CS304 2012 W2 Midterm 2

Common mistakes and marking explanations

Question 2A

Student (snum, sname, major, level, age)

Enrolled (snum, cname)

Class (name, meets_at, room, fid)

Faculty (fid, fname, deptid)

Q2A. Find the name of the faculty members who taught the most number of unique students and how many students they taught.

Analysis: This question can be answered in three steps:

1) Create a view taught (**fid, cnt**) where cnt is the total number of unique students taught by faculty with ID=fid.
2) Find the max (cnt) in the above view. (aka. **connection** step).
3) Return the **final result** (fname, cnt) that is asked for in the question, by using the max (cnt) calculated in the above step.

The second step is the **connection** between the first and third.

Although this question can technically be answered in a single nested SQL statement, answering it using multiple SQL statements will make it much easier. The official answer consists of two SQL statements:

The first statement corresponds to step 1.

CREATE VIEW taught AS

SELECT f.fid, count(distinct e.snum) as cnt FROM faculty f, enrolled e, class c

WHERE f.fid = c.fid AND c.name = e.cname GROUP BY f.fid

The second statement corresponds to step 2 and 3, in which the nested sub-statement (highlighted in bold font) corresponds to step 2.

SELECT f.fname, t1.cnt FROM taught t1, faculty f

WHERE f.fid=t1.fid AND t1.cnt **>= ALL (SELECT cnt FROM taught t2)**

General Marking Scheme:

1) Step 1 get (fid, cnt): 4 points in total
2) Step 2 connection: 2 points in total
3) Step 3 get final result: 4 points in total

For each step, you lose 1 point for ***each*** minor mistake you make, 2 points for ***each*** major mistake you make, or all points if your corresponding SQL statement or sub-statement is missing or severely wrong.

In the case that you get 0 in a certain step according to this rule, the marker may give you 1 point, *at his discretion*, if your answer reveals that you have got the basic idea for solving this step **and** how to do it in SQL (but failed to solve it reasonably well).

Due to the facts that the three steps are closely related to each other and there is more than 1000 ways to (approximately) solve this question, it is not always possible to assign a mistake to a specific step. You may also have written something that causes multiples mistakes at the same time. In such situations, the marker will classify and weight the mistake at his discretion, with the general goal of maximizing your scores.

Common minor mistakes:

- Missed "distinct" in count()
- Missed a column or wrongly included an extra column that will cause an error.
- Used "fname" rather than "fid" in the first statement. This is incorrect because faculty members may have duplicate names.
- Minor SQL syntax error. The marker usually highlights this in your answer.

Common major mistakes:

- Mismatch between SELECT and GROUP BY statement when any aggregation function is present.
- Missed part of the statement. (i.e.: your answer is incorrect. But it can be fixed by inserting some short SQL strings into your statement.)
- Wrote the wrong statement that returns something very similar but different.
- Major SQL syntax error. The marker usually highlights this in your answer.

Question 2b

Common Mistakes

****

Common incorrect approaches

****

Approach 1:

```
Select sname
From Student s, Enrolled e
Where s.snum = e.snum and e.cname NOT IN (select c.name
                                from Class c
                                where c.fid = 489456522)
Order by sname;
```

Why is this approach incorrect?

- by using s.snum = e.snum in the main select you are eliminating the subset of students who aren't enrolled in any class
- the second select will get the names of the classes not taught by 489456522.
- by using NOT IN in combination with the 2nd select you will get the classes who are not taught by 489456522.
- However there may be students who are taking both classes with that faculty member and classes taught by someone else. This approach will include these students in the answer.

Example: There are three students in your table S1, S2, S3. S1 isn't enrolled, S2 is enrolled in a class C1 taught by someone other than 489456522 and S3 is enrolled in a class C2 taught by 489456522 and class C1. The correct solution should be S1, S2. Table enrolled looks like

```
snum   | cname
------------------
S2     | C1
```

```
S3     | C1
S3     | C2
```

Table Class contains two things: C1 and C2; C2 is taught by 489456522.

The second select will give you C2. Using NOT IN you see that C2 is in the result but C1 is not. So that will eliminate the last row, thus obtaining S2, S3.

Approach 2:

```
Select sname
From Student s, Enrolled e, Class c
where s.num = e.snum and e.cname = c.name and c.fid <> 489456522
Order by sname;
```

This will give you the list of students, who are enrolled in classes (thus eliminating from the answer those who aren't enrolled in any class) that are not taught by 489456522. However there may be students who are taking both classes with that faculty member and classes taught by someone else. This approach will include these students in the answer.

Example: There are three students in your table S1, S2, S3. S1 isn't enrolled, S2 is enrolled in a class C1 taught by someone other than 489456522 and S3 is enrolled in a class C2 taught by 489456522 and class C1. The correct solution should be S1, S2. Table enrolled looks like

```
snum    | cname
------------------
S2     | C1
S3     | C1
S3     | C2
```

By using e.snum = s.snum (take students who are enrolled) you're eliminating S1 from the answer. Using e.cname = c.name and c.fid <> 489456522 you are eliminating the third row from enrolled, thus getting the answer S2, S3.

Question 2D

Common mistakes

Mistake: get set of classes with conflict and check if a student is enrolled in a conflict class.

Need to check that student is enrolled in both classes involved in conflict.

Mistake: projecting out attributes that do not participate in the group by, this is an illegal operation.

example:

```
select sname
from Student
group by snum
```

Mistake: group by snum (and variations, such as group by snum, meets_at)

snum is unique, no groups will be aggregated.

Mistake: misuse of IN, EXISTS, =

x IN y; checks that an attribute x is in the set y

ex: snum IN (select e1.snum from ...)

EXISTS y; checks that the set y is not null

x = y; checks x is equal to y, x and y should be definite values, not sets