

CS44800 - Project 1

Spring 2016

Due: Monday 1st February, 2016, 11:59PM

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

Introduction

Each student has to do this project individually. You are going to use Oracle to perform some queries against a predefined database. The schema and sample data of the database are provided. The project should be run on 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: 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 testing scenarios. 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.

ProjectsInfo is a database used by a software development company to keep track of its projects. ProjectsInfo keeps track of the projects, the employees/managers working on the projects including the universities they were graduated from. The back-end database of the ProjectsInfo consists of the relations defined in the following schema:

University(UnivId, UnivName)
Department(DeptId, DeptName)
Employee(EmpId, EmpName, DeptId, HomeZipCode)
Project(ProjId, ProjName)
Graduate(EmpId, UnivId, GradYear)
EmpProject(EmpId, ProjId, StartDate, EndDate)
ProjectManager(ProjId, MgrId, StartDate, EndDate)

- Relation University contains information about the universities where the employees graduated from. Attribute UnivId is the primary key.
- Relation Department contains information about the different departments in the company. Attribute DeptId is the primary key. The name of a department is unique.
- Relation Employee contains information about the employees in the company (including managers). Attribute EmpId is the primary key.
- Relation Project contains information about the projects that are running or that have been completed by the company. Attribute ProjId is the primary key. The name of a project is unique.
- Relation Graduate contains information about the university as well as the graduation year of each employee. It is assumed that each employee is graduated by one degree from one university. Thus, the primary key is defined to be EmpId.
- Relation EmpProject contains information about all the projects an employee is/was working on. The primary key is composed of the three attributes: EmpId, ProjId, StartDate as an employee can rejoin a project s/he was released from. A NULL value in the EndDate attribute indicates "Current", i.e., the employee is currently working on that project.
- Relation ProjectManager contains information about all the managers of each project. A project has only one manager at a time, but the project can have different managers at different non-intersecting time frames. A manager is identified by Attribute MgrId and references the EmpId attribute of the Employee relation. The primary key is composed of the three attributes: ProjId, MgrId, StartDate. A NULL value in the EndDate attribute indicates "Current", i.e., the manager is currently managing that project.

Part 1: SQL Queries (60 Points)

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 of the employees who are living in West Lafayette (Zip code 47906 or 47907).
2. Find the names of the projects that are **currently** managed by any manager.
3. For each project, display its name as well as the number of employees who are currently working on it.
4. Find the name(s) of the university/universities that graduated the maximum number of distinct managers.
5. For each employee, say **E**, display the name of **E**, the department name of **E**, and the graduation year of **E**.
6. Display the name of the project that has the maximum number of different employees who worked/"are working" on it. If more than one project qualify, display all the qualified projects. (Hint: refer to EmpProject only and not ProjectManager).

Part 2: SQL Updates and Deletes (40 Points)

1. Employee with Id 2 has moved to an address in 47907. Please update this in the database and select that employee after the update to ensure that the corresponding tuple has been updated.
2. Just as a practical joke, increment the graduation year of every graduate who graduated before 2002 by three.
3. Just another practical joke, decrement the graduation year of every graduate who lives in 47907 by two.
4. For some reasons, the database owner wants to get rid of everything related to the project with Id 2. Please delete all the tuples related to Project 2 (as if that project has never existed before).

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 (call droptables.sql at the end of your submission 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 CS448 SQL Project 1
rem your_first_name    your_last_name
rem your_Purdue_email_address
```

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. However, please do not submit the output file.

How to submit your .sql file

After you are satisfied that your .sql file performs the required functions, you need to create one zip file that contains the sql file as well as a readme file. The readme file should contain your name, your Purdue email address as well as any other information you would like the TAs to know. Submit your zip file from your CS44800 account in BlackBoard before the deadline. For any questions regarding this project, please contact the TAs.