Database Systems
Project 1: Writing SQL Queries in Oracle
Max Total: 15 points

The following six normalized tables will be used for this project. These tables are either directly from Homework 2 solutions or modified by normalizing them.

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

The SQL statements for creating and populating these tables are provided in the file SRSTablesScript.sql, which is available under Project 1 on Brightspace. Copy and save SRSTablesScript.sql file to your harveyv account. Run RBMSTablesScript.sql file to create a database required for this project. **No changes are allowed to these tables for this project.**

There are 15 statements in this project, each of which is 1 point. You need to write a single SQL query for each statement unless specified otherwise. Some questions may ask you to write more than one SQL query in different styles. You are not allowed to create views or other (temporary) tables for this project. Inline views (i.e., select in the "from clause") are allowed. Your query should take into consideration that the tuples currently in the tables may change. In other words, your solution to each question must be correct no matter what valid tuples are in the tables.

I suggest you write and test each query and save it in a different file (with the .sql extension) in your harveyv account (e.g., SQL> save query1.sql). Suppose you have saved your queries in query1.sql, … query15.sql. Follow the steps below to **generate the spool file** after logging into your Oracle account:

```
SQL> set echo on
SQL> spool project1.txt
SQL> start 1
SQL> start 2
.......
SQL> start 15
SQL> spool off
```

For each of the 15 statements, your output in the spool file needs to show both the query and the result of the query. This can be achieved by "set echo on" as shown above. Before you submit the file project1.txt, you need to add your name, B-number, and the following statement "*We have done this project entirely on our own. We have neither copied the work of others nor shared our solution with anyone else. We understand that if we are involved in plagiarism or cheating, we will be required to sign an official form admitting to the violation, which will be added to our official university records. We also understand that we will receive a grade of 0, our letter grades will be reduced by one level for our first offense, and any additional offense of any kind will result in a grade of 'F'.*" at the top of the file. Remember to (digitally) sign the statement before submitting the file.

The 15 query statements for this project follow:

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

2. Find the classid, dept_code, and course# of each course that was not offered in Spring 2024.

3. Find the first name and last name of each student who has never received a C for any course he/she has taken. If you write a nested query, make sure the subquery is uncorrelated.

4. Find the first name of each student who has received a B 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. If multiple students have the same first name, return the first one only, while removing the others with the same first name. Do this without using ‘distinct’ in your query.

5. Find the cid, dept_code, and course# of each course that has been offered the smallest number of times (each record in the classes table corresponds to a course offering). If more than one course satisfy this query condition, retrieve all such courses.

6. Find the sid and first name of every student who has taken more than 2 classes.

7. Find every class (all attributes are needed) offered in the Spring 2025 and has less than 2 students enrolled. For this query, you are not allowed to use the size information from the classes table.

8. Find the sid and last name of every student who has taken all 200-level Math courses. Here we are referring to courses, not classes. (A course, e.g., CS 240, can be offered in different semesters where the different implementations of the course are different classes.)

9. Find the title of each course that has been taken by student B005 but not by student B003.

10. Find the last name of every student who has taken at least one course that has been taken by student B002. Note that here we are talking about taking the same course, not just the same class.

11. 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.

12. Find the sid and first name 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). 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.

13. List the dept_code, course#, and title of every course that has been taken by student B005. For each such course, also list the grade the student received. If the grade is null for a course, output “TBA” as the grade information for the course.

14. Find the dept_code, course# and title of each course whose title contains “data” and that has been taken by all students whose GPA is 3.0 or higher. Note that even though a qualified course is required to be taken by all students whose GPA is 3.0 or higher, it may also be taken by some students whose GPA is not higher than 3.0.

15. 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, last name, and the computed GPA (name column head as cgpa) of the student need be displayed. Display the results in ascending non-null GPA values.