4: STUDENT FACULTY DATABASE

Consider the following database for student enrollment for course :

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

CLASS(cname: 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(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.

create database StudentFaculty;

mysql> use StudentFaculty;

mysql> create table student(snum int,sname varchar(60),major varchar(60),level varchar(6),

age int,primary key(snum));

mysql> desc student;

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| snum | int | NO | PRI | NULL | |

| sname | varchar(60) | YES | | NULL | |

| major | varchar(60) | YES | | NULL | |

| level | varchar(6) | YES | | NULL | |

| age | int | YES | | NULL | |

+-------+-------------+------+-----+---------+-------+

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

mysql> desc faculty;

+--------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------+-------------+------+-----+---------+-------+

| fid | int | NO | PRI | NULL | |

| fname | varchar(60) | YES | | NULL | |

| deptid | int | YES | | NULL | |

+--------+-------------+------+-----+---------+-------+

mysql> create table class(cname varchar(60),meets\_at timestamp,room varchar(60),fid int,

primary key(cname),foreign key(fid) references faculty(fid));

mysql> desc class;

+----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+-------------+------+-----+---------+-------+

| cname | varchar(60) | NO | PRI | NULL | |

| meets\_at | timestamp | YES | | NULL | |

| room | varchar(60) | YES | | NULL | |

| fid | int | YES | MUL | NULL | |

+----------+-------------+------+-----+---------+-------+

mysql> create table enrolled(snum int,cname varchar(60),primary key(snum,cname),foreign key(cname)

references class(cname),foreign key(snum) references student(snum));

mysql> desc enrolled;

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| snum | int | NO | PRI | NULL | |

| cname | varchar(60) | NO | PRI | NULL | |

+-------+-------------+------+-----+---------+-------+

mysql> insert into student values(1,'john','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);

mysql> select \*from student;

+------+-------+-------+-------+------+

| snum | sname | major | level | age |

+------+-------+-------+-------+------+

| 1 | john | 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 |

+------+-------+-------+-------+------+

mysql> insert into faculty values(11,'harish',1000),(12,'manav',1000),

(13,'mira',1001),(14,'shiva',1002),(15,'nupur',1000);

mysql> select\*from faculty;

+-----+--------+--------+

| fid | fname | deptid |

+-----+--------+--------+

| 11 | harish | 1000 |

| 12 | manav | 1000 |

| 13 | mira | 1001 |

| 14 | shiva | 1002 |

| 15 | nupur | 1000 |

+-----+--------+--------+

mysql> insert into class values('class1','12/11/15 10:15:16','R1',14);

('class10','12/11/15 10:15:16','R128',14);

('class2','12/11/15 10:15:20','R2',12);

('class3','12/11/15 10:15:25','R3',12);

('class4','12/11/15 10:15:20','R4',14);

('class5','12/11/15 20:15:20','R3',15);

('class6','12/11/15 13:20:20','R2',14);

('class7','12/11/15 10:10:10','R3',14);

mysql> select\*from class;

+---------+---------------------+------+------+

| 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 | 12 |

| class4 | 2012-11-15 10:15:20 | 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 |

+---------+---------------------+------+------+

mysql> insert into enrolled values(1,'class1'),(2,'class1'),(3,'class3');

(4,'class3'),(5,'class4'),(1,'class5'),(2,'class5'),(3,'class5')'

(4,'class5'),(5,'class5');

mysql> select\*from enrolled;

+------+--------+

| snum | cname |

+------+--------+

| 1 | class1 |

| 2 | class1 |

| 3 | class3 |

| 4 | class3 |

| 5 | class4 |

| 1 | class5 |

| 2 | class5 |

| 3 | class5 |

| 4 | class5 |

| 5 | class5 |

+------+--------+

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

mysql> 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';

+-------+

| sname |

+-------+

| smith |

| rahul |

+-------+

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

mysql> 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);

+---------+

| cname |

+---------+

| class10 |

| class5 |

+---------+

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

mysql> 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.

mysql> select f.fname

-> from faculty f

-> where f.fid in(select fid from class

-> group by fid having count(\*)=(select count(distinct room)from class));

+-------+

| fname |

+-------+

| shiva |

+-------+

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

mysql> 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);

+--------+

| fname |

+--------+

| harish |

| manav |

| mira |

| shiva |

+--------+

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

mysql> select distinct s.sname

-> from student s

-> where s.snum not in (select e.snum from enrolled e);

+-------+

| sname |

+-------+

| rita |

+-------+

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).

mysql> 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));

+------+-------+

| age | level |

+------+-------+

| 19 | sr |

| 20 | jr |

| 21 | sr |

+------+-------+