EE5178
112-2 Homework 6

20240523

Homework 6 - Notes

- You will be given 4 csv files, which you will use in doing this homework.
  - studentE.csv: contain information of all students in our class
  - new_studentE.csv: contain the information of three fake new students, in the same format as student.csv.
  - student_groupE.csv: contain student id and the group number of their final project.
  - groupE.csv: contain the group number of each group, and all members in the group, as leader, member2, member3, and member4.

Homework 6 - MongoDB

- Task 1: (import)
  Load the EE5178 student CSV file, studentE.csv, into collection “students” in database “hw6” in MongoDB, and write a MongoDB query to return the information (the document) about yourself. (10%)
  Note: we did not cover how to import CSV file into MongoDB in class. You are responsible for finding out how to do that by yourself as part of the homework. (Hint: install the MongoDB Database Tools package, and use mongoimport)
- Task 2: (aggregation pipeline)
  Write a query to report the top 10 “dept" with the most number of 學生 in the “students” collection (不算旁聽生). (10%)
  (Hint: Use a $match stage to select only “學生".)
- Task 3: (adding fields)
  Import the students from “new_studentE.csv” into your “students” collection.
  For the each student, add a new field “updated", and set it to “2024-05-23”. Write a query to return students of your department to make sure your updated is effective. (5%)
  Note: You are free to use either string or the ISODate() object as the type of “updated”. However, use ISODate() will make Task 5 easier to solve for you.
- Task 4: (update)
  Update the “updated" field of the new students and yourself to “2024-06-01”. Then write one query to return yourself and these new students. (5%)

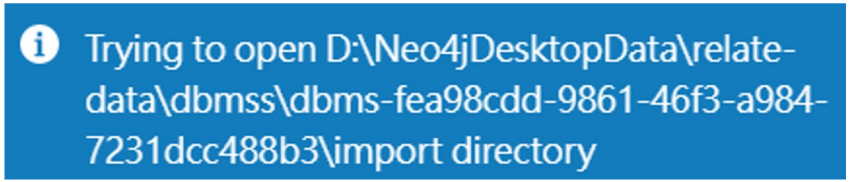
Homework 6 - MongoDB (2)

- Task 5: (Incremental pipeline)
  Design a incremental aggregation pipeline to calculate the number of student for each “dept" for the “updated” field from some starting date to some ending date. Run the pipeline for students with the value of “modified” field from 2024-01-01 to 2024-05-31. Store your result in a “tally” collection, and print out the result.
- Run the pipeline again with “updated” field from “2024-06-01” to 2024-06-30, and print our the “tally” document again. (5%)
- Hint: Make sure you understand the incremental MapReduce example in the lecture.

- Task 6: (Merge)
  Merge information in the student_groupE.csv file with your students collection. That is, import student_groupE.csv into the students collection with the “--mode=merge” option to add the group information into each student. (5%) Hint: study the mongoimport document.
  https://www.mongodb.com/docs/database-tools/mongoimport/
  Or, you can also load the student_groupE CSV file into mongoDB, then use $lookup to merge it with the student collection. Either approach get the same credits.
- Task 7: Write a pipeline to output each group with names of group members in an array. Note: those who do not have a group number (i.e., 旁聽生) should not appear in the output. (5%)

Homework 6 - Neo4J

- Task 1: (Load CSV)
  Load the student CSV, studentE.csv, into Neo4J and create one node for each students. (5%)
  Note: Neo4j is not very friendly when importing data from Windows. This unfortunately is one of the weakness of neo4j, among many of its strengths. To download CSV file in Windows, find out your **neo4j installation location**, and put your CSV in the "import" directory located in a subdirectory a few layers underneath it. Then you can run your LOAD CSV command successfully. Below is my "import" directory shown here as an example.



- You can also find out your "import" directory from within the Neo4jDesktop program.
  https://neo4j.com/docs/getting-started/appendix/tutorials/guide-import-desktop-csv/

- Task 2: (Load CSV and merge into graph)
  Load student_groupE.csv into Neo4J, create one node for each group, and create a relationship between each student and his/her group, if the relationship exists. Accomplish all of these with one query. (10%)
  Hint: Consider combining LOAD CSV and MATCH and MERGE clauses.
  Ref: https://neo4j.com/docs/cypher-manual/current/clauses/load-csv/
  (If you cannot use one query, you can use two queries, but receive a -4% penalty.)

Homework 6 - Neo4J (2)

- Task 3: (Basic)
  Write a Cypher query to return your name, and the names of your group partners, in the format below (5%)
  Note: Receive a -4% penalty if your own name appear in the Partners list.

Self	Partners
王海王	[李火星, 金水星, ...]

- Task 4: (Create and merge)
  Create a new node with appropriate label and properties for each "dept" in your database, and create a relationship for each student to his/her “dept”. Do all of these using one query. (10%)

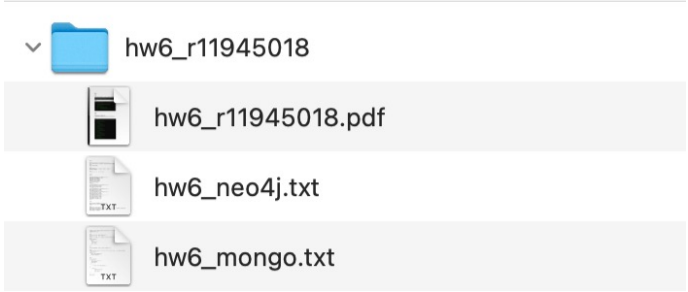
- Task 5: (Aggregation)
  Write a query to report the top 10 “dept" with the most number of students in this graph database. (7%)
  Note: you should only counting “學生", and excluding people such as “旁聽生".
- Task 6: (Advanced)
  Write a query to return the group numbers (not the group nodes) of the top 5 most diversified groups (groups with the most number of different department), with the names of these department, order by group number (ascending) (10%)
- Task 7 (Advanced: Potential partners):Write a query to return the number of students attending this class in the department of each of your group partners (include yourself if you are in that department). The output format should be “parter_name, dept_name, dept_size” (8%)

Version requirements

- MongoDB 6
- Neo4j Community Edition 5.8.0

Homework 6 submission

- Deadline : 6/6 Thur. 23:59 (GMT +8)
- Submission : NTU COOL
- File name & format
  - 上傳檔案: hw6_{id}.zip, e.g. hw6_ r11945018.zip
  - 解壓縮後需包含
    - 最外層資料夾: hw6_{id}, e.g. hw6_r11945018
    - hw6_mongo.txt / hw6_neo4j.txt
      - 各題 mongoDB / neo4j 的 code
    - Report: hw6_{id}.pdf, e.g. hw6_r11945018.pdf
      - 各題的code 截圖 & results，煩請題號標示清楚。
    - 沒有最外層資料夾、檔名錯誤一律扣10分



Homework 6 submission

- Others
  - 程式碼不能跳error，有error 一個扣10 分
- Delay
  - One day: original score * 0.8
  - More than two days: get no points