James Valles
CSC 453
Homework 7

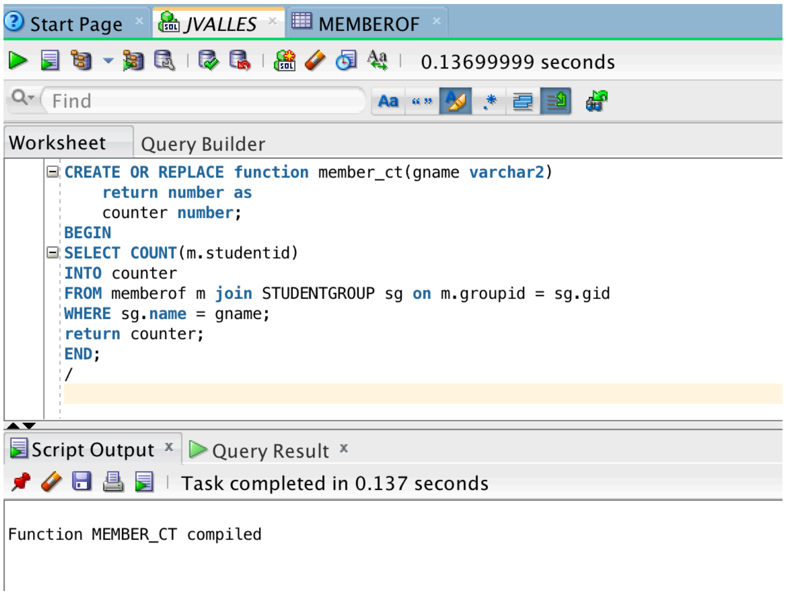
1. Reading done.

2.

a.

Write function takes argument name of student group and returns number of members in group

```
CREATE OR REPLACE function member_ct(gname varchar2)
   return number as
   counter number;
BEGIN
SELECT COUNT(m.studentid)
INTO counter
FROM memberof m join STUDENTGROUP sg on m.groupid = sg.gid
WHERE sg.name = gname;
return counter;
END;
/
```

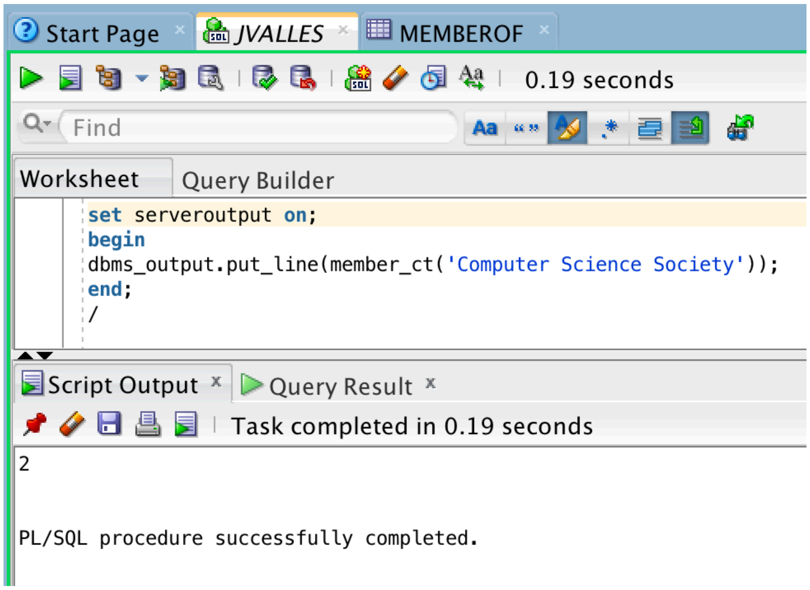


```
CREATE OR REPLACE function member_ct(gname varchar2)
    return number as
    counter number;
BEGIN
SELECT COUNT(m.studentid)
INTO counter
FROM memberof m join STUDENTGROUP sg on m.groupid = sg.gid
WHERE sg.name = gname;
return counter;
END;
/
```

Task completed in 0.137 seconds

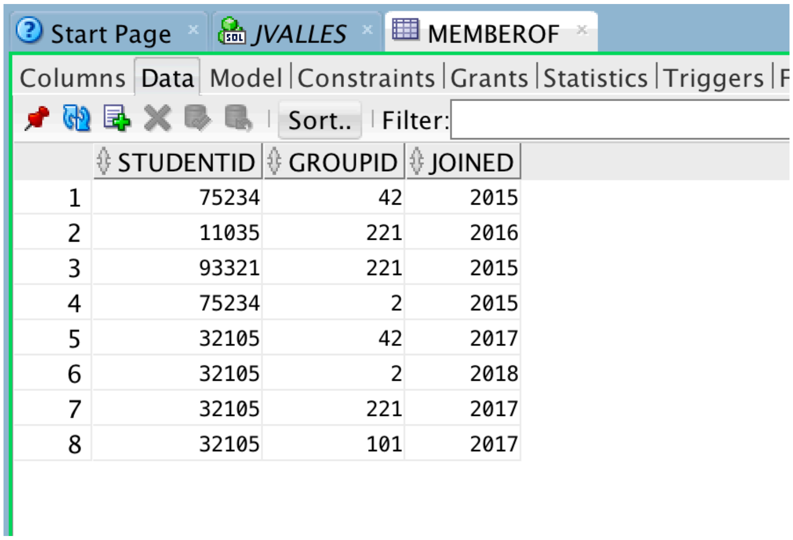
```
Function MEMBER_CT compiled
```

Test: Checking number of members in ‘Computer Science Society’ returns two members as expected.



```
set serveroutput on;
begin
dbms_output.put_line(member_ct('Computer Science Society'));
end;
/
```

```
2


PL/SQL procedure successfully completed.
```



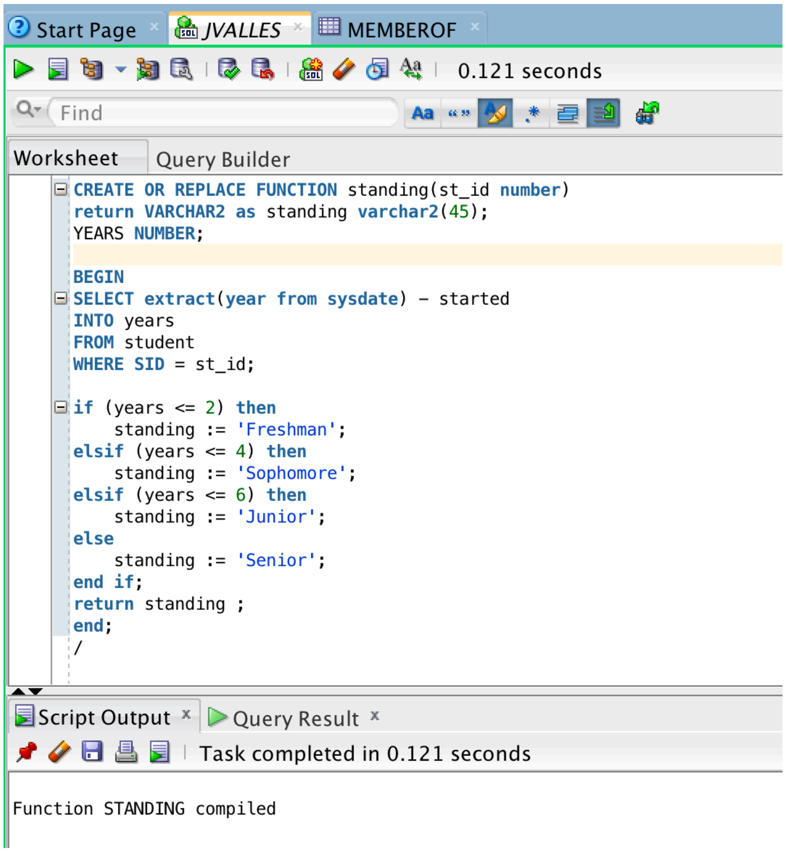
	STUDENTID	GROUPID	JOINED
1	75234	42	2015
2	11035	221	2016
3	93321	221	2015
4	75234	2	2015
5	32105	42	2017
6	32105	2	2018
7	32105	221	2017
8	32105	101	2017

b.

Write function that takes student id as argument and returns student standing

```
CREATE OR REPLACE FUNCTION standing(st_id number)
return VARCHAR2 as standing varchar2(45);
YEARS NUMBER;

BEGIN
SELECT extract(year from sysdate) - started
INTO years
FROM student
WHERE SID = st_id;

if (years <= 2) then
   standing := 'Freshman';
elsif (years <= 4) then
   standing := 'Sophomore';
elsif (years <= 6) then
```

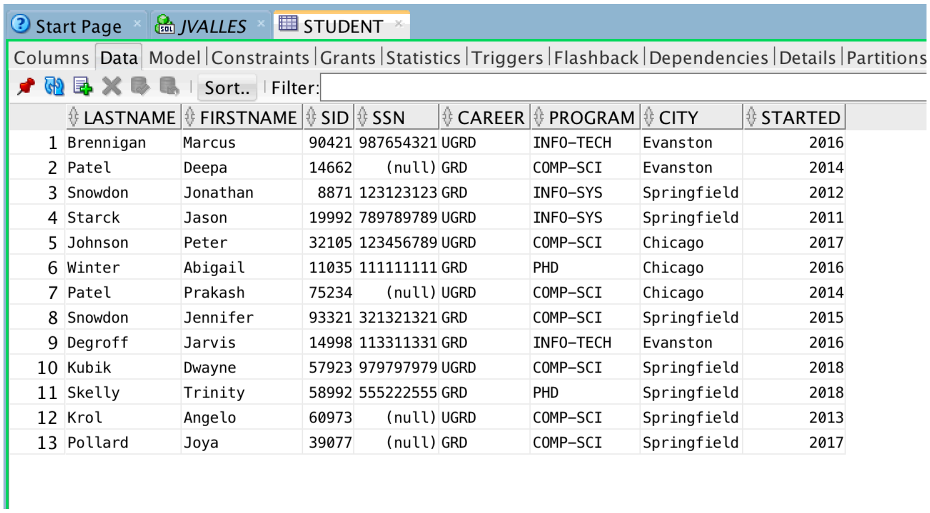
```
   standing := 'Junior';
else
   standing := 'Senior';
end if;
return standing ;
end;
/
```



```
CREATE OR REPLACE FUNCTION standing(st_id number)
return VARCHAR2 as standing varchar2(45);
YEARS NUMBER;

BEGIN
SELECT extract(year from sysdate) - started
INTO years
FROM student
WHERE SID = st_id;

if (years <= 2) then
    standing := 'Freshman';
elsif (years <= 4) then
    standing := 'Sophomore';
elsif (years <= 6) then
    standing := 'Junior';
else
    standing := 'Senior';
end if;
return standing ;
end;
/
```

```
Function STANDING compiled
```

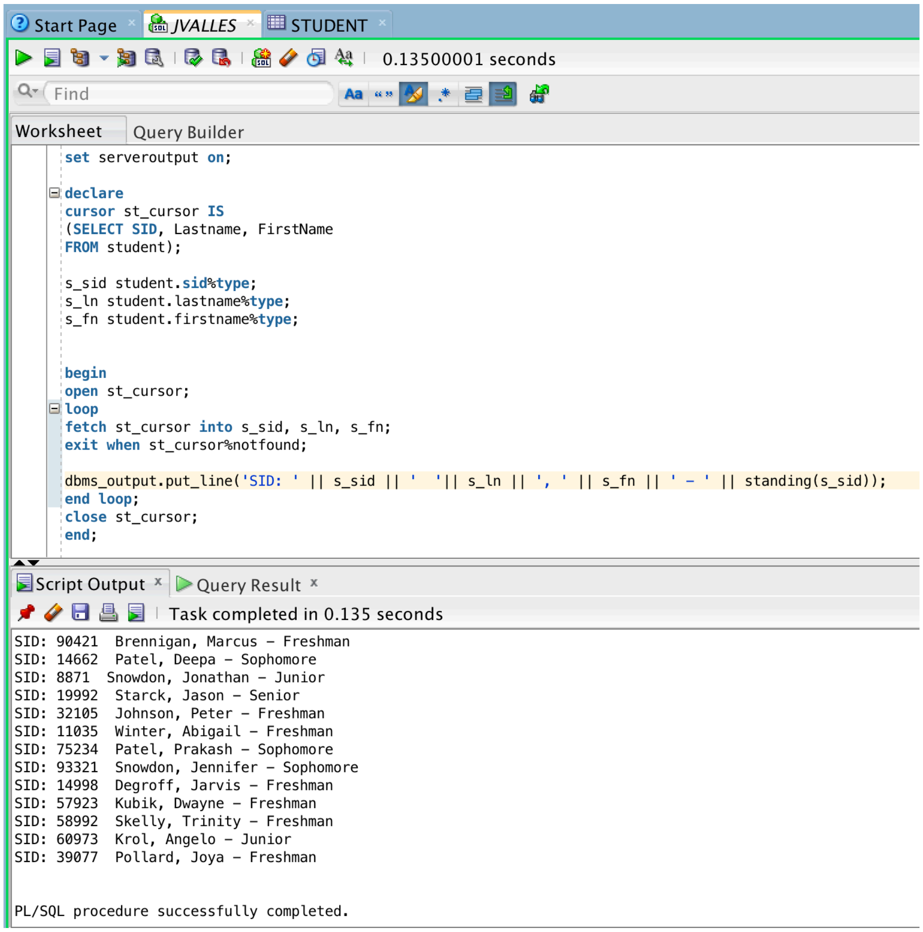
Wrote cursor to test and list all students (SID, Last Name, First Name, Standing)



	LASTNAME	FIRSTNAME	SID	SSN	CAREER	PROGRAM	CITY	STARTED
1	Brennigan	Marcus	90421	987654321	UGRD	INFO-TECH	Evanston	2016
2	Patel	Deepa	14662	(null)	GRD	COMP-SCI	Evanston	2014
3	Snowdon	Jonathan	8871	123123123	GRD	INFO-SYS	Springfield	2012
4	Starck	Jason	19992	789789789	UGRD	INFO-SYS	Springfield	2011
5	Johnson	Peter	32105	123456789	UGRD	COMP-SCI	Chicago	2017
6	Winter	Abigail	11035	111111111	GRD	PHD	Chicago	2016
7	Patel	Prakash	75234	(null)	UGRD	COMP-SCI	Chicago	2014
8	Snowdon	Jennifer	93321	321321321	GRD	COMP-SCI	Springfield	2015
9	Degroff	Jarvis	14998	113311331	GRD	INFO-TECH	Evanston	2016
10	Kubik	Dwayne	57923	979797979	UGRD	COMP-SCI	Springfield	2018
11	Skelly	Trinity	58992	555222555	GRD	PHD	Springfield	2018
12	Krol	Angelo	60973	(null)	UGRD	COMP-SCI	Springfield	2013
13	Pollard	Joya	39077	(null)	GRD	COMP-SCI	Springfield	2017

Results expected: Brennigan, Marcus start 2016 : Freshman; Patel, Deepa start 2014 Sophomore; Snowdon Jonathan start 2012: Junior; Starck, Jason start 2011 Senior

Test cursor:



```
set serveroutput on;

declare
cursor st_cursor IS
(SELECT SID, Lastname, FirstName
FROM student);

s_sid student.sid%type;
s_ln student.lastname%type;
s_fn student.firstname%type;


begin
open st_cursor;
loop
fetch st_cursor into s_sid, s_ln, s_fn;
exit when st_cursor%notfound;

dbms_output.put_line('SID: ' || s_sid || '  '|| s_ln || ', ' || s_fn || ' - ' || standing(s_sid));
end loop;
close st_cursor;
end;
```

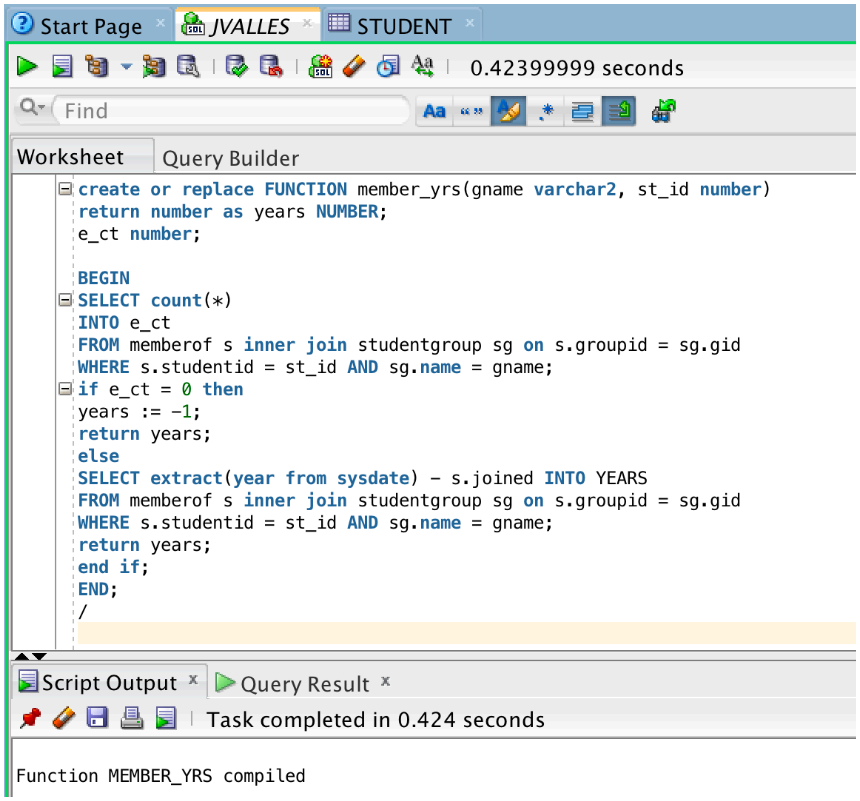
```
SID: 90421  Brennigan, Marcus - Freshman
SID: 14662  Patel, Deepa - Sophomore
SID: 8871  Snowdon, Jonathan - Junior
SID: 19992  Starck, Jason - Senior
SID: 32105  Johnson, Peter - Freshman
SID: 11035  Winter, Abigail - Freshman
SID: 75234  Patel, Prakash - Sophomore
SID: 93321  Snowdon, Jennifer - Sophomore
SID: 14998  Degroff, Jarvis - Freshman
SID: 57923  Kubik, Dwayne - Freshman
SID: 58992  Skelly, Trinity - Freshman
SID: 60973  Krol, Angelo - Junior
SID: 39077  Pollard, Joya - Freshman


PL/SQL procedure successfully completed.
```

c.

Write function taking group name and student id as arguments and returning years student has been a member, return -1 if student not member of group

```
create or replace FUNCTION member_yrs(gname varchar2, st_id number)
return number as years NUMBER;
e_ct number;

BEGIN
SELECT count(*)
INTO e_ct
FROM memberof s inner join studentgroup sg on s.groupid = sg.gid
WHERE s.studentid = st_id AND sg.name = gname;
if e_ct = 0 then
years := -1;
return years;
else
SELECT extract(year from sysdate) - s.joined INTO YEARS
FROM memberof s inner join studentgroup sg on s.groupid = sg.gid
WHERE s.studentid = st_id AND sg.name = gname;
return years;
```

```
end if;
END;
/
```

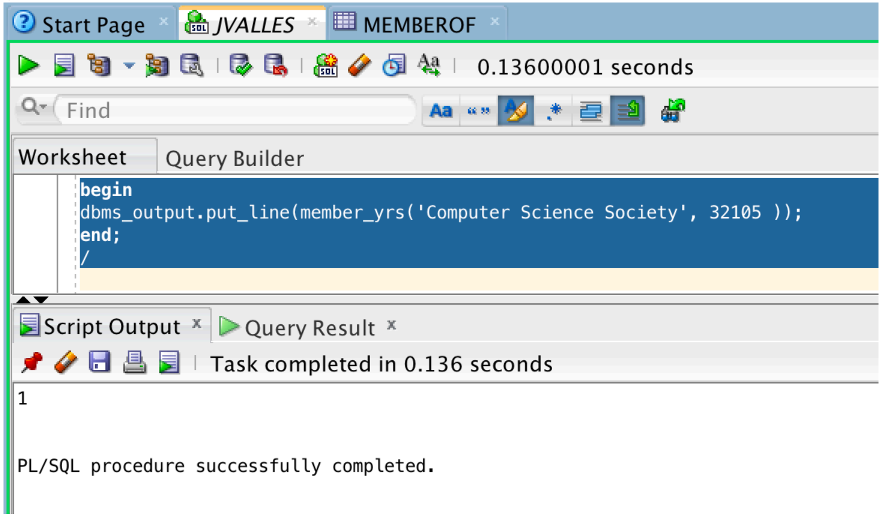


```
create or replace FUNCTION member_yrs(gname varchar2, st_id number)
return number as years NUMBER;
e_ct number;

BEGIN
SELECT count(*)
INTO e_ct
FROM memberof s inner join studentgroup sg on s.groupid = sg.gid
WHERE s.studentid = st_id AND sg.name = gname;
if e_ct = 0 then
years := -1;
return years;
else
SELECT extract(year from sysdate) - s.joined INTO YEARS
FROM memberof s inner join studentgroup sg on s.groupid = sg.gid
WHERE s.studentid = st_id AND sg.name = gname;
return years;
end if;
END;
/
```

```
Function MEMBER_YRS compiled
```

Test: 1

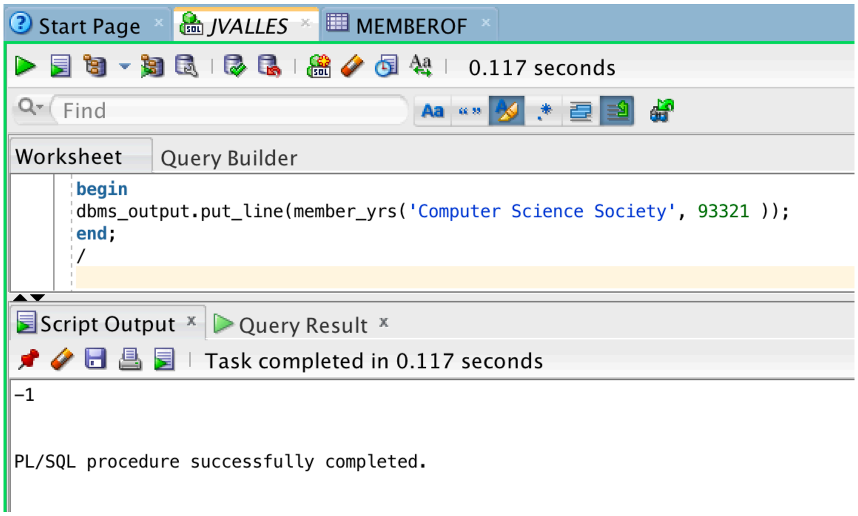
Student 32105 – Peter Johnson Joined ‘Computer Science Society’ in 2017, so should be 1 year.



```
begin
dbms_output.put_line(member_yrs('Computer Science Society', 32105 ));
end;
/
```

```
1


PL/SQL procedure successfully completed.
```

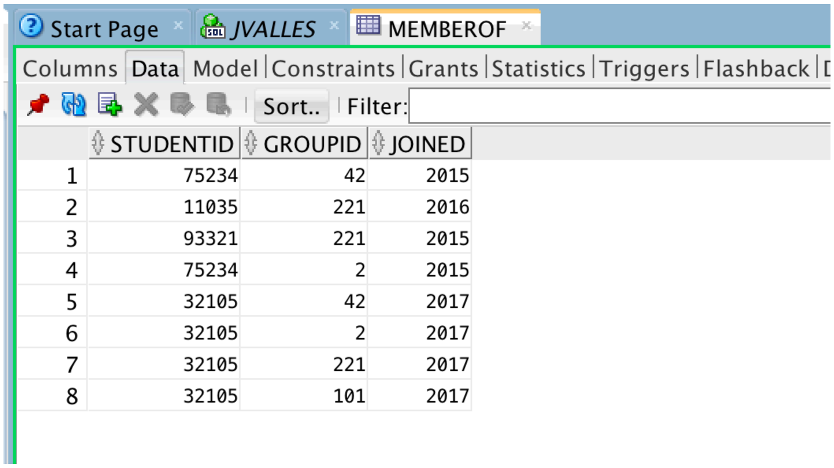
Test 2: Student 93321 – Jennifer Snowdon not in ‘Computer Science Society’ ‘ so should be - 1



```
begin
dbms_output.put_line(member_yrs('Computer Science Society', 93321 ));
end;
/
```

Task completed in 0.117 seconds

```
-1


PL/SQL procedure successfully completed.
```



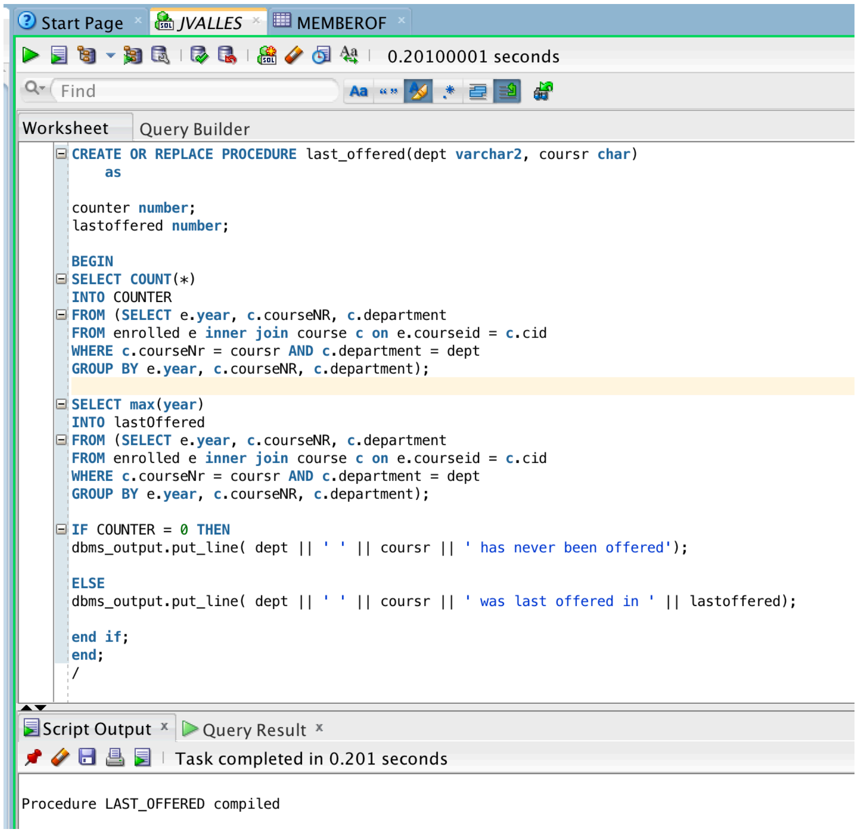
	STUDENTID	GROUPID	JOINED
1	75234	42	2015
2	11035	221	2016
3	93321	221	2015
4	75234	2	2015
5	32105	42	2017
6	32105	2	2017
7	32105	221	2017
8	32105	101	2017

3

a.

Write procedure that takes department and coursenr of a class and prints year class was last offered. If class not offered, print it has never been offered.

```
CREATE OR REPLACE PROCEDURE last_offered(dept varchar2, coursr char)
   as

counter number;
lastoffered number;


BEGIN
SELECT COUNT(*)
INTO COUNTER
FROM (SELECT e.year, c.courseNR, c.department
FROM enrolled e inner join course c on e.courseid = c.cid
WHERE c.courseNr = coursr AND c.department = dept
GROUP BY e.year, c.courseNR, c.department);


SELECT max(year)
INTO lastOffered
FROM (SELECT e.year, c.courseNR, c.department
FROM enrolled e inner join course c on e.courseid = c.cid
WHERE c.courseNr = coursr AND c.department = dept
GROUP BY e.year, c.courseNR, c.department);

IF COUNTER = 0 THEN
dbms_output.put_line( dept || ' ' || coursr || ' has never been offered');

ELSE
dbms_output.put_line( dept || ' ' || coursr || ' was last offered in ' || lastoffered);

end if;
end;
/
```

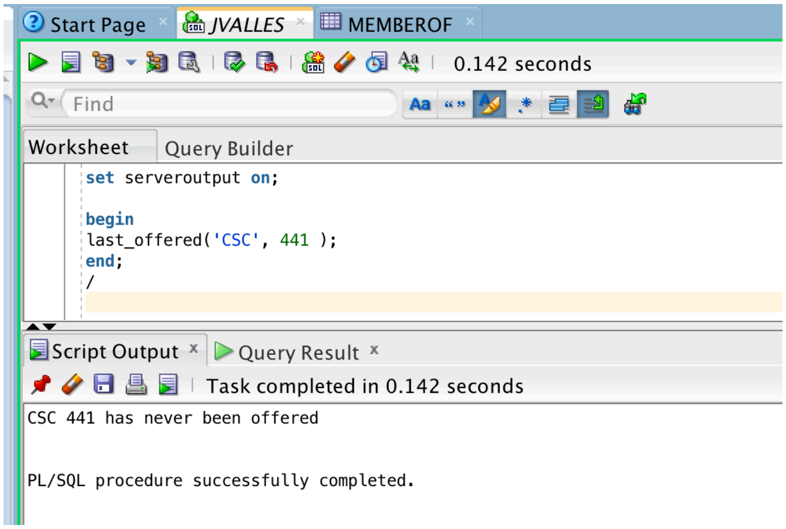


```
CREATE OR REPLACE PROCEDURE last_offered(dept varchar2, coursr char)
    as

counter number;
lastoffered number;

BEGIN
SELECT COUNT(*)
INTO COUNTER
FROM (SELECT e.year, c.courseNR, c.department
FROM enrolled e inner join course c on e.courseid = c.cid
WHERE c.courseNr = coursr AND c.department = dept
GROUP BY e.year, c.courseNR, c.department);

SELECT max(year)
INTO lastOffered
FROM (SELECT e.year, c.courseNR, c.department
FROM enrolled e inner join course c on e.courseid = c.cid
WHERE c.courseNr = coursr AND c.department = dept
GROUP BY e.year, c.courseNR, c.department);

IF COUNTER = 0 THEN
dbms_output.put_line( dept || ' ' || coursr || ' has never been offered');

ELSE
dbms_output.put_line( dept || ' ' || coursr || ' was last offered in ' || lastoffered);

end if;
end;
/
```

```
Task completed in 0.201 seconds

Procedure LAST_OFFERED compiled
```

Test #1:

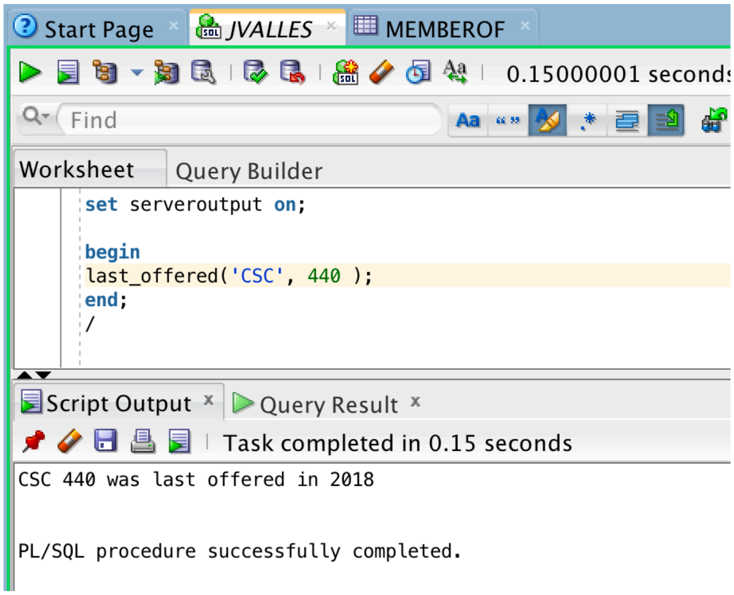
CSC 441 doesn't exist, expected result was never offered



```
set serveroutput on;

begin
last_offered('CSC', 441 );
end;
/
```

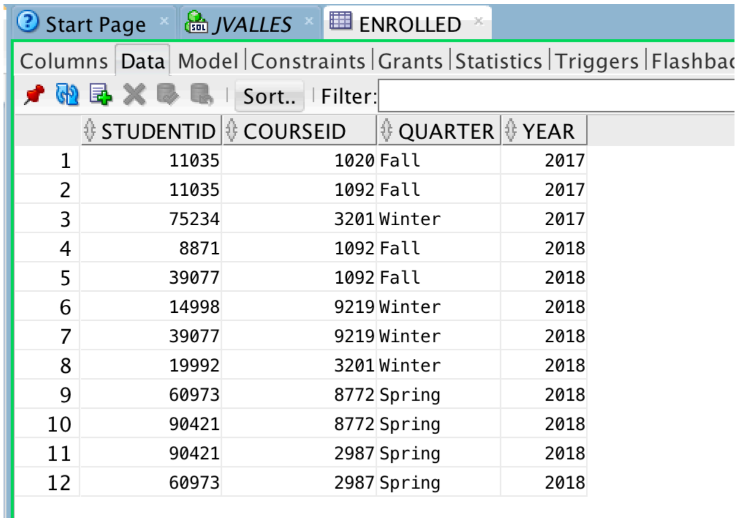
```
CSC 441 has never been offered


PL/SQL procedure successfully completed.
```

Test #2: CSC 440 was last offered in 2018 .



```
set serveroutput on;

begin
last_offered('CSC', 440 );
end;
/
```

```
CSC 440 was last offered in 2018


PL/SQL procedure successfully completed.
```

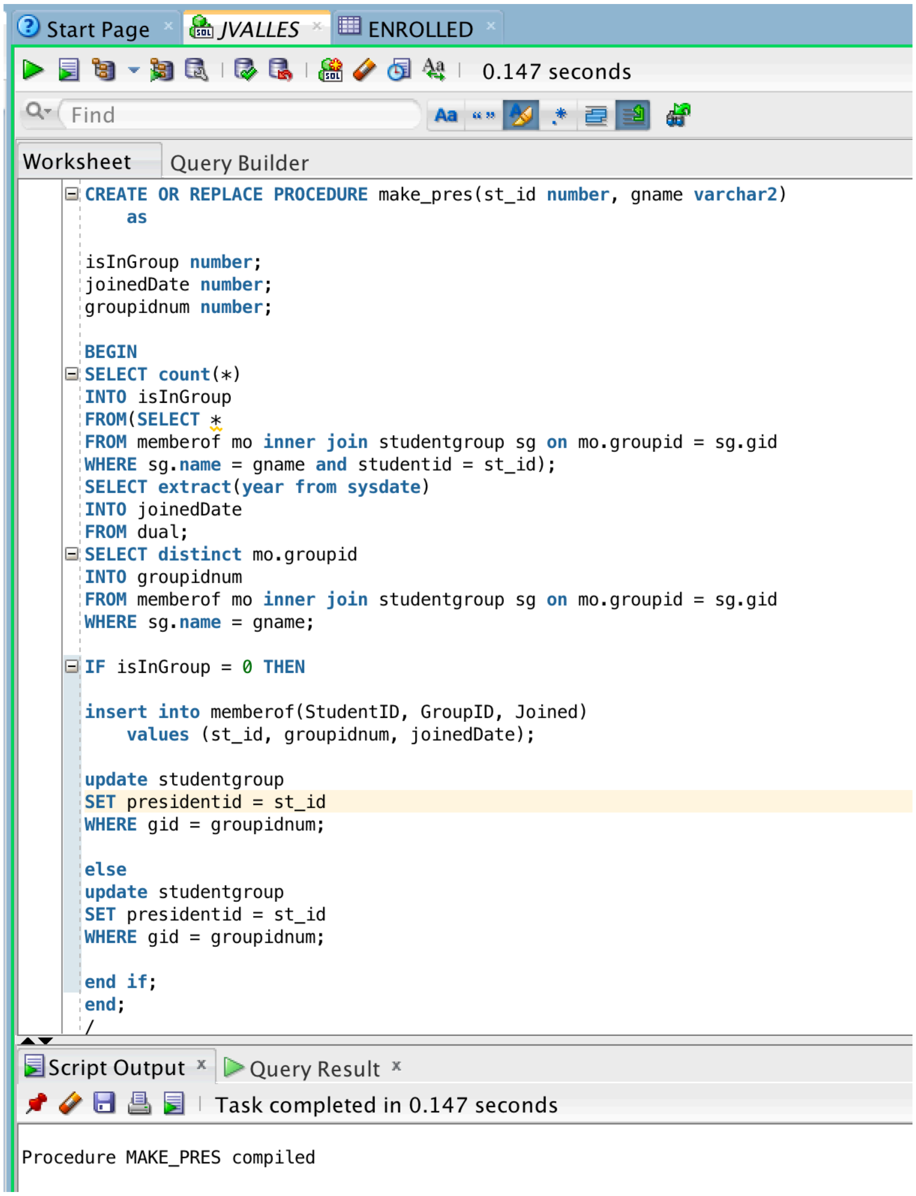


	STUDENTID	COURSEID	QUARTER	YEAR
1	11035	1020	Fall	2017
2	11035	1092	Fall	2017
3	75234	3201	Winter	2017
4	8871	1092	Fall	2018
5	39077	1092	Fall	2018
6	14998	9219	Winter	2018
7	39077	9219	Winter	2018
8	19992	3201	Winter	2018
9	60973	8772	Spring	2018
10	90421	8772	Spring	2018
11	90421	2987	Spring	2018
12	60973	2987	Spring	2018

b.

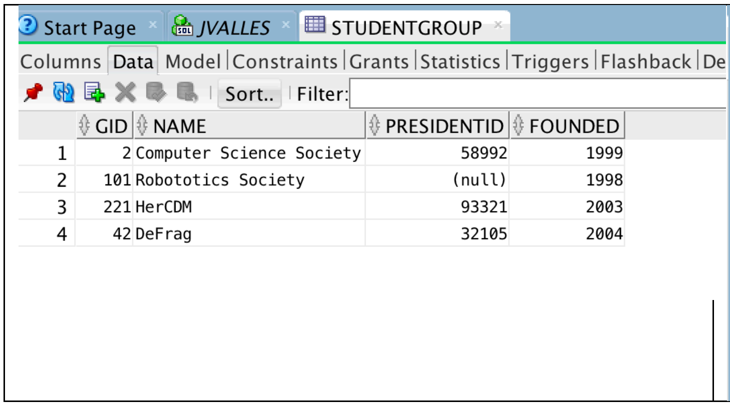
Write procedure takes student id and name of student group and makes student new President of group. If student not a member, make student member of group set joined to current year (2018) and then make student new president of group.

```
CREATE OR REPLACE PROCEDURE make_pres(st_id number, gname varchar2)
   as

isInGroup number;
joinedDate number;
groupidnum number;

BEGIN
SELECT count(*)
INTO isInGroup
FROM(SELECT *
FROM memberof mo inner join studentgroup sg on mo.groupid = sg.gid
WHERE sg.name = gname and studentid = st_id);
SELECT extract(year from sysdate)
INTO joinedDate
FROM dual;
SELECT distinct mo.groupid
INTO groupidnum
FROM memberof mo inner join studentgroup sg on mo.groupid = sg.gid
WHERE sg.name = gname;

IF isInGroup = 0 THEN

insert into memberof(StudentID, GroupID, Joined)
   values (st_id, groupidnum, joinedDate);

update studentgroup
SET presidentid = st_id
WHERE gid = groupidnum;

else
update studentgroup
SET presidentid = st_id
WHERE gid = groupidnum;

end if;
end;
/
```

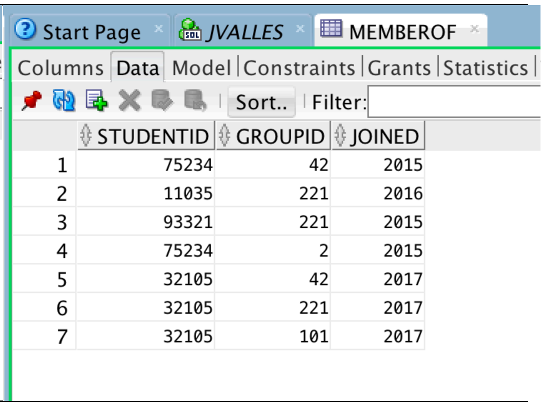


```
CREATE OR REPLACE PROCEDURE make_pres(st_id number, gname varchar2)
    as

isInGroup number;
joinedDate number;
groupidnum number;

BEGIN
SELECT count(*)
INTO isInGroup
FROM(SELECT *
FROM memberof mo inner join studentgroup sg on mo.groupid = sg.gid
WHERE sg.name = gname and studentid = st_id);
SELECT extract(year from sysdate)
INTO joinedDate
FROM dual;
SELECT distinct mo.groupid
INTO groupidnum
FROM memberof mo inner join studentgroup sg on mo.groupid = sg.gid
WHERE sg.name = gname;

IF isInGroup = 0 THEN

insert into memberof(StudentID, GroupID, Joined)
    values (st_id, groupidnum, joinedDate);

update studentgroup
SET presidentid = st_id
WHERE gid = groupidnum;

else
update studentgroup
SET presidentid = st_id
WHERE gid = groupidnum;

end if;
end;
/
```

```
Procedure MAKE_PRES compiled
```

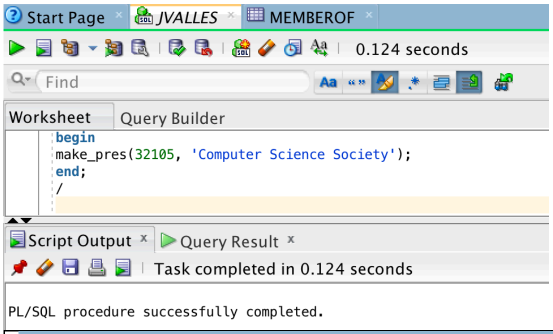


	GID	NAME	PRESIDENTID	FOUNDED
1	2	Computer Science Society	58992	1999
2	101	Robototics Society	(null)	1998
3	221	HerCDM	93321	2003
4	42	DeFrag	32105	2004



	STUDENTID	GROUPID	JOINED
1	75234	42	2015
2	11035	221	2016
3	93321	221	2015
4	75234	2	2015
5	32105	42	2017
6	32105	221	2017
7	32105	101	2017

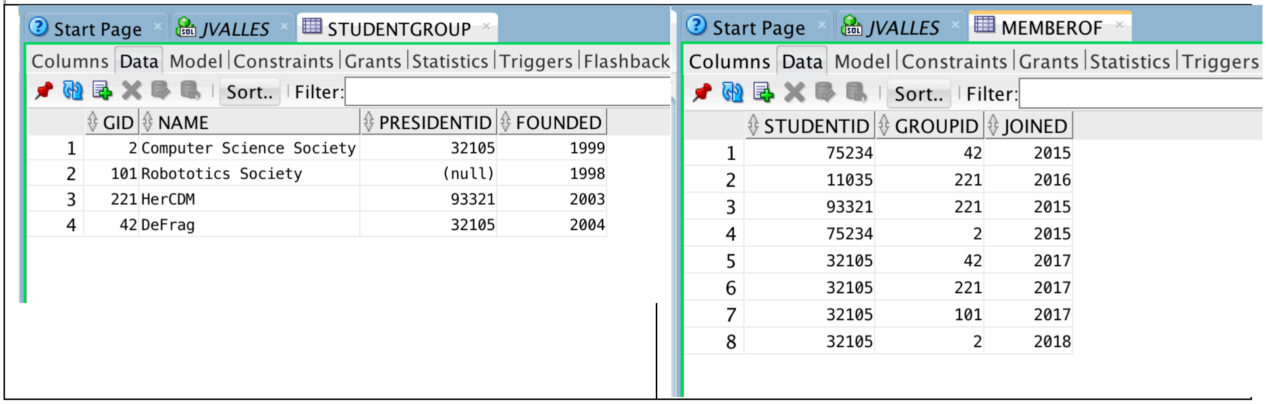
Test: Student 32105 (Peter Johnson) is **not** a member of ‘Computer Science Society’ . So, he was added to group first and joined year is set to system year (2018). He is then made president of group.



```
begin
make_pres(32105, 'Computer Science Society');
end;
/
```

Task completed in 0.124 seconds

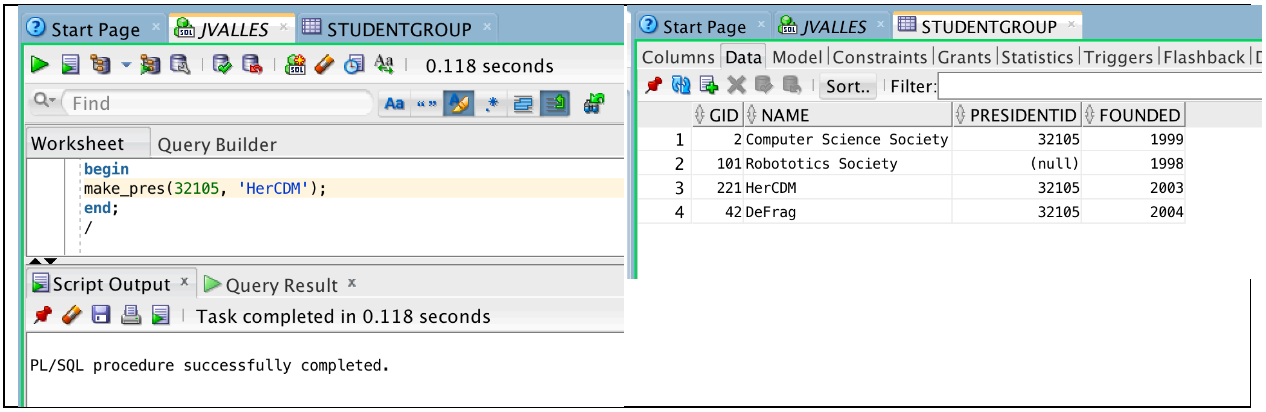
```
PL/SQL procedure successfully completed.
```



	GID	NAME	PRESIDENTID	FOUNDED
1	2	Computer Science Society	32105	1999
2	101	Robototics Society	(null)	1998
3	221	HerCDM	93321	2003
4	42	DeFrag	32105	2004

	STUDENTID	GROUPID	JOINED
1	75234	42	2015
2	11035	221	2016
3	93321	221	2015
4	75234	2	2015
5	32105	42	2017
6	32105	221	2017
7	32105	101	2017
8	32105	2	2018

Test #2: Student 32105 is also already a member of ‘HerCDM’, but will make president.



```
begin
make_pres(32105, 'HerCDM');
end;
/
```

Task completed in 0.118 seconds

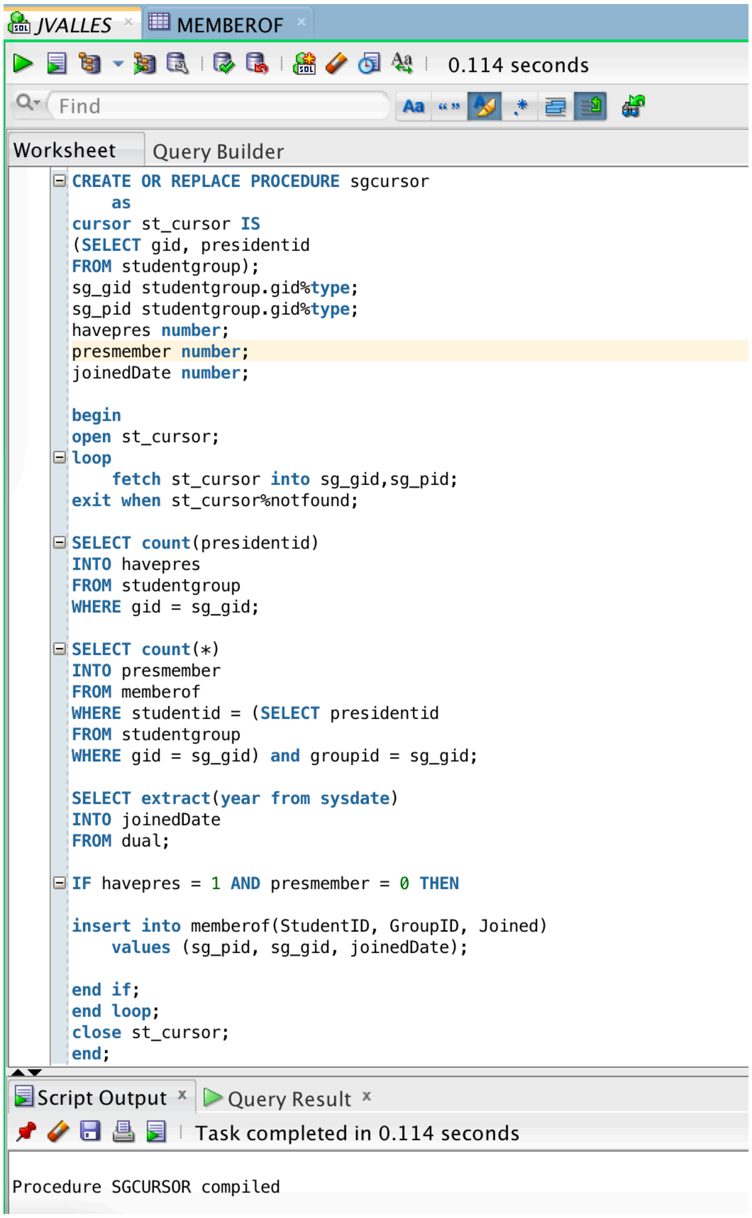
```
PL/SQL procedure successfully completed.
```

	GID	NAME	PRESIDENTID	FOUNDED
1	2	Computer Science Society	32105	1999
2	101	Robototics Society	(null)	1998
3	221	HerCDM	32105	2003
4	42	DeFrag	32105	2004

4.

Write procedure to force every president of student group to be member of group and set joined year to system year (2018) if they are not a group. Ignore student group if no president, or if president is already a member.

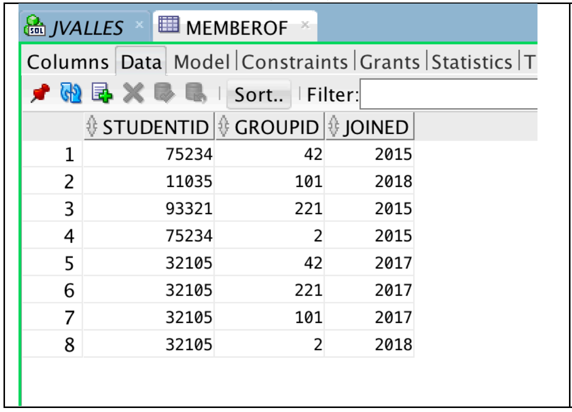
```
CREATE OR REPLACE PROCEDURE sgcursor
   as
cursor st_cursor IS
(SELECT gid, presidentid
FROM studentgroup);
sg_gid studentgroup.gid%type;
sg_pid studentgroup.gid%type;
havepres number;
presmember number;
joinedDate number;

begin
open st_cursor;
loop
   fetch st_cursor into sg_gid,sg_pid;
exit when st_cursor%notfound;

SELECT count(presidentid)
INTO havepres
FROM studentgroup
WHERE gid = sg_gid;

SELECT count(*)
INTO presmember
FROM memberof
WHERE studentid = (SELECT presidentid
FROM studentgroup
WHERE gid = sg_gid) and groupid = sg_gid;

SELECT extract(year from sysdate)
INTO joinedDate
FROM dual;

IF havepres = 1 AND presmember = 0 THEN

insert into memberof(StudentID, GroupID, Joined)
   values (sg_pid, sg_gid, joinedDate);

end if;
end loop;
close st_cursor;
end;
/
```



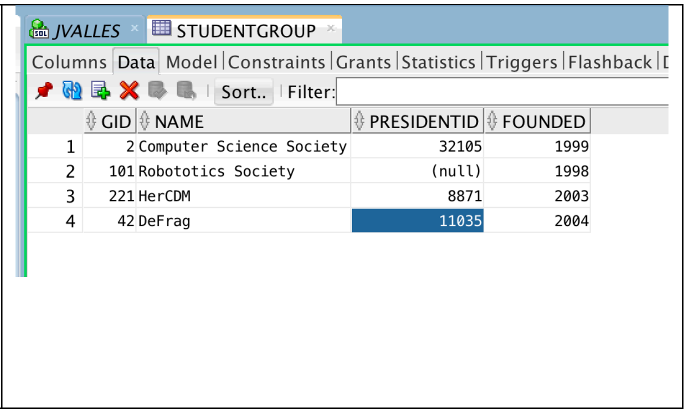
```
CREATE OR REPLACE PROCEDURE sgcursor
    as
cursor st_cursor IS
(SELECT gid, presidentid
FROM studentgroup);
sg_gid studentgroup.gid%type;
sg_pid studentgroup.gid%type;
havepres number;
presmember number;
joinedDate number;

begin
open st_cursor;
loop
    fetch st_cursor into sg_gid,sg_pid;
exit when st_cursor%notfound;

SELECT count(presidentid)
INTO havepres
FROM studentgroup
WHERE gid = sg_gid;

SELECT count(*)
INTO presmember
FROM memberof
WHERE studentid = (SELECT presidentid
FROM studentgroup
WHERE gid = sg_gid) and groupid = sg_gid;

SELECT extract(year from sysdate)
INTO joinedDate
FROM dual;

IF havepres = 1 AND presmember = 0 THEN

insert into memberof(StudentID, GroupID, Joined)
    values (sg_pid, sg_gid, joinedDate);

end if;
end loop;
close st_cursor;
end;
```

```
Task completed in 0.114 seconds

Procedure SGCURSOR compiled
```

Test #1 Student 11035 president of DeFrag and Student 8871 president of Her CDM not members of group

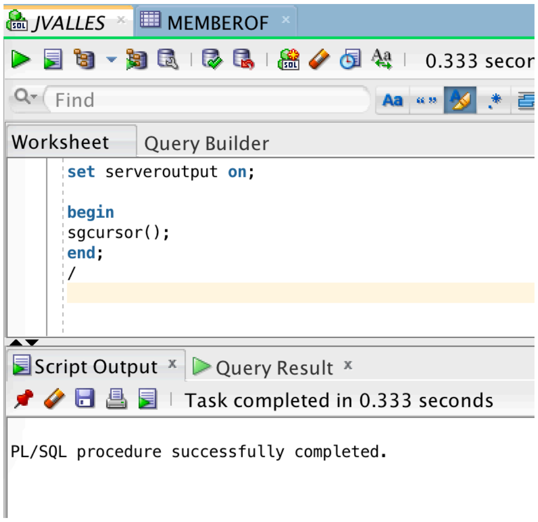


	STUDENTID	GROUPID	JOINED
1	75234	42	2015
2	11035	101	2018
3	93321	221	2015
4	75234	2	2015
5	32105	42	2017
6	32105	221	2017
7	32105	101	2017
8	32105	2	2018



	GID	NAME	PRESIDENTID	FOUNDED
1	2	Computer Science Society	32105	1999
2	101	Robototics Society	(null)	1998
3	221	HerCDM	8871	2003
4	42	DeFrag	11035	2004

Run procedure sgcursor()

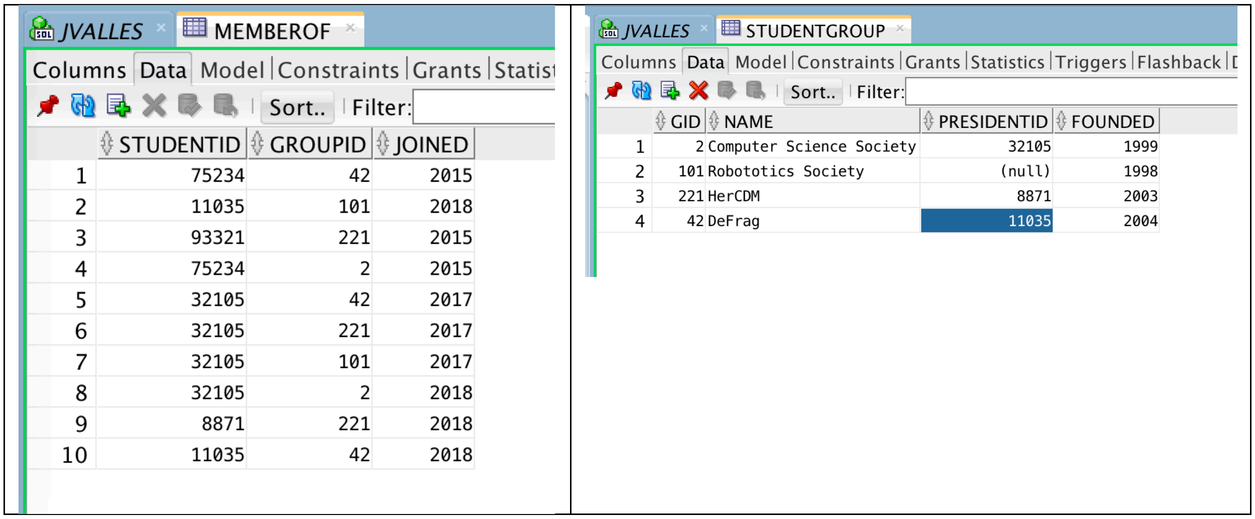


```
set serveroutput on;

begin
sgcursor();
end;
/
```

Task completed in 0.333 seconds

```
PL/SQL procedure successfully completed.
```

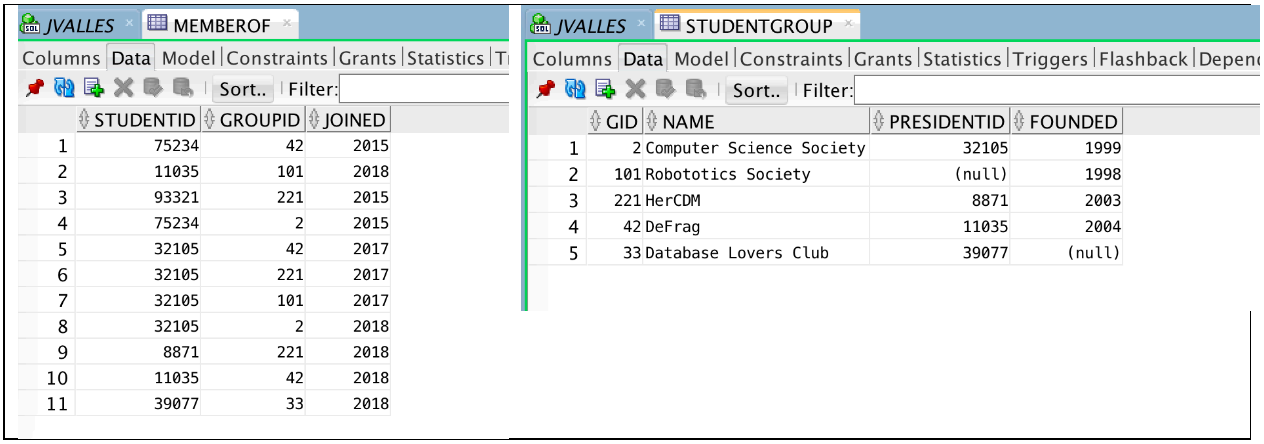
Student 11035 now a member of DeFrag (42) with join year of 2018. Student 8871 now member of HerCDM (221) with join year of 2018. Robotoics Society is ignored. And student 32105 who is President and a member of Computer Science Society is ignored.



	STUDENTID	GROUPID	JOINED
1	75234	42	2015
2	11035	101	2018
3	93321	221	2015
4	75234	2	2015
5	32105	42	2017
6	32105	221	2017
7	32105	101	2017
8	32105	2	2018
9	8871	221	2018
10	11035	42	2018

	GID	NAME	PRESIDENTID	FOUNDED
1	2	Computer Science Society	32105	1999
2	101	Robototics Society	(null)	1998
3	221	HerCDM	8871	2003
4	42	DeFrag	11035	2004

Test 2: Added new group, ‘Database Lovers Club’ , and made Joya Pollard 39077 president. Will run procedure and she should automatically be added to group with year 2018.. We should only see one additional entry in the memberof table compared to last run. This will test that code works for an arbitrary number of student groups.



	STUDENTID	GROUPID	JOINED
1	75234	42	2015
2	11035	101	2018
3	93321	221	2015
4	75234	2	2015
5	32105	42	2017
6	32105	221	2017
7	32105	101	2017
8	32105	2	2018
9	8871	221	2018
10	11035	42	2018
11	39077	33	2018

	GID	NAME	PRESIDENTID	FOUNDED
1	2	Computer Science Society	32105	1999
2	101	Robototics Society	(null)	1998
3	221	HerCDM	8871	2003
4	42	DeFrag	11035	2004
5	33	Database Lovers Club	39077	(null)