Introduction to SQL

University of Iowa

SAS® Users Group

See Us on the Web:

https://uisug.org.uiowa.edu/

Subscribe to the UI SAS ListServ:

Via the web:

https://list.uiowa.edu/scripts/wa.exe?HOME

Via email:

email to: listserv@list.uiowa.edu
subject: *anything*
body: subscribe sas-users *yourfirstname yourlastname*

1. Introduction and basic uses
2. Joins and Views
3. Reporting examples

Introduction to SQL

Patient	Sex	YOB	Blood Type	Visit #	Weight
Joe	male	73	A+	1	150
Joe	male	73	A+	2	153
Joe	male	73	A+	3	151
Sally	female	88	O-	1	128
Sally	female	88	O-	2	127
Sally	female	88	O-	3	126
Tom	male	71	B+	1	163
Tom	male	71	B+	2	165

"Old School" Flat File Structure

"Really Old School" Flat File Structure

Patient	Sex	YOB	Blood Type
Joe	male	73	A+
Sally	female	88	O-
Tom	male	71	B+

Patient	Visit #	Weight
Joe	1	150
Joe	2	153
Joe	3	151
Sally	1	128
Sally	2	127
Sally	3	126
Tom	1	163
Tom	2	165

Relational File Structure

SQL

- Oracle
  - Database
  - MySQL
- Microsoft
  - SQL Server
  - Access
- IBM
  - DB2
  - Informix
- SAP Sybase
- Teradata
- PostgreSQL
- SQLite

SQL

Performs direct data access functions
Retrieves and Updates tables

Proc SQL;

SAS
Native SQL

Uses SQL syntax within the local SAS environment

SAS SQL
Pass-through

Enables you to send DBMS-specific statements to a DBMS and to retrieve DBMS data

```
proc sql;
   connect to oracle as myconn
        (user=smith password=secret path='myoracleserver');

create view salary_view as
   select * from connection to myconn
      (select empid, lastname, firstname, salary
   from employees where salary>75000);

disconnect from myconn;
quit;
```

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

A Simple "select"

```
proc sql;
   select name, sex, age from class;
quit;

- - - - - - - - - -

proc print data=class noobs;
   var name sex age;
run;
```

Result: output

Name	Sex	Age
Alfred	M	14
Alice	F	13
Barbara	F	13
Carol	F	14
Henry	M	14
James	M	12
Jane	F	12
Janet	F	15
Jeffrey	M	13
John	M	12
Joyce	F	11
Judy	F	14
Louise	F	12
Mary	F	15
Philip	M	16
Robert	M	12
Ronald	M	15
Thomas	M	11
William	M	15

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

A Simple "select"

```
proc sql;
   select * from class;
quit;

- - - - - - - - - -

proc print data=class noobs;
run;
```

Result: output

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

“select” with “where”

```
proc sql;
   select * from class
   where Sex='M';
quit;

- - - - - - - - - -

proc print data=class noobs;
   where Sex='M';
run;
```

Result: output

Name	Sex	Age	Height
Alfred	M	14	69.0
Henry	M	14	63.5
James	M	12	57.3
Jeffrey	M	13	62.5
John	M	12	59.0
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

"select" with "where" with nested "select"

```
proc sql;
   select * from class
   where height<(select
      median(height) from class);
quit;
```

Result: output

Name	Sex	Age	Height
Alice	F	13	56.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Louise	F	12	56.3
Thomas	M	11	57.5

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

“select” with “order by”

```
proc sql;
   select * from class
   order by sex, age;
quit;

- - - - - - - - - -

proc sort data=class;
   by sex age;
proc print data=class noobs;
run;
```

Result: output

Name	Sex	Age	Height
Joyce	F	11	51.3
Jane	F	12	59.8
Louise	F	12	56.3
Barbara	F	13	65.3
Alice	F	13	56.5
Carol	F	14	62.8
Judy	F	14	64.3
Mary	F	15	66.5
Janet	F	15	62.5
Thomas	M	11	57.5
James	M	12	57.3
John	M	12	59
Robert	M	12	64.8
Jeffrey	M	13	62.5
Alfred	M	14	69
Henry	M	14	63.5
William	M	15	66.5
Ronald	M	15	67
Philip	M	16	72

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

“create table”

```
proc sql;
    create table ClassCopy as
    select * from class;
quit;

- - - - - - - - - -

data ClassCopy;
    set class;
run;
```

Result: (no output)

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

"insert"

```
proc sql;
   insert into class
     (Name, Sex, Age, Age Height)
   values ('Norm', 'M', 15, 54.2);
quit;
                - - - - - - - - - -
data class;
   set class end=eof;
output;
if eof then do;
   name='Norm';  Sex='M';
   age=15;  height=54.2;
   output;
end;
run;
```

Result: (no output)

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5
Norm	M	15	54.2

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5
Norm	M	15	54.2

“delete”

```
proc sql;
   delete from class
   where name=‘Norm’;
quit;
          - - - - - - - - - -
data class;
   set class;
if name=‘Norm’ then delete;
run;
```

Result: (no output)

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

"update"

```
proc sql;
   update class
   set height=69.3
        where name='Alfred';
   set height=57.9, age=19
        where name='Alice';
quit;
                - - - - - - - - - -
data class;
   set class;
if name='Alfred' then height=69.3;
if name='Alice' then do;
   height=57.9;
   age=19;
end;
run;
```

Result: (no output)

Name	Sex	Age	Height
Alfred	M	14	69.3
Alice	F	19	57.9
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

"functions"

```
proc sql;
   select avg(age) as MeanAge
      from class;
quit;

- - - - - - - - - -

proc means data=class;
   var age;
run;
```

Result: output

13.31579

Table/Data set: class

Name	Sex	Age	Height
Alfred	M	14	69.0
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8
Henry	M	14	63.5
James	M	12	57.3
Jane	F	12	59.8
Janet	F	15	62.5
Jeffrey	M	13	62.5
John	M	12	59.0
Joyce	F	11	51.3
Judy	F	14	64.3
Louise	F	12	56.3
Mary	F	15	66.5
Philip	M	16	72.0
Robert	M	12	64.8
Ronald	M	15	67.0
Thomas	M	11	57.5
William	M	15	66.5

“create a macro variable”

```
proc sql;
   select avg(age) into :mnage
     from class;
quit;

%put Mean age is &mnage ;
```

Result: log

Mean age is 13.31579

More fun things we can do with Proc SQL

1. Create new variables
2. Group data
3. Subset grouped data
4. Conditional Processing
5. Joining Datasets (WITHOUT sorting!)

Proc SQL Syntax: Creating new variables using “as”

Dataset: Baseball_career

Name	Team	CrAtBat	CrHome	YrMajor
Buckner, Bill	Boston	8424	164	18
Speier, Chris	Chicago	6631	98	16
Pasqua, Dan	New York	428	25	2
Matthews, Gary	Chicago	6986	231	15
Orta, Jorge	Kansas City	5779	128	15
Easler, Mike	New York	3400	113	13
Smith, Ozzie	St Louis	4739	13	9
Gedman, Rich	Boston	2131	69	7
Sandberg, Ryne	Chicago	3146	74	6
McGee, Willie	St Louis	2703	32	5
Wilson, Willie	Kansas City	4908	30	11

```
proc sql;
    create table baseball_select as
    select  name,
            CrAtBat,
            YrMajor as years
    from baseball_career;
quit;
```

Dataset: Baseball_select

Name	CrAtBat	years
Buckner, Bill	8424	18
Speier, Chris	6631	16
Pasqua, Dan	428	2
Matthews, Gary	6986	15
Orta, Jorge	5779	15
Easler, Mike	3400	13
Smith, Ozzie	4739	9
Gedman, Rich	2131	7
Sandberg, Ryne	3146	6
McGee, Willie	2703	5
Randolph, Willie	5511	12
Wilson, Willie	4908	11

- only variables you pick will be in output
- you can rename variables with the “as” statement

Proc SQL Syntax: Creating new variables using “as”

Dataset: Baseball_career

Name	Team	CrAtBat	CrHome	YrMajor
Buckner, Bill	Boston	8424	164	18
Speier, Chris	Chicago	6631	98	16
Pasqua, Dan	New York	428	25	2
Matthews, Gary	Chicago	6986	231	15
Orta, Jorge	Kansas City	5779	128	15
Easler, Mike	New York	3400	113	13
Smith, Ozzie	St Louis	4739	13	9
Gedman, Rich	Boston	2131	69	7
Sandberg, Ryne	Chicago	3146	74	6
McGee, Willie	St Louis	2703	32	5
Wilson, Willie	Kansas City	4908	30	11

```
proc sql;
create table newvar as
    select name,
        (CrAtBat/YrMajor) as avg_atbat,
        (CrAtBat/YrMajor)/162 as atbatpergame
    from baseball_career;
quit;
```

-or-

```
proc sql;
    select name,
        (CrAtBat/YrMajor) as avg_atbat,
        CALCULATED avg_atbat/162 as atbatpergame
    from baseball_career;
quit;
```

Dataset: newvar

Name	avg_atbat
Buckner, Bill	468.000
Speier, Chris	414.438
Pasqua, Dan	214.000
Matthews, Gary	465.733
Orta, Jorge	385.267
Easler, Mike	261.538
Smith, Ozzie	526.556
Gedman, Rich	304.429
Sandberg, Ryne	524.333
McGee, Willie	540.600
Randolph, Willie	459.250
Wilson, Willie	446.182

Proc SQL Syntax: Grouping Data

Dataset: Baseball_career

Name	Team	CrAtBat	CrHome	YrMajor
Buckner, Bill	Boston	8424	164	18
Speier, Chris	Chicago	6631	98	16
Pasqua, Dan	New York	428	25	2
Matthews, Gary	Chicago	6986	231	15
Orta, Jorge	Kansas City	5779	128	15
Easler, Mike	New York	3400	113	13
Smith, Ozzie	St Louis	4739	13	9
Gedman, Rich	Boston	2131	69	7
Sandberg, Ryne	Chicago	3146	74	6
McGee, Willie	St Louis	2703	32	5
Wilson, Willie	Kansas City	4908	30	11

```
proc sql;
    select team,
        sum(crHome) as tot_HR
    from baseball_career
    group by team;
quit;
```

Result: (no output)

Team	tot_HR
Boston	233
Chicago	403
Kansas City	158
New York	177
St Louis	45

Proc SQL Syntax: Grouping Data

Dataset: Baseball_career

Name	Team	CrAtBat	CrHome	YrMajor
Buckner, Bill	Boston	8424	164	18
Speier, Chris	Chicago	6631	98	16
Pasqua, Dan	New York	428	25	2
Matthews, Gary	Chicago	6986	231	15
Orta, Jorge	Kansas City	5779	128	15
Easler, Mike	New York	3400	113	13
Smith, Ozzie	St Louis	4739	13	9
Gedman, Rich	Boston	2131	69	7
Sandberg, Ryne	Chicago	3146	74	6
McGee, Willie	St Louis	2703	32	5
Wilson, Willie	Kansas City	4908	30	11

```
proc sql;
    select team,
        sum(crHome) as tot_HR
    from baseball_career
    group by team
    having calculated tot_HR >200;
quit;
```

Result: (no output)

Team	tot_HR
Boston	233
Chicago	403

Proc SQL Syntax: Grouping Data

Dataset: Baseball_career

Name	Team	CrAtBat	CrHome	YrMajor
Buckner, Bill	Boston	8424	164	18
Speier, Chris	Chicago	6631	98	16
Pasqua, Dan	New York	428	25	2
Matthews, Gary	Chicago	6986	231	15
Orta, Jorge	Kansas City	5779	128	15
Easler, Mike	New York	3400	113	13
Smith, Ozzie	St Louis	4739	13	9
Gedman, Rich	Boston	2131	69	7
Sandberg, Ryne	Chicago	3146	74	6
McGee, Willie	St Louis	2703	32	5
Wilson, Willie	Kansas City	4908	30	11

```
proc sql;
    select *
    from baseball_career
    where upcase(team) like '%ST LOUIS%';
quit;
```

Result: (no output)

Name	Team	CrAtBat	CrHome	YrMajor
Smith, Ozzie	St Louis	4739	13	9
McGee, Willie	St Louis	2703	32	5

Proc SQL Syntax: Conditional Processing

Dataset: Baseball_career

Name	Team	CrAtBat	CrHome	YrMajor
Buckner, Bill	Boston	8424	164	18
Speier, Chris	Chicago	6631	98	16
Pasqua, Dan	New York	428	25	2
Matthews, Gary	Chicago	6986	231	15
Orta, Jorge	Kansas City	5779	128	15
Easler, Mike	New York	3400	113	13
Smith, Ozzie	St Louis	4739	13	9
Gedman, Rich	Boston	2131	69	7
Sandberg, Ryne	Chicago	3146	74	6
McGee, Willie	St Louis	2703	32	5
Wilson, Willie	Kansas City	4908	30	11

```
proc sql;
    create table baseball_HR as
    select name, crhome,
    case
        when crhome <= 50 then 'Low'
        when crhome <= 150 then 'Avg'
        else 'Slugger'
    end as HRcat
    from baseball_career;
quit;
```

Dataset: baseball_HR

Name	CrHome	HRcat
Buckner, Bill	164	Slugger
Speier, Chris	98	Avg
Pasqua, Dan	25	Low
Matthews, Gary	231	Slugger
Orta, Jorge	128	Avg
Easler, Mike	113	Avg
Smith, Ozzie	13	Low
Gedman, Rich	69	Avg
Sandberg, Ryne	74	Avg
McGee, Willie	32	Low
Randolph, Willie	39	Low
Wilson, Willie	30	Low

Proc SQL Syntax: Joining Datasets

Dataset: Baseball_career

Name	Team	CrAtBat	CrHome	YrMajor
Buckner, Bill	Boston	8424	164	18
Speier, Chris	Chicago	6631	98	16
Pasqua, Dan	New York	428	25	2
Matthews, Gary	Chicago	6986	231	15
Orta, Jorge	Kansas City	5779	128	15
Easler, Mike	New York	3400	113	13
Smith, Ozzie	St Louis	4739	13	9
Gedman, Rich	Boston	2131	69	7
Sandberg, Ryne	Chicago	3146	74	6
McGee, Willie	St Louis	2703	32	5
Wilson, Willie	Kansas City	4908	30	11

Dataset: Baseball_1986

Name	nAtBat	nHome	Position
Buckner, Bill	629	18	1B
Speier, Chris	155	6	3S
Pasqua, Dan	280	16	LF
Matthews, Gary	370	21	LF
Orta, Jorge	336	9	DH
Smith, Ozzie	514	0	SS
Gedman, Rich	462	16	C
Sandberg, Ryne	627	14	2B
McGee, Willie	497	7	CF
Randolph, Willie	492	5	2B

- No need for sorting

Proc SQL Syntax: Joining Datasets

Inner Joins

- Uses “where”
- keeps common observations based on keyword

```
proc sql;
    create table innerball as
    select a.name, a.position,
           b.team, b.yrmajor
    from  baseball_1986 a, baseball_career b
    where a.name = b.name;
quit;
```

Dataset = innerball

Obs	Name	Position	Team	YrMajor
1	Buckner, Bill	1B	Boston	18
2	Speier, Chris	3S	Chicago	16
3	Pasqua, Dan	LF	New York	2
4	Matthews, Gary	LF	Chicago	15
5	Orta, Jorge	DH	Kansas City	15
6	Smith, Ozzie	SS	St Louis	9
7	Gedman, Rich	C	Boston	7
8	Sandberg, Ryne	2B	Chicago	6
9	McGee, Willie	CF	St Louis	5

Proc SQL Syntax: Joining Datasets

Outer Joins

- Uses “on”
- keeps all observations from both datasets

```
proc sql;
    create table outerball as
    select COALESCE (a.name,b.name) as name,
           a.position,
           b.team,
           b.YrMajor
    from baseball_1986 a full join baseball_career b
    on a.name = b.name;
quit;
```

Dataset = outerball

Obs	name	Position	Team	YrMajor
1	Buckner, Bill	1B	Boston	18
2	Easler, Mike		New York	13
3	Gedman, Rich	C	Boston	7
4	Matthews, Gary	LF	Chicago	15
5	McGee, Willie	CF	St Louis	5
6	Orta, Jorge	DH	Kansas City	15
7	Pasqua, Dan	LF	New York	2
8	Randolph, Willie	2B		.
9	Sandberg, Ryne	2B	Chicago	6
10	Smith, Ozzie	SS	St Louis	9
11	Speier, Chris	3S	Chicago	16
12	Wilson, Willie		Kansas City	11

Proc SQL Syntax: Joining Datasets

Left/Right Joins

- Uses “on”
- keeps observations in “left” dataset

```
proc sql;
    create table leftball as
    select a.name, a.team,
    b.position
    from baseball_career a left join baseball_1986 b
    on a.name = b.name
    order by team;
quit;
```

Dataset = leftball

Obs	Name	Team	Position
1	Gedman, Rich	Boston	C
2	Buckner, Bill	Boston	1B
3	Matthews, Gary	Chicago	LF
4	Speier, Chris	Chicago	3S
5	Sandberg, Ryne	Chicago	2B
6	Orta, Jorge	Kansas City	DH
7	Wilson, Willie	Kansas City	
8	Pasqua, Dan	New York	LF
9	Easler, Mike	New York	
10	McGee, Willie	St Louis	CF
11	Smith, Ozzie	St Louis	SS

PROC SQL and the Cloud

- Canvas Data
  - Star schema database in the Amazon Cloud
  - Fact and dimension tables
  - Page view table > 1 billion rows

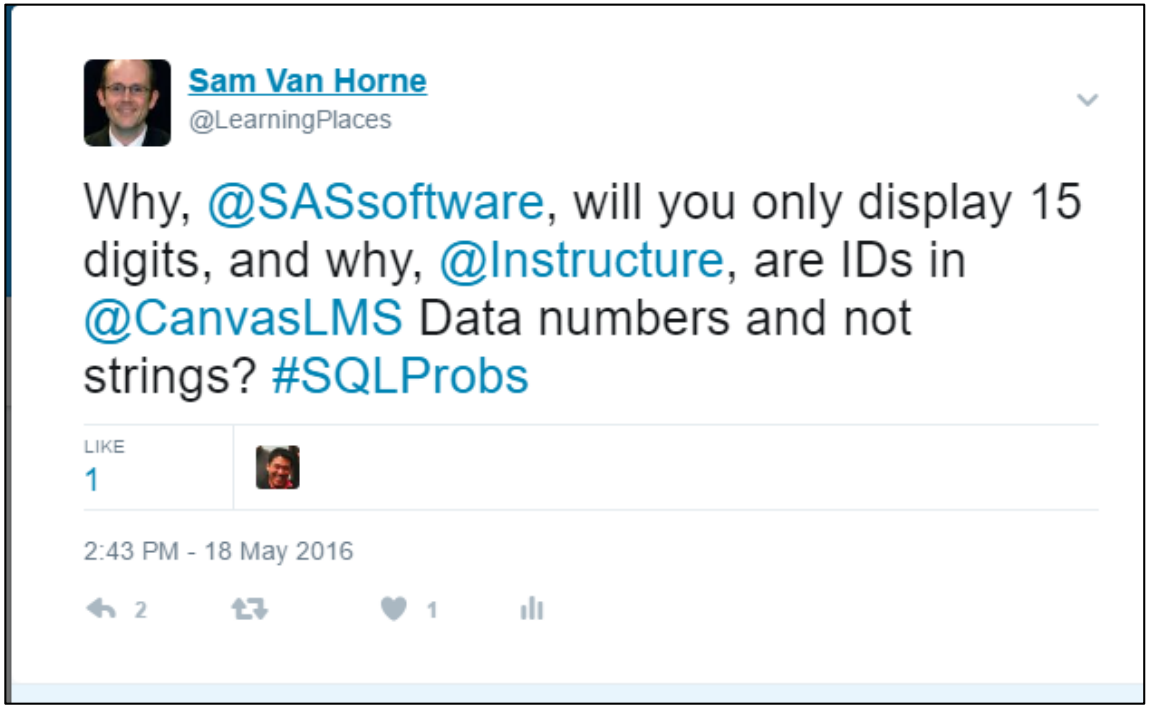
Practical Need for PROC SQL

- Data Reports for Instructors
  - What are the distributions of scores?
  - Which students are participating in discussion forums?

Using the Unizin Data Platform

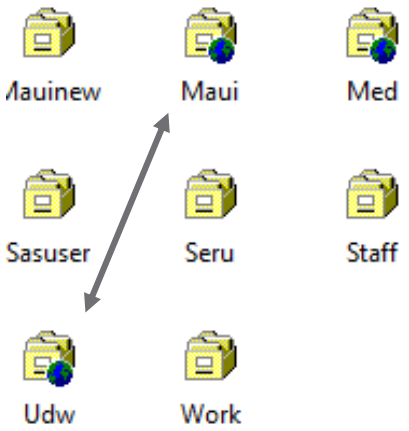
- The Office of Teaching, Learning and Technology configured connections to UDP via SAS
  - SAS provides robust data management tools
  - SAS provides a variety of statistical analyses and tools for creating reports
  - Very simple to use ODBC connection to connect to Redshift
  - SQL in SAS can be read by DBAs or other application developers
- Some downsides of SAS…
  - Not able to process Requests table for many courses at once
  - The Big Integer is Pythonic
  - Not commonly used by developers

Overcoming some hurdles…



Reporting the Outcomes of Mandatory Quiz

- The Department of Art and Art History requires students to take a mandatory quiz about safety.
- An Auditing and Compliance Department must collect identities of students who have passed the quiz.
- We connect to UDP and our SIS in same programming steps.



Efficiency of assignment_dim

- **Ability to find assignments by their title**
- **Effective because submission_dim does not have course_id values**

```
PROC SQL;
create table ELEMENTS_quiz as
SELECT A.TITLE, A.ID AS
ASSIGNMENT_ID,
(dbsastype = (id='char(20)'
course_id = 'char(20)')) A
INNER JOIN WORK.ELEMENTS B
ON A.COURSE_ID=B.ID
WHERE title = 'SAAH Quiz';
QUIT;
```

Report for College of Medicine

Goal: Report of scores on all major assignments for students in the medical school

1. Pull all courses for the Med School from Account_dim and inner join with Course_dim
   - Where course_dim.workflow_state = 'available'
   - Enrollment_term_id is current semester
2. Pull all exams from assignment_dim table

```
proc sql noprint;
create table ASSIGNMENTS as select *, id as
assignment_id label = "assignment_id",
course_id  from u.ASSIGNMENT_DIM
where course_id in (select course_id from
MED_V2) and prxmatch('m/exam/oi', title) > 0
and workflow_state = 'published';
quit;
```

Account_dim
- Parent_account_id
- Course_id

Course_dim
- Course_id
- Enrollment_term_id
- Workflow_state

Assignment_dim
- Course_id
- Assignment_id
- Title

Submission_dim
- Assignment_id
- User_id
- Grade
- Workflow_state

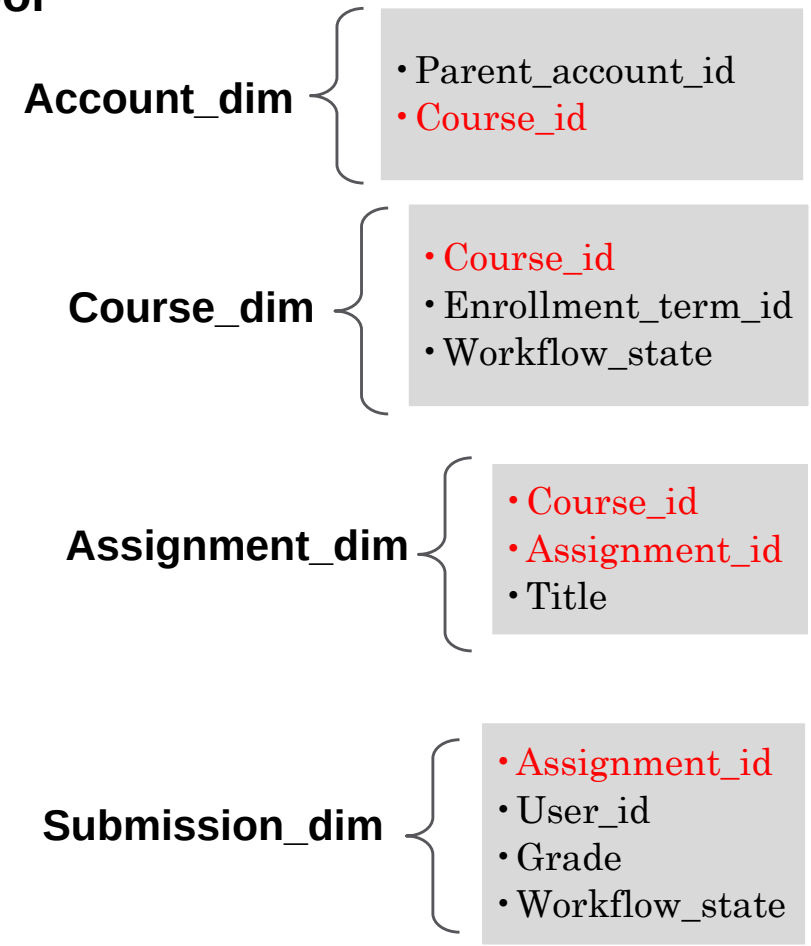
Report for College of Medicine

Goal: Report of scores on all major assignments for students in the medical school

3. Inner join assignment_dim with submission_dim table
   - Workflow_state = 'graded'
   - User_id ^= " "
   - Remove % signs using TRANWRD function

4. Inner join with pseudonym_dim table to get SIS_user_id

```
select TRANWRD(grade, %, " ")
```

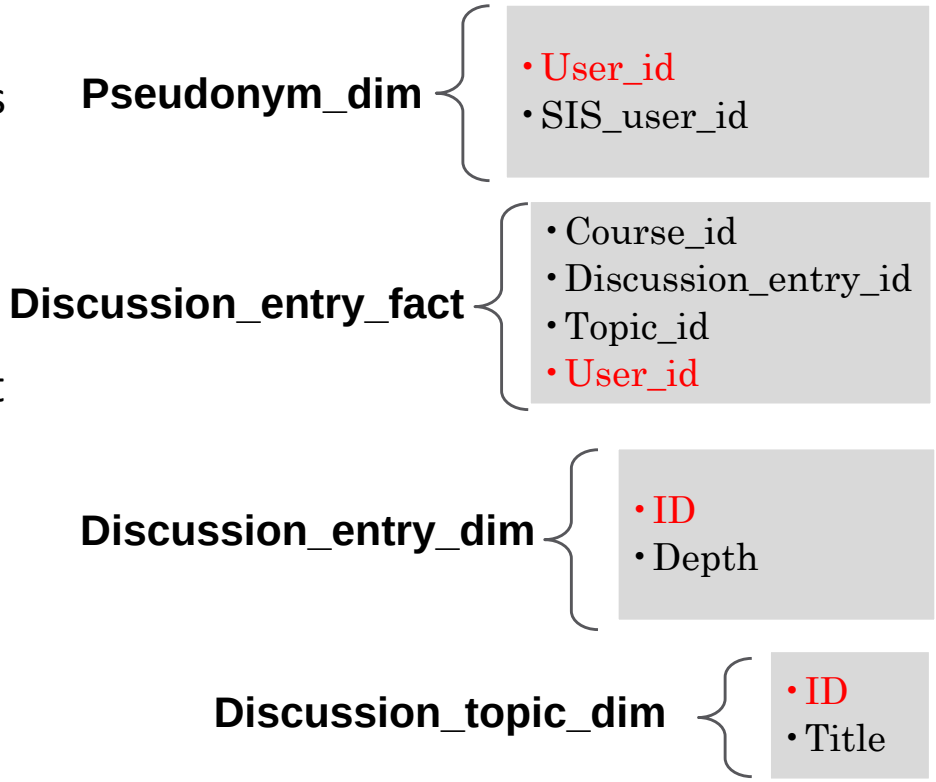


Preparation of a Discussion Report

Goal: Report of discussion posts, replies, and 'views' for each student

Getting Discussion Posts and Replies

1. Match user_id to university_id using Pseudonym_dim table
2. Pull Discussion_entry_fact by course_id
3. Inner join Discussion_entry_fact with Discussion_entry_dim
   - **If depth > 1 output to Replies**
   - **If depth = 1 output to Posts**
4. Sort by User_id
   - Count overall user posts & replies
5. Inner join with Discussion_topic_dim
6. Sort by User_id & Topic_id
   - Count topic posts and replies by user



Preparation of a Discussion Report

Goal: Report of discussion posts, replies, and 'views' for each student

Note: Views were defined as rows in the requests table related to discussions, as long as the subsequent rows were for different topic_ids or > 15 minutes apart

1. Pull from requests by course_id

2. Extract discussion_id from URL

3. Sort by user_id, topic_id_abbr, timestamp

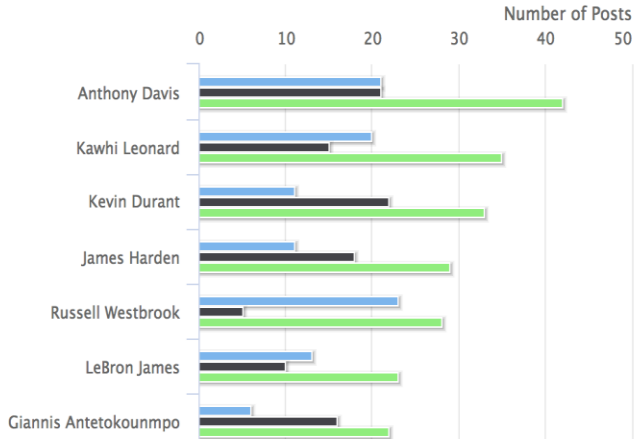
4. Count views by user_id with conditional statements and lag function

```
proc sql noprint;
    create table requests2_&course_number as
    select *, input(substr(url, 33, 6), 12.)
    as topic_id_abbr
    label = 'topic_id_abbr'
    from requests_&course_number;
quit;
```

```
timestamp2 = timestamp - (lag(timestamp));
    if first.sis_user_id then posts_read = 1;
    if sis_user_id = lag(sis_user_id) and
    topic_id_abbr = lag(topic_id_abbr) and timestamp2 < 900
    then posts_read = posts_read + 0 ;
```

Discussion Post Activity

*Data in these reports may be up to two days old



Name	University Id	Posts	Replies	Total
Anthony Davis		21	21	42
Kawhi Leonard		15	20	35
Kevin Durant		22	11	33
James Harden		18	11	29
Russell Westbrook		5	23	28
LeBron James		10	13	23
Giannis Antetokounmpo		16	6	22

sis_user_id	VIEWS	REPLIES	POSTED	user_id	STUDENT_FIRST_NAME	STUDENT_LAST_NAME
00000001	48	21	14	-2999999999999	Prince	Hamlet
00000002	16	19	10	88888888888888	King	Lear
00000003	23	10	10	-777777777777	Lord	Montague
00000004	37	13	12	-5555555555555	Lord	Capulet
00000005	68	26	15	4444444444444	Duke	Albany
00000006	16	11	13	-3333333333333	Duke	Cornwall