CLASS - 18

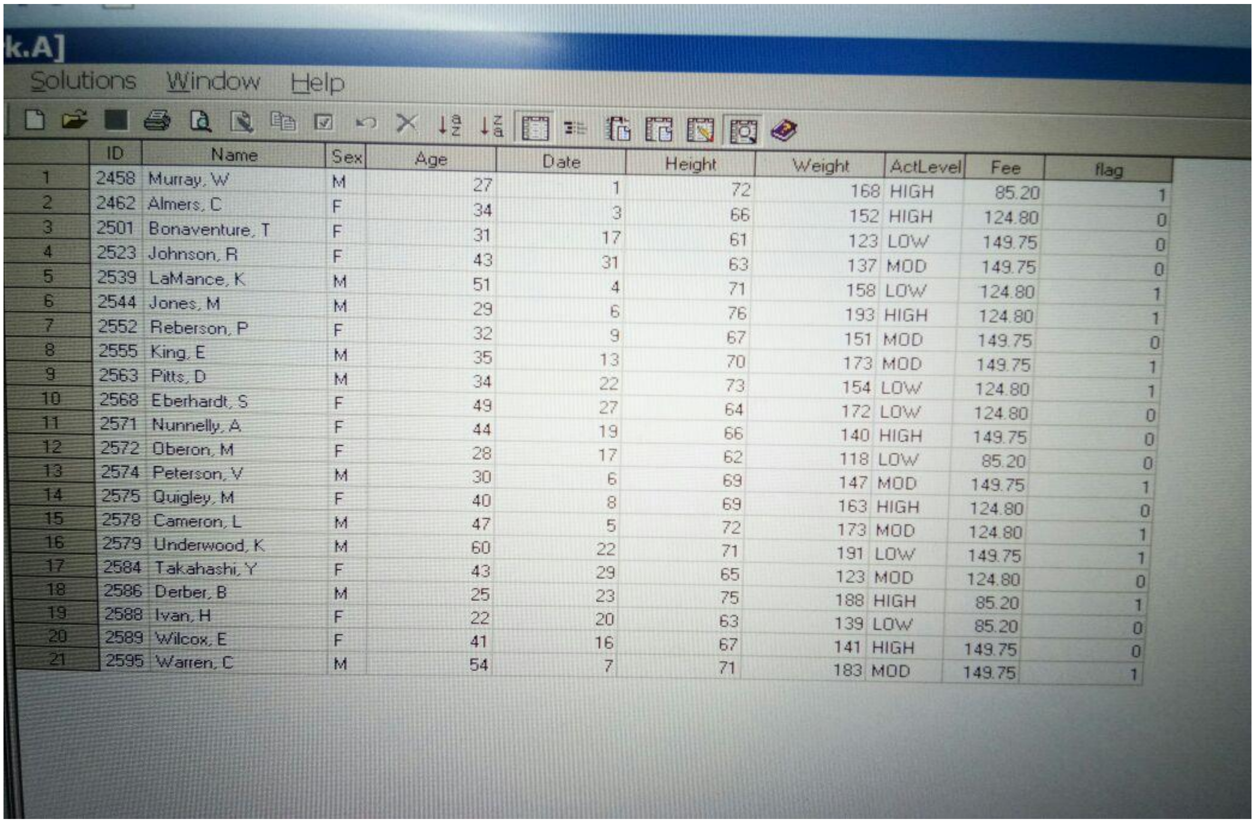
Base SAS

```
Data a;
Set sasuser.admit;
If sex=”M” then flag=1;
Else flag=0;
Run;
```

SQL

```
proc sql;
create table a as select*,case
when sex=”M” then 1
else 0 end as flag from SA;
 quit;
```

Note: In SQL we have case statements(**also called switch statements**). As we know that there is no “ if ” in SQL, so we use “when” in place of “if”.
‘If then’ in base SAS is replaced by ‘when then’ in SQL.
Flag is the new variable created and will take the value as ‘1’ wherever sex will b “M” else will print 0.
As we have started the case statement so it will be ended by *‘end as.’*



	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee	flag
1	2458	Murray, W	M	27	1	72	168	HIGH	85.20	1
2	2462	Almers, C	F	34	3	66	152	HIGH	124.80	0
3	2501	Bonaventure, T	F	31	17	61	123	LOW	149.75	0
4	2523	Johnson, R	F	43	31	63	137	MOD	149.75	0
5	2539	LaMance, K	M	51	4	71	158	LOW	124.80	1
6	2544	Jones, M	M	29	6	76	193	HIGH	124.80	1
7	2552	Reberson, P	F	32	9	67	151	MOD	149.75	0
8	2555	King, E	M	35	13	70	173	MOD	149.75	1
9	2563	Pitts, D	M	34	22	73	154	LOW	124.80	1
10	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80	0
11	2571	Nunnelly, A	F	44	19	66	140	HIGH	149.75	0
12	2572	Oberon, M	F	28	17	62	118	LOW	85.20	0
13	2574	Peterson, V	M	30	6	69	147	MOD	149.75	1
14	2575	Quigley, M	F	40	8	69	163	HIGH	124.80	0
15	2578	Cameron, L	M	47	5	72	173	MOD	124.80	1
16	2579	Underwood, K	M	60	22	71	191	LOW	149.75	1
17	2584	Takahashi, Y	F	43	29	65	123	MOD	124.80	0
18	2586	Derber, B	M	25	23	75	188	HIGH	85.20	1
19	2588	Ivan, H	F	22	20	63	139	LOW	85.20	0
20	2589	Wilcox, E	F	41	16	67	141	HIGH	149.75	0
21	2595	Warren, C	M	54	7	71	183	MOD	149.75	1

Another example

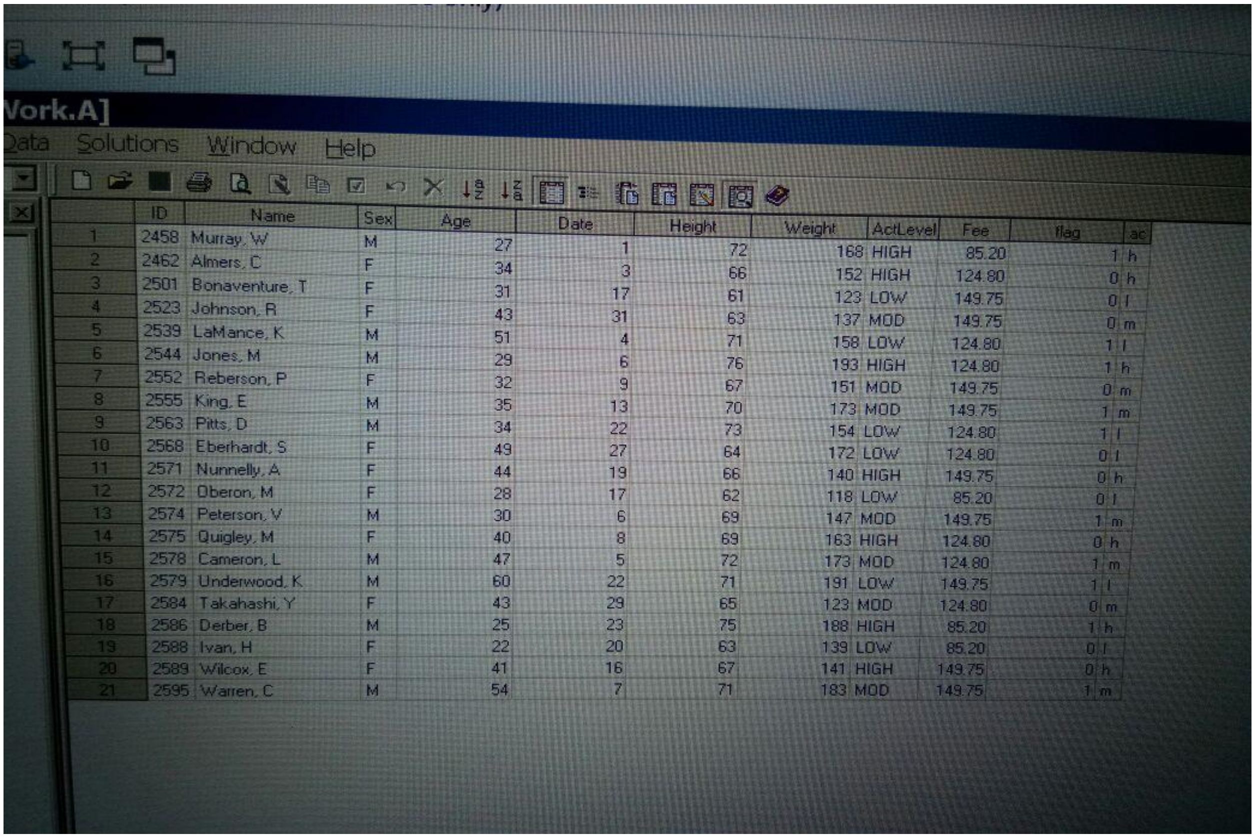
Base SAS

Data a;
Set sasuser.admit;
If sex="M" then flag=1;
Else flag=0;
If actlevel="HIGH" then ac="h";
Else if actlevel="MOD" then ac="m";
Else ac="1";
Run;

SQL

proc sql;
create table a as select * ,case
when sex="M" then 1

```
else 0 end as flag, case
when actlevel=”HIGH” then “h”;
when actlevel=”MOD” then “m”;
else “l” end as ac from SA;
quit;
```



	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee	flag	ac
1	2458	Murray, W	M	27	1	72	168	HIGH	85.20	1	h
2	2462	Almers, C	F	34	3	66	152	HIGH	124.80	0	h
3	2501	Bonaventure, T	F	31	17	61	123	LOW	149.75	0	l
4	2523	Johnson, R	F	43	31	63	137	MOD	149.75	0	m
5	2539	LaMance, K	M	51	4	71	158	LOW	124.80	1	l
6	2544	Jones, M	M	29	6	76	193	HIGH	124.80	1	h
7	2552	Reberson, P	F	32	9	67	151	MOD	149.75	0	m
8	2555	King, E	M	35	13	70	173	MOD	149.75	1	m
9	2563	Pitts, D	M	34	22	73	154	LOW	124.80	1	l
10	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80	0	l
11	2571	Nunnelly, A	F	44	19	66	140	HIGH	149.75	0	h
12	2572	Oberon, M	F	28	17	62	118	LOW	85.20	0	l
13	2574	Peterson, V	M	30	6	69	147	MOD	149.75	1	m
14	2575	Quigley, M	F	40	8	69	163	HIGH	124.80	0	h
15	2578	Cameron, L	M	47	5	72	173	MOD	124.80	1	m
16	2579	Underwood, K	M	60	22	71	191	LOW	149.75	1	l
17	2584	Takahashi, Y	F	43	29	65	123	MOD	124.80	0	m
18	2586	Derber, B	M	25	23	75	188	HIGH	85.20	1	h
19	2588	Ivan, H	F	22	20	63	139	LOW	85.20	0	l
20	2589	Wilcox, E	F	41	16	67	141	HIGH	149.75	0	h
21	2595	Warren, C	M	54	7	71	183	MOD	149.75	1	m

Important : In base SAS, if the new variable is created with the already existing variable name, then it overwrites the previous one.

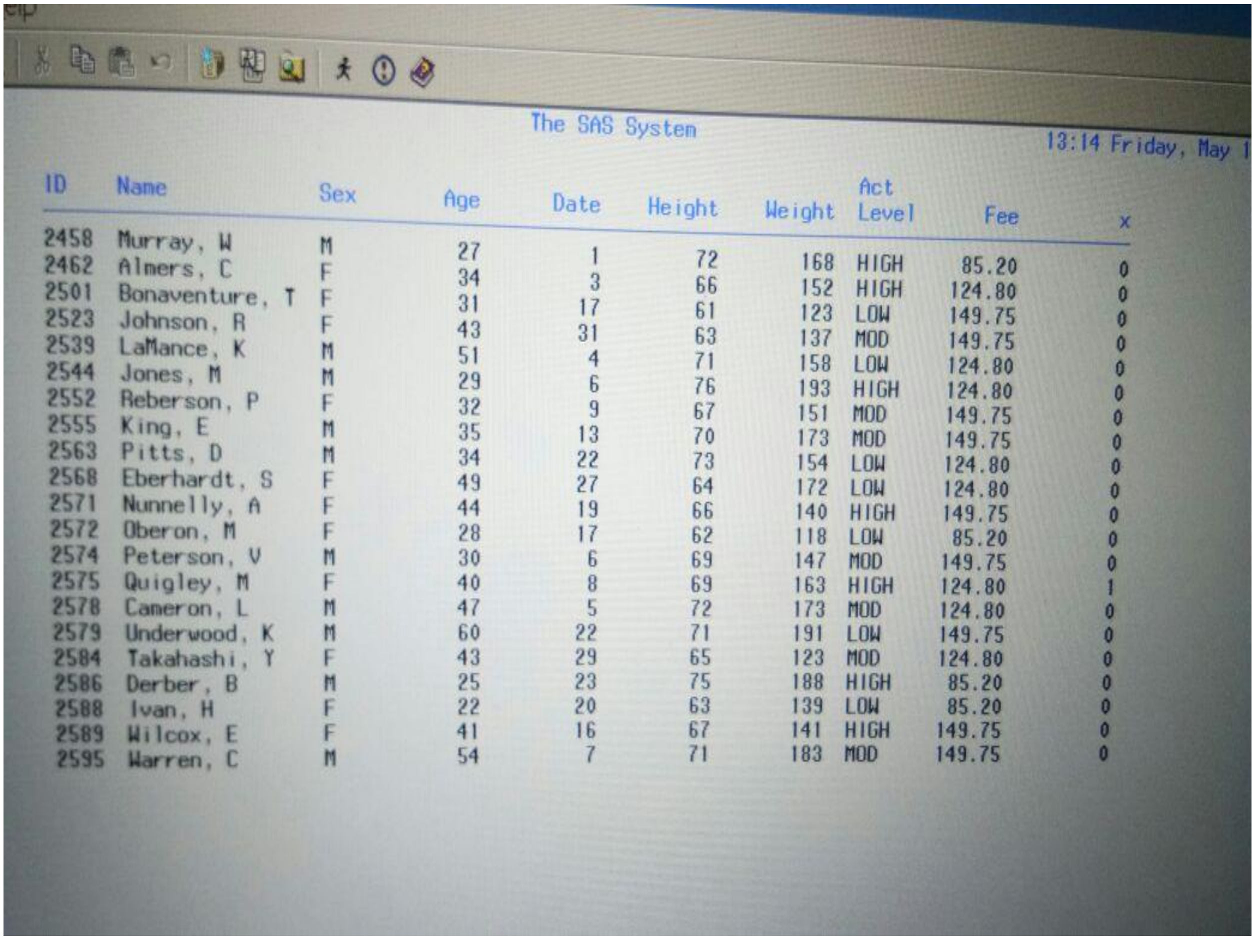
But in SQL the new variable created should have a different name as it does not overwrite the already existing variable and will show error.

Boolean expression

```
Data a;
Set sasuser.admit;
```

x=age=40;
Run;

Proc sql;
Create table a as select * , age=40 as x from Sasuser.admit;
Quit;

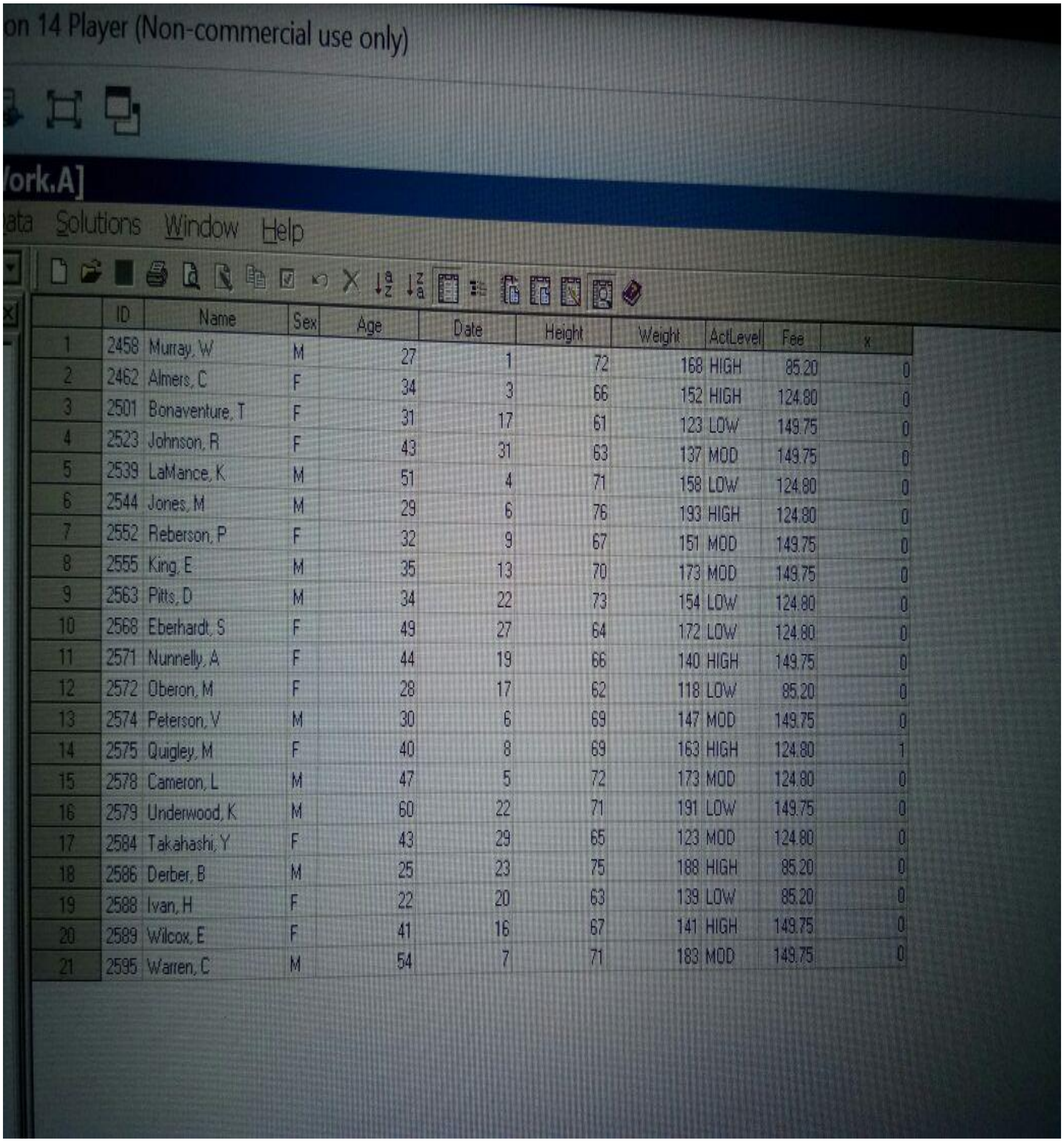


```
                              The SAS System
                                                              13:14 Friday, May 1

ID    Name              Sex    Age    Date   Height   Weight  Act      Fee      x
                                                              Level
2458  Murray, W         M       27      1      72      168    HIGH     85.20    0
2462  Almers, C         F       34      3      66      152    HIGH    124.80    0
2501  Bonaventure, T    F       31     17      61      123    LOW     149.75    0
2523  Johnson, R        F       43     31      63      137    MOD     149.75    0
2539  LaMance, K        M       51      4      71      158    LOW     124.80    0
2544  Jones, M          M       29      6      76      193    HIGH    124.80    0
2552  Reberson, P       F       32      9      67      151    MOD     149.75    0
2555  King, E           M       35     13      70      173    MOD     149.75    0
2563  Pitts, D          M       34     22      73      154    LOW     124.80    0
2568  Eberhardt, S      F       49     27      64      172    LOW     124.80    0
2571  Nunnelly, A       F       44     19      66      140    HIGH    149.75    0
2572  Oberon, M         F       28     17      62      118    LOW      85.20    0
2574  Peterson, V       M       30      6      69      147    MOD     149.75    0
2575  Quigley, M        F       40      8      69      163    HIGH    124.80    1
2578  Cameron, L        M       47      5      72      173    MOD     124.80    0
2579  Underwood, K      M       60     22      71      191    LOW     149.75    0
2584  Takahashi, Y      F       43     29      65      123    MOD     124.80    0
2586  Derber, B         M       25     23      75      188    HIGH     85.20    0
2588  Ivan, H           F       22     20      63      139    LOW      85.20    0
2589  Wilcox, E         F       41     16      67      141    HIGH    149.75    0
2595  Warren, C         M       54      7      71      183    MOD     149.75    0
```

age=40 as x : creates a boolean variable x and where the value of age will be, x will take the value as 1 else 0.

Proc sql;
<u>Select *</u> , age =40 as x from Sasuser.admit;
Quit;

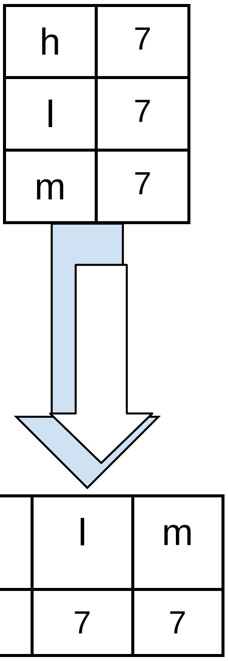
Note: In the second example by using <u>Create table a</u> , table is created and in this example using only <u>Select *</u> will print the data.



	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee	x
1	2458	Murray, W	M	27	1	72	168	HIGH	85.20	0
2	2462	Almers, C	F	34	3	66	152	HIGH	124.80	0
3	2501	Bonaventure, T	F	31	17	61	123	LOW	149.75	0
4	2523	Johnson, R	F	43	31	63	137	MOD	149.75	0
5	2539	LaMance, K	M	51	4	71	158	LOW	124.80	0
6	2544	Jones, M	M	29	6	76	193	HIGH	124.80	0
7	2552	Reberson, P	F	32	9	67	151	MOD	149.75	0
8	2555	King, E	M	35	13	70	173	MOD	149.75	0
9	2563	Pitts, D	M	34	22	73	154	LOW	124.80	0
10	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80	0
11	2571	Nunnelly, A	F	44	19	66	140	HIGH	149.75	0
12	2572	Oberon, M	F	28	17	62	118	LOW	85.20	0
13	2574	Peterson, V	M	30	6	69	147	MOD	149.75	0
14	2575	Quigley, M	F	40	8	69	163	HIGH	124.80	1
15	2578	Cameron, L	M	47	5	72	173	MOD	124.80	0
16	2579	Underwood, K	M	60	22	71	191	LOW	149.75	0
17	2584	Takahashi, Y	F	43	29	65	123	MOD	124.80	0
18	2586	Derber, B	M	25	23	75	188	HIGH	85.20	0
19	2588	Ivan, H	F	22	20	63	139	LOW	85.20	0
20	2589	Wilcox, E	F	41	16	67	141	HIGH	149.75	0
21	2595	Warren, C	M	54	7	71	183	MOD	149.75	0

HOW TO TRANSPOSE DATA IN SQL (sum case)

Output Generated



h	7
l	7
m	7

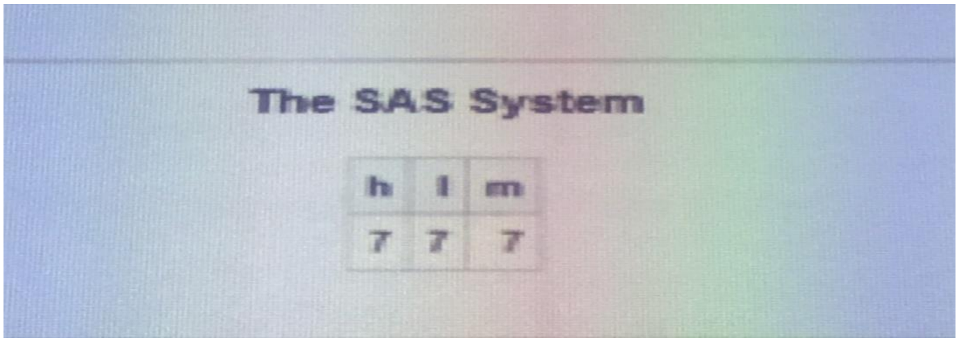
h	l	m
7	7	7

Required Output

Suppose we need to transpose our data (see from the above figure). In this situation we will use “sum case “ Here sum is behaving like count.

Code:

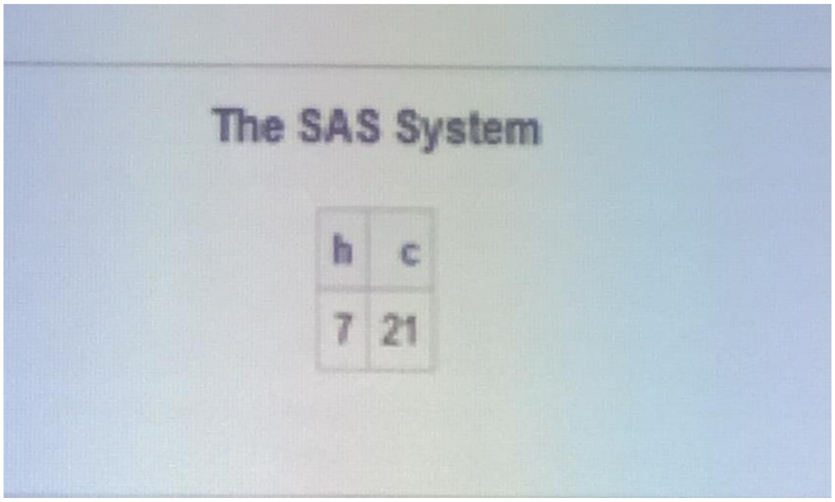
```
Proc sql;
Select  sum  (actlevel=”HIGH”) as h, sum (actlevel=”LOW”) as l , sum(actlevel=”MOD”) as m
from sasuser.admit;
Quit;
```



The SAS System

h	l	m
7	7	7

```
Proc sql;
Select sum(actlevel=”HIGH”) as h, count (actlevel=”HIGH”) as c from sasuser.admit;
Quit;
```

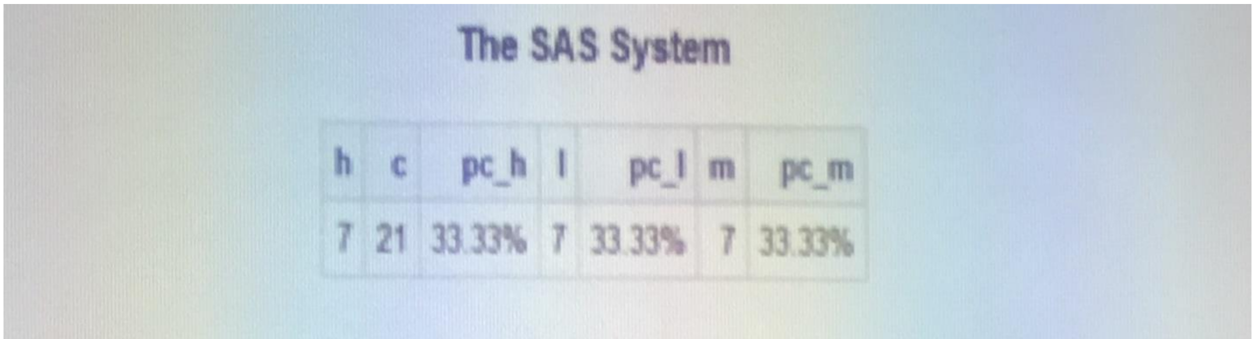


The SAS System

h	c
7	21

Note: “ **count “** will always give the number of rows whether there is any value in rows or not.

```
Proc sql;
Select sum(actlevel=”HIGH”) as h, count (actlevel=”HIGH”) as c, calculated  h/calculated c as
pc_h format percent 9.2;

Select sum(actlevel=”LOW”) as h, count (actlevel=”LOW”) as c, calculated  l/calculated c as pc_l
format percent  9.2;

Select sum(actlevel=”MOD”) as h, count (actlevel=”MOD”) as c, calculated  m/calculated c as
pc_m format percent  9.2;
from sasuser.admit;
quit;
```

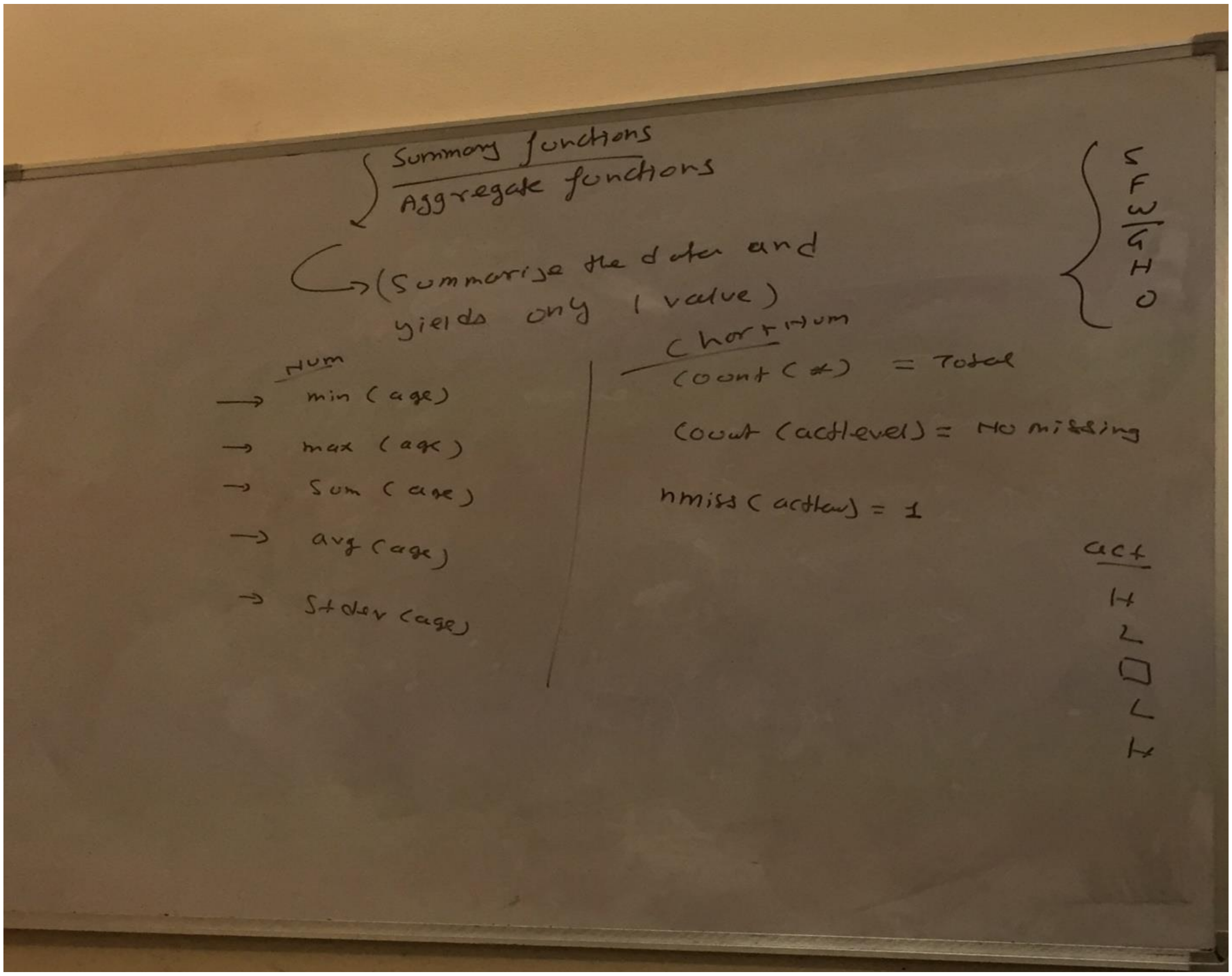


The SAS System

h	c	pc_h	l	pc_l	m	pc_m
7	21	33.33%	7	33.33%	7	33.33%

SUMMARY FUNCTIONS / AGGREGATE FUNCTIONS

Summary functions are used to summarize the data and it yields only **one** output.



Summary functions
Aggregate functions

(Summarise the data and yields ony 1 value)

Num
- min (age)
- max (age)
- sum (age)
- avg (age)
- Stdev (age)

Char+Num
count (*) = Total
count (actlevel) = No missing
nmiss (actlev) = 1

S
F
W
G
H
O

act
H
L
□
L
H

Count (*) : will always give total number of rows whether there is any value in row or not.

Count (any variable) : It will give non-missing values.
Eg: Suppose in count(actlevel) the values are H,L,(null value),H,L
In this count(*) will be 5 and count(actlevel) will be 4.

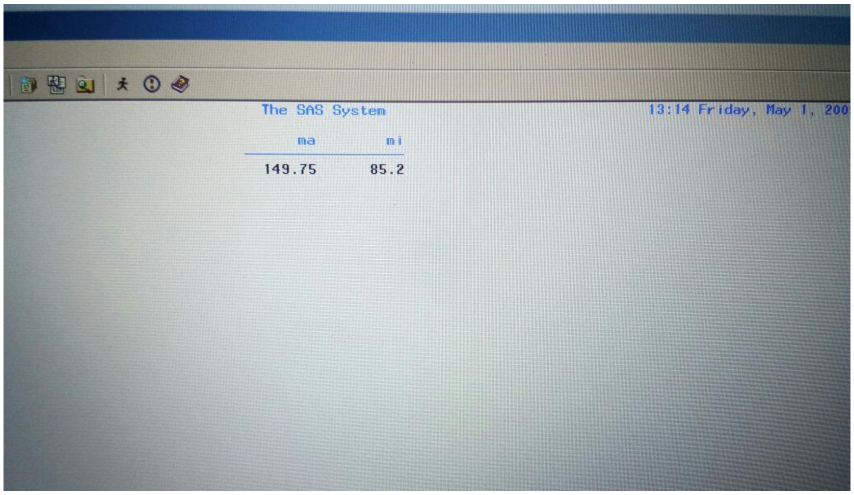
Note: count(*) = Count (any variable**) ,** when there will be no null value in the variable**.**

Code:

```
Proc sql;
select sum(fee) as f from sasuser.admit;
```

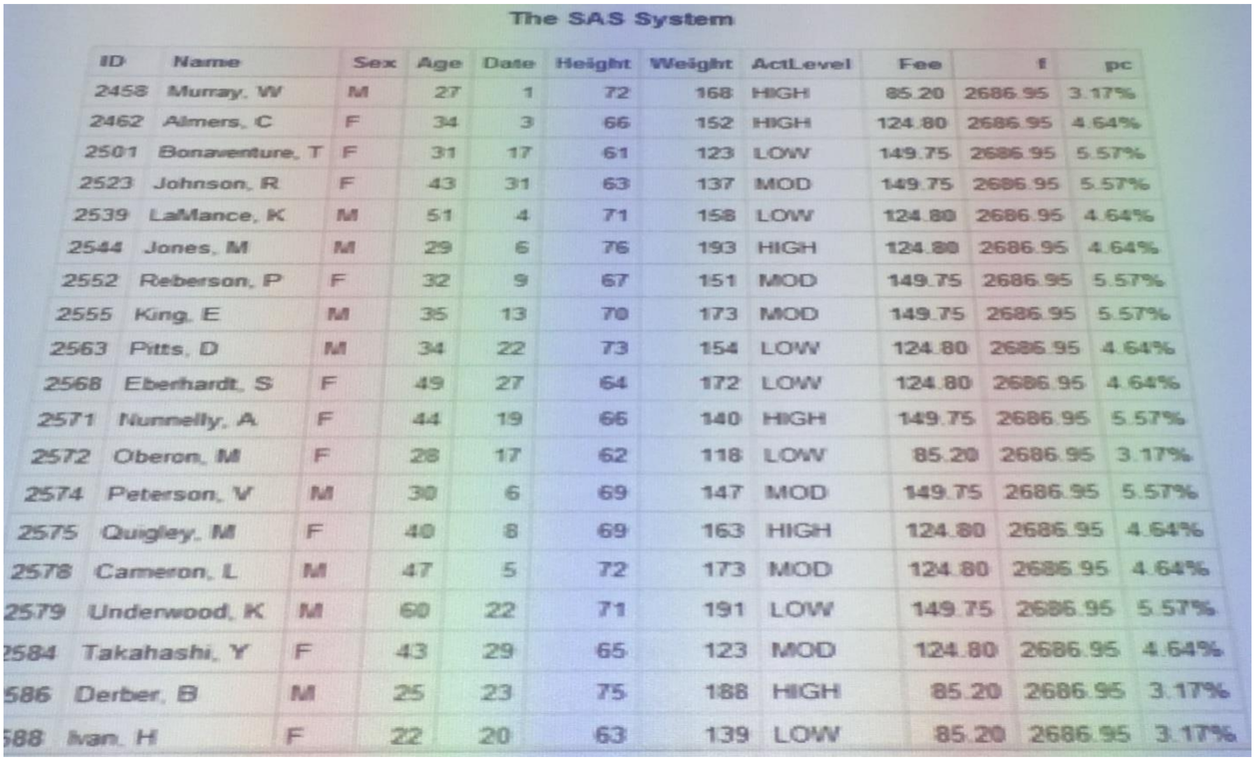
```
Quit;
```

```
Proc sql;
select max(fee) as ma, min (fee) as mi from sasuser.admit;
Quit;
```



```
                         The SAS System                            13:14 Friday, May 1, 200

                              ma          mi
                        ----------------------
                          149.75        85.2
```

```
Proc sql;
Select * , sum(fee) as f, fee/calculated f as pc format percent 9.2 from sasuser.admit;
Quit;
```



The SAS System

ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee	f	pc
2458	Murray, W	M	27	1	72	168	HIGH	85.20	2686.95	3.17%
2462	Almers, C	F	34	3	66	152	HIGH	124.80	2686.95	4.64%
2501	Bonaventure, T	F	31	17	61	123	LOW	149.75	2686.95	5.57%
2523	Johnson, R	F	43	31	63	137	MOD	149.75	2686.95	5.57%
2539	LaMance, K	M	51	4	71	158	LOW	124.80	2686.95	4.64%
2544	Jones, M	M	29	6	76	193	HIGH	124.80	2686.95	4.64%
2552	Reberson, P	F	32	9	67	151	MOD	149.75	2686.95	5.57%
2555	King, E	M	35	13	70	173	MOD	149.75	2686.95	5.57%
2563	Pitts, D	M	34	22	73	154	LOW	124.80	2686.95	4.64%
2568	Eberhardt, S	F	49	27	64	172	LOW	124.80	2686.95	4.64%
2571	Nunnelly, A	F	44	19	66	140	HIGH	149.75	2686.95	5.57%
2572	Oberon, M	F	28	17	62	118	LOW	85.20	2686.95	3.17%
2574	Peterson, V	M	30	6	69	147	MOD	149.75	2686.95	5.57%
2575	Quigley, M	F	40	8	69	163	HIGH	124.80	2686.95	4.64%
2578	Cameron, L	M	47	5	72	173	MOD	124.80	2686.95	4.64%
2579	Underwood, K	M	60	22	71	191	LOW	149.75	2686.95	5.57%
2584	Takahashi, Y	F	43	29	65	123	MOD	124.80	2686.95	4.64%
586	Derber, B	M	25	23	75	188	HIGH	85.20	2686.95	3.17%
588	Ivan, H	F	22	20	63	139	LOW	85.20	2686.95	3.17%

Note: using select* will print the whole data . This is also called **remerging of statistics** as the value of fee sum is getting remerged with every row.
Using format percent 9.2 will calculate the percentage.

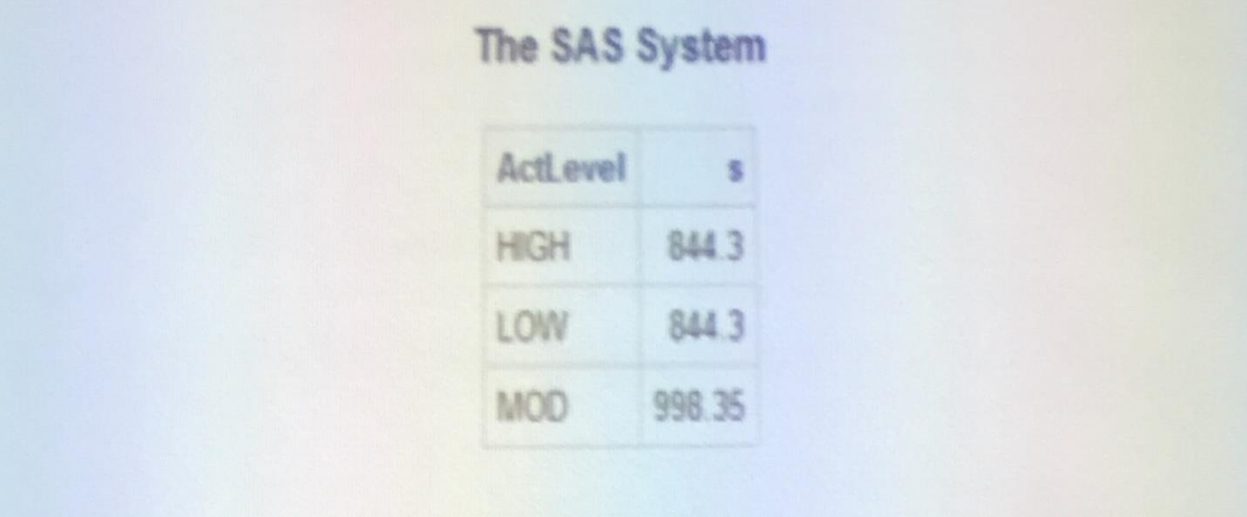
Group By: *classifies the data into groups based on the specified columns*

Suppose we have a class of students and we want to make groups on the basis of gender. So, two groups will be formed i.e. male and female and in that also we want the count of males and females. So for this we use **group by**.

So, basically **group by** is used to calculate summary function of each distinct group.

Code:

```
Proc  sql;
Select  actlevel ,  sum(fee)   as  s  from  sasuser.admit  group  by  actlevel;
Quit;
```



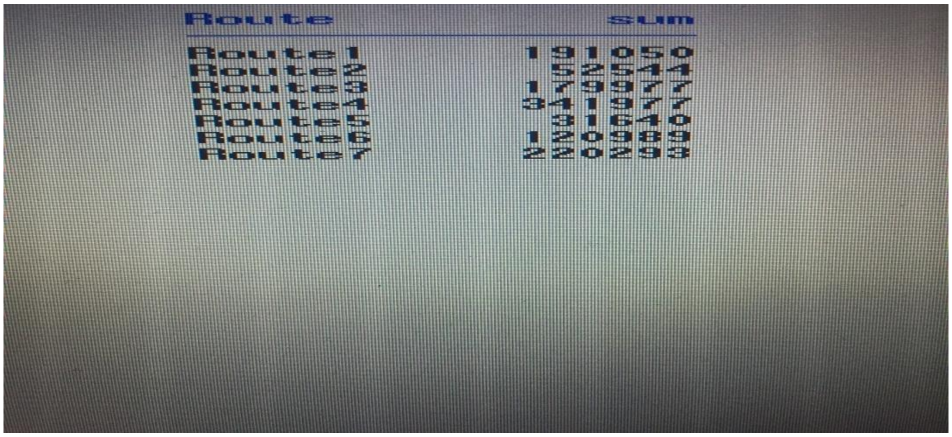
The SAS System

ActLevel	s
HIGH	844.3
LOW	844.3
MOD	998.35

Another example:

```
Proc  sql;
Select  route ,  sum(revcargo)   as  sum from  sasuser.cargorev  group  by  route;
Quit;
```

Note: **group by** route - will create 6 groups (route1 - route 6)and sum of each group will be printed.



```
Route                      sum
Route1                  191050
Route2                   52544
Route3                  179977
Route4                  341977
Route5                   31640
Route6                  120989
Route7                  220293
```

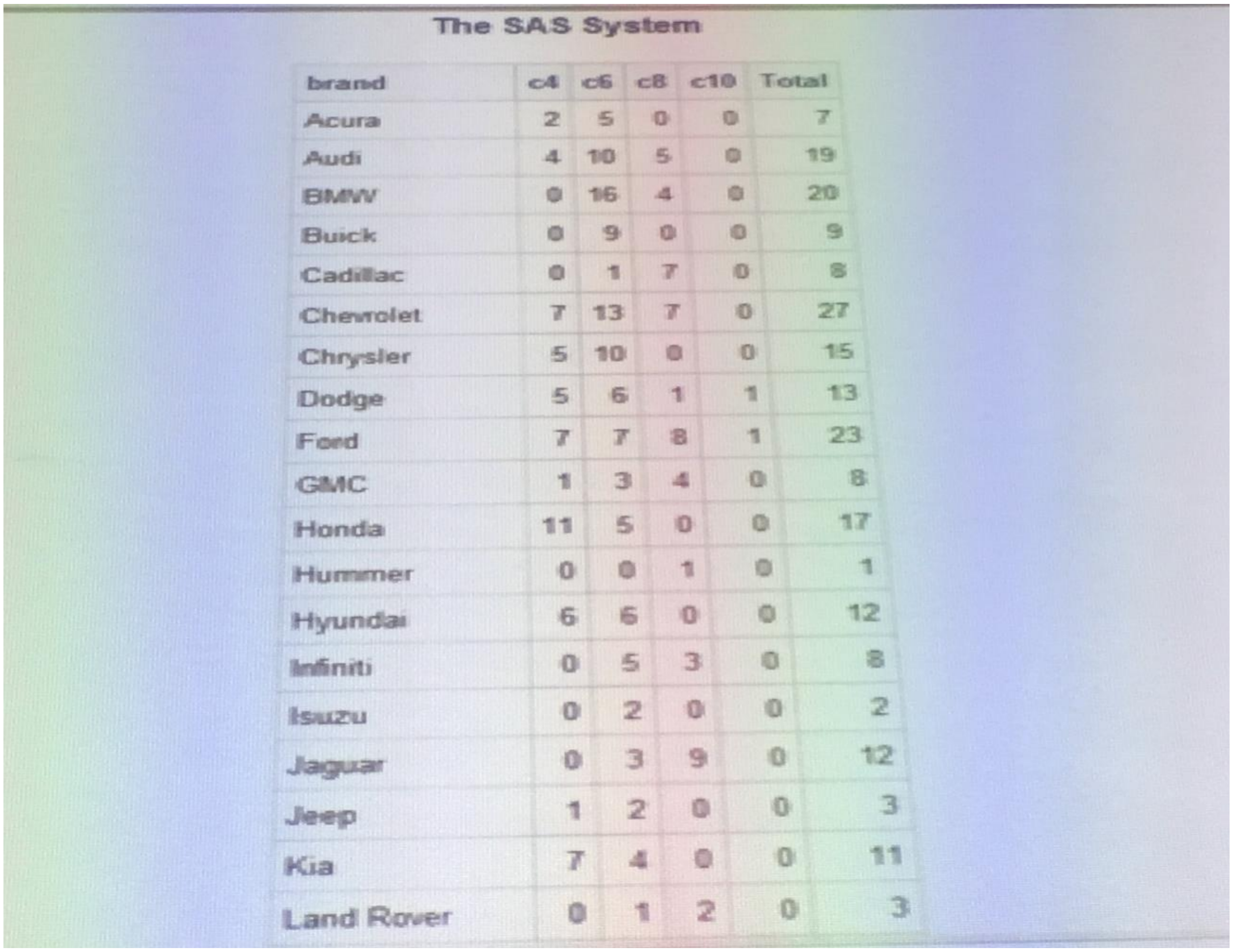
*****************;

Proc sql;

Select distinct make as brand, sum(cylinders=4) as c4, sum(cylinders=6) as c6, sum(cylinders=8) as c8, sum(cylinders=10) as c10, count (*) as Total from sashelp.cars **group by** make;

Quit;

Note: The keyword distinct is used to eliminate duplicate rows (observations) from your query results.
The output generated is called portfolio analysis.



The SAS System

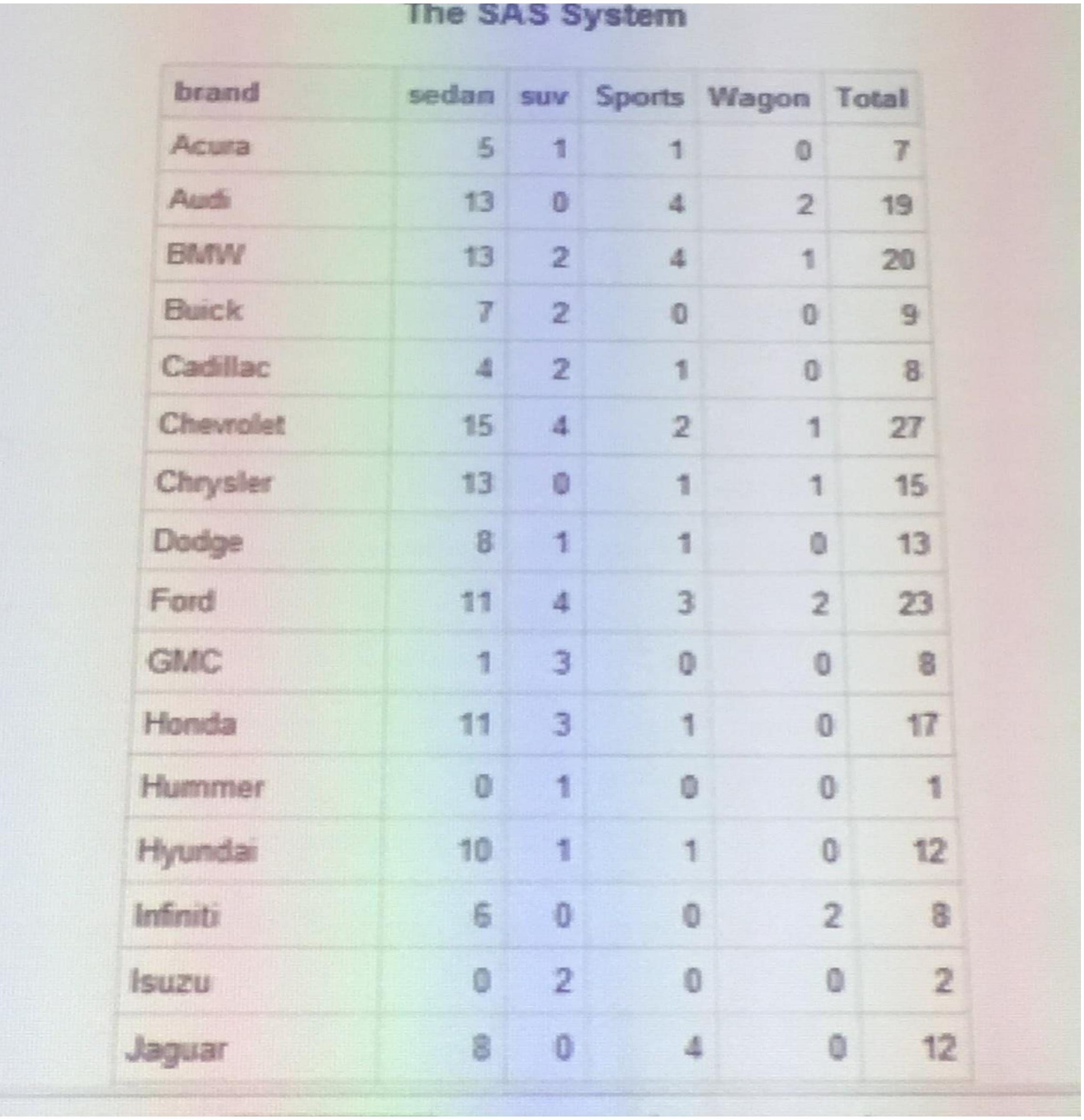
brand	c4	c6	c8	c10	Total
Acura	2	5	0	0	7
Audi	4	10	5	0	19
BMW	0	16	4	0	20
Buick	0	9	0	0	9
Cadillac	0	1	7	0	8
Chevrolet	7	13	7	0	27
Chrysler	5	10	0	0	15
Dodge	5	6	1	1	13
Ford	7	7	8	1	23
GMC	1	3	4	0	8
Honda	11	5	0	0	17
Hummer	0	0	1	0	1
Hyundai	6	6	0	0	12
Infiniti	0	5	3	0	8
Isuzu	0	2	0	0	2
Jaguar	0	3	9	0	12
Jeep	1	2	0	0	3
Kia	7	4	0	0	11
Land Rover	0	1	2	0	3

****************;

the same way we can find the count of type of cars.

Proc sql;

Select make as brand , sum(type=”Sedan”) as sedan , sum(type=”SUV”) as suv , sum (type=”Sports”) as Sports, sum(type=”Wagon”) as Wagon, count (*) as Total from Sashelp.cars **group by** make;
Quit;



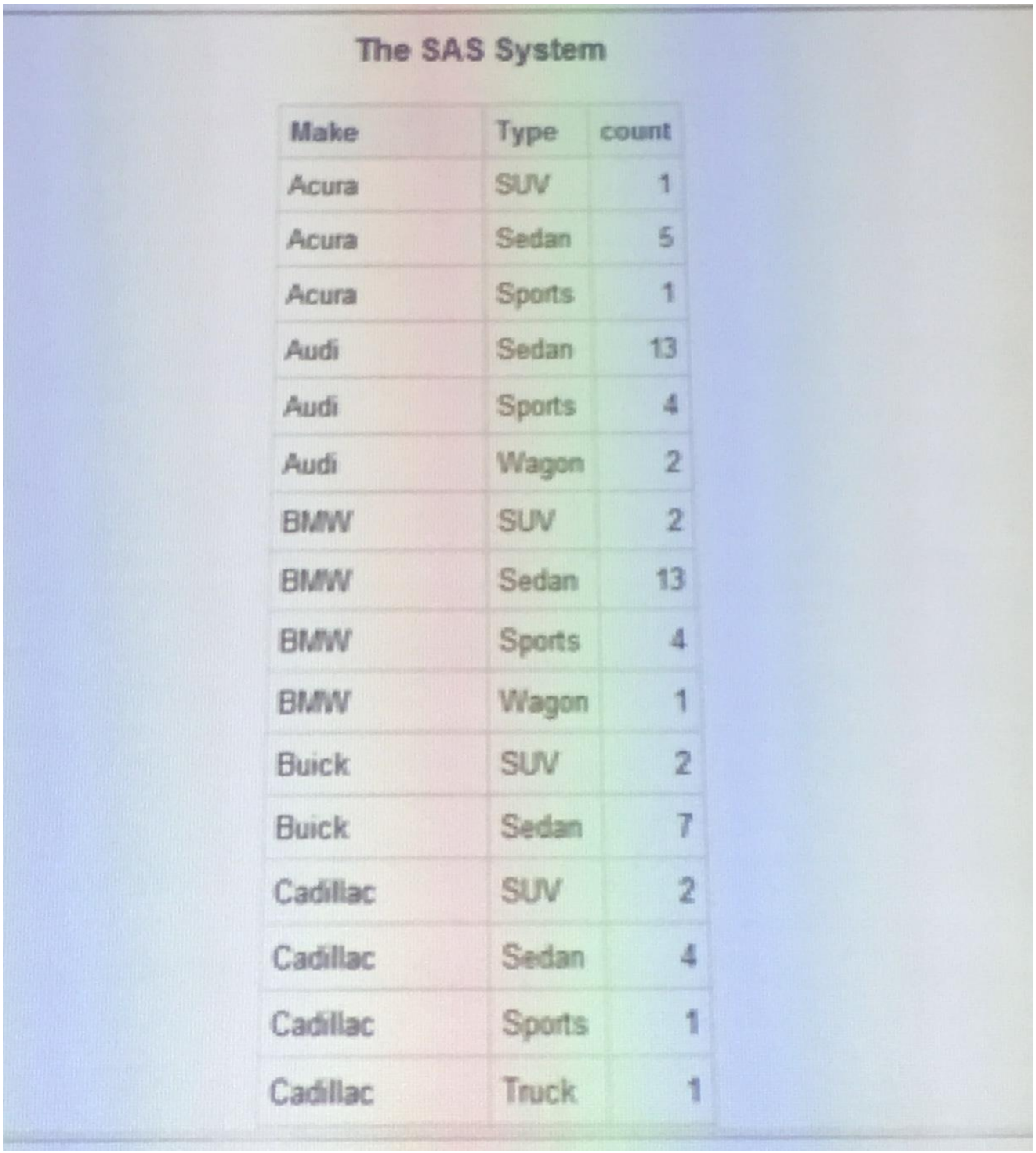
The SAS System

brand	sedan	suv	Sports	Wagon	Total
Acura	5	1	1	0	7
Audi	13	0	4	2	19
BMW	13	2	4	1	20
Buick	7	2	0	0	9
Cadillac	4	2	1	0	8
Chevrolet	15	4	2	1	27
Chrysler	13	0	1	1	15
Dodge	8	1	1	0	13
Ford	11	4	3	2	23
GMC	1	3	0	0	8
Honda	11	3	1	0	17
Hummer	0	1	0	0	1
Hyundai	10	1	1	0	12
Infiniti	6	0	0	2	8
Isuzu	0	2	0	0	2
Jaguar	8	0	4	0	12

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

Sum case- data is printed horizontally

```
Proc sql;
Select make, type,count(*) as count from sashelp.cars group by make,type;
Quit;
```



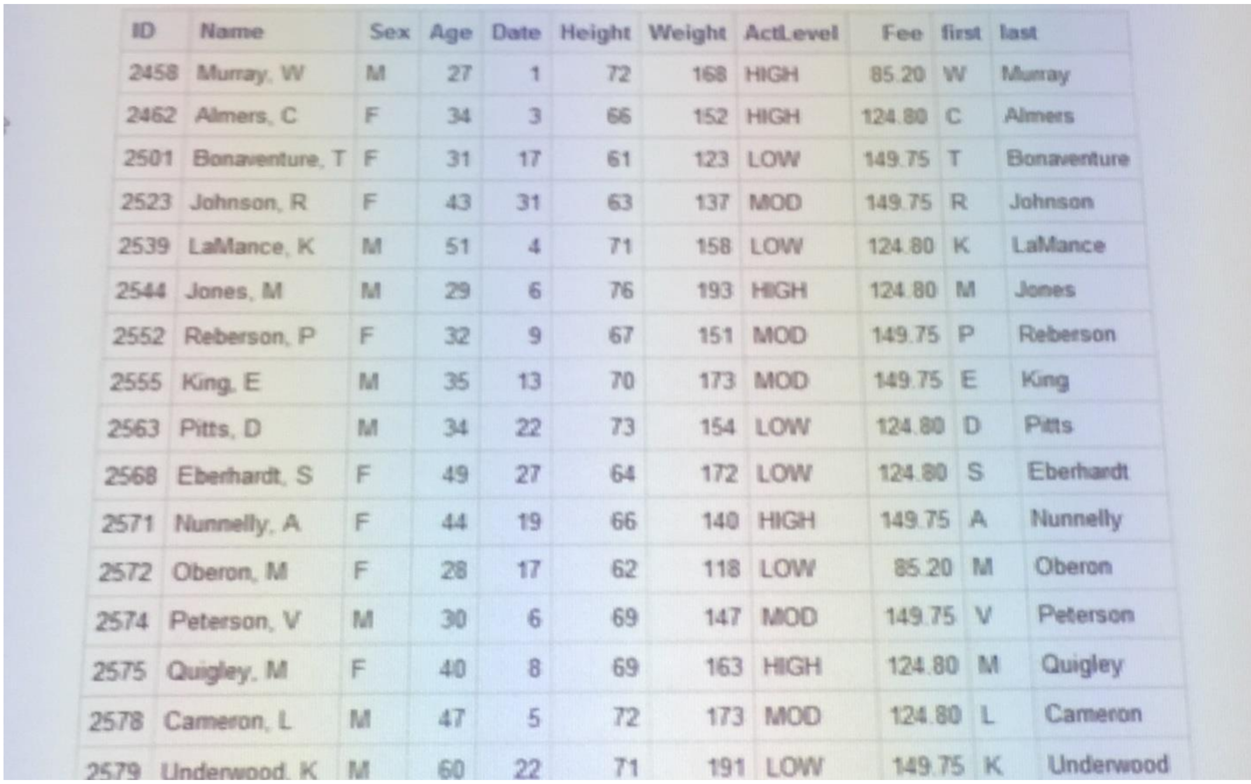
The SAS System

Make	Type	count
Acura	SUV	1
Acura	Sedan	5
Acura	Sports	1
Audi	Sedan	13
Audi	Sports	4
Audi	Wagon	2
BMW	SUV	2
BMW	Sedan	13
BMW	Sports	4
BMW	Wagon	1
Buick	SUV	2
Buick	Sedan	7
Cadillac	SUV	2
Cadillac	Sedan	4
Cadillac	Sports	1
Cadillac	Truck	1

All SAS functions can also be used in SQL

Example - To separate the first and last name of a person:

```
Proc sql;
Select * , scan ( name , 2 , ”,” ) as first , scan ( name , 1 , ”,” ) as last from sasuser.admit;
quit;
```

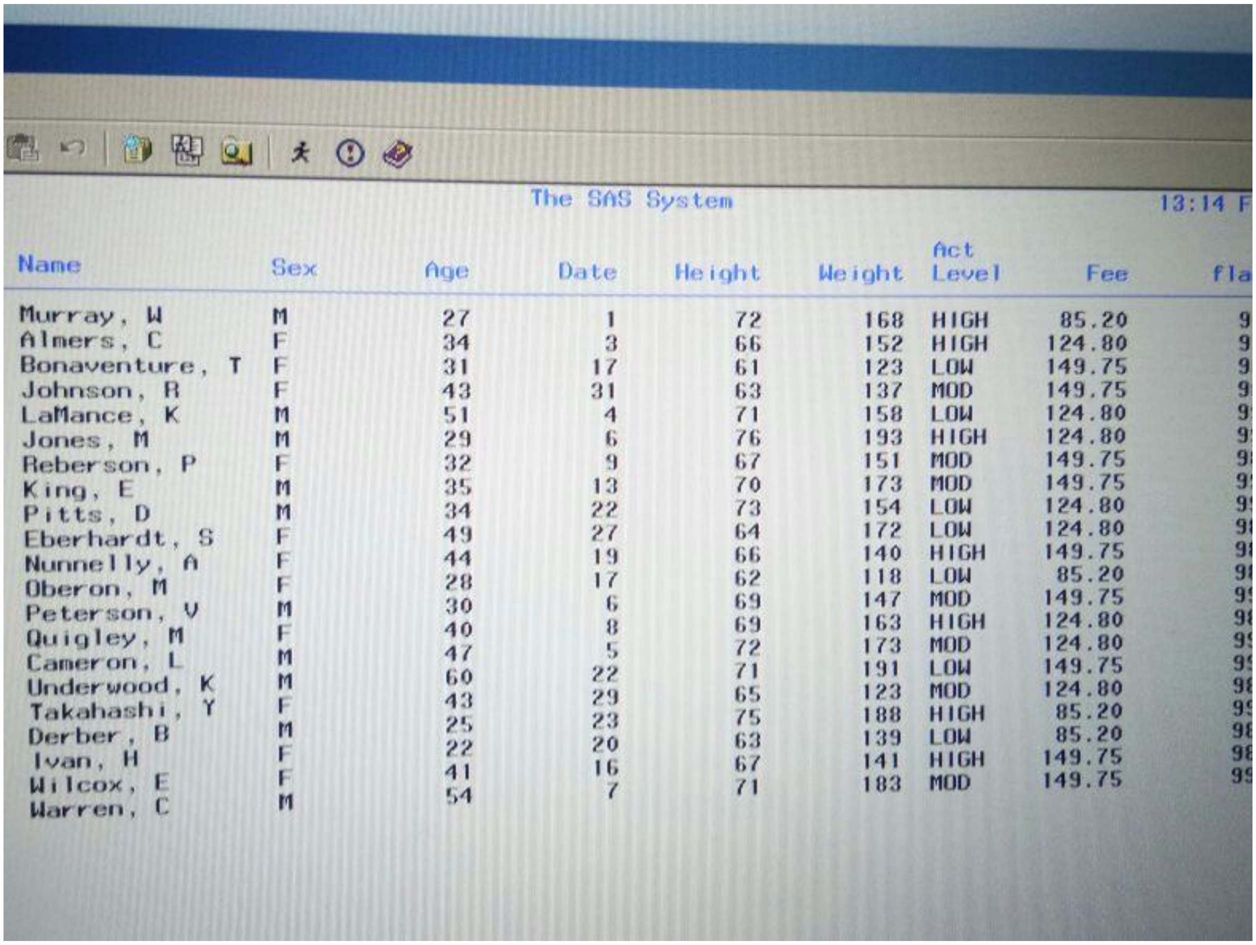


ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee	first	last
2458	Murray, W	M	27	1	72	168	HIGH	85.20	W	Murray
2462	Almers, C	F	34	3	66	152	HIGH	124.80	C	Almers
2501	Bonaventure, T	F	31	17	61	123	LOW	149.75	T	Bonaventure
2523	Johnson, R	F	43	31	63	137	MOD	149.75	R	Johnson
2539	LaMance, K	M	51	4	71	158	LOW	124.80	K	LaMance
2544	Jones, M	M	29	6	76	193	HIGH	124.80	M	Jones
2552	Reberson, P	F	32	9	67	151	MOD	149.75	P	Reberson
2555	King, E	M	35	13	70	173	MOD	149.75	E	King
2563	Pitts, D	M	34	22	73	154	LOW	124.80	D	Pitts
2568	Eberhardt, S	F	49	27	64	172	LOW	124.80	S	Eberhardt
2571	Nunnelly, A	F	44	19	66	140	HIGH	149.75	A	Nunnelly
2572	Oberon, M	F	28	17	62	118	LOW	85.20	M	Oberon
2574	Peterson, V	M	30	6	69	147	MOD	149.75	V	Peterson
2575	Quigley, M	F	40	8	69	163	HIGH	124.80	M	Quigley
2578	Cameron, L	M	47	5	72	173	MOD	124.80	L	Cameron
2579	Underwood, K	M	60	22	71	191	LOW	149.75	K	Underwood

Question 1 : create a new variable flag and wherever sex=”M” , value of flag is 99 else 98.

Code

Proc sql;
Select * , case
When sex=”M” then 99
Else 98 end as flag from sasuser.admit;
Quit;

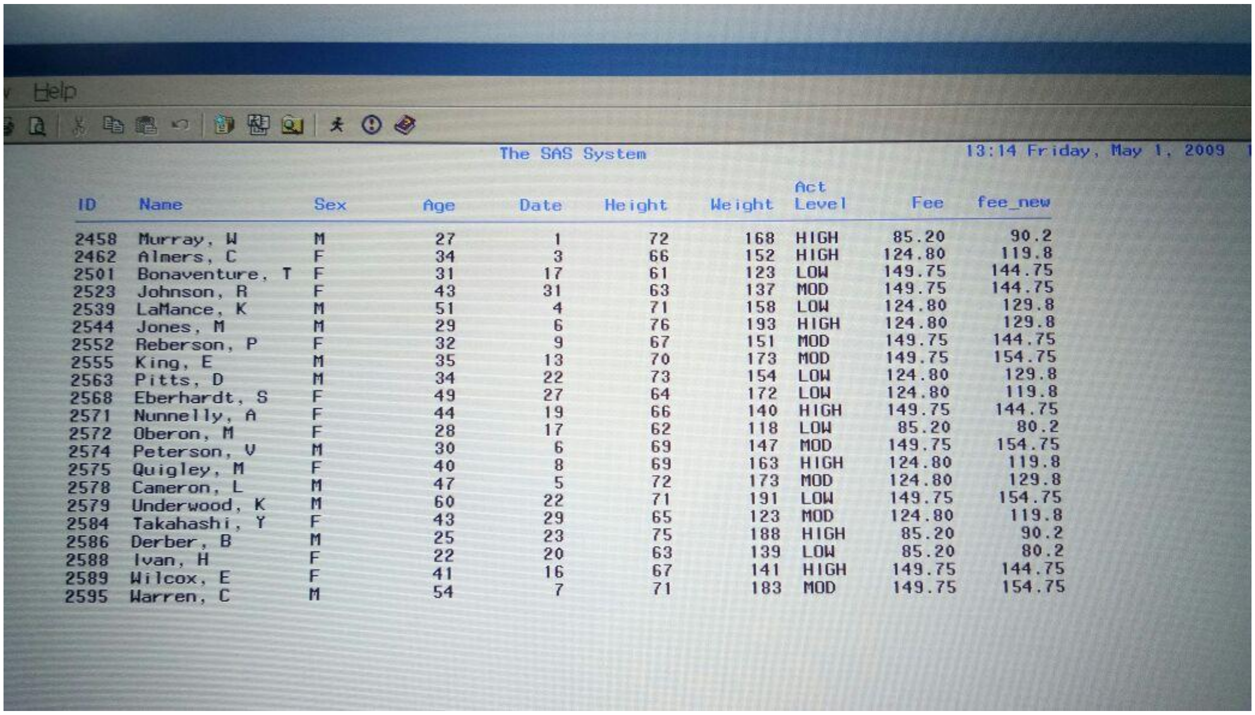


```
                                     The SAS System                                13:14 F

                                                                   Act
Name               Sex        Age      Date     Height    Weight   Level       Fee       fla

Murray, W          M           27         1         72       168   HIGH       85.20        9
Almers, C          F           34         3         66       152   HIGH      124.80        9
Bonaventure, T     F           31        17         61       123   LOW       149.75        9
Johnson, R         F           43        31         63       137   MOD       149.75        9
LaMance, K         M           51         4         71       158   LOW       124.80        9
Jones, M           M           29         6         76       193   HIGH      124.80        9
Reberson, P        F           32         9         67       151   MOD       149.75       98
King, E            M           35        13         70       173   MOD       149.75       99
Pitts, D           M           34        22         73       154   LOW       124.80       99
Eberhardt, S       F           49        27         64       172   LOW       124.80       98
Nunnelly, A        F           44        19         66       140   HIGH      149.75       98
Oberon, M          F           28        17         62       118   LOW        85.20       98
Peterson, V        M           30         6         69       147   MOD       149.75       99
Quigley, M         F           40         8         69       163   HIGH      124.80       98
Cameron, L         M           47         5         72       173   MOD       124.80       99
Underwood, K       M           60        22         71       191   LOW       149.75       99
Takahashi, Y       F           43        29         65       123   MOD       124.80       98
Derber, B          M           25        23         75       188   HIGH       85.20       99
Ivan, H            F           22        20         63       139   LOW        85.20       98
Wilcox, E          F           41        16         67       141   HIGH      149.75       98
Warren, C          M           54         7         71       183   MOD       149.75       99
```

Question 2 : Along with flag create a new variable “fee_new” and increase the fee of male by 5 , else in other case decrease by 5.

Code

```
Proc sql;
Select * , case
When sex=”M” then 99
Else 98 end as flag,
Select * , case
When sex=”M” then fee+5
Else fee-5 end as fee_new from sasuser.admit;
Quit;
```



```
                                          The SAS System                    13:14 Friday, May 1, 2009

                                                                           Act
 ID    Name              Sex        Age       Date     Height     Weight  Level        Fee    fee_new

 2458  Murray, W          M          27          1         72        168  HIGH       85.20       90.2
 2462  Almers, C          F          34          3         66        152  HIGH      124.80      119.8
 2501  Bonaventure, T     F          31         17         61        123  LOW       149.75     144.75
 2523  Johnson, R         F          43         31         63        137  MOD       149.75     144.75
 2539  LaMance, K         M          51          4         71        158  LOW       124.80      129.8
 2544  Jones, M           M          29          6         76        193  HIGH      124.80      129.8
 2552  Reberson, P        F          32          9         67        151  MOD       149.75     144.75
 2555  King, E            M          35         13         70        173  MOD       149.75     154.75
 2563  Pitts, D           M          34         22         73        154  LOW       124.80      129.8
 2568  Eberhardt, S       F          49         27         64        172  LOW       124.80      119.8
 2571  Nunnelly, A        F          44         19         66        140  HIGH      149.75     144.75
 2572  Oberon, M          F          28         17         62        118  LOW        85.20       80.2
 2574  Peterson, V        M          30          6         69        147  MOD       149.75     154.75
 2575  Quigley, M         F          40          8         69        163  HIGH      124.80      119.8
 2578  Cameron, L         M          47          5         72        173  MOD       124.80      129.8
 2579  Underwood, K       M          60         22         71        191  LOW       149.75     154.75
 2584  Takahashi, Y       F          43         29         65        123  MOD       124.80      119.8
 2586  Derber, B          M          25         23         75        188  HIGH       85.20       90.2
 2588  Ivan, H            F          22         20         63        139  LOW        85.20       80.2
 2589  Wilcox, E          F          41         16         67        141  HIGH      149.75     144.75
 2595  Warren, C          M          54          7         71        183  MOD       149.75     154.75
```