

# CS54100 - Programming Assignment 1

## Fall 2016

Due: September 30, 11:59 PM

(There will be a 10% penalty for each late day up to five days. The assignment will not be accepted afterwards.)

---

### Part1: (50 Points)

#### Introduction

Each student has to do this project individually. You are going to use Oracle to perform some queries and create views for a database. The schema and sample data of the database are provided. The project should be run on CS sun workstations.

Your login information can be found on [my.cs.purdue.edu](http://my.cs.purdue.edu), under the *Local Accounts* page. Information about general initial configuration is available in: <https://www.cs.purdue.edu/resources/facilities/oracle.html>

---

#### Your assignment

In this project you will use the file: db.sql. Copy this file into your working directory. Create and populate tables by the following command:

```
SQL>@db
```

File db.sql will create the tables needed for this assignment. It will also fill the tables with some sample data. This will help you test your queries. The current data does not cover all the possible scenarios that the queries address and they are not completely matching the examples shown in this handout. So, feel free to add additional tuples to test some corner cases. In order to grade the assignment, the TA will be using a different data set for testing your queries. So, make sure to cover all cases.

The following section shows the schema of the database. Study the schema carefully.

**CourseRegistry** is a database used by a university to keep track of its enrollment information. CourseRegistry keeps track of the courses, faculty, students, and which students are enrolled in which courses. The back-end of the CourseRegistry database consists of the relations defined in the following schema:

Students(snum, sname, major, standing, age, gpa)

Faculty(fid, fname, deptid)

Courses(cnum, cname, course\_level, credits)

Offerings(onum, cnum, day, starttime, endtime, room, max\_occupancy, fid)

Enrolled(snum, onum)

- Relation Students contains information about each student. The snum attribute is the primary key. The standing attribute records the grade of the student (e.g. 'FR', 'SO', 'JR', 'SR', 'GR' = Freshman, Sophomore, Junior, Senior and Graduate, respectively).
- Relation Faculty contains information about the faculty and which department they are in. The fid attribute is the primary key.
- Relation Courses contains information about each course in the course catalog for the university. course\_level is either 'UG' (undergraduate) or 'GR' (graduate). An undergraduate student cannot enroll in a graduate-level course. Credits is how many credit hours the course is worth. The cnum attribute is the primary key.
- Relation Offerings records information about the specific course offerings for the current semester. The cnum attribute is a foreign key referencing Courses, and fid is a foreign key referencing Faculty. day is the day of the week the offering is held, either MWF or TR. starttime and endtime are stored as the DATE datatype. room is the location of the course. max\_occupancy is how many students can be allowed to enroll in the course. You can assume that no courses will meet in the same room at overlapping times, no faculty member will teach courses held at overlapping times, and that no course offering will have more than the maximum occupancy currently enrolled.
- Relation Enrolled records information about which students are enrolled in a course. The snum and onum attributes are foreign keys referencing Students and Offerings, respectively. You can assume that no student will be enrolled in courses with overlapping times, and no student will be enrolled in multiple offerings of the same course.

## Queries

Write SQL queries that answer the questions below (one SQL query per question but you are allowed to use nested queries and/or the "WITH" clause of Oracle) and run them on the Oracle system. The query answers should be duplicate-free, but you should use distinct only when necessary.

If you are making any assumptions, state them clearly and document your queries.

1. Find the names as well as the GPAs of the students who are enrolled in only one course this semester.
2. Find the names of the courses where there exists at least one course offering that only has enrolled students that are all in the same major.
3. For each course, display its name as well as the number of offerings of that course this semester.
4. Find the name(s) of the course(s) that have an offering with the largest number of students enrolled in this semester. If there are ties, display all courses that qualify.
5. Find the name(s) of the undergraduate course(s) with the most graduate students. You can assume that at least one undergraduate course has at least one graduate student.
6. For each student **S**, display the name of **S** as well as the number of courses **S** is enrolled in. Also display the total number of credits hours **S** is taking this semester.
7. For each faculty member **F**, display **F**'s name as well as the average number of students currently enrolled in the offerings **F** teaches this semester.

8. Find the names of the faculty who only teach offerings with full occupancy this semester.

## Drop

Drop all tables. Use statement “select \* from user\_catalog;” to make sure that all the objects are dropped. You can use the droptables.sql file.

---

## What to submit

The result of your work is a file named **p1\_your\_career\_login.sql** which contains all the SQL statements you used in this assignment. Not naming your files as your **p1\_your\_career\_login.sql** may result in penalty points.

---

## A useful strategy

Here are some useful approaches for doing the project

1. Follow the introduction about the environment setup, connect to the Oracle server with your assigned Oracle account.
2. Try a few simple SQL statements until you are comfortable interacting with sqlplus.
3. Workout the SQL statements you need to solve the above queries
4. Use a text editor you are familiar with to create a .sql file that contains the necessary SQL statements for this project.
5. Test your .sql file
6. Add the following lines in the beginning of your .sql file (**mandatory step**):

```
rem CS541 SQL Project 11
rem your_first_name    your_last_name
rem your_Purdue_email_address
```

7. Work incrementally. Test your .sql file continuously as you add the new SQL statements. You can use the oracle command spool to direct the output to a file that you can check. However, please do not submit the output file.
- 

## PART2 (50 points)

### Description:

Writing Oracle Functions and Procedures to Process Data.

You are going to use PL/SQL (Oracle’s procedural extension to SQL) to write a few functions and procedures to process data. The result of this part of the project should be a file named **p2\_your\_career\_login.sql** that will contain all the PL/SQL statements you develop to handle

the data processing tasks described below. Your grade depends on how you use the PL/SQL statements and the result of running the .sql file. You will use the same database used in PART1 of this project.

**It is required that you perform the following tasks (notice that the results shown in these exercises do not necessarily correspond to the actual values obtained from the given database):**

Create and populate all tables (This should be already done after Part1).

1. Write Procedure sp\_univ\_students that generates information about the student body of the university

| Standing | NumOfStudents | AvgGPA | AvgCreditHours |
|----------|---------------|--------|----------------|
| FR       | 300           | 3.21   | 12.0           |
| SO       | ...           | ...    | ...            |
| JR       | ...           | ...    | ...            |
| SR       | ...           | ...    | ...            |
| GR       | ...           | ...    | ...            |

This procedure gives a list of information of each class standing in the university. The **NumOfStudents** indicates the number of students of each standing. **AvgGPA** and **AvgCreditHours** indicates the average GPA and average number of credit hours taken, respectively, of the students in that standing.

2. Write Procedure sp\_course\_registry that generates the following report about the course offerings and enrollment information for the semester

Faculty Member: Ivana Teach

| OfferingNumber | CourseName       | Day             | Time       | Room     |
|----------------|------------------|-----------------|------------|----------|
| 7              | Database Systems | MWF             | 12:30-1:45 | 1320 DCL |
|                | StudentID        | StudentName     | Standing   |          |
|                | 1                | Joseph Thompson | SO         |          |
|                | ...              |                 |            |          |
| OfferingNumber | CourseName       | Day             | Time       | Room     |
| ...            |                  |                 |            |          |

Faculty Member: Mary Johnson

...  
.....

This procedure generates a report of all faculty, the offerings they teach, and the students enrolled in those offerings. Faculty are ordered by faculty ID, course offerings should be ordered by course name and then offering number, and enrolled students should be ordered by student number. Time consists of the starttime and endtime, formatted as an interval.

3. Write Procedure `sp_avail_course` that generates for each student, say **S**, the course offerings that **S** is able to enroll in. **S** is allowed to enroll in a course offering if the number of current students enrolled is not already at the maximum occupancy, **S** is not currently enrolled in another offering of the same course, and if **S** is not enrolled in any other course offering that has an overlapping meeting time. In addition, undergraduates cannot enroll in graduate-level courses.  
The report should be in the following format:

Student ID: 1  
Student Name: Joseph Thompson  
Standing: SO  
Course: Database Systems

| OfferingID | Instructor   | Day | Time       | Room     |
|------------|--------------|-----|------------|----------|
| 7          | Ivana Teach  | MWF | 12:30-1:20 | 1320 DCL |
| 12         | Mary Johnson | TR  | 1:30-2:45  | 1320 DCL |

Course: ...  
.....  
...

Student ID: 2  
Student Name: Angela Martinez  
Standing: SO

...  
.....

This report is ordered by student ID, then by course ID, then by offering ID. If a course has no offerings available that the student can register in, you should not include it.

## DROP

Use the command DROP to drop all procedures and tables.

---

## A useful strategy

Here is a useful approach for doing the project:

1. Work through the examples in Chapter 3 of “Oracle 10g Programming: A Primer, Rajshekhar Sunderman” to become familiar with the PL/SQL syntax and creation of functions and procedures.
2. In sqlplus command mode, type “set serveroutput on size 32000”
3. Create and run simple procedures to study their behavior.
4. Always have at least two terminal windows open so that you can run sqlplus in one window and text editor on the other. Use the text editor to create .sql files to try out the functions and procedures.
5. When you build a procedure, make sure that your .sql file starts with the “CREATE OR REPLACE PROCEDURE ... ” statement, ends with the “/” to compile the procedure, and nothing else. This will make it much easier to debug your procedure. If it fails to compile, you can use the SQL command “show error” to get the detailed compilation error message that gives the line number and column number of your PL/SQL statement that has syntax problems.
6. Add the following lines to the beginning of your .sql:

```
rem CS 541 SQL Project 12
rem your_first_name    your_last_name
rem your_purdue_email_address
```

so that the TA can assign the grades.

(Note: Your grade may be affected if you do not follow this)

---

## What to submit

You are going to submit ONLY ONE file for this part, p2\_your\_career\_login.sql. It should contain the following:

- @ command to run db.sql
- Function and procedure definitions
- Execute statements to execute the procedures in order
- Drop statements to clean up your database (you can just run droptables.sql)

The TA will run your submitted .sql file to grade your project.

---

## How to submit your .sql files for Parts 1 and 2

After you are satisfied that your .sql files for both Part1 and Part2 perform the required functions, you need to create one zip file that contains both sql files for Parts1 and 2. Submit your zip file from your CS54100 account in BlackBoard before the deadline. For any questions regarding this project, please contact the TA.