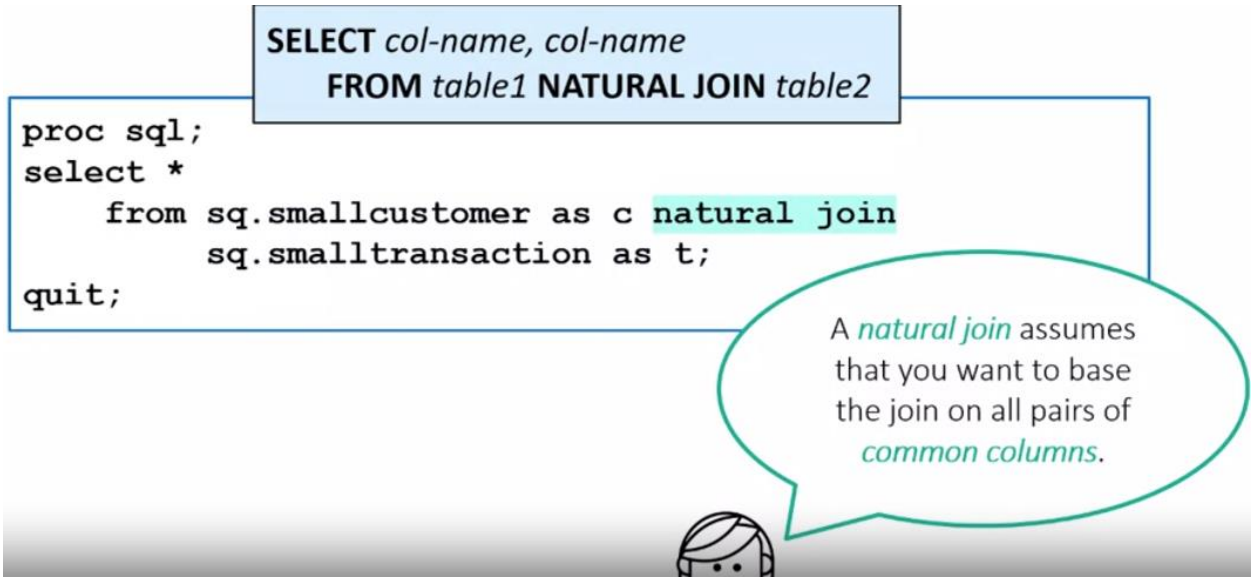
Name	PopEstimate1	PopEstimate1	PopEstimate1
AL	4864745	4864745	4864745
AK	741504	741504	741504
AZ	6945452	6945452	6945452
AR	2990410	2990410	2990410
CA	39209127	39209127	39209127
CO	5540921	5540921	5540921
CT	3578674	3578674	3578674
DE	949216	949216	949216
DC	686575	686575	686575
FL	20629982	20629982	20629982

Table: STATECODE

StateCode	StateName
AL	Alabama
AK	Alaska
AZ	Arizona
AR	Arkansas
CA	California
CO	Colorado
CT	Connecticut
DE	Delaware
DC	District of Columbia
FL	Florida

```
*******************************;
*COMPLETE THE INNER JOIN     *;
*******************************;
proc sql number;
select Name, StateName, PopEstimate1, PopEstimate2, PopEstimate3
        from sq.statepopulation as p INNER JOIN sq.statecode as c
        on p.Name = c.StateCode
   order by StateName;
quit;
```

Row	Name	StateName	PopEstimate1	PopEstimate2	PopEstimate3
1	AL	Alabama	4864745	4875120	4887871
2	AK	Alaska	741504	739786	737438
3	AZ	Arizona	6945452	7048876	7171646
4	AR	Arkansas	2990410	3002997	3013825
5	CA	California	39209127	39399349	39557045
6	CO	Colorado	5540921	5615902	5695564
7	CT	Connecticut	3578674	3573880	3572665
8	DE	Delaware	949216	957078	967171
9	DC	District of Columbia	686575	695691	702455
10	FL	Florida	20629982	20976812	21299325
11	GA	Georgia	10304763	10413055	10519475
12	HI	Hawaii	1428105	1424203	1420491




```
************************************************************;
*  Activity 3.03                                           *;
*  1) Complete the first query by adding the BANKID        *;
*     column name in the WHERE clause. How many tables     *;
*     contain the BankID column?                           *;
*  2) Replace BANKID with MERCHANTID. How many tables      *;
*     contain the MerchantID column?                       *;
************************************************************;

proc sql;
```

```
select memname, name
        from dictionary.columns
        where libname="SQ" and
      upcase(name)='BANKID';
quit;
```

Member Name	Column Name
BANK	BankID
CUSTOMER	BankID
SMALLCUSTOMER	BankID
SMALLCUSTOMER2	BankID
SMALLTRANSACTION	BankID
SMALLTRANSACTION2	BankID
TRANSACTION	BankID
TRANSACTIONFULL	BankID

```
proc sql;
select memname, name
        from dictionary.columns
        where libname="SQ" and
                upcase(name)='MERCHANTID';
quit;
```

Member Name	Column Name
MERCHANT	MerchantID
SMALLTRANSACTION	MerchantID
SMALLTRANSACTION2	MerchantID
TRANSACTION	MerchantID
TRANSACTIONFULL	MerchantID

```
************************************************************;
*  Performing an Inner Join with Four Tables          *;
************************************************************;
*  Syntax                                  *;
*                                        *;
*  PROC SQL;                               *;
*  SELECT col-name, col-name                    *;
```

```
*    FROM table1 INNER JOIN table2                  *;
*    ON table1.col-name=table2.col-name INNER JOIN      *;
*      table3                                       *;
*    ON join-criteria INNER JOIN                    *;
*      table4                                       *;
*    ON join-criteria;                              *;
*  QUIT;                                            *;
************************************************************;

************************************************************;
*  Demo                                             *;
*  1) Open the s103d02.sas program in the demos folder    *;
*    and find the Demo section of the program. Under the *;
*    Explore the Tables section, run the queries to      *;
*    explore the sq.smallcustomer, sq.smalltransaction,  *;
*    sq.bank, and sq.merchant tables. Describe the       *;
*    relationships between the tables.                   *;
*  2) Find the Joining Data from More Than Two Tables     *;
*    section. Highlight and run the query to join        *;
*    sq.smallcustomer with sq.transaction. Examine the   *;
*    results.                                            *;
*  3) Add a second inner join and join the MerchantID     *;
*    column from the sq.merchant table with the          *;
*    MerchantID column of the previous join. Replace     *;
*    MerchantID in the SELECT clause with MerchantName.  *;
*    Highlight and run the query. Examine the results.   *;
*  4) Add a third inner join and join the BankID column   *;
*    from the sq.bank table with the BankID column of    *;
*    the previous join. Replace BankID in the SELECT     *;
```

```
*     clause with the bank name. Highlight and run the    *;
*     query. Examine the results.                         *;
************************************************************;


************************;
*EXPLORE THE TABLES    *;
************************;
proc sql inobs=5;
title "Table: SMALLCUSTOMER";
select *
          from sq.smallcustomer;
title "Table: SMALLTRANSACTION";
select *
          from sq.smalltransaction;
title "Table: MERCHANT";
select *
          from sq.merchant;
title "Table: BANK";
select *
          from sq.bank;
title;
quit;
```

Table: SMALLCUSTOMER

FirstName	LastName	State	BankID	Income	AccountID
Gary	Sienkiewicz	NY	101010101	67210.91	1010159565
Sergio	Lefeld	CA	101010101	86859.07	1010367330
John	Oliver	CA	202020202	43623.75	2020012887
Iva	Bower	NY	303030303	67949.96	3030085224
Janet	Sienkiewicz	NY	303030303	50111.59	3030101942

Table: SMALLTRANSACTION

AccountID	DateTime	BankID	MerchantID	Amount	Services
.	07MAY18:15:35:02	.	542058	58.79	Bar
1010159565	16SEP18:14:57:08	101010101	568268	107.16	Lawn Care
1010183063	24FEB18:17:27:42	101010101	562326	370.53	Fancy Restaurant
1010367330	15MAY18:17:54:21	101010101	542058	23.39	Bar
1010367330	17OCT18:11:02:38	101010101	525576	21.02	Economy

Table: MERCHANT

Merchant ID	Merchant Name	City	State	Zip	Phone	Type	Service
501976	Coruscating Carwash, LLC	New York	NY	10001	(212)7909623	Auto	Carwash
502136	Good Service Auto Repair	New York	NY	10001	(212)2988674	Auto	Service and Repair
505780	Local Gas & Oil Co.	Los Angeles	CA	90001	(323)8422053	Auto	Gas
509793	Comfortable Coach	New York	NY	10001	(212)8465690	Business Travel	Bus
513178	Alar Air, Inc.	New York	NY	10001	(212)5212018	Business Travel	Air

Table: BANK

BankID	Name	Address	City	State	Zip	Domain	Customer Service	Company Email
101010101	Biggest Bank, Inc.	620 14th Avenue	New York	NY	10036	bbfake.com	(800)5550100	BiggestBankInc@bbfake.com
202020202	Sailors Credit Union	404 East Shoreline Drive	Long Beach	CA	90802	scufake.org	(800)5550155	SailorsCredit@scufake.org
303030303	Wheatberry Bank, Inc.	1354 Southwest Central Park Ave	Topeka	KS	66604	wbbfake.com	(800)5550177	WheatberryBank@wbbfake.com
505050505	GMD Federal Bank	234 Main Street	Cary	NC	27513	gmdfake.com	(800)5550166	gmdfederal@gmdfake.com

```
*****************************************;
*Joining Data From More Than Two Tables   *;
*****************************************;
proc sql;
select FirstName, LastName, c.State, Income, DateTime, MerchantID, Amount,
      c.AccountID, c.BankID
   from sq.smallcustomer as c inner join
        sq.smalltransaction as t
   on c.AccountID = t.AccountID
          /*Join the results with the Merchant table*/
          /*Join the results with the Bank table*/;
```

quit;

FirstName	LastName	State	Income	DateTime	MerchantID	Amount	AccountID	BankID
Gary	Sienkiewicz	NY	67210.91	16SEP18:14:57:08	568268	107.16	1010159565	101010101
Sergio	Lefeld	CA	86859.07	15MAY18:17:54:21	542058	23.39	1010367330	101010101
Sergio	Lefeld	CA	86859.07	17OCT18:11:02:38	525576	21.02	1010367330	101010101
John	Oliver	CA	43623.75	23FEB18:09:25:37	525576	108.22	2020012887	202020202
Iva	Bower	NY	67949.96	27JUL18:12:05:48	525576	26.1	3030085224	303030303
Janet	Sienkiewicz	NY	50111.59	18SEP18:12:13:40	549940	37.38	3030101942	303030303
Olga	Comstock	NY	31896.96	11MAR18:10:07:14	580881	319.95	3030165207	303030303

```
*****************************************;
*Joining Data From More Than Two Tables   *;
*****************************************;
proc sql;
select FirstName, LastName, c.State, Income, DateTime, MerchantName, Amount,
    c.AccountID, b.Name
  from sq.smallcustomer as c inner join
     sq.smalltransaction as t
  on c.AccountID = t.AccountID inner join
        sq.Merchant as m
  on t.MerchantID = m.MerchantID inner join
        sq.Bank as b
  on t.BankID = b.BankID
        /*Join the results with the Merchant table*/
        /*Join the results with the Bank table*/;
quit;
```

FirstName	LastName	State	Income	DateTime	Merchant Name	Amount	AccountID	Name
Sergio	Lefeld	CA	86859.07	17OCT18:11:02:38	Economical Superstore	21.02	1010367330	Biggest Bank, Inc.
Iva	Bower	NY	67949.96	27JUL18:12:05:48	Economical Superstore	26.1	3030085224	Wheatberry Bank, Inc.
John	Oliver	CA	43623.75	23FEB18:09:25:37	Economical Superstore	108.22	2020012887	Sailors Credit Union
Sergio	Lefeld	CA	86859.07	15MAY18:17:54:21	Happy Sour Bar & Grill	23.39	1010367330	Biggest Bank, Inc.
Janet	Sienkiewicz	NY	50111.59	18SEP18:12:13:40	Big Burgers, Inc.	37.38	3030101942	Wheatberry Bank, Inc.
Gary	Sienkiewicz	NY	67210.91	16SEP18:14:57:08	Livable Landscaping, LLC	107.16	1010159565	Biggest Bank, Inc.
Olga	Comstock	NY	31896.96	11MAR18:10:07:14	Pebble Cable, Inc.	319.95	3030165207	Wheatberry Bank, Inc.

```
*****************************************;
```

```
*Alternate Solution Using WHERE          *;
*****************************************;
proc sql;
select FirstName, LastName, c.State, Income, DateTime, MerchantName, Amount,
      c.AccountID, b.Name
   from sq.smallcustomer as c,
       sq.smalltransaction as t,
       sq.bank as b,
       sq.merchant as m
           where c.AccountID = t.AccountID and
               t.MerchantID = m.MerchantID and
        t.BankID = b.BankID;
quit;
```

```
proc sql;
   select *
   from sq.smallcustomer2 as c inner join
        sq.smalltransaction2 as t
   on c.AccountID = t.AccountID and
      c.AccountID is not null;
quit;
```

Adding the IS NOT NULL operator to the ON clause prevents the missing values from joining.

FirstName	LastName	State	BankID	Income	AccountID	AccountID
Sergio	Lefeld	CA	101010101	86859.07	1010367330	1010367330
Sergio	Lefeld	CA	101010101	86859.07	1010367330	1010367330
John	Oliver	CA	202020202	43623.75	2020012887	2020012887
Iva	Bower	NY	303030303	67949.96	3030085224	3030085224
Janet	Sienkiewicz	NY	303030303	50111.59	3030101942	3030101942
Olga	Comstock	NY	303030303	31896.96	3030165207	3030165207

smallcustomer

FirstName	LastName	State	BankID	Income	AccountID
Gary	Sienkiewicz	NY	101010101	67210.91	1010159565
Sergio	Lefeld	CA	101010101	86859.07	1010367330
John	Oliver	CA	202020202	43623.75	2020012887
Iva	Bower	NY	303030303	67949.96	3030085224
Janet	Sienkiewicz	NY	303030303	50111.59	3030101942
Olga	Comstock	NY	303030303	31896.96	3030165207

taxbracket

TaxBracket	LowIncome	HighIncome
10%	0	9524.99
12%	9525	38699.99
22%	38700	82499.99
24%	82500	157499.99
32%	157500	199999.99
35%	200000	499999.99

```
select FirstName, LastName, Income,
       TaxBracket
    from sq.smallcustomer as c inner join
         sq.taxbracket as t
    on c.Income >= t.LowIncome and
       c.Income <= t.HighIncome;
```

Use comparison operators in the ON clause instead of equality.

FirstName	LastName	Income	TaxBracket
Olga	Comstock	31896.96	12%
Ada	Vieyra	29586.44	12%
Samantha	Carney	25476.14	12%
Gary	Sienkiewicz	67210.91	22%
John	Oliver	43623.75	22%
Iva	Bower	67949.96	22%
Janet	Sienkiewicz	50111.59	22%
Sergio	Lefeld	86859.07	24%

```
************************************************************;
*  Activity 3.04                                           *;
*  1) Complete the ON clause to join on rows where         *;
*     customer Income is greater than the LowIncome        *;
*     range, and less than or equal to the HighIncome      *;
*     range using the BETWEEN-AND where operator.          *;
*  2) What tax bracket is Olga Comstock in?                *;
```

```
*  3) View your log. What note do you see?              *;
************************************************************;


***************;
*NON-EQUI JOIN *;
***************;
proc sql;
select FirstName, LastName, Income format=dollar16.,
      TaxBracket
           from sq.smallcustomer as c inner join
       sq.taxbracket as t
   on c.Income between t.LowIncome and t.HighIncome
           order by TaxBracket desc, Income desc;
quit;
```

FirstName	LastName	Income	TaxBracket
Sergio	Lefeld	$86,859	24%
Iva	Bower	$67,950	22%
Gary	Sienkiewicz	$67,211	22%
Janet	Sienkiewicz	$50,112	22%
John	Oliver	$43,624	22%
Olga	Comstock	$31,897	12%
Ada	Vieyra	$29,586	12%
Samantha	Carney	$25,476	12%

/*Practice Level 1: Performing an Inner Join

If necessary, start SAS Studio before you begin.

If you restarted your SAS session, submit your libname.sas program to access the practice data.

Open s103p01.sas from the practices folder.

Modify the program to generate a report that shows the breakdown of employment and marital status for customers in New York City.

Add a PROC SQL step to create a table named work.nyc that combines sq.customer and sq.maritalcode. Use the following requirements:

This table should include only FirstName, LastName, Employed, and MaritalStatus.

Perform an inner join on the Married column in the sq.customer table and MaritalCode column in the sq.maritalcode table.

Filter the Zip column for customers in the 10001 Zip code.

Run the query and view the results.

Run the PROC FREQ step to generate the crosstabulation of MaritalStatus and Employed.

View the results.

Overall, are NYC customers likely to be employed or unemployed?

Does this employment status vary across marital status?

*/

/*s103p01.sas*/

/*Insert the PROC SQL step here*/

proc sql outobs=5;

select * from sq.customer;

select * from sq.maritalcode;

quit;

First Name	Middle Name	Last Name	Gender	Date of Birth	Employed	Race	Married	StreetNumber	StreetName	City	State	Zip	HomePhone	CellPhone	StateID	User ID	Customer ID	BankID	Income
Rodney	Matthew	Joyner	M	2202	Y	W	M	28	Davis Place	Greenfield	WI	53001	(920)6982806	(920)6491939	WI62748437	rodmajoyner6611@n/a.com	1902986359	.	63583.22
Jeanne	Carol	Ballenger	F	1254	N	H		236	Hillcrest Court	Sammamish	WA	98074			WA56580527	jeacaballenger638@fakeemail.com	1935367360	.	.
Brian	Dallas	Harper	M	-4584	N	W	M	57	Oak Stanley Boulevard	Milwaukee	WI	53201	(414)7077277	(414)9037075	WI12094855	bridaharper4714@invalid.com	1455003144	.	.
Thomas	Eric	Henderson	M	1421	N	W	S	127	Marshall Arbor	Seattle	WA	98101	(206)5134695		WA59465008	thoerhenderson6322@ismissing.com	1979102386	.	.
Becky	Danna	Cheers	F	-5365	N	W	M	502	Meadow Lane	Altoona	WI	54720	(715)4864642	(715)0238456	WI66464214	becdacheers4524@n/a.com	1914860679	.	.

MaritalCode	MaritalStatus
D	Divorced
M	Married
S	Single
W	Widowed

proc sql;

create table nyc as

select FirstName, LastName, Employed, MaritalStatus

from sq.customer as c inner join

sq.maritalcode as m

on c.married = m.maritalcode

where zip = 10001;

quit;

Total rows: 4629 Total columns: 4 Rows

	FirstName	LastName	Employed	MaritalStatus
1	Janet	Turner	Y	Single
2	Lawrence	Athas	Y	Married
3	Robert	Freydel	Y	Single
4	Mary	Remaklus	Y	Divorced
5	Lisa	Brunson	N	Married
6	Wanita	Rankin	Y	Married

/*Run the Frequency Procedure on the Newly Created Table*/

title 'Marital Status by Employment for NYC Customers';

proc freq data=work.NYC order=Freq;

tables MaritalStatus*employed;

run;

title;

Marital Status by Employment for NYC Customers

The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of MaritalStatus by Employed

	Employed		
MaritalStatus	Y	N	Total
Married	1693 36.57 60.01 64.20	1128 24.37 39.99 56.63	2821 60.94
Single	430 9.29 43.43 16.31	560 12.10 56.57 28.11	990 21.39
Divorced	424 9.16 63.28 16.08	246 5.31 36.72 12.35	670 14.47
Widowed	90 1.94 60.81 3.41	58 1.25 39.19 2.91	148 3.20
Total	2637 56.97	1992 43.03	4629 100.00

/*Practice Level 2: Joining on Inequality

If necessary, start SAS Studio before you begin.

If you restarted your SAS session, submit your libname.sas program to access the practice data.

Open s103p02.sas from the practices folder.

Modify the program to join the sq.customer and sq.agegroup tables based on a customer's year of birth.

Add a PROC SQL step to the top of the program to create a table named work.generation that

combines sq.customer and sq.agegroup. Use the following requirements:

Select FirstName and LastName.

Create a column named Year that uses the DOB (Date of Birth) column in the sq.customer to determine the year the customer was born.

Select the Name column from the sq.agegroup table.

The StartYear and EndYear columns in the sq.agegroup table indicate the starting and ending years for each generation. Use these columns to perform a non-equijoin using the calculated Year value from the previous step.

Run the query and view the results.

Run the PROC SGPLOT step below your query to generate the bar chart.

Which age group has the most customers?

*/

/*s103p02.sas*/

/*Insert the PROC SQL step here*/

proc sql outobs=5;

select * from sq.customer;

select * from sq.agegroup;

quit;

First Name	Middle Name	Last Name	Gender	Date of Birth	Employed	Race	Married	StreetNumber	StreetName	City	State	Zip	HomePhone	CellPhone	StateID	User ID	Customer ID
Rodney	Matthew	Joyner	M	2202	Y	W	M	28	Davis Place	Greenfield	WI	53001	(920)6982806	(920)6491939	WI62748437	rodmajoyner6611@n/a.com	1902986359
Jeanne	Carol	Ballenger	F	1254	N	H		236	Hillcrest Court	Sammamish	WA	98074			WA56580527	jeacaballenger638@fakeemail.com	1935367360
Brian	Dallas	Harper	M	-4584	N	W	M	57	Oak Stanley Boulevard	Milwaukee	WI	53201	(414)7077277	(414)9037075	WI12094855	bridaharper4714@invalid.com	1455003144
Thomas	Eric	Henderson	M	1421	N	W	S	127	Marshall Arbor	Seattle	WA	98101	(206)5134695		WA59465008	thoerhenderson6322@ismissing.com	1979102386
Becky	Danna	Cheers	F	-5365	N	W	M	502	Meadow Lane	Altoona	WI	54720	(715)4864642	(715)0238456	WI66464214	becdacheers4524@n/a.com	1914860679

Name	StartYear	EndYear
G.I. Generation	1900	1924
Silent Generation	1925	1945
Baby Boomers	1946	1964
Thirteeners or Generation X	1965	1979
Millennials or Generation Y	1980	2000

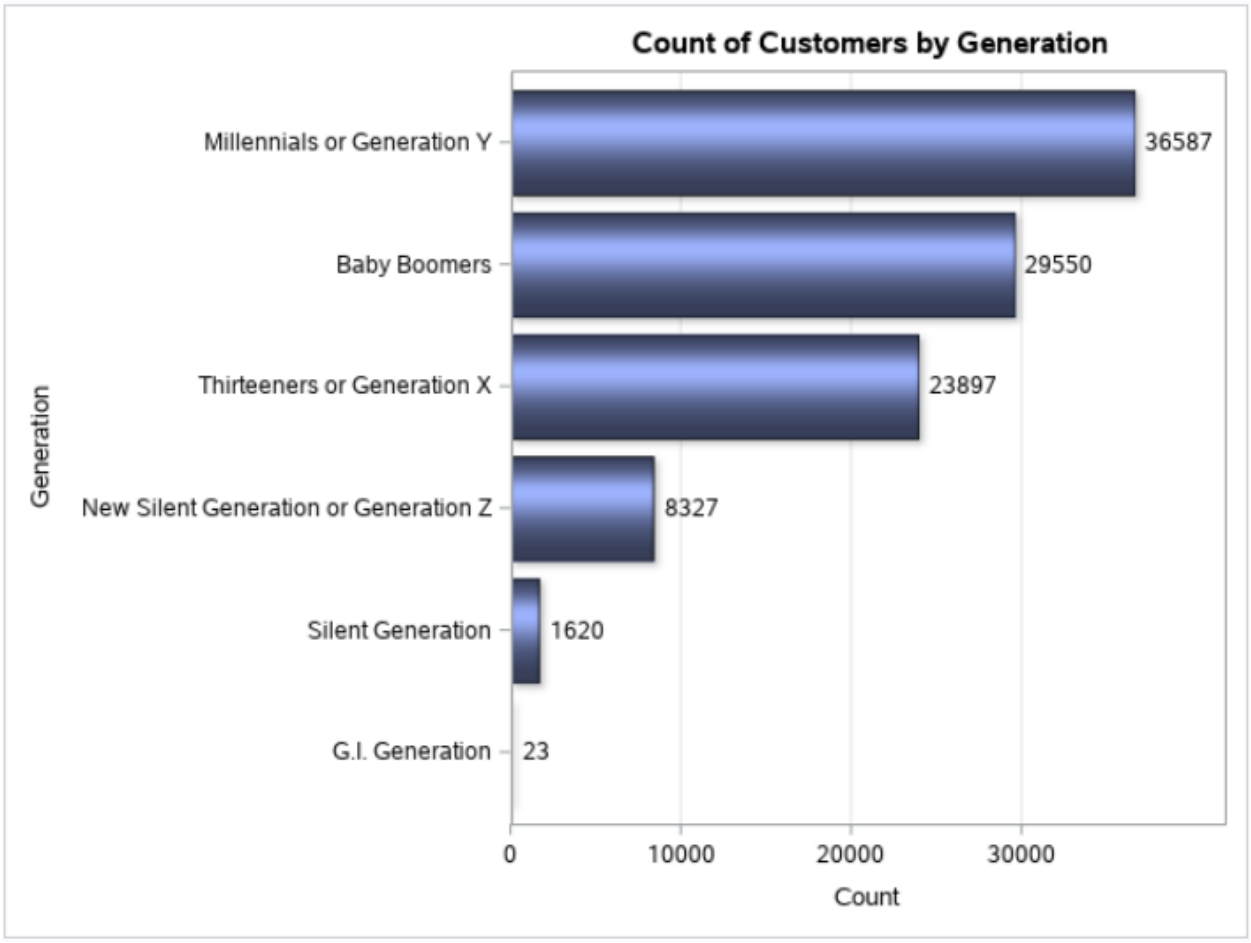
```
proc sql;
create table generation as
select FirstName, LastName, year(DOB) as Year, Name
from sq.customer as c inner join
        sq.agegroup as a
        on year(c.DOB) between a.StartYear and a.EndYear;
quit;
```

Total rows: 100004 Total columns: 4 Rows

	FirstName	LastName	Year	Name
1	Becky	Cheers	1945	Silent Generation
2	Kathryn	Mathews	1940	Silent Generation
3	James	Frederick	1945	Silent Generation
4	Byron	Pray	1934	Silent Generation
5	Lauren	Mitchell	1934	Silent Generation
6	Pat	Johnson	1943	Silent Generation

```
/*Run the Visualization on the Newly Created Table*/
title 'Count of Customers by Generation';
proc sgplot data=work.generation noautolegend;
   hbar Name /
        stat=freq
        dataskin=sheen
        categoryorder=respdesc
        datalabel
        datalabelattrs=(size=9pt)
        FILLATTRS=(color=cx6f7eb3);
        yaxis label="Generation";
        xaxis grid label="Count";
run;
title;
```

```
/*Alternate Solution*/
proc sql;
create table work.generation as
select c.FirstName,
       c.LastName,
       year(c.DOB) as Year,
       a.Name
    from sq.Customer as c, sq.AgeGroup as a
    where calculated Year between a.StartYear and a.EndYear;
quit;
```



```
************************************************************;
*  Activity 3.05                                           *;
*  1) Run the query to create a left join between the      *;
*     sq.smallcustomer and sq.smalltransaction tables.     *;
*     Notice the difference within the AccountID columns   *;
*     in rows 8 and 9.                                     *;
*  2) Remove the column t.AccountID in the SELECT clause.  *;
*     Run the query and examine the results. How many      *;
*     missing AccountID values are in the results?         *;
*  3) Replace c.AccountID with t.AccountID. Replace the c  *;
*     in the column label with a t. How many missing       *;
*     AccountID values are in the results?                 *;
************************************************************;

proc sql number;
          select FirstName, LastName, Income,
        c.AccountID "c.AccountID", t.AccountID "t.AccountID",
        DateTime, MerchantID, Amount
          from sq.smallcustomer as c left join
       sq.smalltransaction as t
          on c.AccountID = t.AccountID;
quit;
```

Row	FirstName	LastName	Income	c.AccountID	t.AccountID	DateTime	MerchantID	Amount
1	Gary	Sienkiewicz	67210.91	1010159565	1010159565	16SEP18:14:57:08	568268	107.16
2	Sergio	Lefeld	86859.07	1010367330	1010367330	17OCT18:11:02:38	525576	21.02
3	Sergio	Lefeld	86859.07	1010367330	1010367330	15MAY18:17:54:21	542058	23.39
4	John	Oliver	43623.75	2020012887	2020012887	23FEB18:09:25:37	525576	108.22
5	Iva	Bower	67949.96	3030085224	3030085224	27JUL18:12:05:48	525576	26.1
6	Janet	Sienkiewicz	50111.59	3030101942	3030101942	18SEP18:12:13:40	549940	37.38
7	Olga	Comstock	31896.96	3030165207	3030165207	11MAR18:10:07:14	580881	319.95
8	Ada	Vieyra	29586.44	4040164206	.	.	.	.
9	Samantha	Carney	25476.14	5540174271	.	.	.	.

```
************************************************************;
*  Performing a Full Join with PROC SQL                    *;
************************************************************;
*  Syntax                                                  *;
*                                                          *;
*  PROC SQL;                                               *;
*  SELECT col-name, col-name                               *;
*    FROM table1 FULL JOIN table2                          *;
*    ON table1.col-name=table2.col-name;                   *;
*  QUIT;                                                   *;
*
          *;
*  COALESCE(argument-1, argument-2<, ...argument-n>)       *;
************************************************************;


************************************************************;
*  Demo                                                    *;
*  1) Open the s103d03.sas program in the demos folder     *;
*    and find the Demo section of the program. Highlight   *;
*    and run the query. Examine the results. Discuss the   *;
*    values in both AccountID columns.                     *;
*  2) Modify the SELECT clause and remove the t.AccountID  *;
*    column. Highlight and run the query. Examine the      *;
*    results. Discuss the missing values in the            *;
*    c.AccountID column.                                   *;
*  3) Modify the SELECT clause and replace c.AccountID     *;
*    with t.AccountID. Highlight and run the query.        *;
*    Examine the results. Discuss the missing values in    *;
*    the t.AccountID column.                               *;
```

```
*  4) Modify the SELECT clause, use the COALESCE          *;
*     function, and add c.AccountID and t.AccountID as    *;
*     arguments. Add the alias AccountID and the          *;
*     FORMAT=10. column modifier to the newly created     *;
*     column. Highlight and run the query. Examine the    *;
*     results.                                            *;
************************************************************;;


proc sql;
select FirstName, LastName, Income, c.AccountID, t.AccountID,
      DateTime, MerchantID, Amount
   from sq.smallcustomer as c full join
        sq.smalltransaction as t
   on c.AccountID = t.AccountID;
quit;
```

FirstName	LastName	Income	AccountID	AccountID	DateTime	MerchantID	Amount
		.	.	.	07MAY18:15:35:02	542058	58.79
Gary	Sienkiewicz	67210.91	1010159565	1010159565	16SEP18:14:57:08	568268	107.16
		.	.	1010183063	24FEB18:17:27:42	562326	370.53
Sergio	Lefeld	86859.07	1010367330	1010367330	17OCT18:11:02:38	525576	21.02
Sergio	Lefeld	86859.07	1010367330	1010367330	15MAY18:17:54:21	542058	23.39
		.	.	1010367364	18OCT18:17:52:51	549940	37.24
John	Oliver	43623.75	2020012887	2020012887	23FEB18:09:25:37	525576	108.22
Iva	Bower	67949.96	3030085224	3030085224	27JUL18:12:05:48	525576	26.1
Janet	Sienkiewicz	50111.59	3030101942	3030101942	18SEP18:12:13:40	549940	37.38
Olga	Comstock	31896.96	3030165207	3030165207	11MAR18:10:07:14	580881	319.95
		.	.	3030231909	13FEB18:16:48:05	513178	115.48
		.	.	3030231909	27JAN18:13:52:35	536123	1.76
Ada	Vieyra	29586.44	4040164206	.	.	.	.
Samantha	Carney	25476.14	5540174271	.	.	.	.

```
proc sql;
select FirstName, LastName, Income,
            coalesce(c.AccountID, t.AccountID) as AccountID format=15.,
      DateTime, MerchantID, Amount
```

```
   from sq.smallcustomer as c full join
      sq.smalltransaction as t
   on c.AccountID = t.AccountID;
quit;
```

FirstName	LastName	Income	AccountID	DateTime	MerchantID	Amount
		.	.	07MAY18:15:35:02	542058	58.79
Gary	Sienkiewicz	67210.91	1010159565	16SEP18:14:57:08	568268	107.16
		.	1010183063	24FEB18:17:27:42	562326	370.53
Sergio	Lefeld	86859.07	1010367330	17OCT18:11:02:38	525576	21.02
Sergio	Lefeld	86859.07	1010367330	15MAY18:17:54:21	542058	23.39
		.	1010367364	18OCT18:17:52:51	549940	37.24
John	Oliver	43623.75	2020012887	23FEB18:09:25:37	525576	108.22
Iva	Bower	67949.96	3030085224	27JUL18:12:05:48	525576	26.1
Janet	Sienkiewicz	50111.59	3030101942	18SEP18:12:13:40	549940	37.38
Olga	Comstock	31896.96	3030165207	11MAR18:10:07:14	580881	319.95
		.	3030231909	13FEB18:16:48:05	513178	115.48
		.	3030231909	27JAN18:13:52:35	536123	1.76
Ada	Vieyra	29586.44	4040164206	.	.	.
Samantha	Carney	25476.14	5540174271	.	.	.

```
**********************************************************;
*  Activity 3.06                                         *;
*  1) Run the query to create a left join between the    *;
*     sq.smalltransaction2 and sq.smallcustomer2 tables. *;
*     Examine the report. Notice that the rows with      *;
*     missing values in AccountID have been joined.      *;
*  2) In the ON clause, add the expression AND           *;
*     t.AccountID is not null. Run the query. Confirm    *;
*     that missing values were not joined.               *;
*  3) Add a WHERE clause with the expression c.AccountID *;
*     is NULL to filter for all transactions without a   *;
*     documented customer. Run the query and examine the *;
*     report. How many transactions do not have a        *;
*     customer associated with them?                     *;
**********************************************************;
```

```
proc sql;
select FirstName, LastName, Income,
      c.AccountID "c.AccountID",
      t.AccountID "t.AccountID",
      DateTime, MerchantID
   from sq.smalltransaction2 as t left join
       sq.smallcustomer2 as c
     on c.AccountID = t.AccountID;
quit;
```

FirstName	LastName	Income	c.AccountID	t.AccountID	DateTime	MerchantID
Alejandro	Garcia	86324.38	.	.	07MAY18:15:35:02	542058
Alejandro	Garcia	86324.38	.	.	09MAY20:12:30:08	549940
Alejandro	Garcia	86324.38	.	.	16SEP18:14:57:08	568268
Sai	Nair	51256.02	.	.	07MAY18:15:35:02	542058
Sai	Nair	51256.02	.	.	09MAY20:12:30:08	549940
Sai	Nair	51256.02	.	.	16SEP18:14:57:08	568268
Samantha	Carney	25476.14	.	.	07MAY18:15:35:02	542058
Samantha	Carney	25476.14	.	.	09MAY20:12:30:08	549940
Samantha	Carney	25476.14	.	.	16SEP18:14:57:08	568268
		.	.	1010183063	24FEB18:17:27:42	562326
Sergio	Lefeld	86859.07	1010367330	1010367330	17OCT18:11:02:38	525576
Sergio	Lefeld	86859.07	1010367330	1010367330	15MAY18:17:54:21	542058
		.	.	1010367364	18OCT18:17:52:51	549940
John	Oliver	43623.75	2020012887	2020012887	23FEB18:09:25:37	525576
Iva	Bower	67949.96	3030085224	3030085224	27JUL18:12:05:48	525576
Janet	Sienkiewicz	50111.59	3030101942	3030101942	18SEP18:12:13:40	549940
Olga	Comstock	31896.96	3030165207	3030165207	11MAR18:10:07:14	580881
		.	.	3030231909	13FEB18:16:48:05	513178
		.	.	3030231909	27JAN18:13:52:35	536123

```
proc sql;
select FirstName, LastName, Income,
      c.AccountID "c.AccountID",
      t.AccountID "t.AccountID",
      DateTime, MerchantID
```

```
   from sq.smalltransaction2 as t left join
        sq.smallcustomer2 as c
     on c.AccountID = t.AccountID
     and t.AccountID is not null;
quit;
```

FirstName	LastName	Income	c.AccountID	t.AccountID	DateTime	MerchantID
		.	.	.	07MAY18:15:35:02	542058
		.	.	.	09MAY20:12:30:08	549940
		.	.	.	16SEP18:14:57:08	568268
		.	.	1010183063	24FEB18:17:27:42	562326
Sergio	Lefeld	86859.07	1010367330	1010367330	17OCT18:11:02:38	525576
Sergio	Lefeld	86859.07	1010367330	1010367330	15MAY18:17:54:21	542058
		.	.	1010367364	18OCT18:17:52:51	549940
John	Oliver	43623.75	2020012887	2020012887	23FEB18:09:25:37	525576
Iva	Bower	67949.96	3030085224	3030085224	27JUL18:12:05:48	525576
Janet	Sienkiewicz	50111.59	3030101942	3030101942	18SEP18:12:13:40	549940
Olga	Comstock	31896.96	3030165207	3030165207	11MAR18:10:07:14	580881
		.	.	3030231909	13FEB18:16:48:05	513178
		.	.	3030231909	27JAN18:13:52:35	536123

```
proc sql;
select FirstName, LastName, Income,
       c.AccountID "c.AccountID",
       t.AccountID "t.AccountID",
       DateTime, MerchantID
   from sq.smalltransaction2 as t left join
        sq.smallcustomer2 as c
     on c.AccountID = t.AccountID
     and t.AccountID is not null
   where c.AccountID is null;
quit;
```

FirstName	LastName	Income	c.AccountID	t.AccountID	DateTime	MerchantID
		.	.	.	07MAY18:15:35:02	542058
		.	.	.	09MAY20:12:30:08	549940
		.	.	.	16SEP18:14:57:08	568268
		.	.	1010183063	24FEB18:17:27:42	562326
		.	.	1010367364	18OCT18:17:52:51	549940
		.	.	3030231909	13FEB18:16:48:05	513178
		.	.	3030231909	27JAN18:13:52:35	536123

/*Practice Level 1: Using Outer Joins to Find Nonmatches

If necessary, start SAS Studio before you begin.

If you restarted your SAS session, submit your libname.sas program to access the practice data.

Join the sq.globalpop and sq.globalmetadata tables to create the work.meta table.

Use the work.meta table to generate a report showing the country codes for countries in the sq.globalpop table that

do not have any country metadata in the sq.globalmetadata table.

Write a PROC SQL step to join the sq.globalpop and sq.globalmetadata tables and create a table named work.meta.

Use the following requirements:

Select the CountryCode, SeriesName, EstYear1, and EstYear3 columns from the sq.globalpop table and

the ShortName and IncomeGroup columns from the sq.globalmetadata table.

Perform a left join on the sq.globalpop and sq.globalmetadata tables.

Use the CountryCode column in both tables for the join criteria.

Run the program and view the results.

Create a report showing the unique country codes for which there is no global metadata using the work.meta table.

Select the CountryCode column from the work.meta table and eliminate duplicate values.

Filter for rows where the ShortName column is missing. The ShortName column contains values from the sq.globalmetadata table.

If the results are missing, then the row did not retrieve information from sq.globalmetadata.

Order the results by CountryCode.

Run the program and view the results.

What is the last CountryCode value in your results?

```
*/
proc sql;
create table work.meta as
select p.CountryCode, p.SeriesName, p.EstYear1, p.EstYear3, m.ShortName, m.IncomeGroup
        from sq.globalpop as p left join
                sq.globalmetadata as m
                on p.CountryCode = m.CountryCode;
quit;
```

Total rows: 3978 Total columns: 6 Rows 1-1

	CountryCode	SeriesName	EstYear1	EstYear3	ShortName	IncomeGroup
1	ABW	Population 70-74	3179	3641		
2	ABW	Population 65-69	4638	5339		
3	ABW	Population 60-64	6247	6983		
4	ABW	Population 00-04	5810	5394		
5	ABW	Population 10-14	7231	7060		
6	ABW	Population 40-44	7645	7236		
7	ABW	Population 80+	2048	2355		
8	ABW	Population 45-49	8485	7899		
9	ABW	Population 30-34	5857	5399		
10	ABW	Population 50-54	9264	8999		
11	ABW	Population 25-29	5127	6063		
12	ABW	Population 20-24	6979	7679		
13	ABW	Population 35-39	6821	6325		
14	ABW	Population 75-79	2290	2435		
15	ABW	Population 55-59	7783	8610		
16	ABW	Population 05-09	6813	6396		
17	ABW	Population 15-19	7581	7443		
18	AFG	Population 25-29	2412180	2699261	Afghanistan	Low income
19	AFG	Population 45-49	1031057	1160666	Afghanistan	Low income
20	AFG	Population 65-69	373492	411313	Afghanistan	Low income

```
title "Country Codes without global metadata";
proc sql;
select distinct CountryCode
        from work.meta
```

where ShortName is null

order by CountryCode;

run;

title;

Country Codes without global metadata

CountryCode
ABW
ATG
BHS
BRB
BRN
CEB
CHI
CPV
CSS
CUB
CUW
EAP
EAR
ECA
EUU
FCS
FJI

```
*********************************************;
*Alternative Solution without Temporary Table  *;
*********************************************;
title 'Countries with no Metadata';
title2 'No Temporary Table';
proc sql;
select distinct p.CountryCode
    from sq.globalpop as p left join
         sq.globalmetadata as m
     on p.CountryCode= m.CountryCode
     where ShortName is null;
quit;
title;
```

/*Practice Level 2: Using Outer Joins to Summarize Data

If necessary, start SAS Studio before you begin.

If you restarted your SAS session, submit your libname.sas program to access the practice data.

Generate a report showing the count of customer marital status descriptions for each primary bank.

The sq.customer table contains a marital code (Married) and a primary bank ID.

The final results should contain BankID, MaritalStatus, Name (name of bank), and Count.

Write a PROC SQL step to join the sq.customer and sq.maritalcode tables. Use the following requirements:

Select BankID from the sq.customer table and MaritalStatus value from the sq.maritalcode table.

Create a new column named Count to count the number of customers. Format the new column using commas.

Use a left join to select all customers from the sq.customer table, with or without matches in the sq.maritalcode table.

Use the Married column in the sq.customer table and the MaritalCode column in the sq.maritalcode table as the join criteria.

Filter rows where the customer's bank ID is not missing.

Group the data by BankID and MaritalStatus.

Order the table by descending Count.

As a checkpoint, run the query.

In the same PROC SQL step, add the descriptive bank Name column to the results.

After the MaritalStatus column in the SELECT clause, add Name from the sq.bank table based on matching BankID values,

again using a left join.

Add Name in the GROUP BY clause after the MaritalStatus column.

Correctly reference the BankID columns in the query.

Add an appropriate title.

Run the program and view the results.

Which combination of MaritalStatus and Name has the lowest count of customers?

*/

proc sql;

select c.BankID, m.MaritalStatus, count(c.BankID) as Count format=comma16.

```
        from sq.customer as c left join
                sq.maritalcode as m
                on c.married = m.maritalcode
        where c.BankID is not null
        group by c.BankID, m.MaritalStatus
        order by Count desc;;
quit;
```

BankID	Marital Status	Count
101010101	Married	22,933
202020202	Married	16,997
303030303	Married	14,258
101010101	Single	8,793
202020202	Single	6,589
303030303	Single	5,458
101010101	Divorced	4,896
202020202	Divorced	3,644
303030303	Divorced	3,054
101010101		2,814
202020202		2,162
303030303		1,802
101010101	Widowed	739
202020202	Widowed	549
303030303	Widowed	362

```
title "Count of Customer Marital Status Description for each primary bank";
proc sql;
select c.BankID, m.MaritalStatus, b.Name, count(c.BankID) as Count format=comma16.
        from sq.customer as c
                left join sq.maritalcode as m
                on c.married = m.maritalcode
                left join sq.bank as b
                on c.BankID = b.BankID
        where c.BankID is not null
        group by c.BankID, m.MaritalStatus, b.Name
```

```
        order by Count desc;;
quit;
title;
```

Count of Customer Marital Status Description for each primary bank

BankID	MaritalStatus	Name	Count
101010101	Married	Biggest Bank, Inc.	22,933
202020202	Married	Sailors Credit Union	16,997
303030303	Married	Wheatberry Bank, Inc.	14,258
101010101	Single	Biggest Bank, Inc.	8,793
202020202	Single	Sailors Credit Union	6,589
303030303	Single	Wheatberry Bank, Inc.	5,458
101010101	Divorced	Biggest Bank, Inc.	4,896
202020202	Divorced	Sailors Credit Union	3,644
303030303	Divorced	Wheatberry Bank, Inc.	3,054
101010101		Biggest Bank, Inc.	2,814
202020202		Sailors Credit Union	2,162
303030303		Wheatberry Bank, Inc.	1,802
101010101	Widowed	Biggest Bank, Inc.	739
202020202	Widowed	Sailors Credit Union	549
303030303	Widowed	Wheatberry Bank, Inc.	362

```
************************************************************;
* Performing a Reflexive Join                    *;
************************************************************;
* Syntax                                  *;
*                                      *;
* PROC SQL;                                  *;
*   SELECT col-name, col-name                    *;
*   FROM table1 INNER JOIN table1                  *;
*   ON table1.col-name=table1.col-name;            *;
* QUIT;                                  *;
************************************************************;


************************************************************;
```

```
*  Demo                                    *;
*  1) Open the s103d04.sas program in the demos folder    *;
*     and find the Demo section of the program. Highlight *;
*     and run the query. Examine the results.             *;
*  2) Modify the query to create a reflexive join. In the *;
*     FROM clause, add an inner join followed by the      *;
*     sq.employee table again. Add the alias m to the     *;
*     second sq.employee table. Add the ON clause and set *;
*     e.ManagerID equal to m.EmployeeID.                  *;
*  3) Add the EmployeeName column in the SELECT clause.   *;
*     Qualify the new EmployeeName column with the table  *;
*     alias m. Highlight and run the query. Examine the   *;
*     results.                                            *;
*  4) Add the column alias ManagerName to the             *;
*     m.EmployeeName column and an ORDER BY clause to     *;
*     sort by ManagerName. Highlight and run the query.   *;
*     Examine the results.                                *;
************************************************************;

proc sql;
select e.EmployeeID, e.EmployeeName, e.StartDate format=date9., e.ManagerID
        from sq.employee as e;
quit;
```

EmployeeID	EmployeeName	StartDate	ManagerID
121044	Abbott, Ray	01AUG1987	121144
120145	Aisbitt, Sandy	01JUN1997	120103
120761	Akinfolarin, Tameaka	01JUL2018	120746
120656	Amos, Salley	01MAR2011	120660
121107	Anger, Rose	01JUL2018	121143
121038	Anstey, David	01AUG2018	121144
120273	Antonini, Doris	01MAR2018	120271
120759	Apr, Nishan	01JAN1995	120746

```
proc sql;
select e.EmployeeID, e.EmployeeName, e.StartDate format=date9., e.ManagerID,
        m.EmployeeName as ManagerName
    from sq.employee as e inner join
        sq.employee as m
    on e.ManagerID = m.EmployeeID
    order by ManagerName;
quit;
```

EmployeeID	EmployeeName	StartDate	ManagerID	ManagerName
120803	Droste, Victor	01JAN1990	120798	Ardskin, Elizabeth
120804	Zied, Ahmed	01JAN1986	120798	Ardskin, Elizabeth
120808	Dupree, Marcel	01JUN1996	120798	Ardskin, Elizabeth
120791	Chiseloff, Richard	01OCT1998	120798	Ardskin, Elizabeth
120805	Walker, Robert	01APR2012	120798	Ardskin, Elizabeth
120806	Ousley, Lorna	01FEB2004	120798	Ardskin, Elizabeth
120801	Kennedy, Kathryn	01JUL2011	120798	Ardskin, Elizabeth
120810	Esguerra, Loyal	01MAR1992	120798	Ardskin, Elizabeth
120809	Marion, Chiorene	01JAN1986	120798	Ardskin, Elizabeth
120802	Parker, U'Vonda	01JAN1990	120798	Ardskin, Elizabeth
120807	Peppers, Gerlinde	01JAN1994	120798	Ardskin, Elizabeth
120794	Cross, Samantha	01JUL2015	120800	Benyami, Fred
120814	Scroggin, Victor	01SEP1991	120800	Benyami, Fred
120799	Stefandonovan, Jeffery	01NOV2010	120800	Benyami, Fred
120798	Ardskin, Elizabeth	01JAN1999	120800	Benyami, Fred
120118	Hartshorn, Darshi	01JUL1996	120104	Billington, Kareen
120115	Nichollas, Hugh	01AUG2017	120104	Billington, Kareen

```
************************************************************;
*  Activity 3.07                                  *;
*  1) Examine and run the query. Did you receive a syntax *;
*     error?                                      *;
*  2) In the ON clause, use the SUBSTR function on        *;
*     t.StateID to extract the first two characters. Run  *;
*     the query. Which StateName is Caberto, Glen Daniel  *;
*     from?                                       *;
************************************************************;


proc sql inobs=100 number;
select StateID, CustomerName, StateName
   from sq.transactionfull as t inner join
        sq.statecode as s
     on substr(t.StateID, 1, 2) = s.StateCode;
quit;
```

Row	StateID	CustomerName	StateName
1	CA37492351	Caberto, Glen Daniel	California
2	CA53344918	Lefeld, Sergio Vance	California
3	CA95831948	Lefeld, Linda Erica	California
4	NY67246023	Bowers, Margaret Katie	New York
5	CA57669199	Kennedy, Lisa Diane	California
6	CA95831948	Lefeld, Linda Erica	California
7	NY14984651	Balo, Cynthia Patricia	New York
8	NY22390152	Sienkiewicz, Janet Elisa	New York
9	NY72156740	Balo, Edna Sherry	New York
10	NY42392270	Bower, Omar Randy	New York
11	NY14984651	Balo, Cynthia Patricia	New York
12	CA53344918	Lefeld, Sergio Vance	California
13	NY93959176	Comstock, Olga Cathy	New York
14	CA65581238	Kennedy, Daniel Eric	California
15	CA95831948	Lefeld, Linda Erica	California
16	NY55955333	Pennacchio, Joan Lynn	New York
17	CA28413396	Oliver, John Paul	California
18	CA95831948	Lefeld, Linda Erica	California
19	NY98630507	Alexander, Ruth Helen	New York
20	CA65581238	Kennedy, Daniel Eric	California
21	CA38929875	Maiden, Pamela Melissa	California
22	NY68565522	Bowers, Douglas Tim	New York
23	NY85467857	Balo, Crystal Diane	New York
24	NY98630507	Alexander, Ruth Helen	New York
25	NY93959176	Comstock, Olga Cathy	New York
26	CA13587032	Caberto, Robert Jason	California
27	NY68565522	Bowers, Douglas Tim	New York
28	CA13587032	Caberto, Robert Jason	California
29	CA37492351	Caberto, Glen Daniel	California
30	CA53344918	Lefeld, Sergio Vance	California

```
************************************************************;
* Activity 3.09                                            *;
* 1) If you have not run the queries in the Create a       *;
*    Table and Insert Values section, run those now.       *;
* 2) Use the PUT function to convert z.Zip in the ON       *;
*    clause to a character value using the z5 format.      *;
*    Run the query.                                        *;
* 3) What city does the ZipCode value 14216 represent?     *;
```

```
************************************************************;
proc sql;
select c.CustomerID, c.ZipCode, c.Gender,
      z.Zip, z.City, z.StateCode
   from customerzip as c inner join
        sashelp.zipcode as z
     on c.ZipCode = put(z.Zip, z5.);
quit;
```

CustomerID	ZipCode	Gender	The 5-digit ZIP Code	Name of city/org	Two-letter abbrev. for state name.
2	04429	M	04429	Holden	ME
5	14216	M	14216	Buffalo	NY
1	14580	M	14580	Webster	NY
4	27519	M	27519	Cary	NC
3	50101	M	50101	Galt	IA