

CSci 4707

Lab 1

Fall 2016

Due Thursday, 10/13/2016 12:55pm

Instructions

- a. Please read this document carefully for the things to do and submission guidelines.
- b. Not following the submission guidelines will result in penalty.
- c. This lab must be done individually.
- d. The best way to get your questions answered is to ask them on moodle forums. That way it helps other students who are facing similar problems.

Goal

In this lab, you will experience writing SQL queries in Oracle.

This lab is done individually. You should have received an email from Harshada Chavan about your Oracle account. If you haven't received them, please email at chava057@umn.edu as soon as possible.

Oracle Guidelines

1. To load Oracle using your CSELABS account, you will need to run the following module at the terminal: **`module load soft/oracle`**
Note: you will need to run this command every time you login. You can add `soft/oracle` in the `.cshrc` file on your home directory in order to skip this step every time you login. Also, this command will only run on CSELABS machines. Therefore, please make sure that you are logged into one of the CSELABS machines before executing it.

2. To login into Oracle use: `"sqlplus <userid>@o11g"`

Then, enter your password. **The password was initially set as the <userid>.**

3. **First time only!** When you login to Oracle for the first time, please change your password by using the following command:

```
"ALTER      USER      <userid>      IDENTIFIED      BY  
<new_password>;"
```

4. To quit Oracle, type: `"quit;"` or `"exit;"`

IMPORTANT: Remember to always use those commands to exit Oracle. Don't exit by just closing the terminal or exit from the SSH terminal since your account might get locked by Oracle and you cannot login for several days.

Don't forget the semicolon (;) after each SQL statement. If backspace doesn't work, use Ctrl + H instead.

Suggestion: Writing the query in the terminal directly is a pain. You can try to create an SQL script (xxx.sql) and write the SQL command in the script. To test, just copy the command from the script to the terminal and execute it.

Important for submission: To run the whole command in a SQL script, type the following command: `"@<script_path>.sql"`

Things to do

Schema

Consider the following database schema (primary keys are underlined):

- Emp(eid:integer, ename: VARCHAR(15), age:integer, salary:integer)
- Dept(did:integer, budget:integer, managerid:integer), managerid is a Foreign Key to Emp(eid).
- Works(eid:integer, did:integer) , both are foreign keys.

Part A

Provide a CREATE TABLE command for all tables above. Note: You must also capture the FOREIGN KEY constraint.

Please download the Lab1.tar.gz from moodle. It contains **insert.sql** file which has insert statements that will load data into your tables. Execute this file first before running any queries from PART B.

Part B

Provide SQL queries for the following queries:

1. Print the names and ids of each employees who work on both did = 1 and did = 2.
2. For each department with more than 10 employees, print the *did* together with the number of employees that work in that department.
3. Print the name of each employee whose salary exceeds the budget of all of the departments that he or she works in.
4. Find the *managerid* of managers who manage only departments with budgets greater than \$50 million.
5. Find the *ename* of managers who manage the departments with the largest budgets.
6. If a manager manages more than one department, he or she *controls* the sum of all the budgets for those departments. Find the *managerid* who control more than \$50 million.

7. Find the *managerid* of managers who control the largest amounts.
8. Find the *ename* of managers who manage only departments with budgets larger than \$30 million, but at least one department with budget less than \$50 million.

Submission Guidelines

- The lab **must** be submitted in “.zip” format via Moodle with the following **naming convention**: “lab1.zip”
- The zip file should contain four files (again, please follow the naming convention):
 1. “partA.sql”: It contains all CREATE TABLE commands for PART A of the assignment. The script must be able to be executed in Oracle.
 2. “partB.sql”: It contains all SQL queries for PART B of the assignment. The script must be able to be executed in Oracle using “@<script_path>.sql” command.
 3. “README.txt”: This should include your name and your X500.

Grading

The grading will be done in the following way:

- 10 Points: Submit the assignment in the correct naming convention.
- 10 Points: PART A Script.
- 80 Points: PART B Script.

Both PART A and each SQL query in PART B will not have any partial credit in case of an incorrect result.