

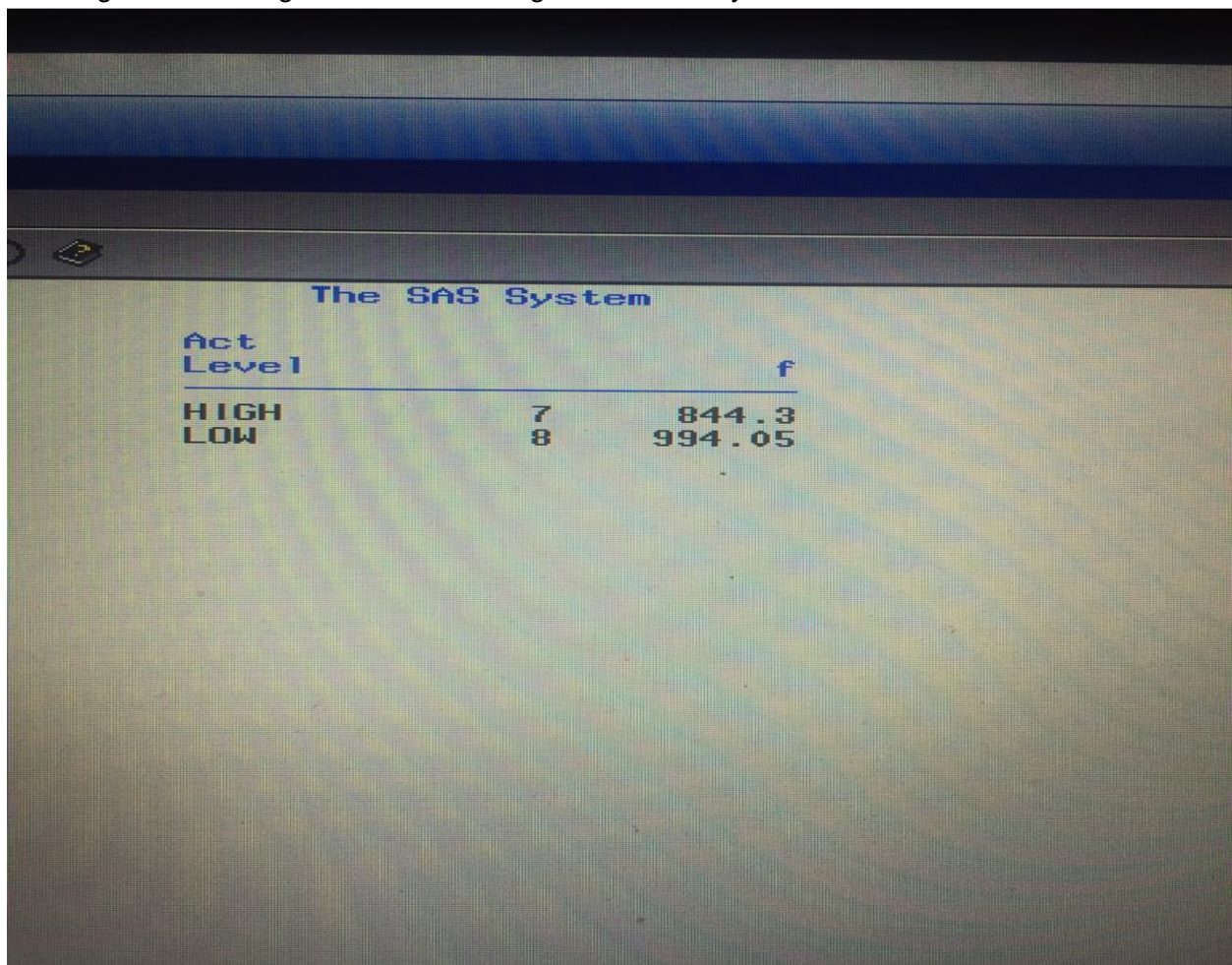
CLASS - 20

INLINE VIEW : It is like imaginary table that does not exist. It does not take physical memory.

```
Proc sql;  
Select * from  
(select actlevel , count (*) , sum (fee) as f from sasuser.admit group by actlevel) { this is inline  
view as it is not a physical table }  
Where actlevel IN("HIGH","LOW");  
Quit;
```

Explanation : The outside query (**Select * from**) running on inline view now has three variables available i.e. actlevel , count , f (sum fee).

Working on something that we are making simultaneously is called inline view.



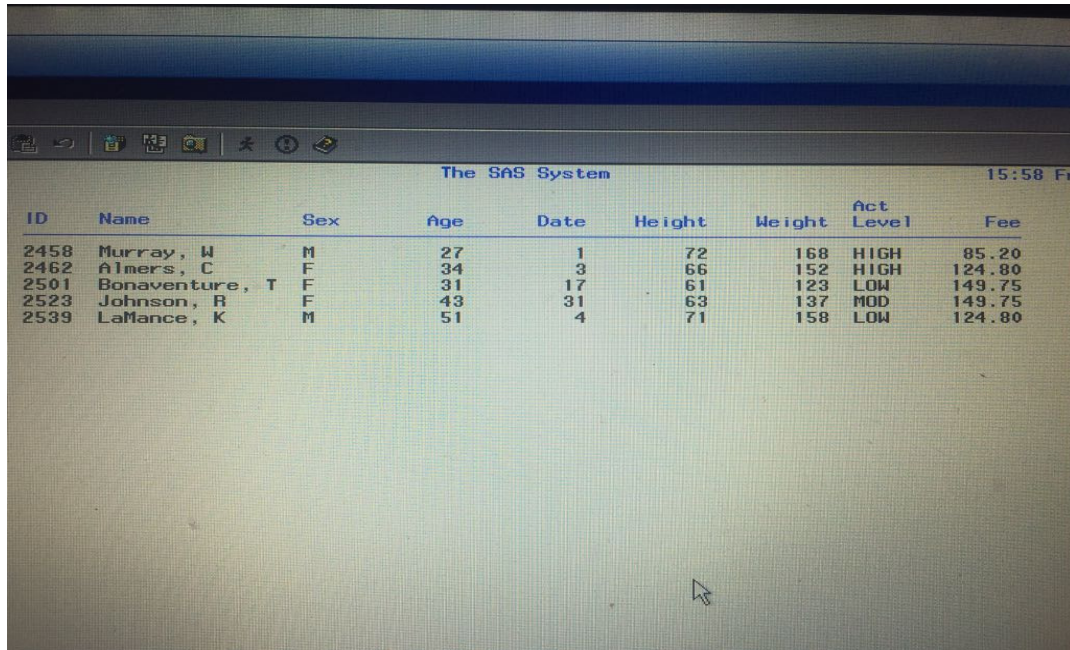
Act Level		f
HIGH	7	844.3
LOW	8	994.05

***** Another example we can take from the example sasuser.cargorev in order to find the best performing route.*****

Question : Give the top 5 observations from sasuser.admit using SQL.

Code:

```
Proc sql inobs=5;  
Select * from sasuser.admit;  
Quit;
```



The screenshot shows a SAS System window with a table of 5 observations. The table has columns: ID, Name, Sex, Age, Date, Height, Weight, Act Level, and Fee. The data is as follows:

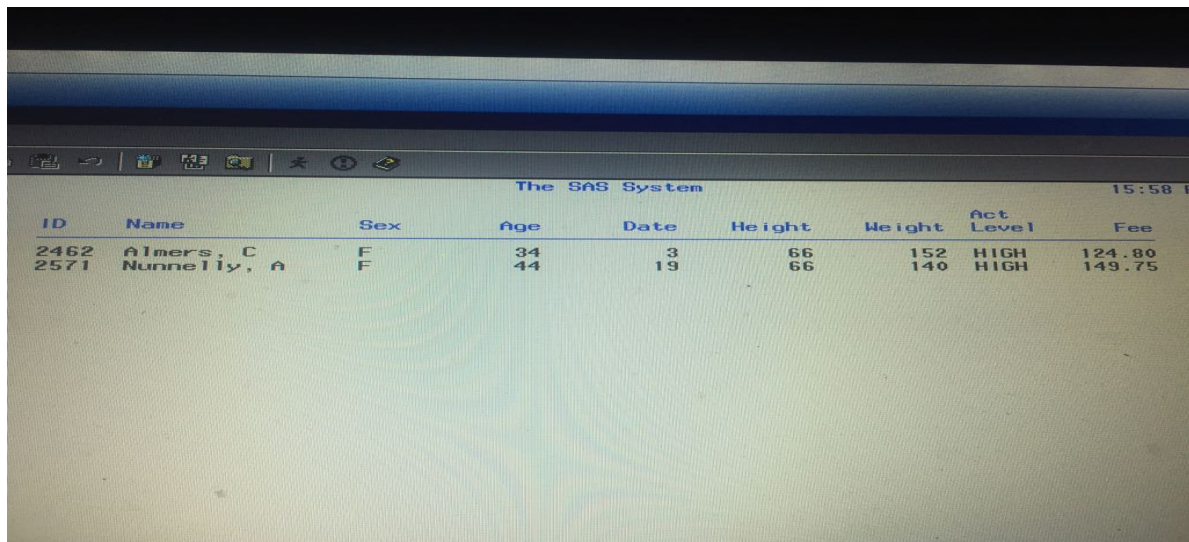
ID	Name	Sex	Age	Date	Height	Weight	Act Level	Fee
2458	Murray, W	M	27	1	72	168	HIGH	85.20
2462	Almers, C	F	34	3	66	152	HIGH	124.80
2501	Bonaventure, T	F	31	17	61	123	LOW	149.75
2523	Johnson, R	F	43	31	63	137	MOD	149.75
2539	LaMance, K	M	51	4	71	158	LOW	124.80

Explanation : inobs=5 / outobs=5 will give the top 5 observations.

Question : Give the data of the customers from sasuser.admit that have capital 'A' anywhere in their names using functions.

Code:

```
Proc sql;  
Select * from sasuser.admit where index(name,'A') > 0;  
Quit;
```

ID	Name	Sex	Age	Date	Height	Weight	Act Level	Fee
2462	Almers, C	F	34	3	66	152	HIGH	124.80
2571	Nunnelly, A	F	44	19	66	140	HIGH	149.75

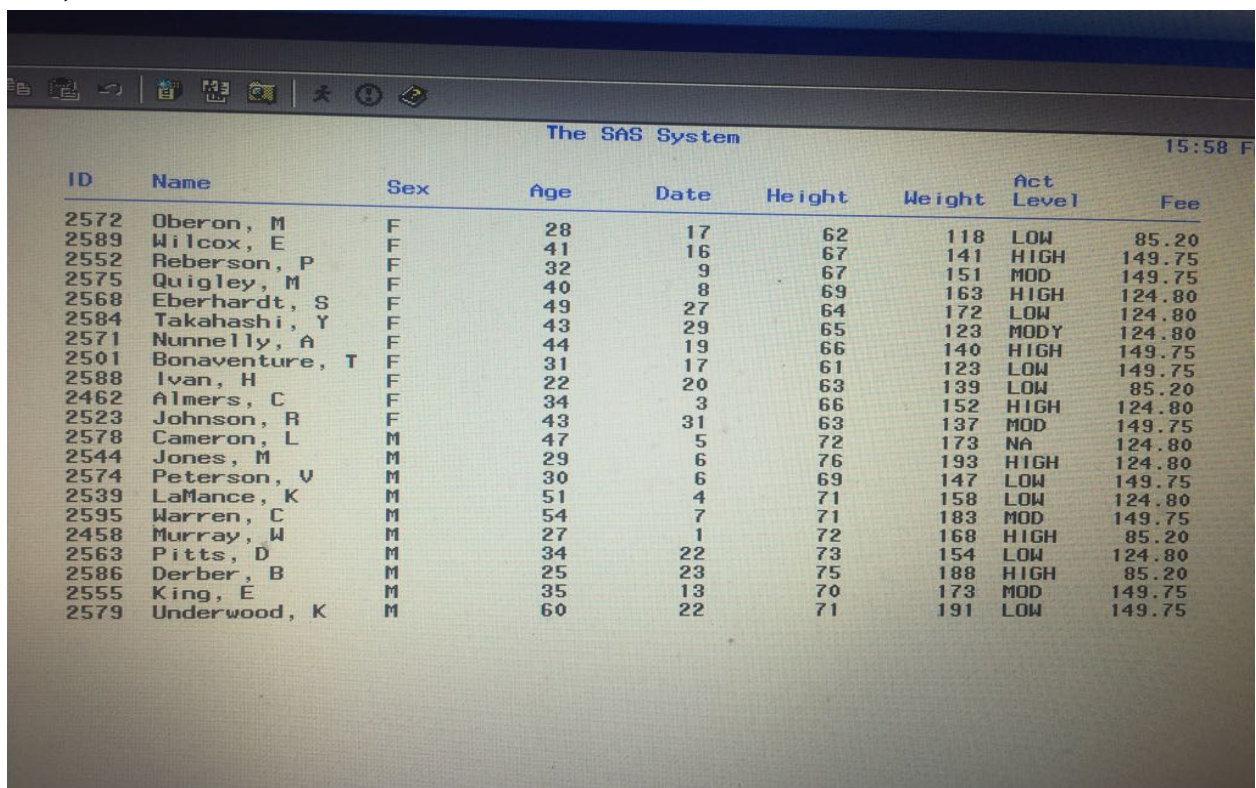
Interview Questions:

***** **1. Group by , no summary function:** Using "group by" without summary function will sort the data. It will behave like "order by".

Proc sql;

Select * from sasuser.admit group by sex;

Quit;



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

Note : Now, different ways of sorting we know till now - **proc sort , group by , order by.**

*****2. **Having** *****

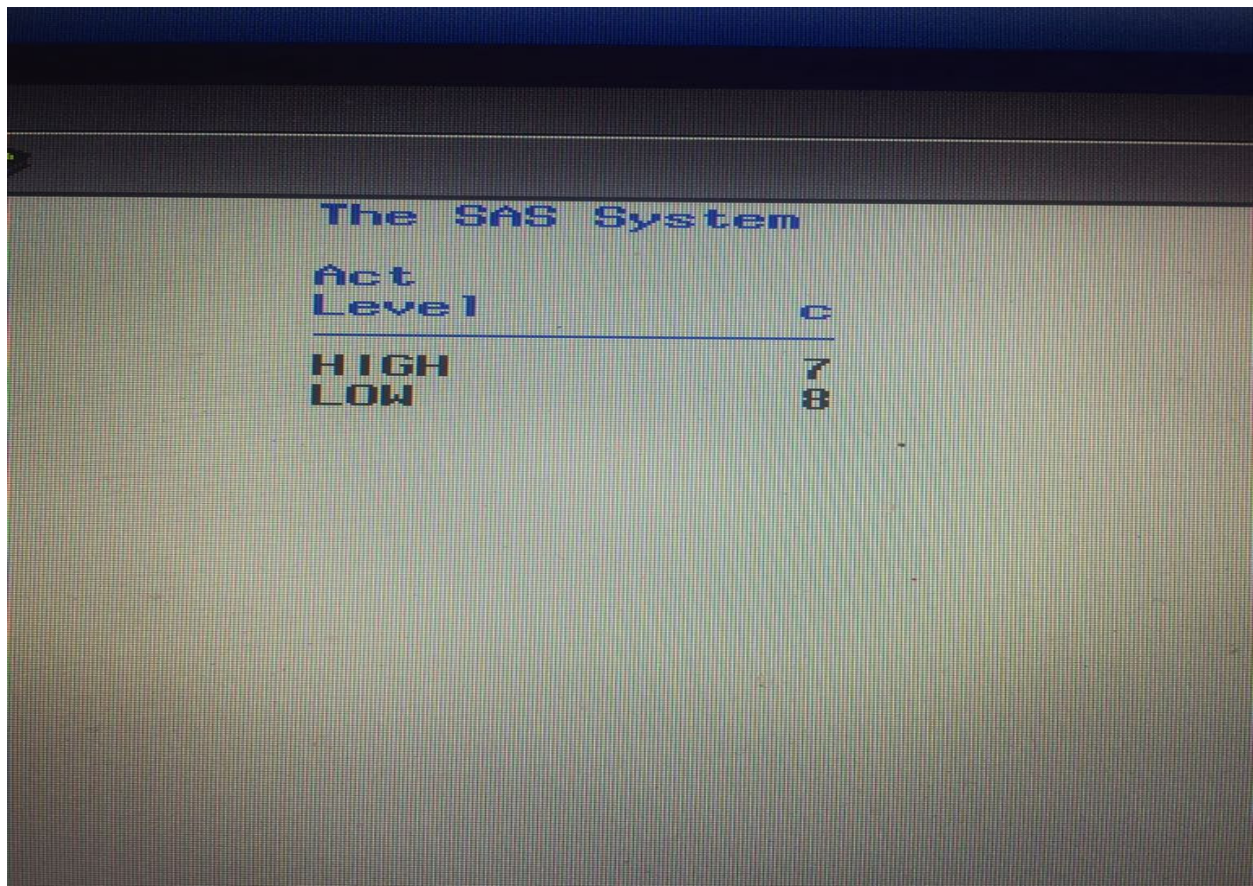
Having is used to apply filter on groups , groups are made by “ group by “ and with “ group by “ we have summary functions.

For applying filter on rows, we use “ where “. (**Interview Question** : Difference between having and where)

Note :Using having with summary function will work , we do not require subquery in this case.

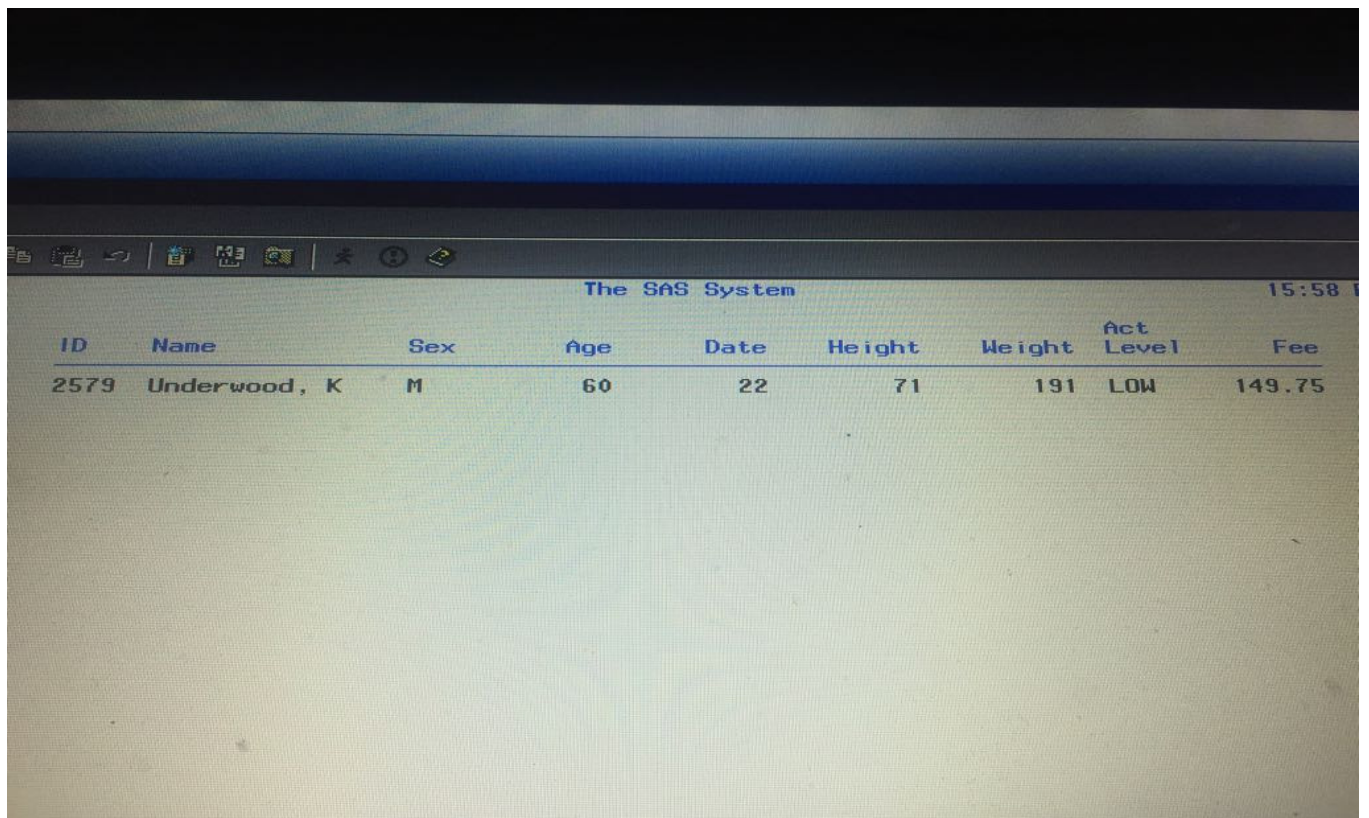
Code:

```
Proc sql;  
select actlevel , count (*) as c from sasuser.admit group by actlevel having actlevel  
IN(“HIGH”,“LOW”);  
Quit;
```



The SAS System	
Act Level	c
HIGH	7
LOW	8

```
Proc sql;  
Select * from sasuser.admit having age=max(age);  
Quit;
```

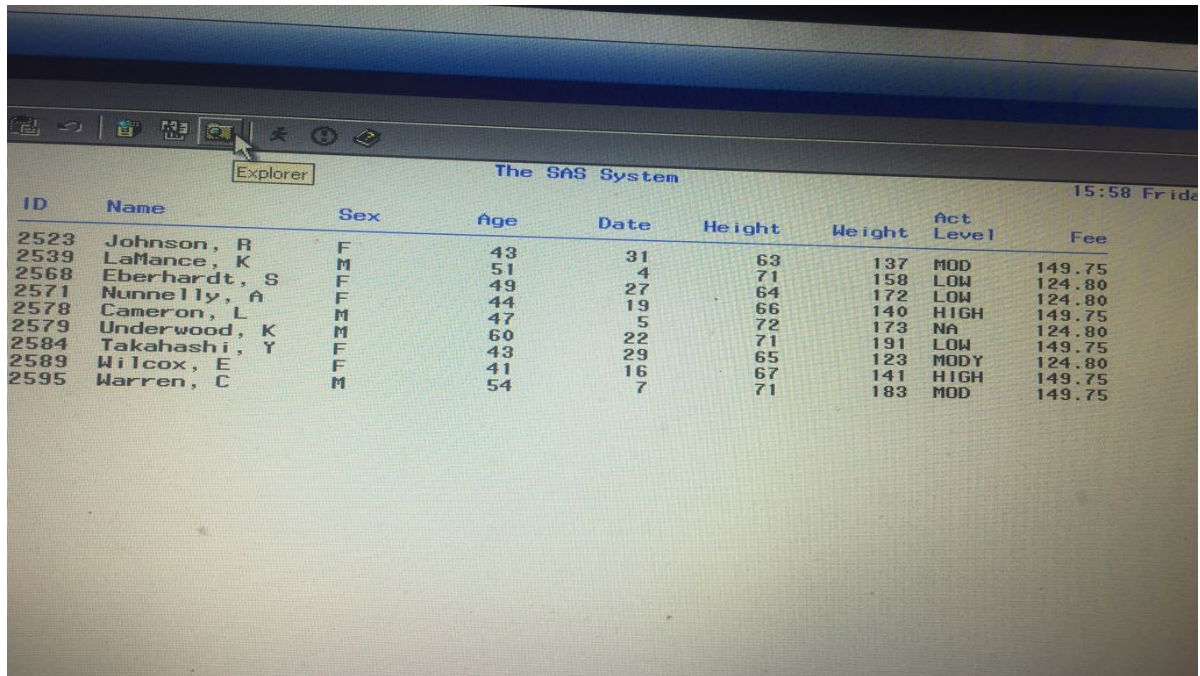
The SAS System									15:58
ID	Name	Sex	Age	Date	Height	Weight	Act Level	Fee	
2579	Underwood, K	M	60	22	71	191	LOW	149.75	

Explanation : This will give the max age i.e. 60.

***** 3. **Having without group by and summary function:** Having will behave like “where”

Code:

```
Proc sql;
Select * from sasuser.admit having age gt 40;
Quit;
```



The screenshot shows the SAS Explorer window with a table of data. The table has the following columns: ID, Name, Sex, Age, Date, Height, Weight, Act Level, and Fee. The data is as follows:

ID	Name	Sex	Age	Date	Height	Weight	Act Level	Fee
2523	Johnson, R	F	43	31	63	137	MOD	149.75
2539	LaMance, K	M	51	4	71	158	LOW	124.80
2568	Eberhardt, S	F	49	27	64	172	LOW	124.80
2571	Nunnally, A	F	44	19	66	140	HIGH	149.75
2578	Cameron, L	M	47	5	72	173	NA	124.80
2579	Underwood, K	M	60	22	71	191	LOW	149.75
2584	Takahashi, Y	F	43	29	65	123	MODY	124.80
2589	Wilcox, E	F	41	16	67	141	HIGH	149.75
2595	Warren, C	M	54	7	71	183	MOD	149.75

Order Of Query	Order Of Execution
1.Select	1. Where
2. From	2. From
3. Where	3. Select
4. Group By	4. Group By
5. Having	5. Having
6. Order By	6. Order By

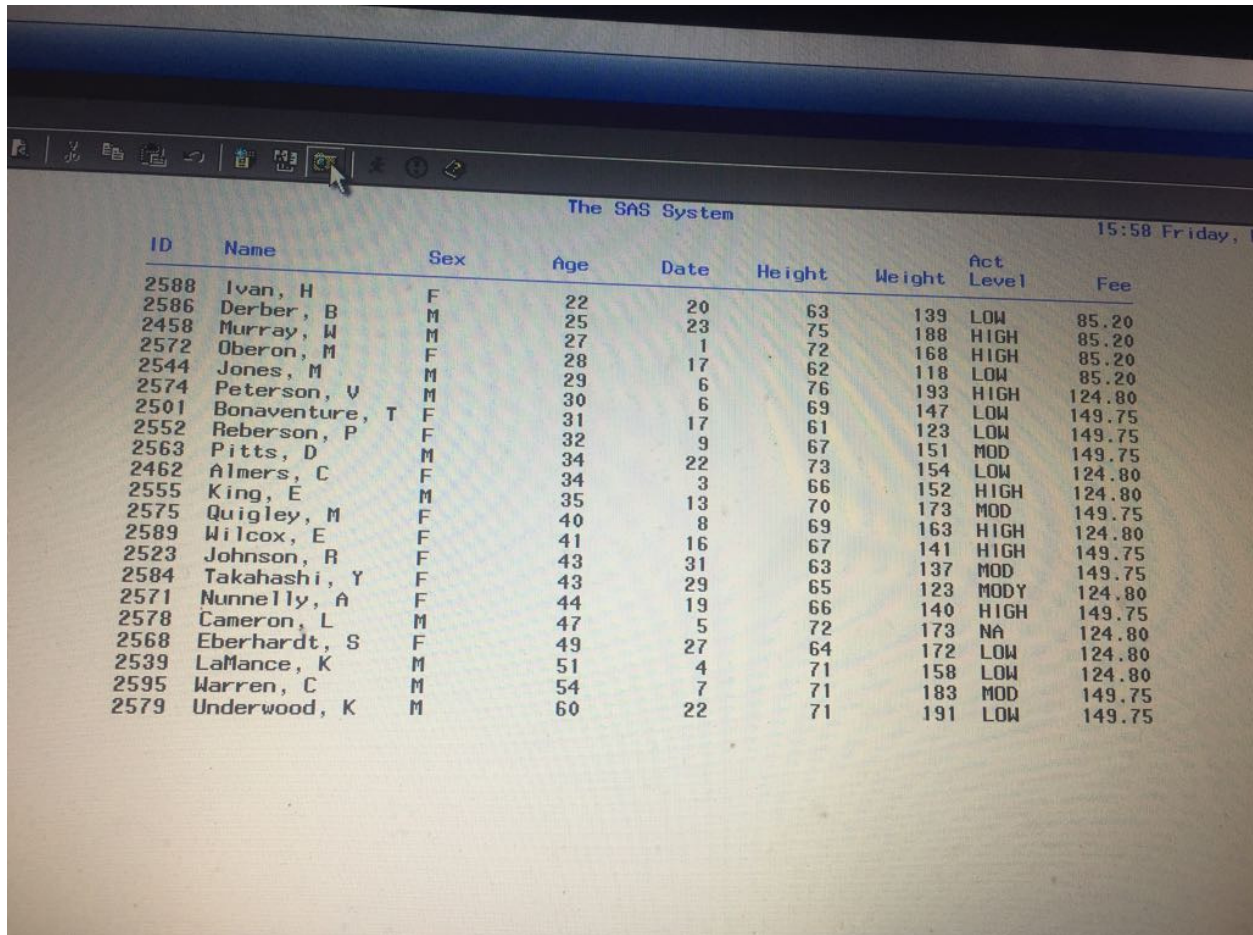
Order of execution : first where, then the data will come, then variables will get selected , then groups will made and so on.

***** Order By = proc sort;

“Order by” is used to sort the data.

Code:

```
Proc sql;  
Select * from sasuser.admit order by age;  
quit;
```



The screenshot shows the SAS output window titled "The SAS System" with a timestamp of "15:58 Friday, 1". The output is a table with the following columns: ID, Name, Sex, Age, Date, Height, Weight, Act Level, and Fee. The data is sorted by age in ascending order.

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

Explanation : Data is sorted by age in ascending order.

Code:

```
Proc sql;  
Select * from sasuser.admit order by name;  
Quit;
```


The SAS System 15:58 Friday,

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

Explanation : Data is sorted by name in alphabetical order.

Code:

```
Proc sql;
Select * from sasuser.admit order by sex,actlevel;
Quit;
```


The SAS System 15:58 Friday, May 1,

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

Explanation : Data is sorted by sex and in every sex it is sorted by actlevel.

Code:

```
Proc sql;
Select * from sasuser.admit order by age desc; { In SQL, desc is written after variable name
whereas in base SAS it is written as by descending age, also we have to write full descending in
base SAS whereas in SQL it can work by writing just "desc" }
quit;
```

The SAS System 15:58 Friday

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

Code:

```
Proc sql;
Select * from sasuser.admit order by 5;
Quit;
```

Explanation : Data is sorted by 5th variable i.e. " Date " from the table.

The SAS System 15:58 Friday,

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

Code:

```
Proc sql;
```

```
Select * from sasuser.admit order by 6,5;
```

```
Quit;
```

Explanation : Data is sorted first by 6th variable and then 5th from the table.

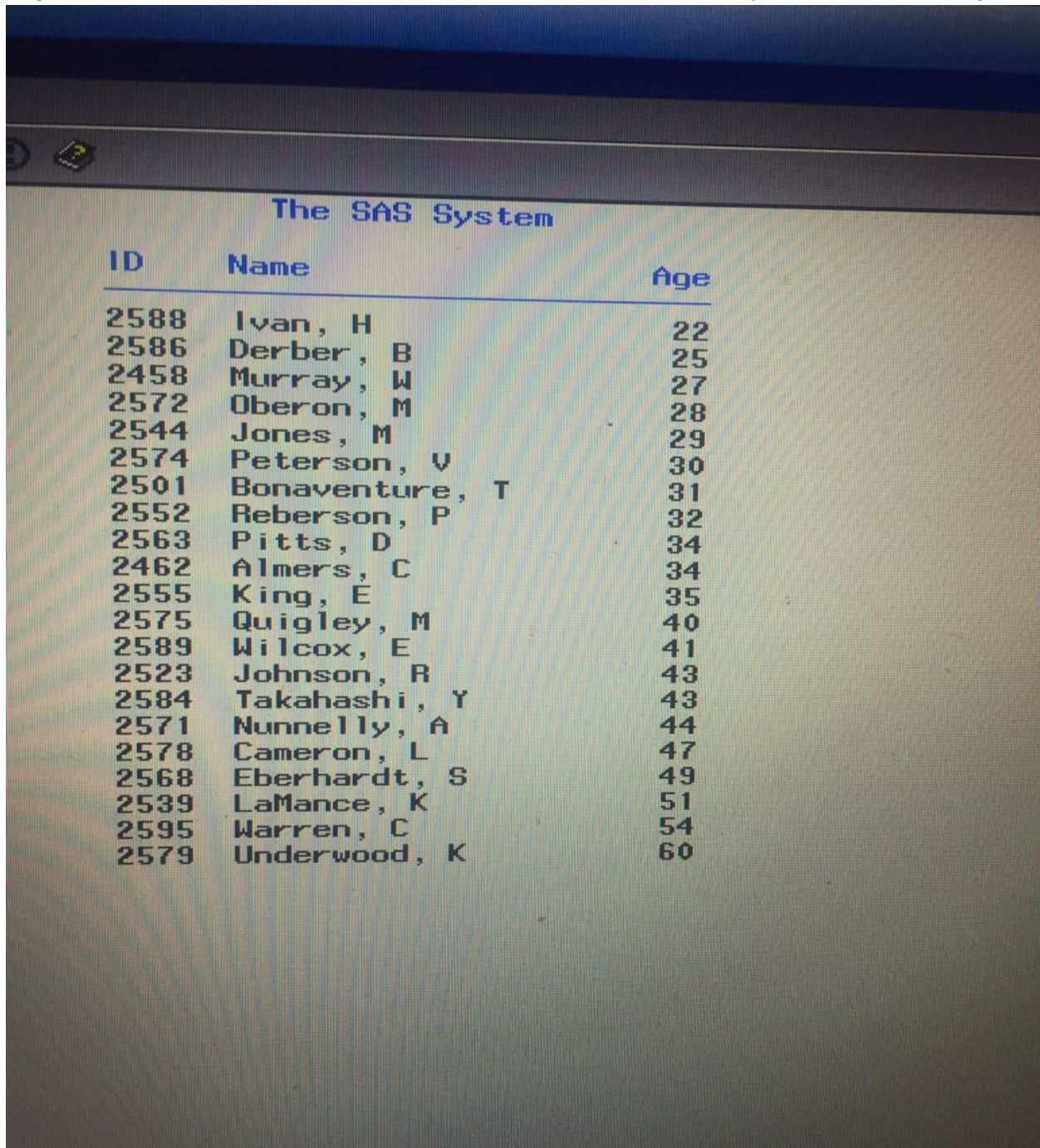
The SAS System 15:58 Friday,

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

Code:

```
Proc sql;  
Select id,name,age from sasuser.admit order by 3;  
Quit;
```

Explanation : Out of the three variables mentioned , Data is sorted by 3rd variable i.e. “ Age “



The screenshot shows a window titled "The SAS System" displaying a table of data. The table has three columns: ID, Name, and Age. The data is sorted by Age in ascending order. The table contains 20 rows of data.

ID	Name	Age
2588	Ivan, H	22
2586	Derber, B	25
2458	Murray, W	27
2572	Oberon, M	28
2544	Jones, M	29
2574	Peterson, V	30
2501	Bonaventure, T	31
2552	Reberson, P	32
2563	Pitts, D	34
2462	Almers, C	34
2555	King, E	35
2575	Quigley, M	40
2589	Wilcox, E	41
2523	Johnson, R	43
2584	Takahashi, Y	43
2571	Nunnelly, A	44
2578	Cameron, L	47
2568	Eberhardt, S	49
2539	LaMance, K	51
2595	Warren, C	54
2579	Underwood, K	60

Code:

```
Proc sql;  
Select * from sasuser.admit order by sex , actlevel desc;  
Quit;
```

Explanation : Data is sorted first by “sex” in ascending order and “actlevel” in descending order.

Question : Country wise, location wise count and the country is USA and the output should be sorted by count and the maximum count should be on the top.

```
Proc sql;  
Select country, location, count(*) as count from sasuser.empdata where country="USA"  
Group by country,location order by count desc;  
Quit;
```

Output

Employee Country of Residence	Employee office Location	count
USA	CARY	37
USA	PHOENIX	1
USA	AUSTIN	1
USA	BOSTON	1
USA	HOUSTON	1

Question : Country wise, location wise print only the row with maximum count i.e. CARY row.

Code:

```
Proc sql;  
Select country, location, count(*) as count from sasuser.empdata where country="USA"  
Group by country,location ) having count=max(count);  
Quit;
```

Error: Summary functions nested in this way are not supported.

having count=max(count) - count is made from count(*) , as we cannot write nested function in a query so, to resolve this inline view will be used.

Summary functions cannot be used in nested way. It cannot calculate two things in one iteration.

We cannot use two summary functions (i.e. max and count(*) on one variable).

Alternative to above program using inline view

Code:

Options **nolabel;** { By default SQL print labels, nolabel will remove the labels and will print variable name }

```
Proc sql;
Select * from
(
Select country, location, count(*) as count from sasuser.empdata where country="USA"
Group by country,location // from this inline view "count" is created and variable is available now//
)
having count=max(count);
Quit;
```

Output

Country	Location	count
USA	CARY	37

Question : Give the the count of cylinders i.e. c4,c6,c8 of brand "Audi" based on the technology Quattro.

Code:

```
proc sql;

select distinct make as brand,sum(cylinders=4) as c4, // sum(cylinders=4) -wherever the
cylinder value will be 4, it will make as 1 and doing their sum will give the count and same will
happen in case of c6 and c8 //
```



```
sum(cylinders=6) as c6, sum(cylinders=8) as c8
from sashelp.cars where make="Audi" and index(model,"Quattro") gt 0;
```

Quit;

Output

brand	c4	c6	c8
Audi	1	6	4

Question : Print the total also.

Code:

```
proc sql;
```

```
select make as brand,cylinders,count(*) as count,
(select count(*) from sashelp.cars where make="Audi") as Total
from sashelp.cars where make="Audi" and index(model,"Quattro") gt 0
group by brand,cylinders;
```

quit;

brand	cylinders	count	Total
Audi	4	1	19
Audi	6	6	19
Audi	8	4	19

Question : From the output of cylinders we want the values 4,6,8 to be printed as C4,C6,C8. For this we will use "cat" function.

Code:

```
proc sql;
```

```
select make as brand,cat("C",cylinders)as cyl,count(*) as count,  
(select count(*) as count from sashelp.cars where make="Audi") as Total  
from sashelp.cars where make="Audi" and index(model,"Quattro") gt 0  
group by brand,cylinders;  
quit;
```

brand	cyl	count	Total
Audi	C4	1	19
Audi	C6	6	19
Audi	C8	4	19

Question : From the above output print only the row with maximum count.

Code:

```
proc sql;  
select * from  
(  
select make as brand,cat("C",cylinders)as cyl, count(*) as count,  
(select count(*) from sashelp.cars where make="Audi") as Total  
from sashelp.cars where make="Audi" and index(model,"Quattro") gt 0  
group by brand,cylinders  
) having count=max(count);  
quit;
```

brand	cyl	count	Total
Audi	C6	6	19

Code:

```
proc sql;
```

```
select * from
(
select make as brand,cat("C",cylinders)as cyl, count(*) as count,
( select count(*) from sashelp.cars )as Total
From sashelp.cars group by brand,cylinders
) having count=max(count);
quit;
```

brand	cyl	count	Total
-------	-----	-------	-------

BMW	C6	16	428
-----	----	----	-----

Explanation: Out of all the brands BMW is the one making maximum number of cars of 6 cylinders and 428 is grand total..