CompSci 516: Homework 3

Spring 2022

Total Points : 100

- Posted on : Thursday March 17 (EST)
- Due on : Tuesday April 5 12 noon (EST)

Course Policy Reminder

Please remember that you need to **strictly** follow the following rules while you are working on the homework:

1. ☐ For this homework, you can ONLY work with one other student (and that student with you) and must submit as a group to the assignment.
2. ☐ Besides the above, you can NOT show your solution to any question to any other student.
3. ☐ You can NOT see the solution to any question from any other student.
4. ☐ You can NOT find the solution to any question from the Internet or any other sources. Note that you can and need to learn programming languages, frameworks, and libraries using online tutorials, and then need to write your own code.
5. ☐ If you are unsure about what is allowed, you need to send the instructor an email and ask about it.
6. ☐ Of course, feel free to post questions on Ed (as many times as you need) if anything is confusing or if you need help! Anonymous questions are fine but do not copy your solutions. If you have specific questions about your solution, you can send private questions on Ed.

1 Overview

In HW1, we have used Postgres as a database management system to store and query the DBLP dataset. In this assignment, you are given the DBLP dataset and a parser to store it in a local MongoDB instance. You will write MongoDB queries, in particular aggregations, to solve problems related to the dataset.

2 Concept

The provided parser.py (explained in next section) only creates Article and Inproceedings collections, whereas HW1 had an Authorship table as well. This is a brief conceptual explanation as to why.

SQL vs NoSQL

- In HW1, we have created three tables in Postgres: Article, Inproceedings, and Authorship. Why did we create Authorship table? Because one record in Article or Inproceedings can have multiple authors. If author records are inserted into Article or Inproceedings, information originally in Article and Inproceedings (pubkey, title, booktitle/journal, year) has to be duplicated for many times, which is a waste of space and violates relational database normal form.
- In this HW, only the Article and Inproceedings collections are necessary. Why do we not need a separate Authorship collection? Author information can be stored directly with the documents in a collection, avoiding redundancy and especially unnecessary joins.

3 [100 pts] Programming

There is a video of a tutorial session on Sakai.

3.1 Download Template File

Download `hw3_code.tgz` from Sakai. This file contains the template code, which you will need to fill in.

3.2 Install MongoDB

Install MongoDB on your local machine. Here are the official websites for downloading and installing MongoDB.

- `https://docs.mongodb.com/manual/administration/install-community/`

3.3 Using Template

1. Get MongoDB running on localhost with port 27017.

- Notice in `parser.py` that self.client connects to localhost with port 27017 and the code uses a database 'dblp'. For most users installing MongoDB and getting it started will require no further configuration. If, however, there is a conflict with either running on localhost with 27017 or you have a 'dblp' dataset in MongoDB you'd like to save, then you will need to make further configurations.

2. Create a python virtual environment (ie python 3.8-3.10), activate it, and install requirements.txt.
   - Note that python, pip typically refer to python2 and python3, pip3 typically refer to python3.
   - Check how to check what version of python(3) you use here.
   - Check how to manage python environments and change your local and/or global environment **before** creating a virtual environment here.
   - Check how to create a virtual environment here.
   - Check how to install requirements.txt **after** activating virtual environment here.
3. Run parser.py with a path to the provided dblp.xml document (i.e. 'python3 parser.py dblp.xml). **If it does not error out immediately and seems to be taking a long time, do not panic, it has been tested to take anywhere from 100-500 secs depending on laptop speed.**
   - Ensure that the dblp.xml is in the same directory as the dblp.dtd.
   - Ensure that whatever python environment you are using is using the proper python version 3 and has all the requirements.txt up to date.
   - Ensure that you have the path to the proper dblp.xml document.
4. Check the local MongoDB database that the parser.py has executed successfully. It should have created a database 'dblp' and two collections 'Article' and 'Inproceedings'.
   - One way of doing this is to use the MongoDB Shell that typically should have been installed with MongoDB in an earlier step.
5. Fill in `answers.py`. When you run it (ie python3 answers.py) it will prompt you for what question(s) to run. As long as you adhere to the output format requirement within the question, in particular that it should be at minimum an empty array, it will be successful.
   - For the question(s) that you decide to run, and if the return output format is 'correct' (it only checks that it is a list), it will automatically create .json files in the directory that it is located in (if this fails then check your permissions for python scripts) that can be submitted to the Gradescope assignment.
   - The Gradescope assignment will give you immediate feedback whether the .json files match the expected results. It will check both differences in length and the first difference in object. **Ensure that you follow the output format given and/or specified in each question otherwise the autograder will likely mark you wrong.**
   - The answers.py has been made to be easy to use so you can focus on filling in the queries. Working on it in order and checking each question in order is a good idea. In particular, you should check the counts with Q3 before moving on, and the schema verification in Q2 will help you understand how to do Q4-Q5.

- **Do not change any of the existing method names. The answers.py will execute any and all methods starting with 'q' and create related .json files with the names.**
- **There is a test 'q' available for you to sample run answers.py and ensure everything is running smoothly. Please remove it after you've tested since it is unnecessary.**
- **Check the final section on Submission and Grading for more information.**

3.4 [100 pts] Analyze DBLP Dataset

In this subsection, you will be filling in `answers.py`.

Here are some websites that may be useful to you:

- `https://pymongo.readthedocs.io/en/stable/`
- `https://docs.mongodb.com/manual/aggregation/`

Q1. [5 pts] Schema

Find the schema of both collections and write the schema in `q1.txt`.

Q2. [10 pts] Counts

Get counts of the total number of documents in both collections.

Q3. [15 pts] Change schema

Add a new field, **area**, in the Inproceedings collection, and populate the field fowllowing the table below. If there is no match, then set it to **UNKNOWN**.

After updating, your output should be the number of documents by area including UNKNOWN. This means answer.py for q4 should contain both the updating and number of documents queries.

You are required to update the existing collections in MongoDB, instead of creating new ones.

Area	Conference Name
Database	SIGMOD Conference VLDB ICDE PODS
Theory	STOC FOCS SODA ICALP
Systems	SIGCOMM ISCA HPCA PLDI
ML-AI	ICML NIPS AAAI IJCAI

Q4. [20 × 2 + 30 = 70 pts] Queries

- Q4a. Find the top-20 authors who published the most number of **Database** papers.
- Q4b. Find the number of authors who published in exactly two areas (Inproceedings) excluding 'UNKNOWN.'
- Q4c. Find the top 5 authors who published the maximum number of journal papers since 2000 among the top 20 authors who published at least one conference 'Database' paper.

4 Submission and Grading

Please submit to Gradescope `answers.py`, `q1.txt`, and the `.json files` from running answers.py with 'all'.

The .json files will have an autograder that will immediately tell you if you are correct, but the answers.py will be checked and if inconsistent with the .json results you will lose all your points for those problems, so focus on the query foremost. In other words, if the autograder marks you incorrect then that problem is incorrect, and if it marks you correct, then it depends on answers.py whether it is marked correct. You will also lose points if your answers.py are modified unreasonably from the skeleton and/or do not follow the instructions in the comment of a problem as well.