Battle of the Titans

DATA Step versus PROC SQL

MINSUG

30 Jul 2025

Battle of the Titans - DATA Step versus PROC SQL

Charu Shankar, SAS® Institute

With a background in computer systems management. SAS Instructor Charu Shankar engages with logic, visuals, and analogies to spark critical thinking since 2007.

Charu curates and delivers unique content on SAS, SQL, Viya, etc. to support users in the adoption of SAS software.

When not coding, Charu teaches yoga and loves to explore Canadian trails with her husky Miko.

Agenda

DATA Step and PROC SQL Basics

Battle #1: SAS Data Set Creation

Battle #2: Data Access, Retrieval, and Manipulation

Battle #3: Logic Scenarios

Battle #4: By-group Processing

Battle #5: Combining Data Sets (or Tables)

Handy Links

DATA Step and PROC SQL Basics

DATA Step and PROC SQL Basics

DATA Step

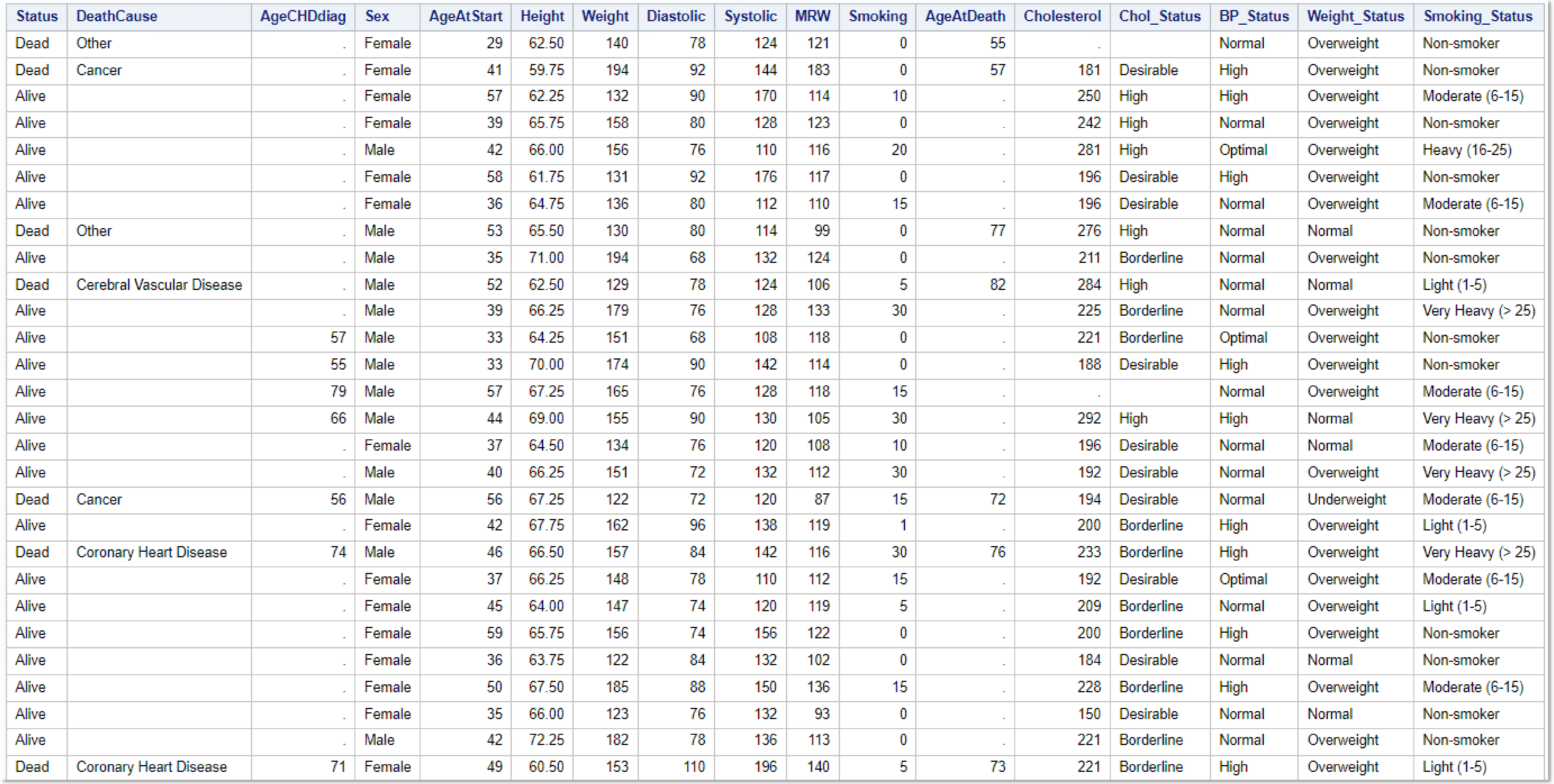
- The power of the DATA step has enabled SAS users to leverage many capabilities and features since the SAS software was conceived.
- Programmers, developers, data engineers, data scientists, statisticians, and users from various industries utilize the DATA step to create, retrieve, update, and process SAS data sets with various SAS statements.

PROC SQL

- PROC SQL is a powerful tool for data access, retrieval, and analysis, and a popular technique for combining data from multiple data sets or relational database tables.

Presentation Data Sets

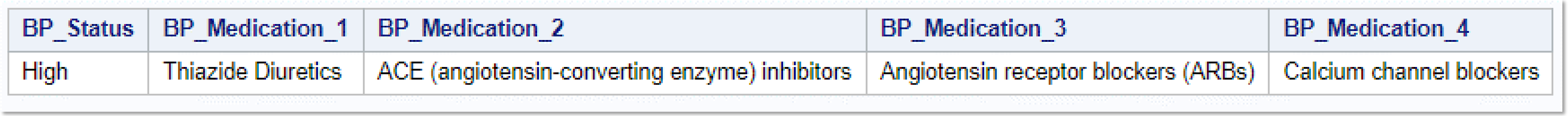
SASHELP.Heart Data Set



Status	DeathCause	AgeCHDdiag	Sex	AgeAtStart	Height	Weight	Diastolic	Systolic	MRW	Smoking	AgeAtDeath	Cholesterol	Chol_Status	BP_Status	Weight_Status	Smoking_Status
Dead	Other	.	Female	29	62.50	140	78	124	121	0	55	.		Normal	Overweight	Non-smoker
Dead	Cancer	.	Female	41	59.75	194	92	144	183	0	57	181	Desirable	High	Overweight	Non-smoker
Alive		.	Female	57	62.25	132	90	170	114	10	.	250	High	High	Overweight	Moderate (6-15)
Alive		.	Female	39	65.75	158	80	128	123	0	.	242	High	Normal	Overweight	Non-smoker
Alive		.	Male	42	66.00	156	76	110	116	20	.	281	High	Optimal	Overweight	Heavy (16-25)
Alive		.	Female	58	61.75	131	92	176	117	0	.	196	Desirable	High	Overweight	Non-smoker
Alive		.	Female	36	64.75	136	80	112	110	15	.	196	Desirable	Normal	Overweight	Moderate (6-15)
Dead	Other	.	Male	53	65.50	130	80	114	99	0	77	276	High	Normal	Normal	Non-smoker
Alive		.	Male	35	71.00	194	68	132	124	0	.	211	Borderline	Normal	Overweight	Non-smoker
Dead	Cerebral Vascular Disease	.	Male	52	62.50	129	78	124	106	5	82	284	High	Normal	Normal	Light (1-5)
Alive		.	Male	39	66.25	179	76	128	133	30	.	225	Borderline	Normal	Overweight	Very Heavy (> 25)
Alive		57	Male	33	64.25	151	68	108	118	0	.	221	Borderline	Optimal	Overweight	Non-smoker
Alive		55	Male	33	70.00	174	90	142	114	0	.	188	Desirable	High	Overweight	Non-smoker
Alive		79	Male	57	67.25	165	76	128	118	15	.	.		Normal	Overweight	Moderate (6-15)
Alive		66	Male	44	69.00	155	90	130	105	30	.	292	High	High	Normal	Very Heavy (> 25)
Alive		.	Female	37	64.50	134	76	120	108	10	.	196	Desirable	Normal	Normal	Moderate (6-15)
Alive		.	Male	40	66.25	151	72	132	112	30	.	192	Desirable	Normal	Overweight	Very Heavy (> 25)
Dead	Cancer	56	Male	56	67.25	122	72	120	87	15	72	194	Desirable	Normal	Underweight	Moderate (6-15)
Alive		.	Female	42	67.75	162	96	138	119	1	.	200	Borderline	High	Overweight	Light (1-5)
Dead	Coronary Heart Disease	74	Male	46	66.50	157	84	142	116	30	76	233	Borderline	High	Overweight	Very Heavy (> 25)
Alive		.	Female	37	66.25	148	78	110	112	15	.	192	Desirable	Optimal	Overweight	Moderate (6-15)
Alive		.	Female	45	64.00	147	74	120	119	5	.	209	Borderline	Normal	Overweight	Light (1-5)
Alive		.	Female	59	65.75	156	74	156	122	0	.	200	Borderline	High	Overweight	Non-smoker
Alive		.	Female	36	63.75	122	84	132	102	0	.	184	Desirable	Normal	Normal	Non-smoker
Alive		.	Female	50	67.50	185	88	150	136	15	.	228	Borderline	High	Overweight	Moderate (6-15)
Alive		.	Female	35	66.00	123	76	132	93	0	.	150	Desirable	Normal	Normal	Non-smoker
Alive		.	Male	42	72.25	182	78	136	113	0	.	221	Borderline	Normal	Overweight	Non-smoker
Dead	Coronary Heart Disease	71	Female	49	60.50	153	110	196	140	5	73	221	Borderline	High	Overweight	Light (1-5)

5,209 observations and 17 variables

High_Blood_Pressure_Medications Data Set



BP_Status	BP_Medication_1	BP_Medication_2	BP_Medication_3	BP_Medication_4
High	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers

1 observation and 5 variables

Battle #1
SAS Data Set Creation

SAS Data Set Creation

- In this first battle we will demonstrate how to create SAS data sets (or tables) using the DATA step and PROC SQL.
- We will highlight the advantage(s) and disadvantage(s) between the DATA step versus PROC SQL as we demonstrate, compare, and contrast between both approaches.

SAS Data Set Creation

DATA Step

```
LIBNAME MYDATA "/home/username/Data Sources" ;
DATA MYDATA.High_Blood_Pressure_Medications ;
  LENGTH BP_Status    $7.
         BP_Medication_1
         BP_Medication_2
         BP_Medication_3
         BP_Medication_4 $50. ;
  BP_Status = "High" ;
  BP_Medication_1 = "Thiazide Diuretics" ;
  BP_Medication_2 = "ACE (angiotensin-converting
enzyme) inhibitors" ;
  BP_Medication_3 = "Angiotensin receptor
blockers (ARBs)" ;
  BP_Medication_4 = "Calcium channel blockers" ;
RUN ;
PROC PRINT
DATA=MYDATA.High_Blood_Pressure_Medications
NOOBS ;
RUN ;
```

PROC SQL

```
LIBNAME MYDATA "/home/username/Data Sources" ;
PROC SQL ;
  CREATE TABLE MYDATA.High_Blood_Pressure_Medications
    (BP_Status CHAR(7)
    ,BP_Medication_1 CHAR(50)
    ,BP_Medication_2 CHAR(50)
    ,BP_Medication_3 CHAR(50)
    ,BP_Medication_4 CHAR(50)) ;
  INSERT INTO MYDATA.High_Blood_Pressure_Medications
    VALUES("High"
          ,"Thiazide Diuretics"
          ,"ACE (angiotensin-converting enzyme)
inhibitors"
          ,"Angiotensin receptor blockers (ARBs)"
          ,"Calcium channel blockers") ;
  SELECT * FROM MYDATA.High_Blood_Pressure_Medications ;
QUIT ;
```

SAS Data Set Creation – Discussion

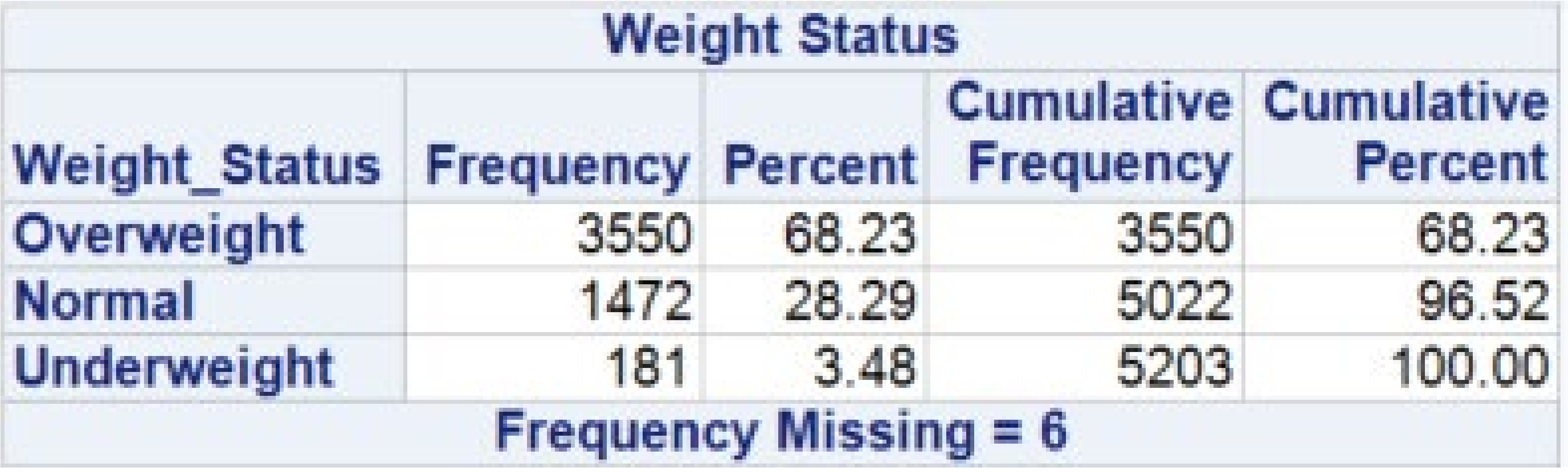
- While both techniques yield the same result, to display the data set results with the DATA step requires using the PRINT procedure (or another procedure).
- However, with PROC SQL, the SELECT statement automatically displays the data set results.
- Thus, with PROC SQL, the data set is created, populated, and displayed in one procedure.

Create Multiple Data Sets

- In the previous battles, at the outset, it appears that both the DATA step and PROC SQL produces the same outcome and indeed that is correct.
- However, it is important to point out one edge the DATA step has. That is the ability to create multiple output data sets in one read of the input table, thereby saving valuable input/output resources.
- Let's take the weight-status column to do some conditional processing. In the next example, a FREQ procedure is used to identify unique values of Weight_status.

Identify Unique Values with PROC FREQ

```
PROC FREQ DATA=SASHELP.Heart ORDER=FREQ ;
      TABLES weight_status ;
RUN ;
```



Weight Status

Weight_Status	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Overweight	3550	68.23	3550	68.23
Normal	1472	28.29	5022	96.52
Underweight	181	3.48	5203	100.00

Frequency Missing = 6

Create Multiple Data Sets

DATA Step

```
DATA WORK.overWt
     WORK.normalWt
     WORK.underWt ;
  SET SASHELP.Heart ;
  IF weight_status="Overweight" THEN
     OUTPUT WORK.overWt ;
  ELSE IF weight_status="Normal" THEN
       OUTPUT WORK.normalWt ;
  ELSE IF weight_status="Underweight" THEN
       OUTPUT WORK.underWt ;
RUN ;
PROC CONTENTS DATA=WORK._ALL_ ;
RUN ;
```

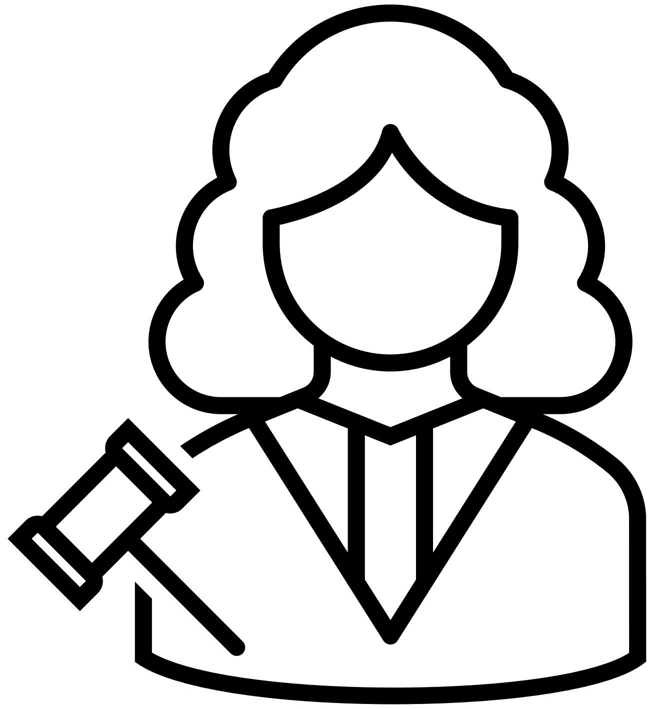
PROC SQL

```
PROC SQL ;
  CREATE TABLE WORK.overWt AS
   SELECT * FROM SASHELP.Heart
    WHERE weight_status="Overweight" ;

  CREATE TABLE WORK.normalwt AS
   SELECT * FROM SASHELP.Heart
    WHERE weight_status="Normal" ;

  CREATE TABLE WORK.underWt AS
   SELECT * FROM SASHELP.Heart
    WHERE weight_status="Underweight" ;
QUIT ;
PROC CONTENTS DATA=WORK._ALL_ ;
RUN ;
```

SAS Data Set Creation – Decision Point



While both techniques yield the same result, it is important to point out one edge the DATA step has over PROC SQL – that is the ability to create multiple output data sets in one read of the input table, thereby saving valuable input/output resources.

Battle #2
Data Access, Retrieval, and Manipulation

Read an XLSX Data File

- In this second battle we turn our attention to data access, data retrieval, and data manipulation techniques.

Read an XLSX Data File

In this next example, we'll access and read an Excel spreadsheet by specifying a LIBNAME statement with the XLSX engine and PROC SQL.

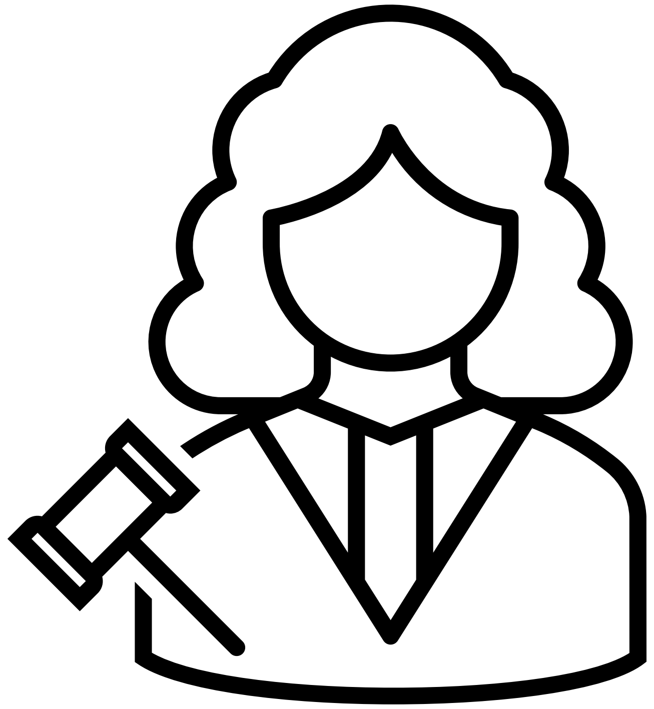
```
LIBNAME MYDATA XLSX "/home/Data
Sources/Heart.xlsx" ;


DATA WORK.Heart_DATA_Step ;
      SET MYDATA.Heart ;
RUN ;
```

```
LIBNAME MYDATA XLSX "/home/Data
Sources/Heart.xlsx" ;

PROC SQL ;
  CREATE TABLE WORK.Heart_PROC_SQL
AS
    SELECT *
      FROM MYDATA.Heart ;
QUIT ;
```

Read an XLSX Data File– Decision Point



- The result of running both approaches is that there is no appreciable difference between the DATA step and PROC SQL approach.

Battle #3
Logic Scenarios

Logic Scenarios

- Both the DATA step and PROC SQL provide us with the ability to apply logic scenarios in our programs so they can conditionally do or perform the operations we desire – “if one condition is true, then do X but if another condition is true, then do Y.”
- But before engaging in this battle of logic scenarios we’ll introduce comparison and logical operators.

Operators

Comparison operators are used in the DATA step and PROC SQL to compare one character or numeric value to another. Comparison operators, mnemonics, and their descriptions are displayed in the following table.

SAS Operator	Mnemonic Operator	Description
=	EQ	Equal to
^=	NE	Not equal to
<	LT	Less than
<=	LE	Less than or equal to
>	GT	Greater than
>=	GE	Greater than or equal to

Logical operators are used to connect two or more expressions together. The three operators include AND, OR, and NOT as is displayed in the following table.

SAS Operator	Mnemonic Operator	Description
AND	&	All expressions (conditions) must be true.
OR	|	Any of the expressions (conditions) can be true.
NOT	^ or ~	Negate or reverse the logic of a comparison.

SELECT-WHEN-OTHERWISE with the DATA Step

In this battle we’ll explore the application of a “SELECT-WHEN-OTHERWISE” conditional logic scenario used with the DATA step.

```
DATA WORK.Smoke ;
SET SASHELP.Heart ;
   SELECT (Smoking_Status) ;
   WHEN (“Non-smoker”)        My_Smoking_Status = “Non-smoker” ;
   WHEN (“Light (1-5)”)       My_Smoking_Status = “Light-smoker” ;
   WHEN (“Moderate (6-15)”)   My_Smoking_Status = “Moderate-smoker” ;
      WHEN (“Heavy (16-25)”)     My_Smoking_Status = “Heavy-smoker” ;
         WHEN (“Very Heavy (> 25)”) My_Smoking_Status = “Very Heavy-smoker”
   ;
         OTHERWISE                  My_Smoking_Status = “Unknown” ;
      END ;
      KEEP Sex Status Smoking_Status My_Smoking_Status ;
   RUN ;
   PROC PRINT DATA = WORK.Smoke NOOBS ;
   RUN ;
```

Select Case Expression with PROC SQL

Case expressions provide a method of reclassifying or regrouping data into separate and unique groups. A PROC SQL SELECT with CASE expression reduces keystrokes when constructing logic scenarios.

```
PROC SQL ;
  SELECT Sex
       , Status
       , Smoking_Status
       , CASE Smoking_Status
           WHEN "Non-smoker"        THEN "Non-smoker"
           WHEN "Light (1-5)"       THEN "Light-smoker"
           WHEN "Moderate (6-15)"   THEN "Moderate-smoker"
           WHEN "Heavy (16-25)"     THEN "Heavy-smoker"
           WHEN "Very Heavy (> 25)" THEN "Very Heavy-smoker"
           ELSE "Unknown"
         END AS My_Smoking_Status
    FROM SASHELP.Heart ;
QUIT ;
```

DATA Step Comparison and Logical Operators

A SELECT-WHEN-OTHERWISE expression supports the application of logic scenarios with comparison and logical operators.

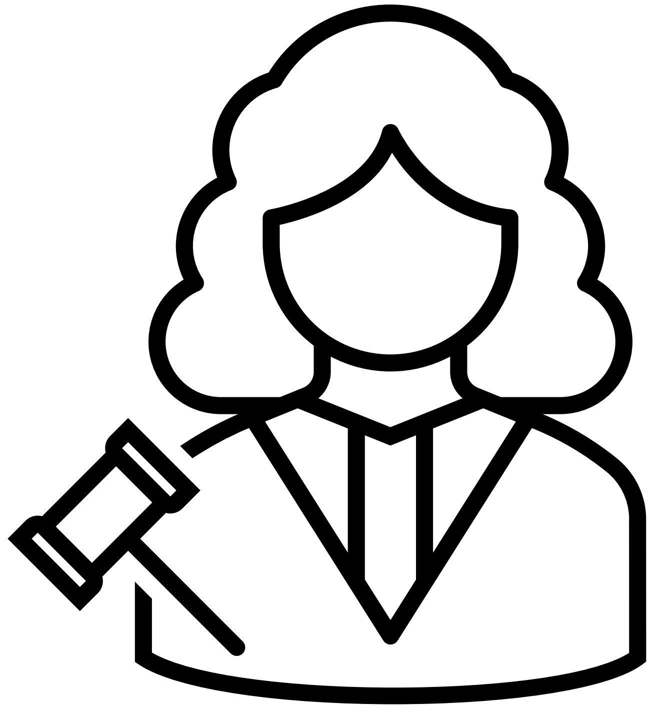
```
DATA WORK.Smoke_BP ;
  SET SASHELP.Heart ;
  LENGTH My_Smoking_Status $50;
  SELECT ;
    WHEN (Smoking_Status = “Non-smoker”)      My_Smoking_Status = “Non-smoker” ;
    WHEN (Smoking_Status = “Light (1-5)”)     My_Smoking_Status = “Light-smoker” ;
    WHEN (Smoking_Status = “Moderate (6-15)”) My_Smoking_Status = “Moderate-smoker” ;
    WHEN (Smoking_Status = “Heavy (16-25)” and BP_Status = “High”)
                  My_Smoking_Status = “Heavy-smoker with High Blood Pressure” ;
    WHEN (Smoking_Status = “Very Heavy (> 25)” and BP_Status = “High”)
                  My_Smoking_Status = “Very Heavy-smoker with High Blood Pressure” ;
    OTHERWISE                                 My_Smoking_Status = “Unknown” ;
  END ;
  KEEP Sex Status Smoking_Status My_Smoking_Status ;
RUN ;
```

PROC SQL Comparison and Logical Operators

A PROC SQL searched CASE expression supports the application of logic scenarios with comparison and logical operators.

```
PROC SQL ;
  SELECT Sex
       , Status
       , BP_Status
       , Smoking_Status
       , CASE
           WHEN Smoking_Status = "Non-smoker"        THEN "Non-smoker"
           WHEN Smoking_Status = "Light (1-5)"       THEN "Light-smoker"
           WHEN Smoking_Status = "Moderate (6-15)"   THEN "Moderate-smoker"
           WHEN Smoking_Status = "Heavy (16-25)"
            AND BP_Status = "High" THEN "Heavy-smoker with High Blood Pressure"
           WHEN Smoking_Status = "Very Heavy (> 25)"
            AND BP_Status = "High" THEN "Very Heavy-smoker with High Blood Pressure"
           ELSE "Unknown"
         END AS My_Smoking_Status
    FROM SASHELP.Heart ;
QUIT ;
```

Comparison and Logical Operators – Decision Point



In this battle, we may have to call it a draw since there is no clear ‘winner’. It comes down to which syntax you are most comfortable with.

Battle #4
By-group Processing

Maximum Value in By -groups with DATA Step

- SAS users often need the ability to identify the first (beginning) and last (ending) observation as well as the between observation(s) in a by-group.
- The DATA step is the "go-to" approach used by many but PROC SQL can also be used to emulate this stalwart DATA step approach.
- In the next example, a PROC SORT and DATA step illustrates a popular BY-group processing technique where two temporary variables: FIRST(dot) and LAST(dot) are automatically created to identify and select the first and last observation in a BY-group.

Maximum Value in By -groups with DATA Step

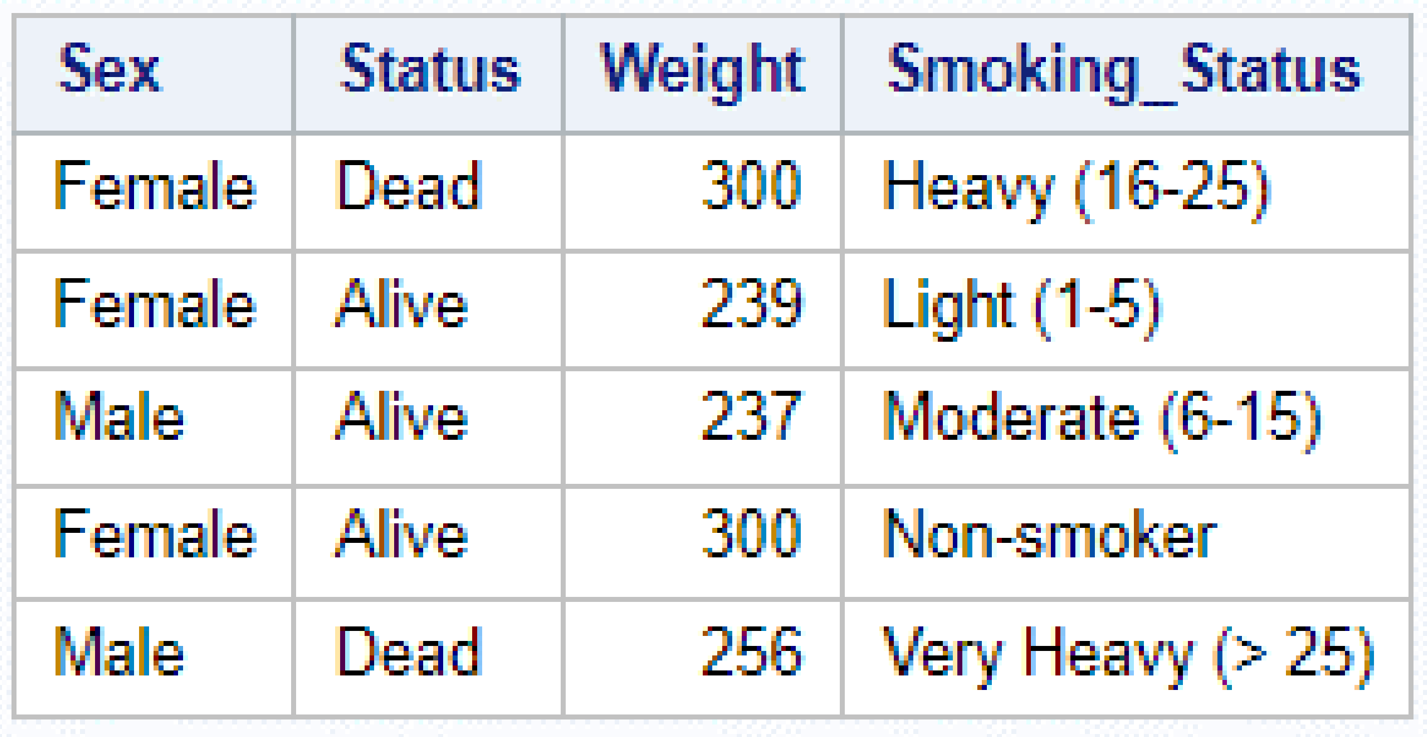
SAS provides users with the ability to identify the maximum value in a By-group with the DATA step.

```
proc sort data=sashelp.heart(keep=sex status weight smoking_status)
           out=work.heart_sorted;
  by smoking_status descending weight;
  where smoking_status ne "" and weight ne .;
run;

data work.by_group_obs;
  set work.heart_sorted;
  by smoking_status;
  if first.smoking_status then output;
run;

proc print data=work.by_group_obs noobs;
  var sex status weight smoking_status;
run;
```

Maximum Value in By -groups with DATA Step



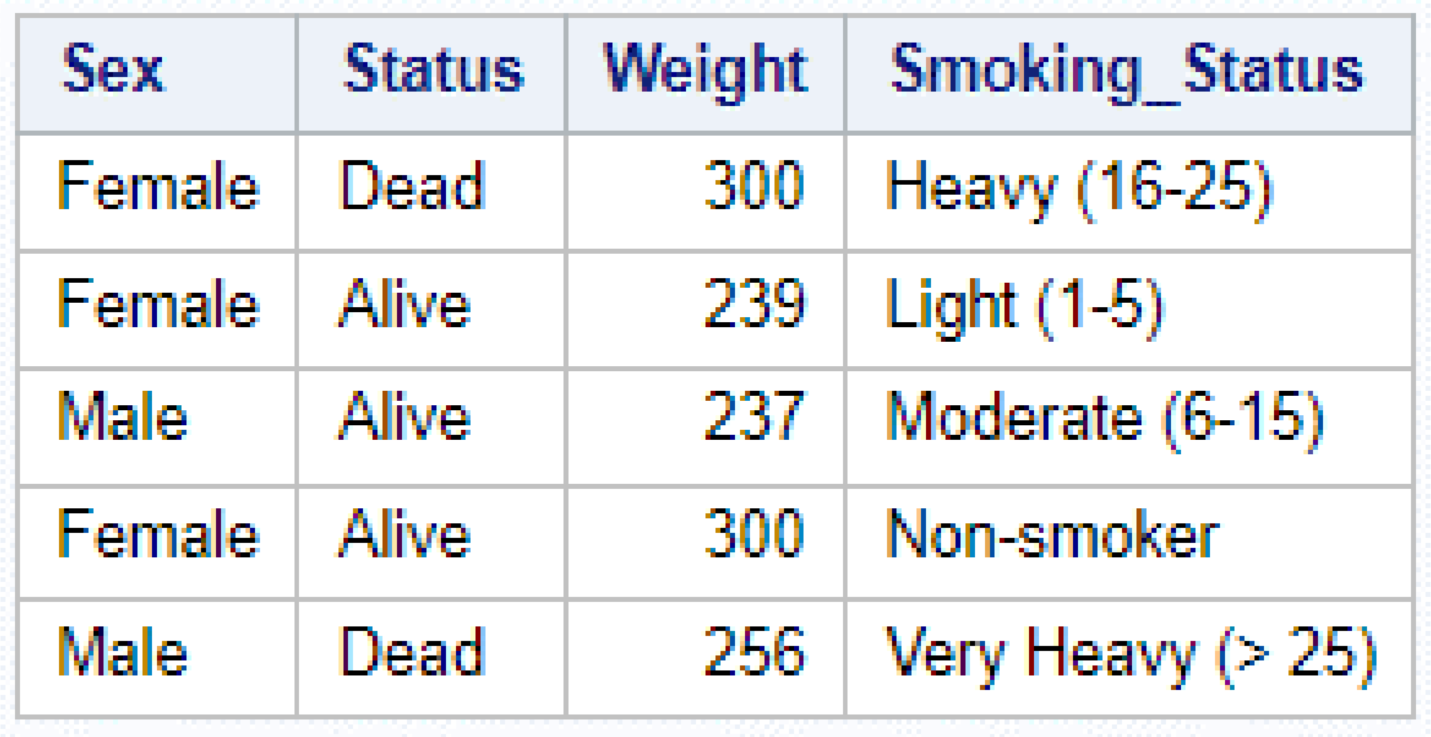
Sex	Status	Weight	Smoking_Status
Female	Dead	300	Heavy (16-25)
Female	Alive	239	Light (1-5)
Male	Alive	237	Moderate (6-15)
Female	Alive	300	Non-smoker
Male	Dead	256	Very Heavy (> 25)

Maximum Value in By -groups with PROC SQL

SAS provides users with the ability to identify the maximum value in a By-group with PROC SQL.

```
proc sql;
  select sex,
         status,
         weight,
         smoking_status
    from sashelp.heart(keep=sex status weight smoking_status)
   group by smoking_status
  having weight = max(weight) and smoking_status ne "";
quit;
```

Maximum Value in By -groups with PROC SQL



Sex	Status	Weight	Smoking_Status
Female	Dead	300	Heavy (16-25)
Female	Alive	239	Light (1-5)
Male	Alive	237	Moderate (6-15)
Female	Alive	300	Non-smoker
Male	Dead	256	Very Heavy (> 25)

By Group Processing– Decision Point

As seen both approaches yield the same results, but PROC SQL is more concise and does not require the pre-sorting that is needed in the DATA step or the display of the results.

Battle #5 -
Combine Data Sets (or Tables)

Combining Data Sets (or Tables)

In the fifth, and final, battle we illustrate various DATA step and PROC SQL techniques to combine two data sets (or tables) together. The following table illustrates merge / join techniques using Venn diagrams, see below.

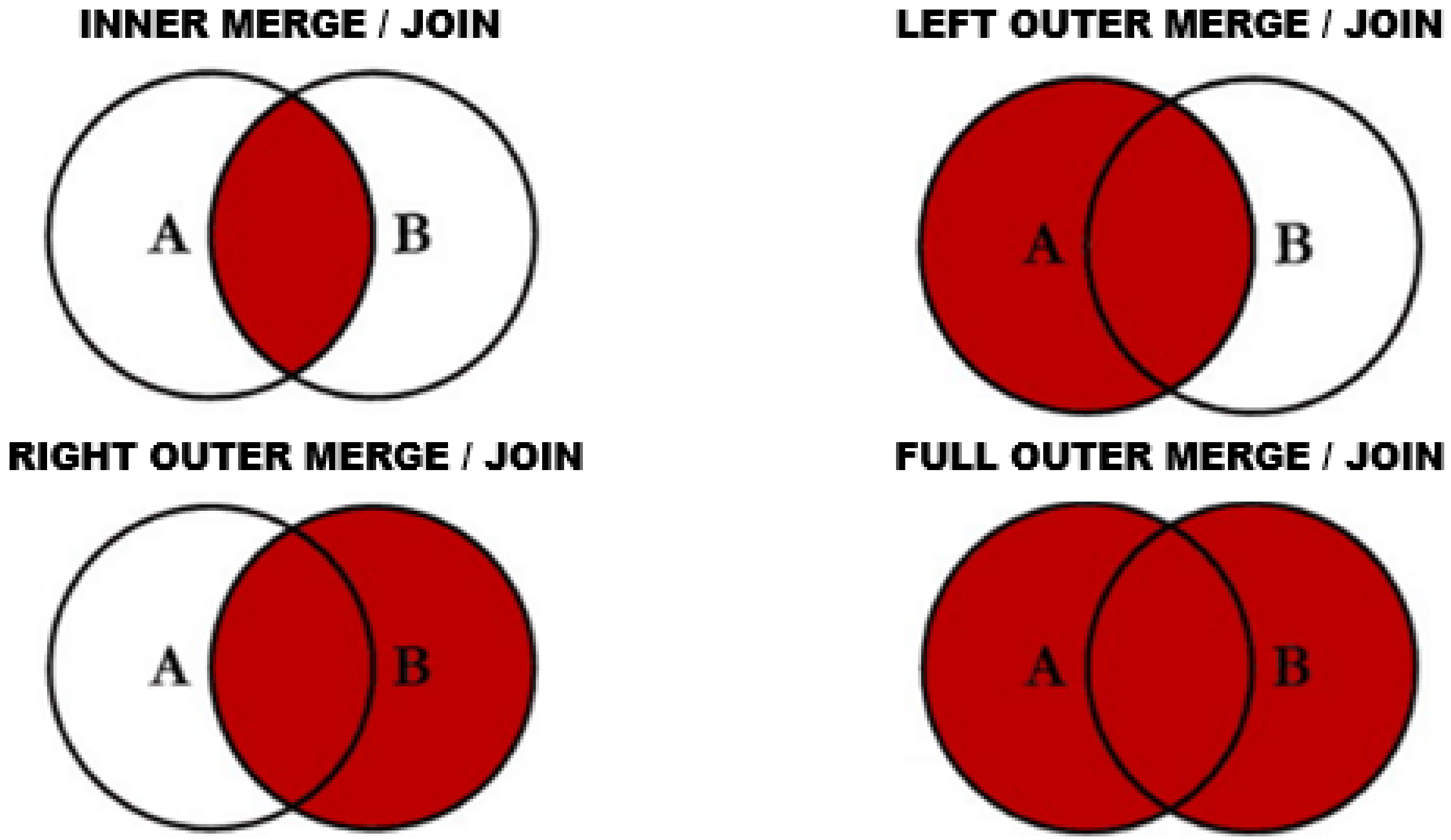


Illustration of Merges / Joins using Venn Diagrams

Match -Merge with the DATA Step

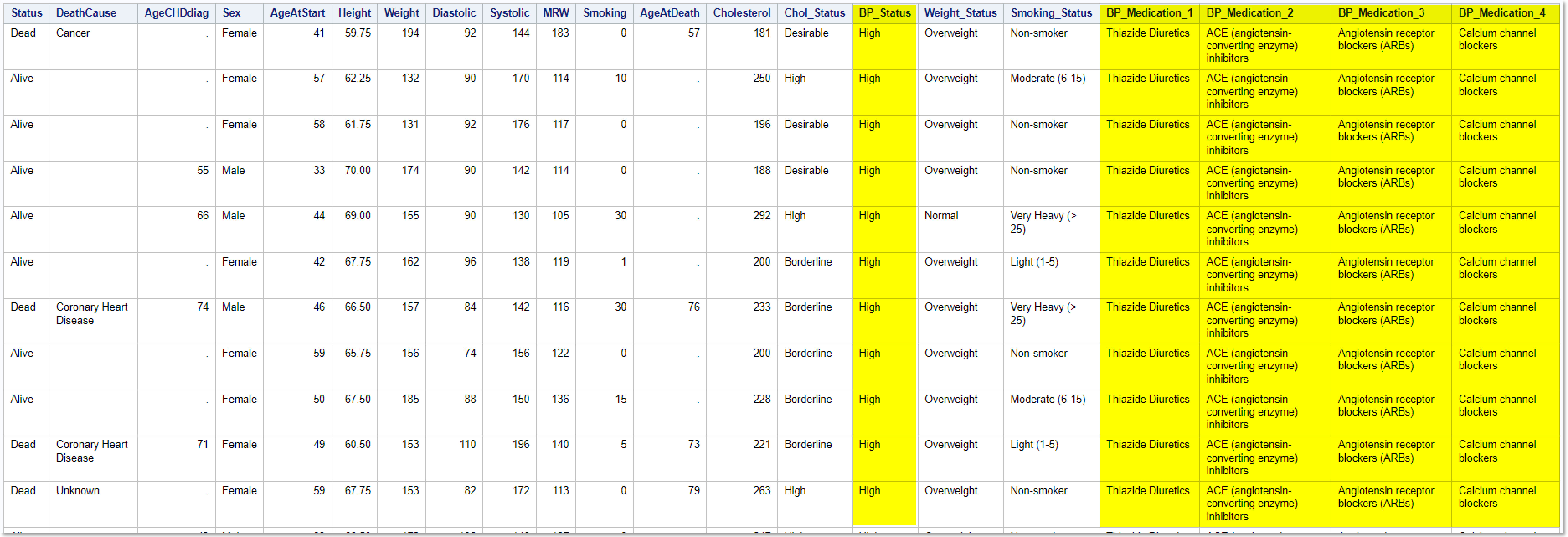
In the next example we illustrate how to combine tables together by constructing a MATCH-MERGE (or Intersect) construct using a DATA step.

```
proc sort data=sashelp.heart
           out=work.heart_sorted;
  by bp_status;
  where bp_status = "High";
run;

data work.heart_hbp_medications_mm;
  merge work.heart_sorted(in=h)
        mydata.high_blood_pressure_medications(in=hbp);
  by bp_status;
  if h and hbp;
run;

proc print data=work.heart_hbp_medications_mm noobs;
run;
```

Match -Merge with the DATA Step



Status	DeathCause	AgeCHDdiag	Sex	AgeAtStart	Height	Weight	Diastolic	Systolic	MRW	Smoking	AgeAtDeath	Cholesterol	Chol_Status	BP_Status	Weight_Status	Smoking_Status	BP_Medication_1	BP_Medication_2	BP_Medication_3	BP_Medication_4
Dead	Cancer	.	Female	41	59.75	194	92	144	183	0	57	181	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	57	62.25	132	90	170	114	10	.	250	High	High	Overweight	Moderate (6-15)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	58	61.75	131	92	176	117	0	.	196	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		55	Male	33	70.00	174	90	142	114	0	.	188	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		66	Male	44	69.00	155	90	130	105	30	.	292	High	High	Normal	Very Heavy (> 25)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	42	67.75	162	96	138	119	1	.	200	Borderline	High	Overweight	Light (1-5)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Coronary Heart Disease	74	Male	46	66.50	157	84	142	116	30	76	233	Borderline	High	Overweight	Very Heavy (> 25)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	59	65.75	156	74	156	122	0	.	200	Borderline	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	50	67.50	185	88	150	136	15	.	228	Borderline	High	Overweight	Moderate (6-15)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Coronary Heart Disease	71	Female	49	60.50	153	110	196	140	5	73	221	Borderline	High	Overweight	Light (1-5)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Unknown	.	Female	59	67.75	153	82	172	113	0	79	263	High	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers

2,267 Observations and 21 Variables

Match -Join with PROC SQL

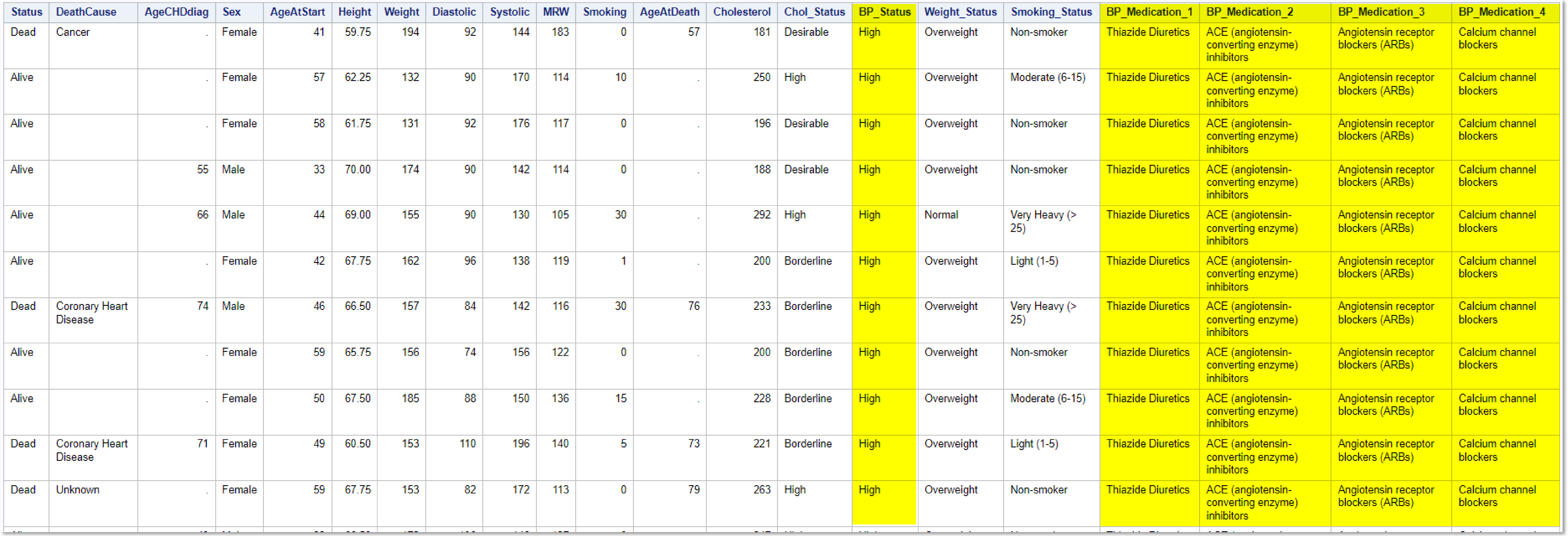
In the next example we illustrate how to combine tables together by constructing a MATCH-JOIN (or Intersect) construct using PROC SQL.

```
libname mydata "/home/username/Data Sources";

proc sql nonumber;
  create table work.heart_hbp_medications_ij as
    select *
      from sashelp.heart h
           inner join
           mydata.high_blood_pressure_medications hbp
        on h.bp_status = hbp.bp_status;

  select *
    from work.heart_hbp_medications_ij;
quit;
```

Match -Join with PROC SQL



Status	DeathCause	AgeCHDdiag	Sex	AgeAtStart	Height	Weight	Diastolic	Systolic	MRW	Smoking	AgeAtDeath	Cholesterol	Chol_Status	BP_Status	Weight_Status	Smoking_Status	BP_Medication_1	BP_Medication_2	BP_Medication_3	BP_Medication_4
Dead	Cancer	.	Female	41	59.75	194	92	144	183	0	57	181	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	57	62.25	132	90	170	114	10	.	250	High	High	Overweight	Moderate (6-15)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	58	61.75	131	92	176	117	0	.	196	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		55	Male	33	70.00	174	90	142	114	0	.	188	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		66	Male	44	69.00	155	90	130	105	30	.	292	High	High	Normal	Very Heavy (> 25)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	42	67.75	162	96	138	119	1	.	200	Borderline	High	Overweight	Light (1-5)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Coronary Heart Disease	74	Male	46	66.50	157	84	142	116	30	76	233	Borderline	High	Overweight	Very Heavy (> 25)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	59	65.75	156	74	156	122	0	.	200	Borderline	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	50	67.50	185	88	150	136	15	.	228	Borderline	High	Overweight	Moderate (6-15)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Coronary Heart Disease	71	Female	49	60.50	153	110	196	140	5	73	221	Borderline	High	Overweight	Light (1-5)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Unknown	.	Female	59	67.75	153	82	172	113	0	79	263	High	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers

2,267 Observations and 21 Variables

Left Outer Merge with the DATA Step

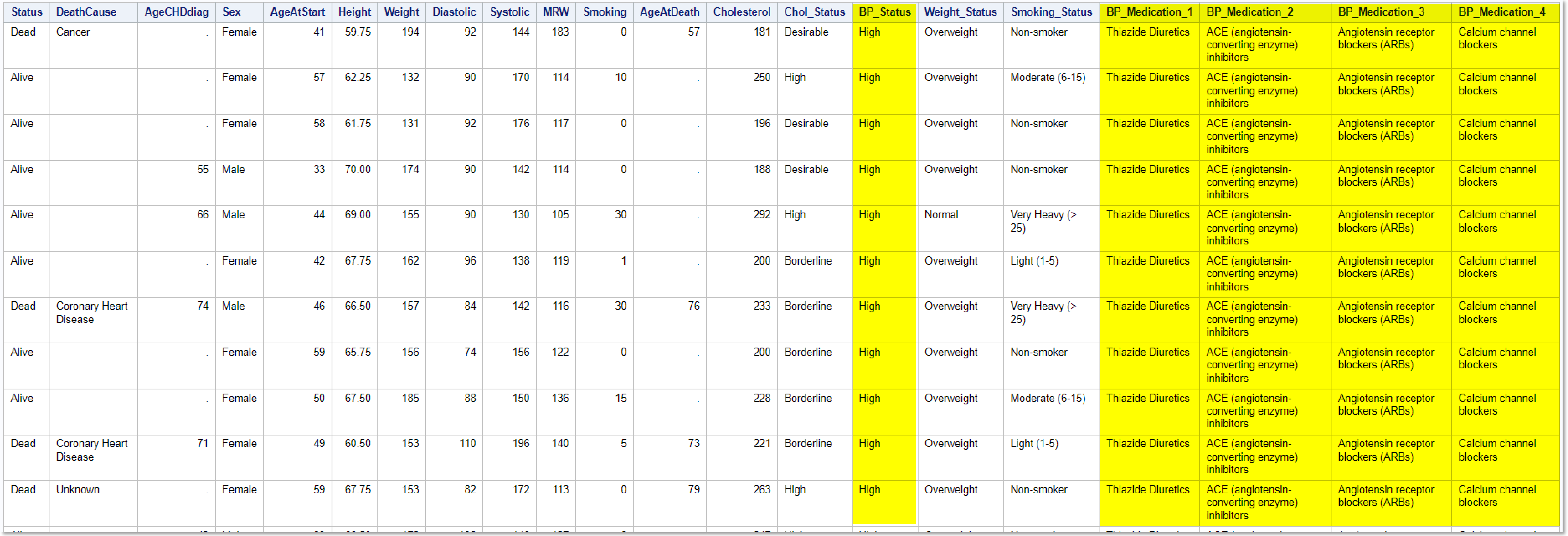
In the next example we illustrate the process of combining two tables together using a LeftOuter Merge in a DATA step.

```
proc sort data=sashelp.heart
           out=work.heart_sorted;
  by bp_status;
run;

data work.heart_hbp_medications_lom;
  merge work.heart_sorted(in=h)
        mydata.high_blood_pressure_medications(in=hbp);
  by bp_status;
  if h;
run;

proc print data=work.heart_hbp_medications_lom noobs;
run;
```

Left Outer Merge with the DATA Step



Status	DeathCause	AgeCHDdiag	Sex	AgeAtStart	Height	Weight	Diastolic	Systolic	MRW	Smoking	AgeAtDeath	Cholesterol	Chol_Status	BP_Status	Weight_Status	Smoking_Status	BP_Medication_1	BP_Medication_2	BP_Medication_3	BP_Medication_4
Dead	Cancer	.	Female	41	59.75	194	92	144	183	0	57	181	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	57	62.25	132	90	170	114	10	.	250	High	High	Overweight	Moderate (6-15)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	58	61.75	131	92	176	117	0	.	196	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		55	Male	33	70.00	174	90	142	114	0	.	188	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		66	Male	44	69.00	155	90	130	105	30	.	292	High	High	Normal	Very Heavy (> 25)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	42	67.75	162	96	138	119	1	.	200	Borderline	High	Overweight	Light (1-5)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Coronary Heart Disease	74	Male	46	66.50	157	84	142	116	30	76	233	Borderline	High	Overweight	Very Heavy (> 25)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	59	65.75	156	74	156	122	0	.	200	Borderline	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	50	67.50	185	88	150	136	15	.	228	Borderline	High	Overweight	Moderate (6-15)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Coronary Heart Disease	71	Female	49	60.50	153	110	196	140	5	73	221	Borderline	High	Overweight	Light (1-5)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Unknown	.	Female	59	67.75	153	82	172	113	0	79	263	High	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers

5,209 Observations and 21 Variables

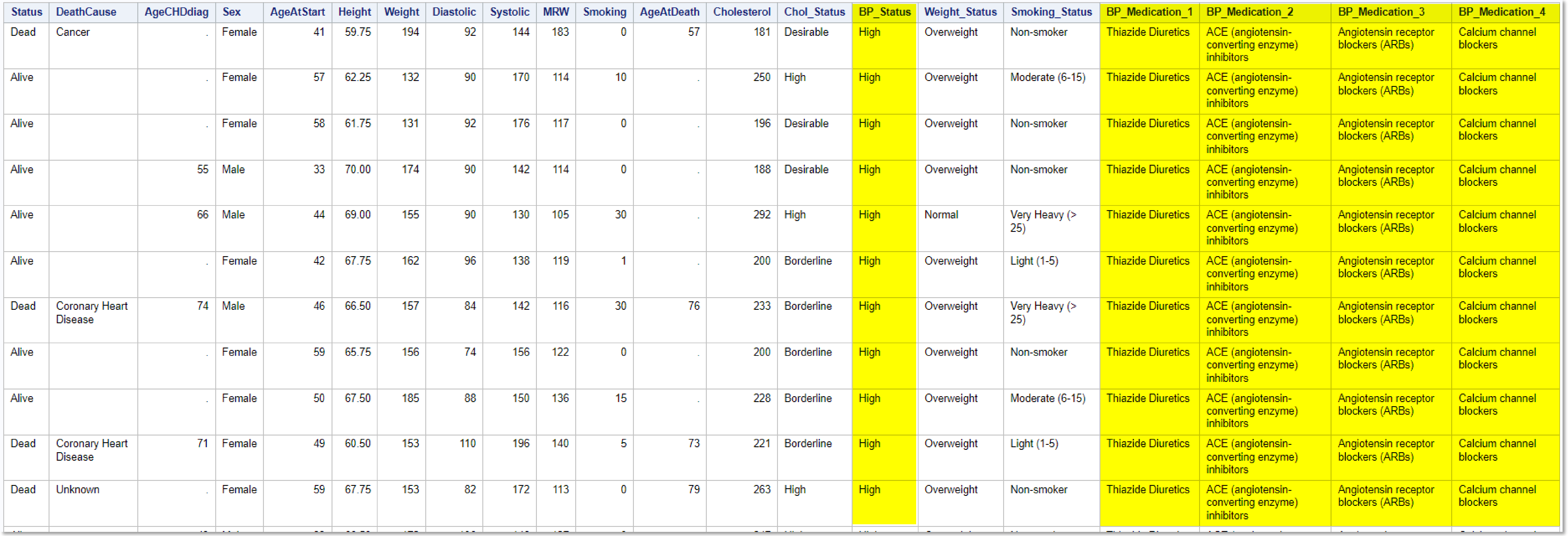
Left Outer Join with PROC SQL

In the next example we illustrate the process of combining two tables together using a LeftOuter Join in PROC SQL.

```
proc sql nonumber;
  create table work.heart_hbp_medications_lj as
    select *
      from sashelp.heart h
           left join
           mydata.high_blood_pressure_medications hbp
        on h.bp_status = hbp.bp_status;

  select *
    from work.heart_hbp_medications_lj;
quit;
```

Left Outer Merge with PROC SQL



Status	DeathCause	AgeCHDdiag	Sex	AgeAtStart	Height	Weight	Diastolic	Systolic	MRW	Smoking	AgeAtDeath	Cholesterol	Chol_Status	BP_Status	Weight_Status	Smoking_Status	BP_Medication_1	BP_Medication_2	BP_Medication_3	BP_Medication_4
Dead	Cancer	.	Female	41	59.75	194	92	144	183	0	57	181	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	57	62.25	132	90	170	114	10	.	250	High	High	Overweight	Moderate (6-15)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	58	61.75	131	92	176	117	0	.	196	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		55	Male	33	70.00	174	90	142	114	0	.	188	Desirable	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		66	Male	44	69.00	155	90	130	105	30	.	292	High	High	Normal	Very Heavy (> 25)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	42	67.75	162	96	138	119	1	.	200	Borderline	High	Overweight	Light (1-5)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Coronary Heart Disease	74	Male	46	66.50	157	84	142	116	30	76	233	Borderline	High	Overweight	Very Heavy (> 25)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	59	65.75	156	74	156	122	0	.	200	Borderline	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Alive		.	Female	50	67.50	185	88	150	136	15	.	228	Borderline	High	Overweight	Moderate (6-15)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Coronary Heart Disease	71	Female	49	60.50	153	110	196	140	5	73	221	Borderline	High	Overweight	Light (1-5)	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers
Dead	Unknown	.	Female	59	67.75	153	82	172	113	0	79	263	High	High	Overweight	Non-smoker	Thiazide Diuretics	ACE (angiotensin-converting enzyme) inhibitors	Angiotensin receptor blockers (ARBs)	Calcium channel blockers

5,209 Observations and 21 Variables

Merge and Joins – Decision Point

While the data step insists that only same named columns can be merged together, SQL has the ability to offer flexibility in joins, by permitting differently named columns to be joined together. In addition, while the datastep can only match on equality, SQL offers an additional flexibility of supporting joins on unequal values as well. However, the data step steps right up with its powerful ability to track row level data with the IN operator. So once again these giants stand evenly matched.

Conclusion

So, “Which colossus approach is better?”, you might ask. Well, our task was not to declare a winner or a loser, but rather to show that the DATA step and PROC SQL could both be used to meet your data processing needs. This paper is intended to illustrate the differences, and the similarities, between the DATA step and PROC SQL. What we have attempted to do is give you an insightful way, and hopefully ignite a curiosity for further discovery about these two SAS “TITANS” (To be clear, we’re talking about the DATA step and SQL procedure here, not the authors.) It is also worth mentioning that each technique has a definite set of strengths over the other depending on the desired result. The tools are available, and they serve as powerful foundations to valuable and productive processing.

Handy Links

- Lafler, Kirk Paul, Joshua Horstman, Ben Cochran, Ray Pass, and Dan Bruns (2023). "Battle of the Titans (Part II): PROC REPORT versus PROC TABULATE," 2023 PharmaSUG Conference.
- Lafler, Kirk Paul (2019). PROC SQL: Beyond the Basics Using SAS, Third Edition, SAS Institute Inc.,
- Lafler, Kirk Paul, Joshua Horstman, Ben Cochran, Ray Pass, and Dan Bruns (2022). "Battle of the Titans (Part II): PROC REPORT versus PROC TABULATE," 2022 SouthEast SAS Users Group (SESUG) Conference.
- SAS 9.4 PROC SQL user's guide
- Video - Step-by-step PROC SQL
- Shanksar, Charu. Go home on time with 5 PROC SQL tips
- Shankar, Charu. Ask The Expert Webinar – Top 5 Handy PROC SQL Tips
- Shankar, Charu Know thy data: Dictionary tables SAS Global Forum Paper
- Shankar, Charu. SAS YouTube Video - Mastering the WHERE clause in PROG SQL
- Shankar, Charu. SAS YouTube Video - Power of SAS SQL –SAS Global Forum 2021
- Shankar, Charu. SAS YouTube Video - Step by step PROC SQL – SAS Global forum 2020
- Shankar, Charu. "Ask the Expert Webinar - Why choose between SAS data Step & PROC SQL When You Can Have Both

Thank You

Charu Shankar
SAS Institute Toronto

EMAIL	Charu.shankar@sas.com
BLOG	https://blogs.sas.com/content/author/charushankar/
TWITTER	CharuYogaCan
LINKEDIN	https://www.linkedin.com/in/charushankar/

✓ Did you enjoy this session, Let us know in the **evaluation**

sas innovate 2025