DBMS LAB 4 :-

```
CREATE TABLE student(
 snum INT,
 sname VARCHAR(20),
 major VARCHAR(10),
 lvl VARCHAR(2),
 age INT,
 PRIMARY KEY(snum)
);
CREATE TABLE class(
 cname VARCHAR(20),
 meets_at TIME(0),
 room VARCHAR(10),
 fid INT,
 PRIMARY KEY(cname),
 FOREIGN KEY(fid) REFERENCES faculty(fid)
 ON UPDATE CASCADE ON DELETE CASCADE
);
CREATE TABLE enrolled(
 snum INT,
 cname VARCHAR(20),
 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(20),
```

```
 deptid INT,
 PRIMARY KEY(fid)
);
INSERT INTO student (snum,sname,major,lvl,age) VALUES
(1, 'hari', 'MATH', 'FR', 18),
(2, 'mahantesh', 'MATH', 'SR', 18),
(3, 'shreshtha', 'TFCS', 'SR', 19),
(4, 'subhas', 'TFCS', 'FR', 19),
(5, 'saquib', 'DBMS', 'JR', 20),
(6, 'krishan', 'DBMS', 'JR', 21),
(7, 'jaques', 'ADA', 'SR', 21),
(8, 'ashutosh' , 'MATH', 'FR', 18),
(9, 'divyanshu' , 'MATH', 'JR', 21),
(10, 'derek' , 'MATH', 'SR', 19);
INSERT INTO class (cname,meets_at,room,fid) VALUES
('A', '01:02:00', 'R124', 1),
('B', '02:02:00', 'R125', 2),
('C', '03:02:00', 'R126', 3),
('D', '03:02:00', 'R125', 4),
('G', '06:02:00', 'R126', 4),
('H', '01:02:00', 'R127', 4),
('F', '05:02:00', 'R124', 4),
('E', '04:02:00', 'R128', 4);INSERT INTO enrolled (snum,cname) VALUES
(1, 'A'),
(2, 'B'),
(3, 'C'),
(3, 'D'),
(4, 'D'),
(5, 'E'),
(6, 'A'),
(7, 'B');
```

INSERT INTO faculty (fid,fname,deptid) VALUES

(1, 'RAM', 1),

(2, 'SHYAM', 2),

(3, 'TOM', 3),

(4, 'DOM', 4);

SELECT * FROM student;

```
1|hari|MATH|FR|18
2|mahantesh|MATH|SR|18
3|shreshtha|TFCS|SR|19
4|subhas|TFCS|FR|19
5|saquib|DBMS|JR|20
6|krishan|DBMS|JR|21
7|jaques|ADA|SR|21
8|ashutosh|MATH|FR|18
9|divyanshu|MATH|JR|21
10|derek|MATH|SR|19

[Program exited with exit code 0]
```

SELECT * FROM class;

```
A|01:02:00|R124|1
B|02:02:00|R125|2
C|03:02:00|R126|3
D|03:02:00|R125|4
G|06:02:00|R126|4
H|01:02:00|R127|4
F|05:02:00|R124|4
E|04:02:00|R128|4

[Program exited with exit code 0]
```

SELECT * FROM enrolled;

```
1|A
2|B
3|C
3|D
4|D
5|E
6|A
7|B

[Program exited with exit code 0]
```

SELECT * FROM faculty;

```
1|RAM|1
2|SHYAM|2
3|TOM|3
4|DOM|4

[Program exited with exit code 0]
```

i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by RAM.
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 = 'RAM' AND s.lvl = 'JR';

```
krishan

[Program exited with exit code 0]
```

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 (e.snum)>4);

```
E

[Program exited with exit code 0]
```

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);

```
shreshtha

[Program exited with exit code 0]
```

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 EXCEPT SELECT c1.room FROM class c1 WHERE c1.fid = f.fid);

```
DOM

[Program exited with exit code 0]
```

v. Find the names of faculty members for whom the combined enrollment 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);

```
RAM
SHYAM
TOM
DOM

[Program exited with exit code 0]
```

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);

```
ashutosh
divyanshu
derek

[Program exited with exit code 0]
```

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.age, s1.lvl HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM student s2 WHERE s1.age=s2.age GROUP BY s2.lvl, s2.age)) ORDER BY s.age;

	age	lvl
1	18	FR
2	19	SR
3	20	JR
4	21	JR