

CSE 4/560 PA 1: Basic SQL Query

Due 23:59 09/20/2020 EST

September 18, 2020

This is an individual programming assignment for writing SQL queries. There are 10 problems with 10 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 5, 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

Find all employees' employee number, birth date, gender. Sort the result by employee's birth date then employ number. The result of query is similar to following table:

emp_no	birth_date	gender
65308	1952-02-01	M
87461	1952-02-01	M
...		

1.2 Problem 2, 1 point

Find all male employees and sort the result by employee number. The result of query is similar to following table:

emp_no	birth_date	first_name	last_name	gender	hire_date
10001	1953-09-02	Georgi	Facello	M	1986-06-26
10003	1959-12-03	Parto	Bamford	M	1986-08-28
...					

1.3 Problem 3, 1 point

Find all employees' last name with their salaries in different periods. Sort the result by last name, from_date, to_date, then salary. The result of query is similar to following table:

last_name	salary	from_date	to_date
Aamodt	81659	1985-02-08	1986-02-08
Aamodt	62378	1985-02-11	1986-02-11
...			
Acton	40000	1985-02-17	1986-02-17
Acton	52117	1985-03-08	1986-03-08
...			

1.4 Problem 4, 1 point

Assuming every employee is supervised by the manager of their current department. Find all employees' current manager and the start date with their employee number and sort the result by employee number. The result of query is similar to following table:

emp	mgr	from_date
10001	110567	1986-06-26
10002	111133	1996-08-03
...		

1.5 Problem 5, 1 point

List the number of current employees in each department and return it only if the number is greater than 20000. Sort the result by department name. The result of query is similar to following table:

dept_name	noe
Development	61386
Production	53304
...	

2 Database: salika

From problem 6 to 10, import the database from salika-db.zip and answer following questions.

2.1 Problem 6, 1 point

Find the actor's first name and last name if first name is KENNETH. The result of query is similar to following table:

first_name	last_name
KENNETH	PALTROW
KENNETH	PESCI
...	

2.2 Problem 7, 1 point

Construct a SQL script with two statements, at the first statement: insert a new actor into the database which first name is KENNETH and last name is OLIVIER. Then do the same query from problem 6. The result of query is similar to following table:

first_name	last_name
KENNETH	PALTROW
KENNETH	PESCI
...	
KENNETH	OLIVIER

2.3 Problem 8, 1 point

Construct a SQL script with two statements, at the first statement: Deactivate the account of customer MARIA MILLER by setting the active column to zero. Show the updated instance with email and active column in the second statement. The result of query is similar to following table:

email	active
MARIA.MILLER@sakilacustomer.org	0

2.4 Problem 9, 1 point

Find out the film with the rental_rate lower than 1, return their film_id and title. The result of query is similar to following table:

```

film_id|title
-----|-----
      1|ACADEMY DINOSAUR
     11|ALAMO VIDEOTAPE
     12|ALASKA PHANTOM
...

```

2.5 Problem 10, 1 point

Given a pair of different films as (f1, f2), find out all pairs of films such that every pair has equal length and rating. The result of query is similar to following table:

```

film_1          |film_2          |length|rating|
-----|-----|-----|-----|
ANYTHING SAVANNAH |ALONE TRIP      |    82|R    |
BLADE POLISH      |ALABAMA DEVIL   |   114|PG-13 |
BORROWERS BEDAZZLED |BONNIE HOLOCAUST |    63|G    |
...

```

3 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.

3.1 Windows

1. Make sure mysql server is running on localhost.
2. Decompress the zip file, the result is a directory named *pa1-grader-win*
3. Edit the *pa1.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 *pa1-grader-win*
5. Execute *pa1_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 *pa1_test.exe* again, grader will show the scores.

3.2 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 *pa1_test.app*
4. Launch a console, change the working directory to *pa1_test.app/Contents/Resources*.
5. Edit the *pa1.cfg*, set the user and password for the mysql server connection.
6. Change the working directory to *pa1_test.app/Contents/MacOS*
7. Execute *pa1_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 *pa1_test.app/Contents/Resources/quiz* directory, each question has one file. e.g., writing the answer for problem 1 in *q1.sql*
9. Run *pa1_test* again, grader will show the scores.

4 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 09/20/2020 11:59PM 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. **No late submission will be accepted.**
- Zip file naming: Use *ubit_pa1* (**NO SPACE!**) for the filename, for example: *jsmith_pa1.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_pa1*, under the folder *ubit_pa1*, there should be 10 SQL files, starting from *q1.sql*, *q2.sql* ... ,*q10.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_pa1.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_pa1.zip`
 - For CSE 560 students:
`submit_cse560 jsmith_pa1.zip`