CS54100 Project1 - SQL Programming

Due: 11:59PM EST, Monday February 9, 2015
(There will be a 10%penalty for each late day)

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 done in CS sun workstations.
Information about getting your Oracle account and 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: script1. Copy this file into your working directory. Create and populate tables by the following command:

```
SQL>@script1
```

File script1.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 may not cover all the possible scenarios that the queries address. So, feel free to add additional tuples to test some corner cases. In order to grade the assignment, the TAs 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.
The CS-Grid is a social network for CS students all over the world. The back-end database of the CS-Grid consists of the relations defined in the seven-schemas as follows:
STUDENT(SID,NAME,SCHOOLID,COLLEGEYEAR)
FRIEND(SID1,SID2,DESCRIPTION)
SCHOOL(SCHID,NAME,ADDRESS,RANK)
COURSE(CID, Title)
STUDIED (CID, SID)
COMPANY(CMPID,Title,ADDRESS)
INTERN(SID,CMPID,STARTDATE,ENDDATE)

- Relation STUDENT contains information about CS students, Attribute SID is the primary key. The SCHOOLID is a foreign key to the SCHOOL table. Attribute COLLEGEYEAR represents the current school level the student is in (i.e., Freshman, Sophomore, Junior, or Senior).
- Relation FRIENDS contains information about every two friends in the CS-GRID. Attribute DESCRIPTION says how students know each other (interns,

friends, family). Attributes SID1 and SID2 are foreign keys to Relation STUDENT.

- Relation SCHOOL contains information about the schools of the students. Attribute SCHID is the primary key of this relation. Attribute RANK represents the ranking of the school. Rank 1 is the top-ranked school, 2 is the second, and so on.
- Relation COURSE contains information about various CS courses. Attribute CID is the primary key of this relation.
- Relation STUDIED contains information about what courses were taken by a specific student. Attribute CID is a foreign key to Relation COURSE, while Attribute SID is a foreign key to Relation STUDENT.
- Relation COMPANY contains information about companies that hire CS students as interns. Attribute CMPID is the primary key for this relation.
- Relation INTERN contains information about companies that students have interned in. A student may have an internship with the same company multiple times.

Queries

Write SQL queries that answer the questions below (one SQL query per question) 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 of the students who took only one course.
2. Find pairs of students (print their names) that have the most number of shared friends while they are not friends.
3. Find the names of companies that only intern students from the three top-ranked schools.
4. For each school, say s, find the number of friendship relationships where both friends are from school s.
5. Find the pair of companies and schools that have the most number of distinct internships (if a student in a certain school interned at the same company more than once, count it as a single internship).
6. Find schools that teach the most number of courses.
7. For each company, find the average internship duration, do not consider internships that do not have an ENDDATE.
8. Find pairs of school names that have the most number of students that interned at the same company.
9. Find course names taken by students that interned at the companies that had the most number of distinct internships (if a student interned at the same company more than once, count it as a single internship).

Drop

Drop all tables. Use statement “select * from user_catalog;” to make sure that all the objects are dropped.

What to submit

The result of your work is a file named **your_career_login_Part1.sql** which contains all the SQL statements you used in this assignment. Not naming your files as your **your_career_login_Part1.sql** will cost you several 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:

   rem CS 541 SQL Project 11
   rem your_first_name your_last_name

   so that the TA can assign the grades.

7. Remember to divide and conquer. 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 it however, please DO NOT submit the output file .DO NOT include spool in your .sql 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 **your_career_login_Part2.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 pro_comp_report that generates company internship information, print the output of your procedure in the following format.

CompanyTitle	NumOfInterns	NumOfShools	AvgInternDuration	TopSchoolInterned	SchoolWithMostIntern
Google	1500	120	MM-DD	Purdue	Illinois
....					
....					

This procedure gives a list of the companies and information about the interns at each company. The **TopSchoolInterned** indicates the top ranked school from which this company had interns. **SchoolWithMostIntern** indicates the school with the highest number of students that have interned at this company. The MM-DD represents a duration in months and days.

2. Write Procedure pro_friend_suggestion that generates the recommendations for friends for every student. Print the output of your procedure in the following format:

Student ID: 1
Student Name: Alex

FriendID	FriendName	NumOfSharedFriends	NumOfSharedCourses	SchoolName
321	Jake	8	2	Purdue

CompaniesInterned:

CompanyID	CompanyName	InternshipDuration
88	Facebook	MM-DD
99	Google	MM-DD

FriendID	FriendName	NumOfSharedFriends	NumOfSharedCourses	SchoolName
568	Jones	7	8	Purdue

CompaniesInterned:

CompanyID	CompanyName	InternshipDuaration
99	LinkedIn	MM-DD

Student ID: 2
Student Name: Kim
…...

This procedure generates friend recommendations for every student. This report is ordered by student ID. MM-DD represents the duration of an internship in months and days. The suggested friend list should be ordered based on the following: (number of shared friends, number of shared courses). The suggested friend list should not contain any of the students' current friends. The suggested friends list should contain only students from the same school and they either have common friends, or that have interned in the same company. In other words, a student, say S1, in the suggested friendship list of another student, say S2, should satisfy at least one of the following criteria: (1) S1 and S2 have a common friend, or (2) Both S1 and S2 have interned in the same company.

3. Write Procedure pro_school_report that generates the following report about schools:

SchoolId: 12
SchoolName: XYZ

TotalNumOfStudents	NumOfStudentsInterned	SchoolRank

541	200	20

CompaniesInterned:

CompanyID	CompanyName	NumOfStudents
88	Facebook	150
99	Google	60

SchoolId: 13
SchoolName: YZW
…….
This procedure generates a report about every school and the companies students go for internships in that school. The **CompaniesInterned** list is ordered based on the number of students that interned at this company. The total **NumOfStudentsInterned** represents the number of students that have at least one internship (Students with multiple internships are counted as a single occurrence). The **NumOfStudents** per company counts distinct internships (Students with multiple internships at the same company are counted as a single occurrence).

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 Sunderrman” 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 behaviour.
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
so that the TA can assign the grades.
(Note: Your grade may be deducted 5% off if your do not follow this)

What to submit

You are going to submit ONLY ONE file for this part, your_career_login_Part2.sql. It should contain the following:

- @ command to create and load the required tables
- Function and procedure definitions
- Execute statements to execute procedures
- Drop statements to clean up your database

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 TAs.