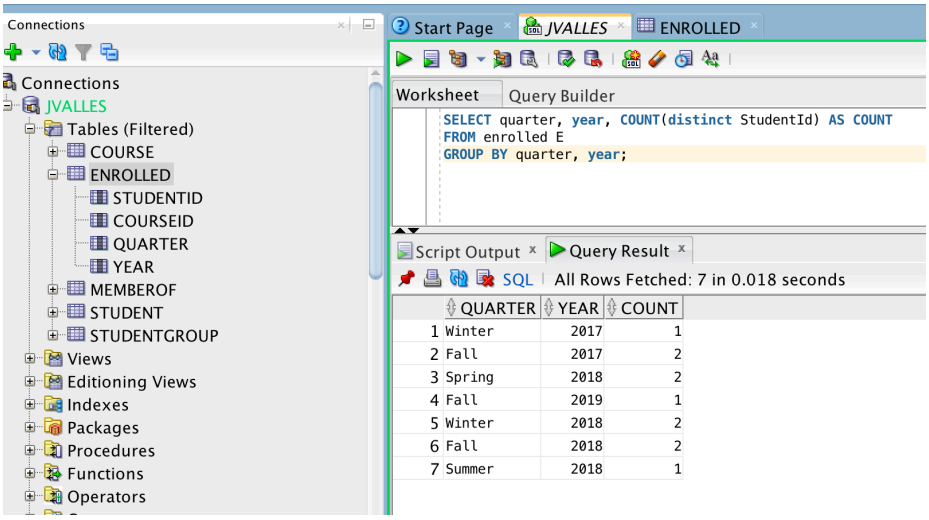
James Valles
CSC 453
Assignment – 4

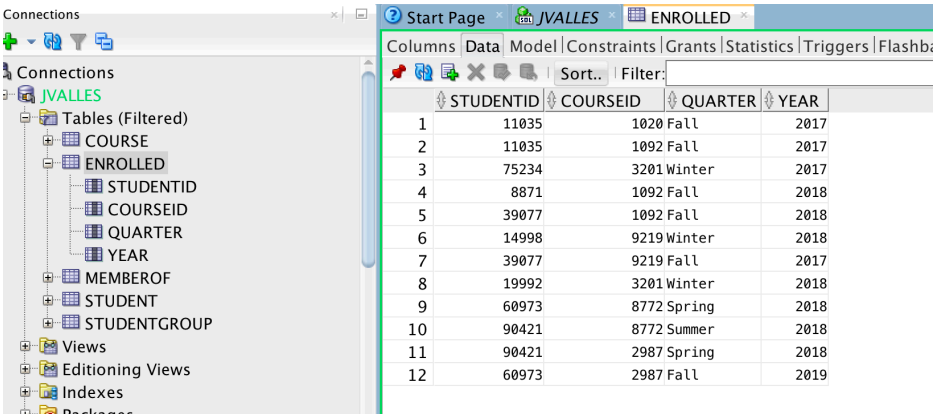
1. Reading done.

2.

a.

```
SELECT quarter, year, COUNT(DISTINCT StudentId) AS COUNT
FROM enrolled E
GROUP BY quarter, year;
```





b.

```
SELECT quarter, year, COUNT(distinct StudentId) AS COUNT
FROM enrolled E
GROUP BY quarter, year
ORDER BY year, CASE quarter
    WHEN 'Fall'   THEN '0'
    WHEN 'Winter' THEN '1'
    WHEN 'Spring' THEN '2'
    ELSE quarter
end;
```

```
SELECT quarter, year, COUNT(distinct StudentId) AS COUNT
FROM enrolled E
GROUP BY quarter, year
ORDER BY year, CASE quarter
    WHEN 'Fall'   THEN '0'
    WHEN 'Winter' THEN '1'
    WHEN 'Spring' THEN '2'
    ELSE quarter
end;
```

All Rows Fetched: 7 in 0.018 seconds

	QUARTER	YEAR	COUNT
1	Fall	2017	2
2	Winter	2017	1
3	Fall	2018	2
4	Winter	2018	2
5	Spring	2018	2
6	Summer	2018	1
7	Fall	2019	1

c.

SELECT SID, LASTNAME, FIRSTNAME,
(SELECT COUNT(*)
FROM STUDENTGROUP WHERE SID = PRESIDENTID) AS COUNT FROM STUDENT;

```
SELECT SID, LASTNAME, FIRSTNAME,
(SELECT COUNT(*)
FROM STUDENTGROUP
WHERE SID = PRESIDENTID) AS COUNT FROM STUDENT;
```

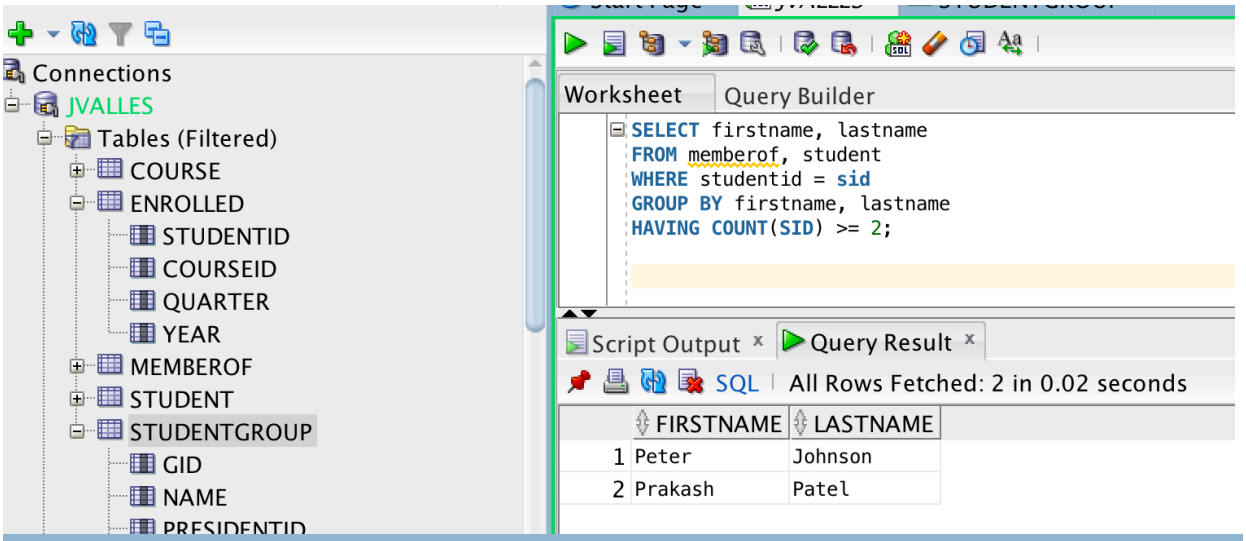
All Rows Fetched: 13 in 0.042 seconds

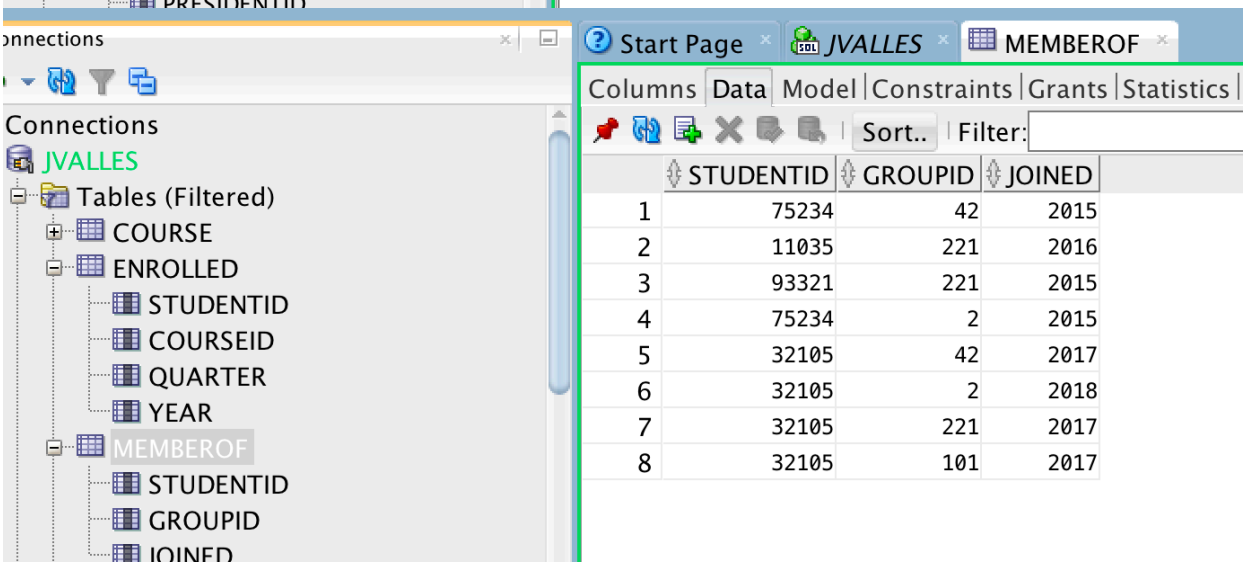
	SID	LASTNAME	FIRSTNAME	COUNT
1	90421	Brennigan	Marcus	0
2	14662	Patel	Deepa	0
3	8871	Snowdon	Jonathan	0
4	19992	Starck	Jason	0
5	32105	Johnson	Peter	1
6	11035	Winter	Abigail	0
7	75234	Patel	Prakash	0
8	93321	Snowdon	Jennifer	1
9	14998	Degroff	Jarvis	0
10	57923	Kubik	Dwayne	0
11	58992	Skelly	Trinity	2
12	60973	Krol	Angelo	0
13	39077	Pollard	Joya	0

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

3.

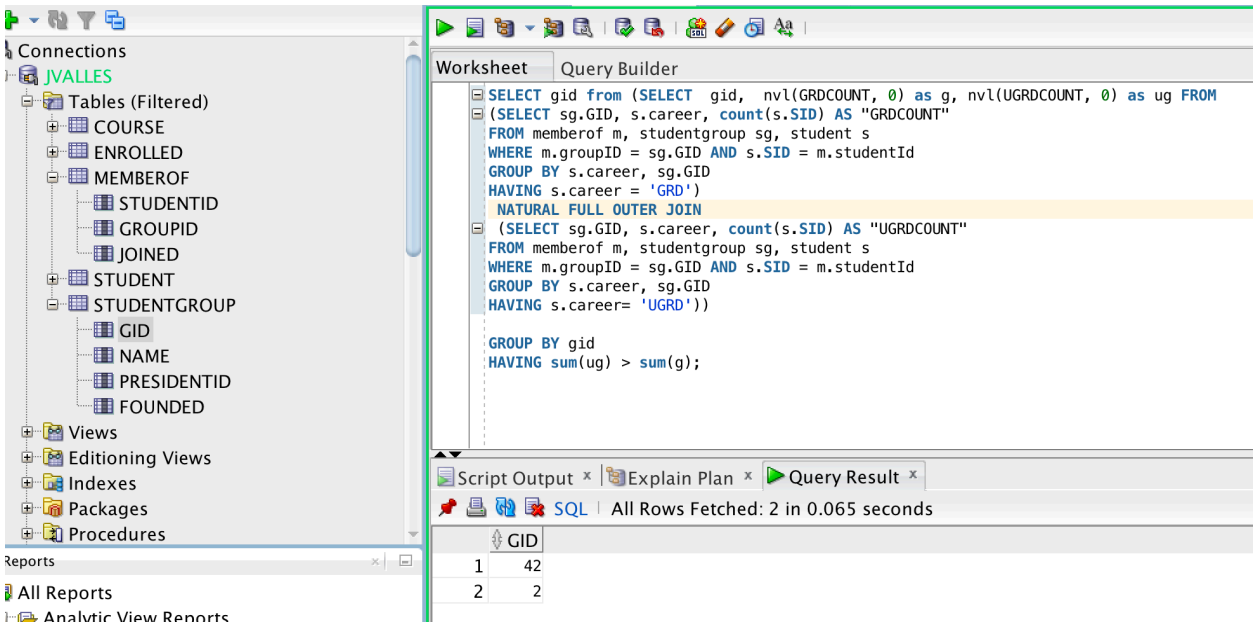
a.

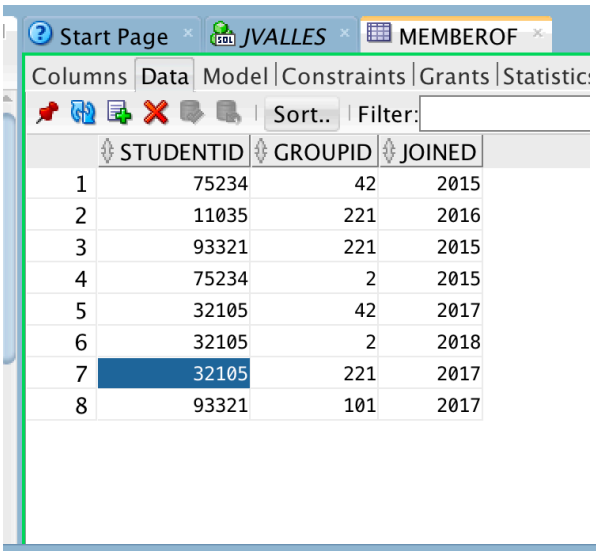




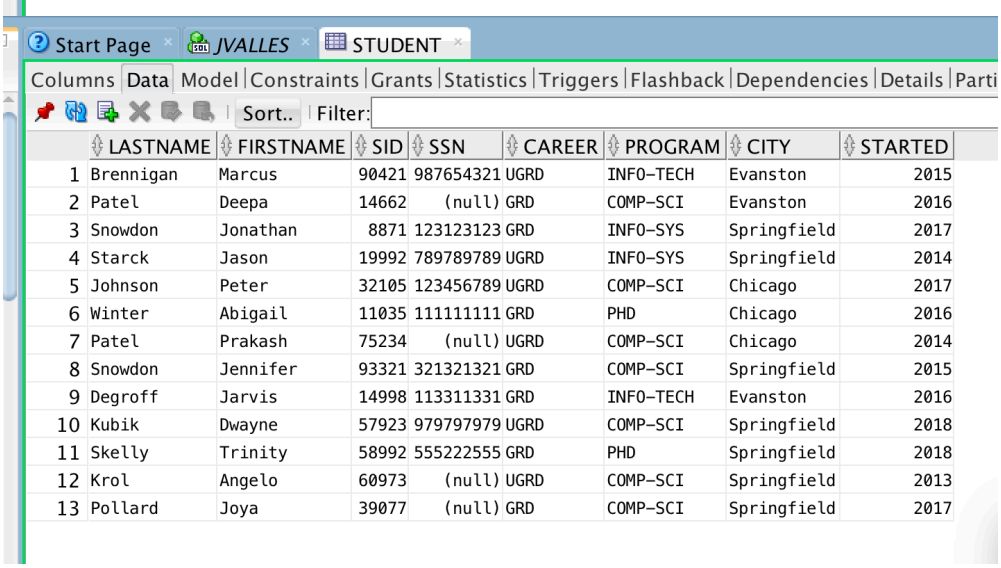
b.

```
select gid from (SELECT  gid,  nvl(GRDCOUNT, 0) as g, nvl(UGRDCOUNT, 0) as ug FROM
(SELECT sg.GID, s.career, count(s.SID) AS "GRDCOUNT"
FROM memberof m, studentgroup sg, student s
WHERE m.groupID = sg.GID AND s.SID = m.studentId
GROUP BY s.career, sg.GID
HAVING s.career = 'GRD')
 natural full outer join

 (SELECT sg.GID, s.career, count(s.SID) AS "UGRDCOUNT"
FROM memberof m, studentgroup sg, student s
WHERE m.groupID = sg.GID AND s.SID = m.studentId
GROUP BY s.career, sg.GID
HAVING s.career= 'UGRD'))

group by gid
having sum(ug) > sum(g)
;
```



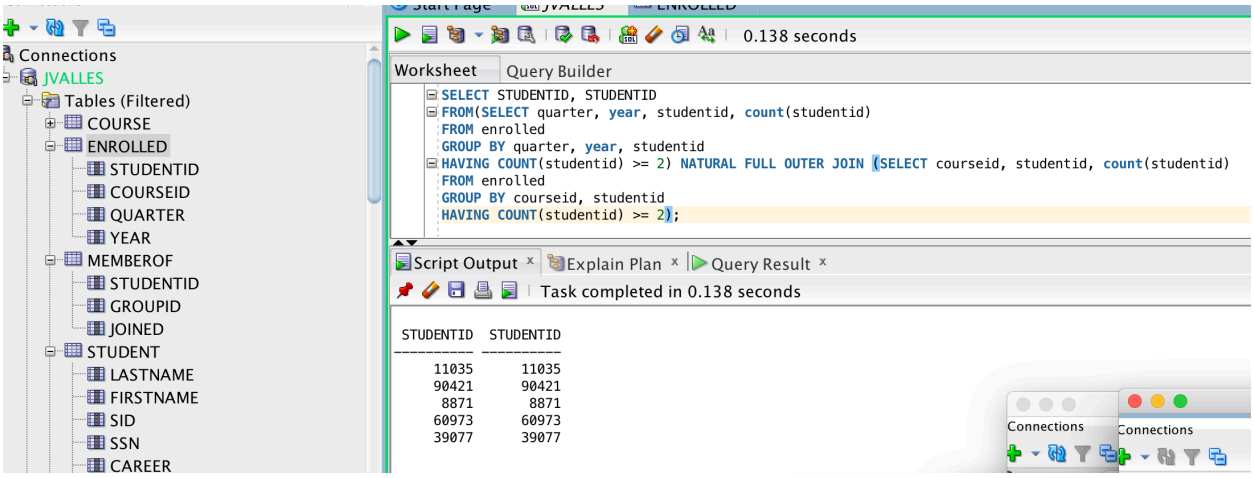


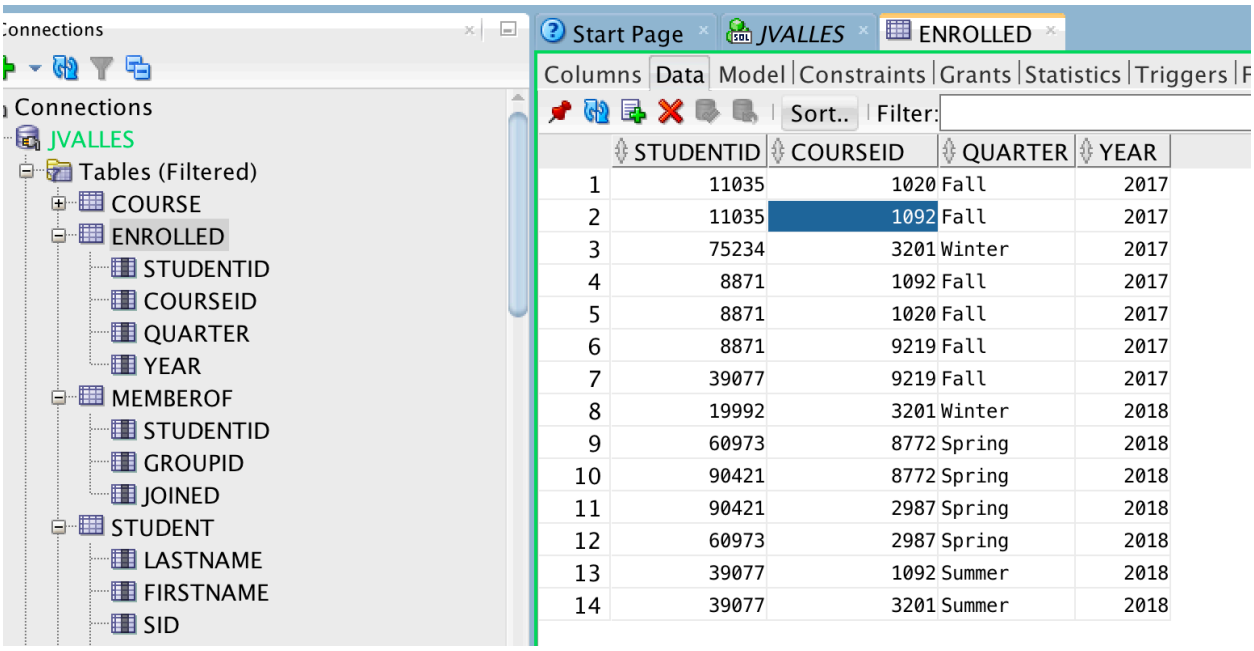
	STUDENTID	GROUPID	JOINED
1	75234	42	2015
2	11035	221	2016
3	93321	221	2015
4	75234	2	2015
5	32105	42	2017
6	32105	2	2018
7	32105	221	2017
8	93321	101	2017



	LASTNAME	FIRSTNAME	SID	SSN	CAREER	PROGRAM	CITY	STARTED
1	Brennigan	Marcus	90421	987654321	UGRD	INFO-TECH	Evanston	2015
2	Patel	Deepa	14662	(null)	GRD	COMP-SCI	Evanston	2016
3	Snowdon	Jonathan	8871	123123123	GRD	INFO-SYS	Springfield	2017
4	Starck	Jason	19992	789789789	UGRD	INFO-SYS	Springfield	2014
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

c.





	STUDENTID	COURSEID	QUARTER	YEAR
1	11035	1020	Fall	2017
2	11035	1092	Fall	2017
3	75234	3201	Winter	2017
4	8871	1092	Fall	2017
5	8871	1020	Fall	2017
6	8871	9219	Fall	2017
7	39077	9219	Fall	2017
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
13	39077	1092	Summer	2018
14	39077	3201	Summer	2018

4.

Can we infer CE -> AB? <u>Yes.</u>
C,E
C -> D = C,D,E
DE -> <u>A</u>,C = C, D, E, A, C
A,C,D -> <u>B</u>
Therefore, CE -> AB

{C, E}+ = {A, B, C, D, E}

Can we infer CF -> AD? <u>No.</u>
CF -> AD
C -> D = C,D,F
We can get to D, but not A.

{C, F)+ = {C, D, F}

KEYS:
F is not functional dependency (left, right), so must be present in every key.
A,B,C,D, E present in left and right, so will consider all combinations.
To be a key: must be minimal, must determine all attributes in relation.

KEY 1: ACF
A, C, F+ = {A, B, C, D, E, F}

C -> D = ACDF
ACD -> B = ABCDF
F determines itself
BC -> AED = ABCDEF

KEY 2: BCF
B, C, F+ = (A, B, C, D, E, F}

BC -> AED = ABCDE
F determines itself .
Therefore, ABCDEF

KEY 3: CEF
C, E, F+ = {A, B, C, D, E, F}

C -> D = CEDF
DE -> AC = ACEDF
ACD -> B = ABCDEF

KEY 4: DEF
D, E, F = {A, B, C, D, E, F}

DE -> AC = ACDEF
ACD -> B = ABCDEF