CSE 4/560 PA 2: SQL Query

Due 23:59 10/09/2020 EST

October 6, 2020

This is an individual programming assignment for writing SQL queries. There are 12 problems with 15 points in total. **Please note that academic integrity is strictly implemented and any violation will lead to a F grade in this course.**

1 Database: Employees

From problem 1 to problem 6, use the employees database, which is imported from PA0, to answer the query. Note that we can treat the date 9999-01-01 as "current" in the value of to_date columns in this database.

1.1 Problem 1, 1 point

List pairs of employee (e_1, e_2) which satisfies ALL following conditions:

1. Both e_1 and e_2's current department number is d002.
2. The year of birthdate for e_1 and e_2 is 1956.
3. The salary of e_2 is greater than e_1.
4. The e_1's employee number is greater than e_2.
5. Both e_1 and e_2 are less than 100000

Sort the result by e_1 then e_2. The result of query is similar to following table:

```
e1   |e2   |
-----|-----|
10180|10132|
11392|10132|
11392|10416|
...
```

1.2 Problem 2, 1 point

List each the current number of employee title respect to every department. Sort the result by department name and title. The result of query is similar to following table:

```
dept_name         |title             |cnt  |
------------------|------------------|-----|
Customer Service  |Assistant Engineer|   68|
Customer Service  |Engineer          |  627|
...
Finance           |Manager           |    1|
Finance           |Senior Staff      | 9545|
...
```

1.3 Problem 3, 1 point

For problem 3, use EXIST/NO EXIST, find out following information. For each department, list out the manager who has the shortest time period in the department. Sort the result by employ number. You may want to use DATEDIFF function provided by MySQL. The result of query is similar to following table:

```
emp_no|dept_name         |
------|------------------|
110022|Marketing         |
110085|Finance           |
...
```

1.4 Problem 4, 1 point

Find out departments which has changed its manager more than once then list out the name of the departments and the number of changes. Sort the result by department name. The result of query is similar to following table:

```
dept_name         |cnt|
------------------|---|
Customer Service  |  3|
Production        |  3|
...
```

1.5 Problem 5, 1 point

For each employee, find out how many times the title has been changed and has a lower salary on the date of title change. e.g. An employee promoted from Engineer to Sr. Engineer with salaries decrease from 10k to 9k. Sort the result by employ number. The result of query is similar to following table:

```
emp_no|on_date   |o_title |n_title        |o_salary|n_salary|
------|----------|--------|---------------|--------|--------|
 82529|1995-10-03|Engineer|Senior Engineer|   50691|   50238|
217678|1998-10-25|Engineer|Senior Engineer|   54816|   54800|
...
```

1.6 Problem 6, 2 point

For each employee with employee number less than 20000, find the employee's salary percentile respect to the rest of employees who is employee number is less than 20000. Sort the result by emp_no. Sort the result by employ number. The result of query is similar to following table:

```
emp_no|percentile|
------|----------|
 10001|   82.9571|
 10002|   56.7229|
 10003|    0.9427|
...
```

2 Database: salika

From problem 6 to 12, import the database from salika-db.zip and answer following questions.

2.1 Problem 7, 1 point

For each actor, list number of category they have played together with their name. Sort the result by actor name. Sort the result by actor name. The result of query is similar to following table:

```
actor_name          |ncat|
--------------------|----|
ADAM GRANT          |  10|
ADAM HOPPER         |  12|
AL GARLAND          |  14|
...
```

2.2 Problem 8, 1 point

For problem 8, use ALL/ANY, find the rental(s) with lowest payment amount. Sort the result by rental id. The result of query is similar to following table:

```
rental_id|payment_id|
---------|----------|
    11676|      5880|
    11782|      9586|
```

```
    12352|      4235|
....
```

2.3 Problem 9, 1 point

Calculate the average rental days for each film. Display them together with the length of film. Sort the result by film id. The result of query is similar to following table:

```
film_id|length|avg_days|
-------|------|--------|
      1|    86|  5.0909|
      2|    48|  5.6667|
      3|    50|  3.4167|
....
```

2.4 Problem 10, 1 point

For each customer, calculate the total amount which the customer spend on the rental and the number of rental. Note that some payment may not have rental_id. We need to exclude such bogus instances. Return only if the total amount is greater than 100. Sort the result by customer id. The result of query is similar to following table:

```
customer_id|total |n_rent|
-----------|------|------|
          1|118.68|    32|
          2|128.73|    27|
          3|135.74|    26|
....
```

2.5 Problem 11, 2 points

Given the definition of a pair of film ids, $(f1, f2)$, where $f2$ has a immediate rental after $f1$'s rental in terms of rental date and both rental belongs to the same customer. In other word, there is no other rental between the rentals of $f1$ and $f2$ from the same customer. Find out all pairs which both $f1$, $f2$, and the count. Sort the result by count (DESCENDING) then $f1$(ASCENDING) and $f2$(ASCENDING). The result of query is similar to following table:

```
f1  |f2 |cnt|
----|---|---|
 285|403|  3|
 621|709|  3|
  10|450|  2|
....
```

2.6 Problem 12, 2 points

Every film has a replacement cost, which is a column in table film. Calculate the total earning for each film from its rental, those amount which is lower than replacement cost is not breakeven yet. List from 6th to 10th films that have the closest amount to breakeven and calculate the difference to breakeven. Sort the result by film id. The result of query is similar to following table:

```
film_id|to_breakeven|
-------|------------|
    105|        0.15|
    399|        1.05|
    538|        0.11|
    613|        0.09|
    947|        1.06|
```

3 Allowed Functions

Since PAs is about implementing functionalities with SQL by ourselves, please note that only following functions are allowed:

- YEAR
- SUM
- COUNT
- AVG
- DATEDIFF
- CONCAT_WS

Functions are not listed above is prohibited in the answer.

4 Offline Grader

Before downloading and using the offline grader, please pay attention to following points:

1. The grader strictly compares the EXACTLY same result and order mentioned in each problem statement.
2. The grader checks DB state on start, make sure the DB state is same as the state which is immediately after importing the employees and salika database.
3. The grader takes the query run time into account, you might get partial or no point if the query is running too slow.

4. In case of quick run or testing purpose, one can append - *-skip-verification* as command line argument to skip the verification.

5. The score is unofficial, we will run the grader with your submission after project due date as the official score.

The grader only supports Windows and Mac operating system. After downloading the zip file, follow the instructions according to the platform.

4.1 Windows

1. Make sure mysql server is running on localhost.

2. Decompress the zip file, the result is a directory named *pa2-grader-win*

3. Edit the *pa2.cfg*, set the user and password for the mysql server connection.

4. Launch a console such as cmd or powershell, change the working directory to *pa2-grader-win*

5. Execute *pa2_test.exe* from console, the result should be a pass on initial state verification and failed on all questions.

6. Write your answer in the files in *quiz* directory, each question has one file. e.g., writing the answer for problem 1 in *q1.sql*

7. Run *pa2_test.exe* again, grader will show the scores.

4.2 Mac OS X

1. Make sure Python 3 is installed at /usr/local/bin/python3

2. Make sure mysql server is running on localhost.

3. Decompress the zip file, the result is a directory named *pa2_test.app*

4. Launch a console, change the working directory to *pa2_test.app/Contents/Resources*.

5. Edit the *pa2.cfg*, set the user and password for the mysql server connection.

6. Change the working directory to *pa2_test.app/Contents/MacOS*

7. Execute *pa2_test* from console, the result should be a pass on initial state verification and failed on all questions.

8. Write your answer in the files in *pa2_test.app/Contents/Resources/quiz* directory, each question has one file. e.g., writing the answer for problem 1 in *q1.sql*

9. Run *pa2_test* again, grader will show the scores.

5 Submission

Failure to comply with the submission specifications will incur penalties for EACH violation.

- What to submit: A zip file has to be submitted through the 'submit_cse460' (if you are CSE460 student) or 'submit_cse560' (if you are CSE560 student) submit script by 23:59 10/09/2020 EST. Only zip extension will be accepted, please **don't** use any other compression methods such as tar or 7zip. You can submit multiple times, note that **only the last submission** will be kept on the server. **Even one minute late is still a late submission. No late submission will be accepted**.

- Zip file naming: Use *ubit_pa2* (**NO SPACE!**) for the filename, for example: *jsmith_pa2.zip*, where *jsmith* is the ubit of submitter. The project is an **INDIVIDUAL** work, so everyone needs to submit ONE zip file.

- Structure of zip file: On unzipping the zip file, there should be a folder named with your ubit *ubit_pa2*, under the folder *ubit_pa2*, there should be 12 SQL files, starting from *q1.sql*, *q2.sql* ... ,*q12.sql* which correspond to SQL query for each problem.

- Follow steps below to submit your work:

  1. copy your file to server, note that there is a dot at the end of the command:
     `scp jsmith_pa2.zip jsmith@timberlake.cse.buffalo.edu:.`
  2. login to server:
     `ssh jsmith@timberlake.cse.buffalo.edu`
  3. submit your file (if you miss this step, we won't be able to see your work and you will NOT receive any score):
     - For CSE 460 students:
       `submit_cse460 jsmith_pa2.zip`
     - For CSE 560 students:
       `submit_cse560 jsmith_pa2.zip`