Isha Singh

1BM19CS218

SECTION-4A

Consider the following database for student enrollment for course :

STUDENT(snum: integer, sname:string, major: string, lvl: string, age: integer) CLASS(cname: string, meetsat: time, room: string, fid: integer)
ENROLLED(snum: integer, cname:string)
FACULTY(fid: integer, fname:string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level(lvl) is a two character code with 4 different values (example: Junior: JR etc)
Write the following queries in SQL.

No duplicates should be printed in any of the answers.

i.Find the names of all Juniors (level = JR) who are enrolled in a class taught by "name"
ii.Find the names of all classes that either meet in room R128 or have five or more Students enrolled.
iii.Find the names of all students who are enrolled in two classes that meet at the same time. iv.Find the names of faculty members who teach in every room in which some class is taught. v.Find the names of faculty members for whom the combined enrollment of the courses
that they teach is less than five.
vi.Find the names of students who are not enrolled in any class.
vii.For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

STUDENT TABLE

snum	sname	major	lvl	age
1	jhon	CS	Sr	19
2	Smith	CS	Jr	20
3	Jacob	CV	Sr	20
4	Tom	CS	Jr	20
5	Rahul	CS	Jr	20
6	Rita	CS	Sr	21
NULL	NULL	NULL	NULL	NULL

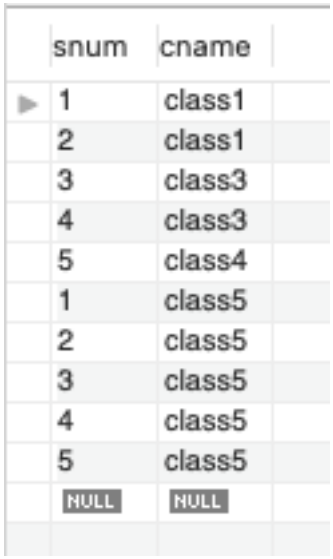
FACULTY TABLE

fid	fname	deptid
11	Harish	1000
12	MV	1000
13	Mira	1001
14	Shiva	1002
15	Nupur	1000
NULL	NULL	NULL

CLASS TABLE

cname	meets_at	room	fid
class1	2012-11-15 10:15:16	R1	14
class10	2012-11-15 10:15:16	R128	14
class2	2012-11-15 10:15:20	R2	12
class3	2012-11-15 10:15:25	R3	11
class4	2012-11-15 20:15:20	R4	14
class5	2012-11-15 20:15:20	R3	15
class6	2012-11-15 13:20:20	R2	14
class7	2012-11-15 10:10:10	R3	14
NULL	NULL	NULL	NULL

ENROLLED TABLE



snum	cname
1	class1
2	class1
3	class3
4	class3
5	class4
1	class5
2	class5
3	class5
4	class5
5	class5
NULL	NULL

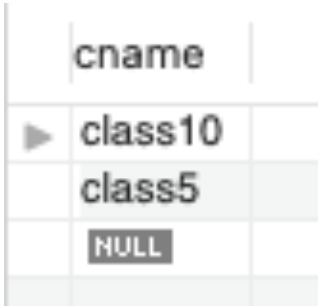
Find the names of all Juniors (level = JR) who are enrolled in a class taught by “name” select s.sname from Student s,Enrolled e,Class c

```
select sname
from student
where lvl='Jr' and snum in
(select snum
from enrolled e,class c,faculty f
where e.cname=c.cname and c.fid=f.fid and f.fname=‘Harish');
```

sname
Tom

Find the names of all classes that either meet in room R128 or have five or more Students enrolled.

```
select distinct c.cname

from class c

where c.room='R128' or cname in

(select cname

from enrolled

group by cname

having count(*)>=5);
```



cname
class10
class5
NULL

Find the names of all students who are enrolled in two classes that meet at the same time.

```
SELECT DISTINCT S.sname
FROM Student S
WHERE S.snum IN (SELECT E1.snum
FROM Enrolled E1, Enrolled E2, Class C1, Class C2
WHERE E1.snum = E2.snum AND E1.cname<> E2.cname
AND E1.cname = C1.cname
AND E2.cname = C2.cname AND C1.meets_at = C2.meets_at);
```

sname
Rahul

Find the names of faculty members who teach in every room in which some class is taught

```
select f.fname
from faculty f,class c
 where c.fid=f.fid
 group by c.fid
 having count(c.fid)=
 (select count(distinct room)
 from class);
```

fname
Shiva

Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

```
select fname
from faculty
where fid not in
(select f.fid
from faculty f,class c,enrolled e
where e.cname=c.cname and c.fid=f.fid
group by e.cname
having count(*)>=5);
```

fname
Harish
MV
Mira
Shiva

Find the names of students who are not enrolled in any class.

```
select sname
from student s
where s.snum not in
(select e.snum
from enrolled e);
```

sname
Rita

For each age value that appears in Students, find the level value that appears most often.

```
select s.age,s.lvl
from student s
group by s.age
having s.lvl in
(select s1.lvl
from student s1
where s1.age=s.age
group by s1.age
having count(*)>=all(select s2.lvl from student s2
where s2.age=s1.age
group by s2.age));
```

age	lvl
19	Sr
20	Jr
21	Sr