Department of Computer Science
New Mexico State University
C S 482/502 Database Management Systems I

Project – Part 2

General requirements:

- You should form a group consisting of 2 or 3 people to work on this part of the project (i.e., the maximum number of students in a group is three). You can change team members when you work on the other parts of the project. Once you form a team for this part of the project, you cannot kick out your team member (in particular at the last minute). If you cannot work with your team member, you will lose points (denoted below).
- You need to finish the tasks stated in three parts. NOTE that the due dates of the three parts are different.

Part 2: Access databases using an application language

Objective:

- Write SQL language to conduct database search.
- Practice calling SQL language from an application language.
- Practice teamwork skills.

Tasks:
Write Java or Python programs to answer the following questions using the database that you created in Project – Part 1.

- Please put all the code into one source file (one java file or one python file).
- Language version
  - Your program should be tested (compiled and run) using the undergraduate or graduate lab machines.
  - Python version should be Python 3.x.
- You can hardcode the database connection parameters (host name, user id, database name, and the database connection password) in your program, or put them in a configuration file.
- Your program should contain a main function
  - The main function should have question no as the first parameter.
  - The main function should also have other necessary parameters to answer individual questions.
- (5 points) Create a README.txt to write down the work allocation of the different team members.
- (5 points) Each team member needs to create a PeerEvaluation_<yourname>.txt to include a peer evaluation to your team members for this part of the project. Your peer evaluation should include a score (1 to 5, with 1 being poorest and 5 being best) and a justification for your score. The justification does not need to be long.

Note:

- If your program cannot be compiled correctly, or cannot be run correctly, this part of your project will be graded as ZERO even though your SQL can be run correctly through the interactive interface of any DBMS.
- The correctness of the program (compiling, taking parameters, connecting to databases, etc.) counts for 10 points. Each of the following questions counts for 10 points:

Queries:

1. (10 points) Find the sites that are on a given street (i.e., the address contains the street name (case insensitive)). Show the detailed information of each site. To get the answer of this question, the command to run is
   ./proj 1 <param_street_name> (for Java)
   python proj.py 1 <param_street_name> (for Python)

2. (10 points) Find the digital displays with a given scheduler system. Show their serial nos, model nos, and the names of technical supports who specialize their models. The scheduler system should be a parameter input through the main program. To get the answer of this question, the command to run is
   ./proj 2 <param_schedular_system> (for Java)
   python proj.py 2 <param_schedular_system> (for Python)

3. (10 points) List the distinct names of all salesmen and the number of salesmen with that name. The output should be in the ascending order of the salesmen name. If multiple salesmen have the same name, show all the attribute values for those salesmen. For instance, if the Salesman relation contains the following 4 records
   (1, 'Peter', 'M'), (2, Mary, 'F'), (3, 'John', 'M'), (4, Mary, 'F').
   The output should be:
   Name cnt

   John 1
   Mary 2 (2,Mary,'F'),(4,Mary,'F')
   Peter 1
   To get the answer of this question, the command to run is
   ./proj 3 (for Java)
   python proj.py 3 (for python)

4. (10 points) Find the clients with a given phone no. The phone no should be a parameter input through the main program. To get the answer of this question, the command to run is
   ./proj 4 <param_phone_no> (for Java)
   python proj.py 4 <param_phone_no> (for Python)

5. (10 points) Find the total working hours of each administrator. Display the administrators' employee ids, names, and total working hours in ascending order of the total working hours. To get the answer of this question, the command to run is
   ./proj 5 (for Java)
   python proj.py 5 (for Python)

6. (10 points) Find the technical supports that specialize a specified model. Display the names of those technical supports. The specified model no should be a parameter input through the main program. To get the answer of this question, the command to run is
   ./proj 6 <param_model_no> (for Java)
   python proj.py 6 <param_model_no> (for Python)

7. (10 points) Order the salesmen with descending order of their average commission rates. Display each salesman's name and the average commission rate. To get the answer of this question, the command to run is
   ./proj 7 (for Java)
   python proj.py 7 (for Python)

8. (10 points) Calculate the number of administrators, salesmen, and technical supports. Display the results in the following format.

```
Role            cnt
-----------------
Administrator   10
Salesmen        40
Technicians     20
```

   To get the answer of this question, the command to run is

   ./proj 8 (for Java)
   python proj.py 8 (for Python)

Submission: You have to submit your assignment electronically (through Canvas). Printed copies are not accepted. See course syllabus for policies on late submission and plagiarism.

- Please create a zip file to contain the source files and the readme file, name the zip file "p2-(your last names).zip". For each team, only submit one copy of the zip file. **Points will be deducted if multiple copies are submitted**.
- Each team member should submit your own PeerEvaluation_<yourname>.txt so that this information is confidential to your team members.