Database Systems
Project 1: Writing SQL Queries in Oracle

The following normalized tables from the Student Registration System (some tables have been simplified) will be used in this project:

> Students(<u>sid</u>, firstname, lastname, status, gpa, email)
> Courses(<u>dept_code, course#</u>, title)
> Course_credit(<u>course#</u>, credits)
> Classes(<u>classid</u>, dept_code, course#, sect#, year, semester, limit, class_size)
> Enrollments(<u>sid, classid,</u> lgrade)
> Grades(lgrade, ngrade)

As a general clarification, we assume that no student takes the same course (including different sections of the same course) more than once. If you have questions about these tables, please contact the instructor for clarification.

1.   Preparation

Save the sql script file proj1_tables_script16.txt as proj1_tables_script16.sql in your bingsuns account.

To run the above script from your Oracle account, use

SQL> start proj1_tables_script18

Then check whether all tables are created correctly in your Oracle account.

2. Query Statements

There are 20 statements in this project and you are asked to write one or more SQL query for each statement. <u>If you are a CS432 student, write queries for statements 1 – 15. CS532 students should write all 20 queries.</u>

Your queries should consider that the tuples currently in the database may change. In other words, your queries must be correct for any valid state of every table. For this project, it is required that each query returns no identical results (i.e., identical rows in the result must be removed). Meanwhile, it is required that you don't use unnecessary "select distinct" for your query. The query (or queries) for each statement is worth 5 points. Unremoved duplicates or unnecessary use of "select distinct" will result in 1-point deduction. No other partial credit will be given in general. You are encouraged to use ways other than "select distinct" to remove/avoid duplicates.

It is suggested that you first test each query individually and save each query in a different file (with extension .sql). After all queries have been tested to your satisfaction, you can run all the queries in a sequence and save the entire session in a spool file. Suppose you have saved your queries in files query1.sql, ..., query20.sql. Follow the steps below to generate the spool file after you have logged on Oracle:

> SQL> set echo on
> SQL> spool project1.txt
> SQL> start query1
> ....
> SQL> start query20

SQL> spool off

Now print the file project1.txt, add your name at the beginning and hand in the printout. Your printout must list each SQL query followed by the result of executing the query.

The 20 query statements follow:

1. Find the dept_code, course# and title of each course that was offered in the Spring semester of 2018. In the output, dept_code and course# of each course should be concatenated under a new column header course_id.

2. Find the first name of each student who has taken at least one CS course and at least one math course.

3. Find the dept_code and course# of each course that was not offered in 2017.

4. Find the sid, firstname and GPA of each undergraduate student who has received an A for at least one course he/she has taken.

5. Find the firstname of each student who has never received an A for any course he/she has taken. If you write a nested query, make sure the subquery is uncorrelated.

6. Find the sid and firstname of each student who has received an A for every course he/she has taken. Count only classes for which he/she received a non-null grade. GPA information is not permitted to be used in this query.

7. Find the dept_code and course# of each course that has been offered the most number of times (each record in the classes table corresponds to a course offering). Note that it is possible that more than one course may satisfy this query condition; in this case, all such courses should be retrieved.

8. Find the classid, dept_code and course# of each class offered in Spring 2018 that is not full and for each such class, also list the number of seats available (computed by limit – class_size) under the header "seats_available'.

9. Find every student (all attributes are needed) who has taken more than 4 classes.

10. Find every class (all attributes are needed) that is offered by the CS department in the Spring semester of 2018 and has less than 3 students enrolled. For this query, you are not allowed to use the size information from the classes table.

11. Find the sid and first name of every student who has taken all 400-level CS courses. Here we are referring to courses, not classes.

12. Find the title of each course that has been taken by student B001 but not by student B002.

13. Find the first name of every student who has taken at least one course that has been taken by student B005. Note that here we are talking about taking the same course, not just the same class. (Clearly, student B005 satisfies the condition and his first name should be included in the output.)

14. Find the dept_code and course# of each course that has two or more classes in the same semester of the same year. The query should also show the semester and year information for each qualified course.

15. Find the sid and firstname of each student who has received at least one highest grade in one of the classes he/she has taken. Suppose all possible grades are (A, B, C, D, F, I). Note that the highest grade given to students in a class is not necessarily A, for example, when all students in the class did poorly.

16. List the dept_code, course# and title of each course that has been taken by student B003. For each such course, also list the grade the student received. If no grade has been assigned for a course, output "to be assigned" as the grade information for the course.

17. Find the dept_code, course# and title of each course whose title contains "systems" and that has been taken by all students whose GPA is higher than 3.25. Note that even though a qualified course is required to be taken by all students whose GPA is higher than 3.25, it may also be taken by some students whose GPA is not higher than 3.25.

18. Find the sid, firstname and the total number of credits that have been earned by each student. If a student has not taken any course, he/she is assumed to have earned zero credits and zero credits should be reported for such a student. Don't count the credits when no grade is given for a class.

19. Find the average total number of credits earned by students (the average is over all students and only one value should be computed) who have taken at least one course. Don't count the credits when no grade is given for a class.

20. Compute the GPA for each student from the student's number grades (ngrade) for all the courses he/she has taken (ignore the GPA values already in the students table). The GPA of a student is computed by dividing the sum of his/her number grades by the number of classes he/she has taken and received a non-null number grade. If a student has not received any non-null grade yet, the student's GPA will be null. For each student, the sid of the student and the computed GPA (name column head as cgpa) should be displayed. Display the results in descending non-null GPA values.