**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.**

CREATE TABLE student(snum INT,sname VARCHAR(10),major VARCHAR(10),lvl VARCHAR(10),age INT, PRIMARY KEY(snum));

CREATE TABLE faculty(fid INT,fname VARCHAR(20),deptid INT,PRIMARY KEY(fid));

CREATE TABLE class(cname VARCHAR(20),meets\_at TIME,room VARCHAR(10),fid INT,PRIMARY KEY(cname),FOREIGN KEY(fid) REFERENCES faculty(fid));

CREATE TABLE enrolled(snum INT,cname VARCHAR(20),PRIMARY KEY(snum,cname),FOREIGN KEY(snum) REFERENCES student(snum),FOREIGN KEY(cname) REFERENCES class(cname));

INSERT INTO STUDENT VALUES(1, 'raj','CS','Sr',19);

INSERT INTO STUDENT VALUES(2, 'Smitha','CS','Jr',20);

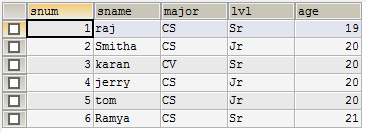
INSERT INTO STUDENT VALUES(3, 'karan','CV','Sr',20);

INSERT INTO STUDENT VALUES(4, 'jerry','CS','Jr',20);

INSERT INTO STUDENT VALUES(5, 'tom','CS','Jr',20);

INSERT INTO STUDENT VALUES(6, 'Ramya','CS','Sr',21);

SELECT \*FROM student;



INSERT INTO FACULTY VALUES(11, 'hamsa', 1000);

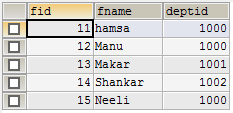
INSERT INTO FACULTY VALUES(12, 'Manu', 1000);

INSERT INTO FACULTY VALUES(13 , 'Makar', 1001);

INSERT INTO FACULTY VALUES(14, 'Shankar', 1002);

INSERT INTO FACULTY VALUES(15, 'Neeli', 1000);

SELECT \*FROM FACULTY;



INSERT INTO class VALUES('class1', '10:15:16', 'R1', 14);

INSERT INTO class VALUES('class10', '10:15:16', 'R128', 14);

INSERT INTO class VALUES('class2', '10:15:20', 'R2', 12);

INSERT INTO class VALUES('class3', '10:15:25', 'R3', 12)

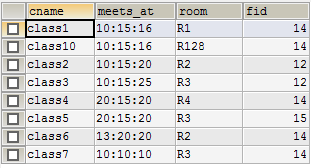
INSERT INTO class VALUES('class4', '20:15:20', 'R4', 14);

INSERT INTO class VALUES('class5', '20:15:20', 'R3', 15);

INSERT INTO class VALUES('class6', '13:20:20', 'R2', 14);

INSERT INTO class VALUES('class7', '10:10:10', 'R3', 14);

SELECT \*FROM CLASS;



INSERT INTO enrolled VALUES(1, 'class1');

INSERT INTO enrolled VALUES(2, 'class1');

INSERT INTO enrolled VALUES(3, 'class3');

INSERT INTO enrolled VALUES(4, 'class3');

INSERT INTO enrolled VALUES(5, 'class4');

INSERT INTO enrolled VALUES(1, 'class5');

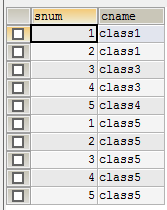
INSERT INTO enrolled VALUES(2, 'class5');

INSERT INTO enrolled VALUES(3, 'class5');

INSERT INTO enrolled VALUES(4, 'class5');

INSERT INTO enrolled VALUES(5, 'class5');

SELECT \*FROM ENROLLED;



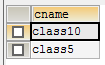
**i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by**

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='manu' AND S.lvl='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 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);



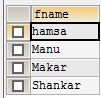
**iv. Find the names of faculty members who teach in every room in which some class is taught.**

SELECT DISTINCT F.fname FROM Faculty F WHERE NOT EXISTS(SELECT C.room FROM Class C WHERE C.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);

****

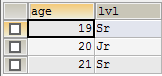
**vi. Find the names of students who are not enrolled in any class.**

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

****