

## More Queries Using SQL (#3)

Code and test the SQL statements needed to generate the following five queries. You should use the SQL test page linked from the class web page. A query should work properly in both Oracle and MySQL. (You may make two queries, one for each, if you cannot reconcile the two.) Place, in comments, any discrepancies that you found between the two that were not covered in class. When these are complete place in a single text file and insert comments to identify yourself and the problem number for each query. You should put in a comment how many rows matched the query. Email the results to the instructor.

1. The college wants to contact students for whom they have no address. Since they cannot send things to such an address they want to contact the teachers who have these students. Generate a query that lists the faculty name and the student name. Is there anything odd about the students with no address?
2. A faculty member hates redundancy and wants to change the name of any course that starts with the words “College” or “University” since all courses should be at that level. Generate a query that finds all such courses and the faculty members that teach them. The query should show the course title, department and number, as well as the faculty member’s name.
3. Make a transcript-like table. It should contain a student’s name and address, the number of courses taken, the sum of the credit hours and the GPA in percent. The GPA will not be on the traditional four point scale but on the 100 point scale that we see in the database. You should round it to just two digits after the decimal. First hint: the GPA is **not** the average of the scores. Instead you need to multiply each score by the credit hours of that course and then divide the sum by the total credit hours. Second hint: You may want to display the students and scores so that you can hand calculate a couple of students that have taken multiple course that have different hours. Third hint: Oracle will require that you include both name and address within the group by, while MySQL will not.

(More problems on back)

Any student that did not receive a C or below in any class. No student should be shown twice. There are two ways to do this and we will do each for the next two problems.

4. The first way is to use a Group By with a Having and the aggregate function min.
  
5. The second way is to use a nested query