

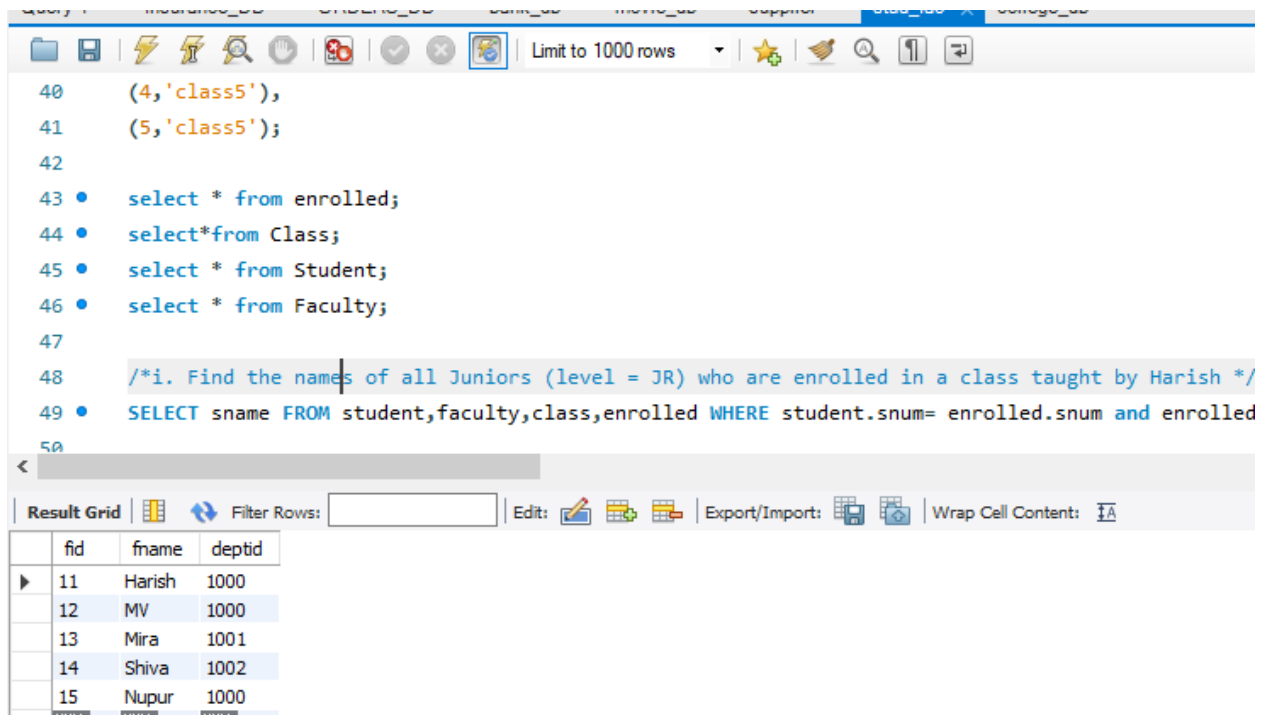
DBMS WEEK 8

Consider the following database for student enrollment 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 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).

OUTPUT:

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



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
40 (4,'class5'),
41 (5,'class5');
42
43 • select * from enrolled;
44 • select*from Class;
45 • select * from Student;
46 • select * from Faculty;
47
48 /*i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Harish */
49 • SELECT sname FROM student,faculty,class,enrolled WHERE student.snum= enrolled.snum and enrolled
50
```

Below the editor is the 'Result Grid' section, which includes a 'Filter Rows' input field and buttons for 'Edit', 'Export/Import', and 'Wrap Cell Content'. The result grid displays the following data:

| | fid | fname | deptid |
|---|-----|--------|--------|
| ▶ | 11 | Harish | 1000 |
| | 12 | MV | 1000 |
| | 13 | Mira | 1001 |
| | 14 | Shiva | 1002 |
| | 15 | Nupur | 1000 |

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

```
50
51 /*ii. Find the names of all classes that either meet in room R128 or have five or more Students
52 • SELECT class.cname FROM class where room='R128' OR class.cname IN(SELECT enrolled.cname FROM enr
53
54 /*iii. Find the names of all students who are enrolled in two classes that meet at the same time
55 • select sname from Student where snum in (select e1.snum from Enrolled e1,Enrolled e2,Class c1,Cl
56
```

<

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

| | |
|---|-------|
| | sname |
| ▶ | Tom |

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

```
57  /*iv. Find the names of faculty members who teach in every room in which some class is taught.*/
58  • SELECT fname FROM faculty WHERE NOT EXISTS(select room from class where room not in(select distir
59
60  /*v. Find the names of faculty members for whom the combined enrollment of the courses that they
61  teach is less than five.*/
62  • SELECT distinct fname FROM faculty WHERE 5>(SELECT count(enrolled snum) from enrolled.class where
```

Result Grid |  Filter Rows: | Export:  | Wrap Cell Content: 

| | |
|---|-------|
| | sname |
| ▶ | Rahul |

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

```
60  /*v. Find the names of faculty members for whom the combined enrollment of the courses that they
61  teach is less than five.*/
62  • SELECT distinct fname FROM faculty WHERE 5>(SELECT count(enrolled.snum) from enrolled,class where
63
64  /*vi. Find the names of students who are not enrolled in any class.*/
65  • select sname from Student where snum not in (select snum from Enrolled);
66
```





Result Grid   Filter Rows: | Export:  | Wrap Cell Content: 

| fname |
|--------|
| Harish |
| MV |
| Mira |
| Shiva |

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

```
59
60  /*v. Find the names of faculty members for whom the combined enrollment of the courses that they
61  teach is less than five.*/
62  • SELECT distinct fname FROM faculty WHERE 5 > (SELECT count(enrolled.snum) from enrolled, class where en
63
64  /*vi. Find the names of students who are not enrolled in any class.*/
65  • select sname from Student where snum not in (select snum from Enrolled);
66
```

<

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

| | 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).

```
67  /*vii. For each age value that appears in Students, find the level value that appears most often. Fo
68  example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you
69  should print the pair (18, FR).*/
70
71  select S.age, S.lvl from Student S group by S.age, S.lvl having S.lvl in (select S1.lvl from Student
72  group by S1.lvl, S1.age having count(*) >= all (select count(*) from Student S2
73  where s1.age = S2.age group by S2.lvl, S2.age));
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

| | | |
|--|-----|-----|
| | age | lvl |
| | 19 | Sr |
| | 20 | Jr |
| | 21 | Sr |

*******WEEK8 ENDS*******