

*HEALTH*  
*QUALITY*  
C O U N C I L



# proc sql

*Nirmal Singh, Research Analyst  
Health Quality Council  
Saskatoon, Canada*

# proc sql

- *SAS' s version of Structured Query Language*
- *Developed by IBM in the early 1970s*
- *It resembles saying some task in English*

# Basic Structure Of sql

```
proc sql;  
create table data1 as select *  
from library.data2  
where drug_name="Benzo"  
order by id;  
quit;
```

We will use words such as **having**, **group by**

# Keeping Limited variables

```
proc sql;  
create table temp3 as select  
                a.id, a.adms_date,  
                b.FSC, b.dos, b.key_doc_num  
from temp1 a, temp2 b  
where a.id=b.id and adms_date LT '31dec1996'd  
order by id desc;  
quit;
```

Note: we do not have `select *`, instead we select . . .

# Counting within proc sql

```
proc sql;  
create table data2 as select *,  
                        count(*) as counterx  
from data1  
group by id, year      ← How would it count?  
having counterx=5;  
quit;
```

- group by classifies data into groups
- having subsets groups of data based on a group condition

# Calculations within proc sql

```
proc sql;
create table temp2 as select *,
                        count(*) as NumOfScript,
                        sum(rx_qty_amt) as NumOfPills
from library.temp1
group by id, gender, year    ←How would it count?
where NumOfScript GE 3 and NumOfPills GE 100;
quit;
```

What else? average, mean, median, formula etc.

## calculation, format, condition, sort

```
proc sql;
create table temp1 as select *,
min(FirstDate_MSB, First_adm_date) as date_of_diag format=date9.,
(calculated date_of_diag-birthday)/365 as age_at_diab format=4.2
from Diabetes_data
having age_at_diab GE 20
order by age_at_diab;
quit;
```

# More Conditions? Yes!

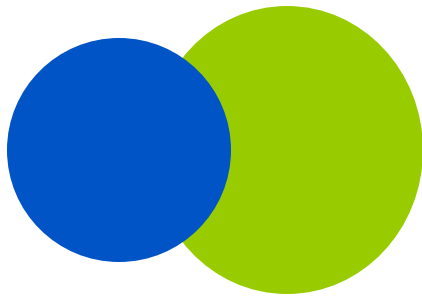
```
proc sql;  
create table case_data as select *  
  from extract.registry  
where key_hsn in  
  (select key_hsn from bcancer.surg_singl)  
and key_hsn in  
  (select key_hsn from bcancer.both)  
and first_primary = 'Y'  
  order by key_hsn;  
quit;
```

# More Conditions? Yes!

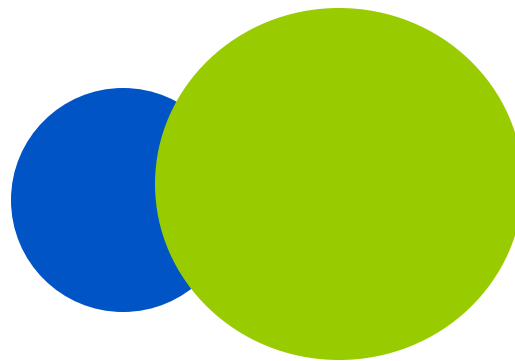
```
proc sql;  
create table data3 as select *  
  from data2  
  where key_HSN in  
  ( select key_HSN from data1  
  where (0 GT gap LE 34) and adm_date LE '01Jan2006'd )  
  order by key_hsn, admit_dt, disch_dt;  
quit;
```

# Types of Joins

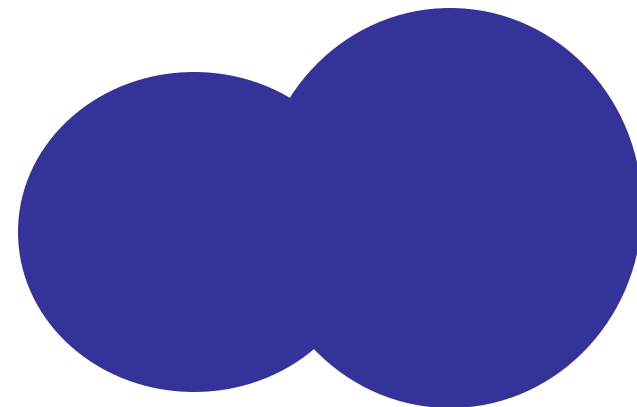
Left Join



Right Join



Inner Join



Full Join

# Left and Right Join

```
data temp1;  
input key Marks1;  
datalines;  
1 11  
2 22  
3 33  
4 44  
4 442  
5 55;  
run;
```

```
data temp2;  
input key Marks2;  
datalines;  
1 111  
2 222  
3 333  
3 133  
4 444  
6 1515;  
run;
```

# Left and Right Join

```
proc sql;  
create table left_data as select *  
from temp1 a left join temp2 b  
on a.key=b.key;  
quit;
```

```
proc sql;  
create table right_data as select *  
from temp1 a right join temp2 b  
on a.key=b.key;  
quit;
```

# Left Join Output

key	Marks1	Marks2
1	11	111
2	22	222
3	33	333
3	<u>33</u>	133
4	442	444
4	44	<u>444</u>
5	55	.

# Right Join Output

key	Marks1	Marks2
1	11	111
2	22	222
3	33	333
3	33	133
4	442	444
4	44	444
.	.	1515

# Inner and Full Join

```
proc sql;  
create table innerdata1 as select *  
from temp1 a inner join temp2 b  
on a.key=b.key;  
quit;
```

```
proc sql;  
create table fulldata1 as select *  
from temp1 a full join temp2 b  
on a.key=b.key;  
quit;
```

# Inner Join Output

key	Marks1	Marks2
1	11	111
2	22	222
3	33	333
3	33	133
4	44	444
4	442	444

# Full Join Output

key	Marks1	Marks2
1	11	111
2	22	222
3	33	333
3	33	133
4	442	444
4	44	444
5	55	.
.	.	1515

# Merge, Left/Right\_Join:Caution

- Whether we do left join or right join, if some variable is common in both data sets, if always keeps that variable from the left data set.
- The same is true even for id.
- Where as while using merge statement, it depends! Keeps that variable, from the data set, written last.

ZXZXZX \*

```

proc sql;
  create table left_join as select *
  from data1 left join data2
  on data1.id=data2.id;
quit;

proc sql;
  create table right_join as select *
  from data1 right join data2
  on data1.id=data2.id;
quit;

data merge1;
  merge data1 data2;
  by id;
run;

```

### VIEWTABLE: Work.Data1

	id	x1
1	1	1

### VIEWTABLE: Work.Data2

	id	x2
1	1	3
2	2	4

### VIEWTABLE: Work.Left\_join

	id	x1	x2
1	1	1	3

### VIEWTABLE: Work.Right\_join

	id	x1	x2
1	1	1	3
2	.	.	4

### VIEWTABLE: Work.Merge 1

	id	x1	x2
1	1	1	3
2	2	.	4

# In / Not In Condition

```
proc sql;  
create table New as select * from Diabetes  
where key not in  
(select key from Pregnancy_filter);  
quit;
```

