CS 340 Programming Assignment 1 - SQLPLUS

Due: 11:59 pm, October 12, 2015

A Simple Oracle Database Design

You are going to use Oracle to design a simple database. You will create tables, views, and implement some queries.

Connecting to Oracle Database

- Open the **.bashrc** file in any text editor (e.g., Emacs).
- Write the following statements at the end of the **.bashrc** file and close it after saving.

```
export PATH=/usr/lib/oracle/11.2/client64/bin:$PATH
export LD_LIBRARY_PATH=/usr/lib/oracle/11.2/client64/lib:$LD_LIBRARY_PATH
```

- Type **source .bashrc** on the Shell prompt to update the PATH and LD_LIBRARY_PATH variables.
- Type echo $PATH on the Shell prompt to print the PATH variable. It should include the path /usr/lib/oracle/11.2/client64/bin
- Type echo $LD_LIBRARY_PATH on the Shell prompt to print the LD_LIBRARY_PATH variable. It should include the path /usr/lib/oracle/11.2/client64/lib
- To connect to the database using SQLPLUS client, use the following statement on the Shell.

```
sqlplus <db_username/password>@localhost:1521/CS340f15
```

Your db_user name is the same as your login to the server (203.135.63.67). Your Oracle DB password is initially set to abcd1234. A student with Registration number 15100281 will use the following statement to connect to the database

```
sqlplus s15100281/abcd1234@localhost:1521/CS340f15
```

Once, you are connected to the database you can create tables, insert record in the tables and query the database.

Currently your password to the CS340f15 database is set to abcd1234. To change the password, use the following command on SQL prompt:

```
ALTER USER user_name IDENTIFIED BY new_password;
```

Please make sure that you remember your password.

To disconnect from the database, type quit at the SQL prompt.

- Use a text editor you are familiar with to create “.sql” files that contain the necessary SQL commands for creating tables, inserting records or querying the database.
- To execute an sql command in the <filename>.sql file, type @<filename> on the SQL prompt. For example, if your .sql file is saved as queries.sql, then type "@" followed by queries without the .sql extension. It looks like:

  ```
  SQL> @queries
  ```

Schema for Programming Assignment 1

- Create the following tables described below. Name these tables STUDENT, CLASS, ENROLLED, FACULTY and PREREQUISITE.
  - STUDENT(sid: number, sname: varchar2(15), major: varchar2(10), slevel:varchar2(10), age: number);
  - CLASS(cnum: varchar2(10), meets_at: date, room : varchar2(10), fid: number);
  - FACULTY(fid: number, fname: varchar2(20), dept: varchar2(20));
  - ENROLLED(cnum:varchar2(10), sid:number)
  - PREREQUISITE(cnum:varchar2(10), prereq:varchar2(10))

The underlined fields are the primary keys of their respective tables

NOTE: The PREREQUISITE table lists the immediate prerequisite course(s) of a given course. These prerequisite courses may have further prerequisites, which are also listed in the same table. We define the terms *immediate perquisite* and *extended prerequisite* with the help of an example given below.

PREREQUISITE Table

Cnum	prereq
EE645	EE600
EE600	EE302
EE302	EE301
EE645	MA532

For the course EE645:

Immediate prerequisite courses are: EE600 and MA532

Extended prerequisite courses are: EE600, MA532, EE302, and EE301.

Implement the following queries:

Queries

1. Find the age of the oldest student who is either a CS major or is enrolled in a course taught by Prof. Brown.
2. Find the names of all classes that either meet in room 115 or have five or more students enrolled.
3. Find the names of all students who are enrolled in two classes that meet at the same time.
4. Find the names of faculty members who teach in every room in which some class is taught.
5. Find the names of faculty members for whom the combined enrollment of the courses that they teach is more than eight.
6. Print the Level and the average age of students for that Level, for all Levels except JR.
7. Find the names of students who are enrolled in the maximum number of classes.
8. Find the names of students who are not enrolled in any class.
9. For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).
10. Compute and print the difference between the average number of students being taught by a EE faculty member and the average number of students taught by CS faculty member.
11. Print the name of the faculty member whose number of students is greater than the average number of students of EE Professor.
12. Find the name(s) of faculty member(s) who can substitute Prof. Wasfi for both of his classes when Prof. Wasfi is out of town. A faculty member can substitute another faculty member if he/she is from the same department and is not teaching at the same time.
13. Find the students who are taking courses that do not have any prerequisite course.
14. Find all courses that can be taken simultaneously in a semester with their immediate prerequisite courses. Two or more courses can be taken simultaneously in a semester if they do not meet at same time.
15. Find an instructor who teaches both a course and its immediate prerequisite courses.
16. Find all courses that have no more than three extended prerequisite courses.
17. Find the names of students who have taken courses which have only one extended prerequisite course. Also list the course and its extended prerequisite course.

Create the following 2 views. Please name them VIEWA and VIEWB.

1. A view that shows the Faculty ID, faculty name and the name of the class he/she teaches.
2. A view that shows the student id, student name, course number of the classes he/she is enrolled in.

- Use the command **SELECT *** to show the content of a view.
- Use the command **DROP** to drop these tables and view.

A useful strategy

Here is a useful approach for doing the Programming Assignment...

1. Connect to the Oracle server with your assigned Oracle account.
2. Try a few simple SQL commands until you are comfortable interacting with sqlplus.
3. Work out the SQL commands you need to solve the STUDENT database problems.
4. Use a text editor you are familiar with to create “.sql” files that contain the necessary SQL commands for Programming Assignment 1.
5. Test your .sql files. For example, if your .sql file is saved as *queries*.sql, then type "@" followed by *queries* without the .sql extension. It looks like:
   ```
   SQL> @queries
   ```
6. Please add the following lines to the beginning of your .sql files so that grades can be assigned:

   ```
   rem   CS 340 Programming Assignment 1
   rem   your_first_name Your_last_name
   rem   Your login
   ```

7. Remember to divide and conquer. Test your .sql file continuously as you add new SQL commands. You can use the Oracle command **spool** to direct the output to a file so that you can check it. However, please **DO NOT** submit the output file.

What to submit

You are going to submit **FOUR** SQL files: ***create.sql, populate.sql, queries.sql***, and ***dropall.sql***.

- create.sql should contain SQL commands to create the tables and defining integrity constraints.
- populate.sql should contain SQL commands (insert into . . .) to populate the tables with your own data.

- queries.sql should contain the SQL commands for the above queries and the SQL commands for displaying the views. (Please use the command DBMS_OUTPUT.PUT_LINE command to display query numbers. E.g., `DBMS_OUTPUT.PUT_LINE('Query #1').` This will help the grader in evaluating the output of your queries.
- dropall.sql should contain the drop table and drop view commands to drop all the tables and views.

How to submit your .sql files

When you are ready to submit your Programming Assignment, put all the files to be submitted in a folder/directory “PA1_[your login].” For example, if your login is “s1510028” put the file in the folder/directory “PA1_s1510028”. Compress the folder into a zip file, and submit it on LMS.

Sample data

Sample data is provided to show the format of data for the corresponding tables. Your queries may not necessarily be evaluated against the sample data.

STUDENT table

Sid	Sname	major	level	age
1	John	EE	FR	18
2	Tim	EE	FR	19
3	Richard	EE	SO	20
4	Edward	EE	SO	21
5	Alber	CS	JR	22
6	Mary	EE	JR	22
7	Jack	EE	SR	23
8	Julian	EE	SR	22
9	Sam	CS	SR	24
10	Ram	EE	SR	23
11	Rick	EE	SR	24

CLASS table

cnum	Meets_at	room	fid
EE101	9:00	117	1
EE102	10:00	117	2
EE104	13:00	117	3
EE151	15:00	117	4
EE261	9:00	118	4
MA365	10:00	118	5
EE347	13:00	118	1
EE404	9:00	115	3
MA448	12:00	115	5
CS480	13:00	115	1

FACULTY table

fid	Fname	dept
1	Prof. James	EE
2	Prof. Brown	CS
3	Prof. Wasfi	EE
4	Prof. Latif	EE
5	Prof. Rutherford	MA

ENROLLED table

cname	sid
EE101	1
EE101	2
EE101	3
EE101	4
CS102	1
CS102	2
CS102	4
EE104	1
EE104	2
EE104	3
EE151	4
EE151	5
EE151	6
EE261	1
EE261	2
EE261	3
EE261	4

EE261	5
EE261	7
MA365	5
MA365	6
MA365	7
MA365	8
EE347	5
EE347	7
EE347	8
EE347	9
EE404	9
EE404	10
EE404	7
MA448	7
MA448	8
MA448	9
MA448	10
CS480	6
CS480	7
CS480	8
CS480	9

Example of Insert Statement for populating Tables

```
INSERT INTO STUDENT (sid, sname, major, slevel, age)
        values (1,'John','EE','FR',18);

INSERT INTO FACULTY (fid, fname, dept)
        values (1,'Prof. James','EE');

INSERT INTO CLASS (cnum, meets_at, room, fid)
        values ('EE101', to_date('9:00','HH24:MI'),'117',1);

INSERT INTO PREREQUISITE(cnum,prereq)
        values ('EE102','EE101');


INSERT INTO ENROLLED (cnum, sid)
        values ('EE101',1);
```