

PROGRAM 4: STUDENT FACULTY DATABASE

Consider the following database for student enrolment for course:

STUDENT (snum: integer, sname: string, major: string, level: string, age: integer)

CLASS (name: string, meets at: 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 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
- 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 enrolment 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).

```
create database Student_faculty;
```

```
use Student_faculty;
```

```
create table student(snum int,sname varchar(60),major varchar(60),level varchar(6),age  
int,primary key(snum));
```

```
create table faculty(fid int,fname varchar(60),deptid int,primary key(fid));
```

```
create table class(cname varchar(60),meets_at timestamp,room varchar(60),fid  
int,primary key(cname),foreign key(fid) references faculty(fid));
```

```
create table enrolled(snum int,cname varchar(60),primary key(snum,cname),foreign
key(cname) references class(cname),foreign key(snum) references student(snum));
```

i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by

```
select distinct s.sname from student s,class c,enrolled e,faculty f where s.snum=e.snum
and e.cname=c.cname and c.fid=f.fid and f.fname='shiva' and s.level='jr';
```

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

```
select c.cname from class c where c.room='R128' or c.cname in(select e.cname from
enrolled e,class c where c.cname=e.cname group by e.cname having count(*) >= 5);
```

iii. 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);
```

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

```
SELECT F.fname FROM Faculty F WHERE NOT EXISTS (( SELECT C.room FROM
Class C ) EXCEPT (SELECT C1.room FROM Class C1 WHERE C1.fid = F.fid ));
```

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

```
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);
```

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

```
SELECT DISTINCT S.sname FROM Student S WHERE S.snum NOT IN (SELECT
E.snum FROM Enrolled E );
```

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).

```
SELECT S.age, S.level FROM Student S GROUP BY S.age, S.level HAVING S.level IN  
(SELECT S1.level FROM Student S1 WHERE S1.age = S.age GROUP BY S1.level,  
S1.age HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM Student S2 WHERE  
s1.age = S2.age GROUP BY S2.level, S2.age));
```