EE5178
112-2 Homework 5

20240502

Part I: SQL Client-Side Programming (50%)

- In this homework, you need to write a Python program with SQL statements inside it to complete various tasks. You should use **mysql.connector** as your mySQL connection package.
- A sample code file is provided for your reference as starting point. A DBMS_student_list.csv file and new_student_list.csv are provided as data input.

- 1. Take DBMS_student_list.csv as data input, and input the data into a “student” table in your database in MySQL. (5%)
- 2. Read the information about yourself from the student table (You can use either your name or your student ID to select out yourself.) (5%)
- 3. Update your own value of the “身分" column from “校內生" to “特優生", then read the updated information about yourself from the database and print it out. (5%)
- 4. Insert three new students into the student table, using the “new_student_list.csv” file provided. (5%)

Part I (2)

- 5. Write a prepared select statement that can read back the information of any student given his/her ID. Execute this prepared statement three times to extract the information of the three new students you inserted just now. (10%)
  Note: You may need to do a little bit research on your own to accomplish this task (look up how to write prepared statement with mysql.connector.)

- 6. (1) Use “alter table” to add two new columns “group” and “captain” into your table (2) input your group number into this column for each your group members; for all other students, input 0 as their group number for this homework. (3) Input “1” (or “Y”) into the “captain” column for the captain of your group. For all other people, the value of this column can be “0” (or “N”) for this homework (4) Select and list the information of your group members, along with the value of “group” and “captain” columns. (select the members using the group number, rather than student ID or name to complete.) (**20%)**

Part II: Server-Side Programming (50%)

- In this part, you re-use the “student” table you created in Part I, and you will need to write SQL statements from the MySQL console. Please use ‘MySQL Command line client - Unicode’.

- 1. **Prepared statement** : (1) Use **prepared statement** to write a SELECT statement on the student table you created in Part I with “系所" select condition, but leave selection condition value as “?”. (2) List the students from your own “系所", by using “set” statement to set the selection condition values, and then run the prepared statement. (3) Set the conditions to a different “系所" (you can pick any one you like, but the output should not be empty) and run the prepared statement again. (10%)

- **2. Stored-function** (10%) The name column (“姓名") contains both people’s Chinese names and English names. Write a **stored-function** that takes the value of this column as input and output only the Chinese name part (2) Write a similar stored-function which output only the English name part. (3) Then write a select statement that uses both of these functions to list the names of all members of your group. The Chinese names and English names should appears as two different columns in the output listing. (10%) (You may want to look up MySQL documentation for how to do text pattern matching.)

Part II (2)

- **3. Stored procedure** (10%) Write a **stored procedure** that take the name of a certain department or graduate institute (系所) and output the count of students in that 系所 into a user variable @STCOUNT. Call this procedure and then print @STCOUNT twice, once with your own 系所 and once with some other 系所 (choose one with none empty output)
- **4. View** (10%) (1) Create a view “new_student” on the “student” table. This view should have the exact same columns as the “student” table, except that the “姓名" column should become two columns “中文名" and “英文名". (Hint: use the stored function you created) (2) Select the 系所, 年級,學號,中文名,英文名 columns of all the student from the same 系所 as you.
- **5. Trigger** (10%) (1) Assume the table “student” is of high level of importance. Whenever someone inserts or delete a row, you want to know who does it and when it is done. Create an new “record table” and write a trigger to record that information. (Hint: you may want to use the user() function call). Also Create two variables “@NUMDEL” and “@NUMINS” to record the number of students deleted and inserted. (2) After you finish writing the trigger, do a select to list the content of the “record table”, and do a select of “@NUMDEL” and “@NUMINS” (to show that it is empty). (3) Then do three insertions and two deletions. List the the “record table” again and then list “@NUMDEL” and “@NUMINS” again to demonstrate the effectiveness of your trigger.

Environment

- Python 3.8
- MySQL 8.0
- Python packages for Part I
    - mysql-connector-python==8.0.33
    - Python Standard Library (such as csv and json)

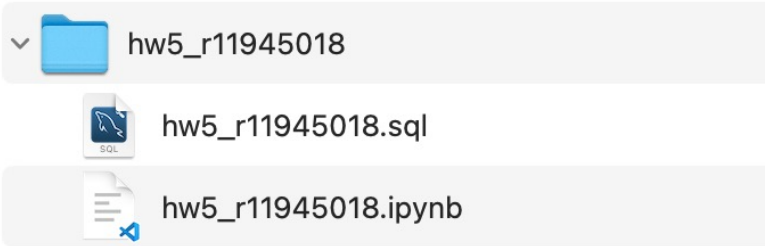
About Part I

- Data
  - DBMS_student_list.csv
  - new_student_list.csv

- mysql.connector example: Pylib_mysql.ipynb 、PylibPandas_mySQL.ipynb
- Submission template: part1_template.ipynb
- You shall NOT hardcode any path and settings in the ipynb notebook.
  - You can modify the hw5_config.json file and load the settings in the notebook.
  - You do NOT need to submit hw5_config.json and csv file.

Homework 5 submission

- Deadline : 5/16 Thur. 23:59 (GMT +8)
- Submission : NTU COOL
- File name & format
    - 上傳檔案: hw5_{id}.zip, e.g. hw5_ r11945018.zip
    - 解壓縮後需包含
        - 最外層資料夾: hw5_{id}, e.g. hw5_r11945018
        - Ipynb 檔: hw5_{id}.ipynb, e.g. hw5_ r11945018.ipynb
        - Sql 檔: hw5_{id}.sql, e.g. hw5_ r11945018.sql
        - 沒有最外層資料夾、檔名錯誤一律扣10分
    - 格式可參考 part1_template.ipynb 及 part2_template.sql



Homework 5 submission

- Others
  - 只要交程式碼，不需要交執行結果
  - 程式碼不能跳error，有error 一個扣10 分
  - 不要更改MySQL 預設設定
- Delay
  - One day: original score * 0.8
  - More than two days: get no points