## **DIVYASHREE K** 1BM19CS054

CSE-4A

Question:-

Program 4:

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(IvI) 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).
Program 4:
create database studentfaculty2;
use studentfaculty2;
create table STUDENT(
snum int,
sname varchar(60),
major varchar(50),
IvI varchar(50),
age int,
primary key(snum)
);
```

```
create table CLASS(
cname varchar(60),
meetsat timestamp,
room varchar(60),
fid int,
primary key (cname)
);
```

create table enrolled(
snum int,
cname varchar(60),
primary key(snum,cname),
foreign key(snum) references
STUDENT(snum)

```
on update cascade on delete cascade,
foreign key(cname) references
CLASS(cname)
on update cascade on delete cascade
);
create table FACULTY(
fid int,
fname varchar(60),
deptid int,
primary key(fid)
);
```

insert into STUDENT values

(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);
insert into CLASS values
('Class1',"12/11/15
10:15:16.00000", 'R1', 14);
select * from CLASS;
delete from CLASS where
cname='Class1';
select * from CLASS;
insert into CLASS values
('Class1',"15/11/12
10:15:16.00000", 'R1', 14);
```

select \* from CLASS;

```
insert into CLASS values
('Class10',"15/11/12
10:15:16.00000",'R128',14),
('Class2',"15/11/12
10:15:20.00000",'R2',12),
('Class3',"15/11/12
10:15:25.00000",'R3',11),
('Class4',"15/11/12
10:15:20.00000",'R4',14),
('Class5',"15/11/12
10:15:20.00000",'R3',15),
('Class6',"15/11/12
13:20:20.00000",'R2',14),
('Class7',"15/11/12
10:10:10.00000",'R3',14);
```

insert into ENROLLED values (1,'Class1'),(2,'Class1'),(3,'Class3'),(4,'Class3'),(5,'Class4');

insert into FACULTY values (11,'Harish',1000),(12,'MV',1000),(13,' Mira',1001),(14,'Shiva',1002),(15,'Nupur',1000);

```
select * from STUDENT;
select * from CLASS;
select * from ENROLLED;
select * from FACULTY;
```

select s.sname, f.fname from STUDENT s, CLASS c, FACULTY f,ENROLLED e

where s.snum=e.snum and s.lvl='Jr' and e.cname=c.cname and f.fid=c.fid;

select c.cname from class c where c.room = 'R128'

or c.cname in (select e.cname from enrolled e group by e.cname having count(e.snum)>5);

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

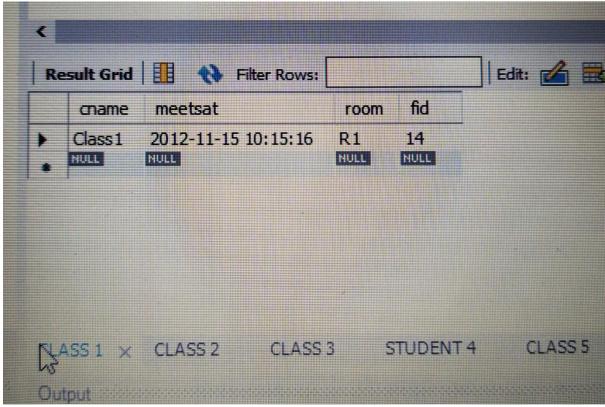
and e2.cname = c2.cname and c1.meetsat = c2.meetsat);

select distinct f.fname from faculty f where 5>(select COUNT(e.snum) from Class c, enrolled e

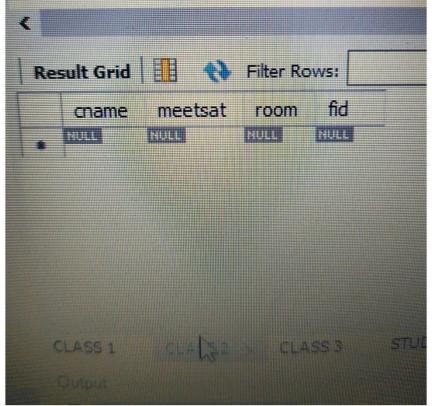
where c.cname = e.cname and c.fid = f.fid);

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

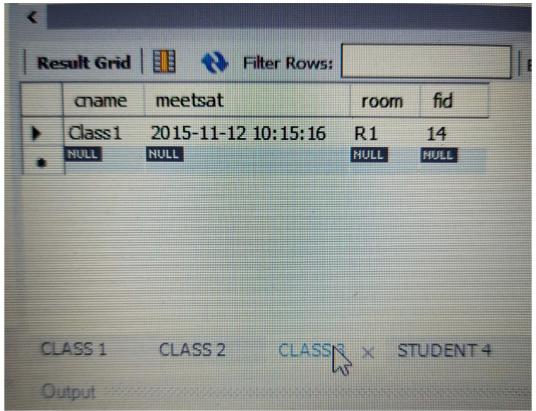
Output:



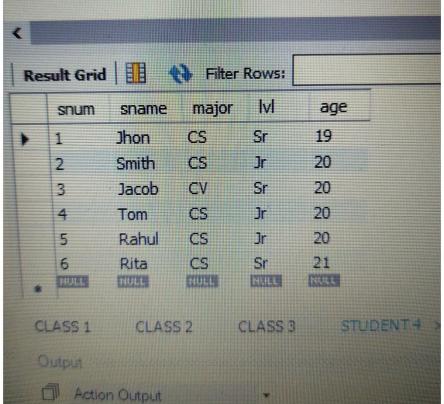
Scanned with CamScanner



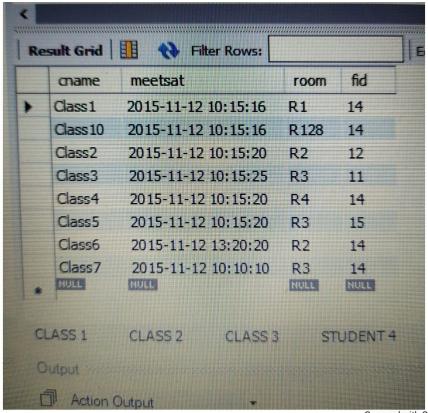
Scanned with CamScanner



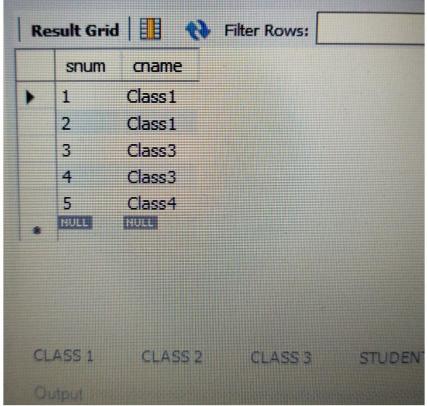
Scanned with CamScanner



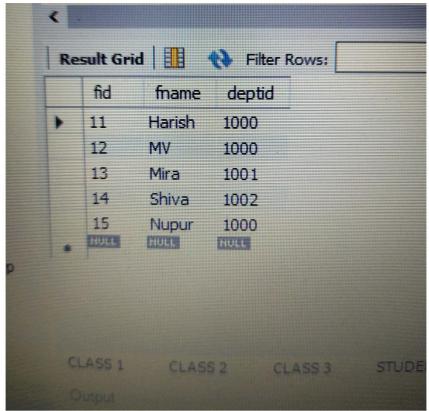
Scanned with CamScanner



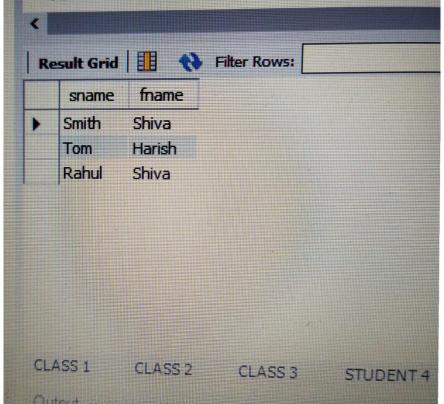
Scanned with CamScanner



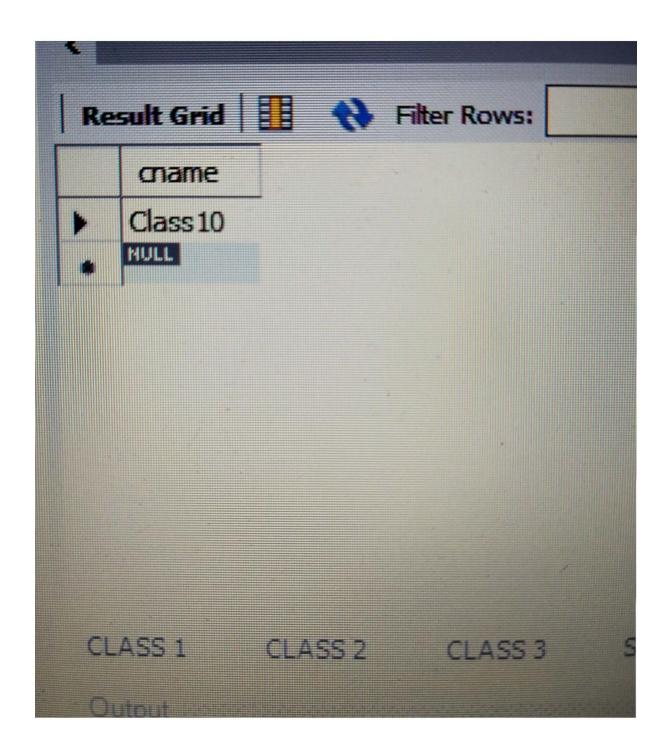
Scanned with CamScanner

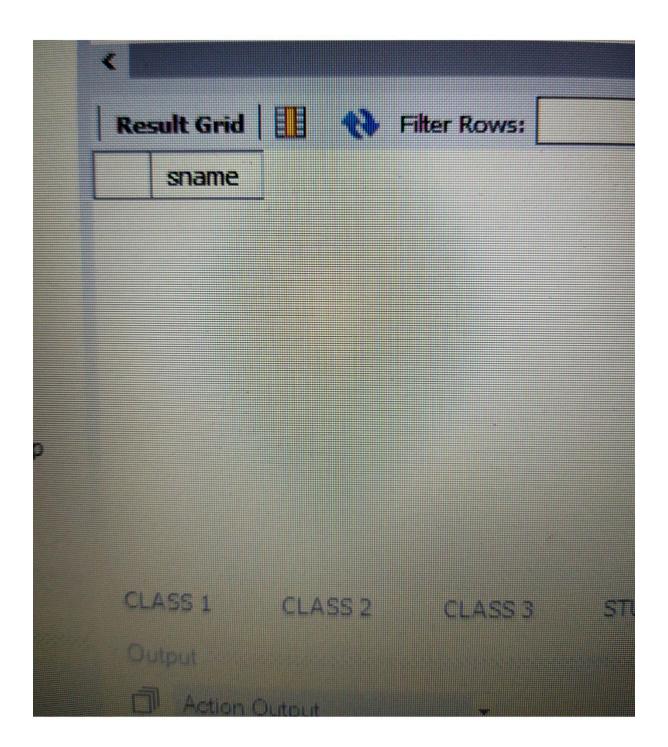


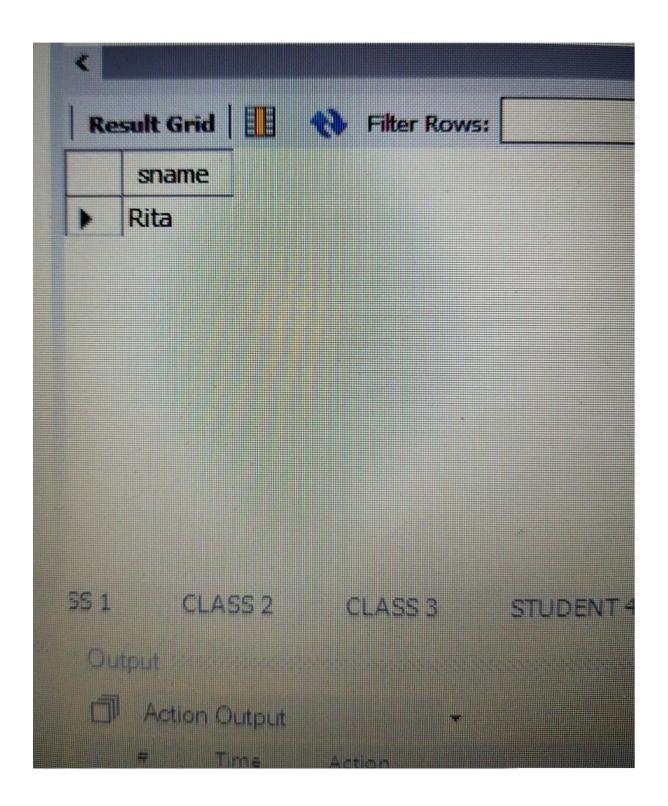
Scanned with CamScanner



Scanned with CamScanner







Scanned with CamScanner