Marked Lab: MongoDB Queries

This lab must be prepared in teams of 2 or 3 students and submitted on Campus. You can check the due date in the submission area.

Contents

1 Work to Do

You have to answer the questions given in the script `queries.js` in the form of JavaScript comments. The questions relate to the database created and populated by the script `database.js`.

To do the lab: (1) download and execute the script `database.js`, (2) download and complete the script `queries.js`. You can use any editor or tool of your choice to complete the script.

The population of the database given as an example in `database.js` is deliberately simple. Some questions in the script might not yield any answer with this population Therefore, you will need to modify the population in order to thoroughly test these queries.

The archive `Example.zip` gives you an example of the files you have to submit ("File to Submit" folder) in accordance with the resource files you are given ("Resources" folder).

If you have a question regarding the lab, please post it on the Questions & Answers forum. I will not reply to emails sent to my email address. Thank you.

2 Beware

2.1 Automatic Marking

Your script will be marked automatically. The answers it outputs will be compared line by line to those given by the solution script when executed on the same database. Therefore, you must be very careful about the following:

- Answer each question precisely, like in the labs: answer each question with only one MongoDB query (most of the time) and conform to the output schema given in the question.

- If (in some rare cases) you need two MongoDB queries to answer a question, **do not use** an temporary variable, as follows:

  ```
  temp = db.myCollection.find( { /* … */ } ).foo
  db.myCollection.find( { bar: temp } )
  ```

  Rather, pass the returned value of the first call directly to the second call, as follows:

  ```
  db.myCollection.find( { bar: db.myCollection.find( { /* … */ } ).foo } )
  ```

- For some questions, the attributes involved in filtering or sorting must not be displayed: this is not a mistake. You can of course display these attributes when testing your queries, but do not forget do remove them before submitting your script.

- Do not remove or alter any line of the provided script, especially the instructions `print("Query xy")`: they help identify the question being answered in the output.

- Your script must output the result of the queries and nothing more (e.g. temporary results, comments, etc.); it must not call or include, in whole or part, the script `database.sql`.

2.2 Executing your Script

Your script will be run using a MongoDB 4.0 server, using the following Unix shell command:

```
mongo [connection parameters] < queries.sql > queries.out 2>&1
```

or, under Windows powershell:

```
get-content queries.js | mongo [paramètres de connexion] > queries.out
```

Before submitting your script, you must check that it executes correctly in your test environment using one of the above two commands.

2.3 Marking Database

A query must not make any assumption about the population of the database: it must return a valid result regardless of the population. Your script will be tested against a database whose population is different from that of the example database.

Moreover, the schema of the marking database will possibly differ from that of the example database in the following respects:

- Some fields (e.g. `writers`) might not exist or they might be set to null in some documents of the database. This case must be treated as the NULL value in SQL.
- Single-element arrays might be expressed as a mere scalar value: `writers: ["Sergio Leone"]` and `writers: "Sergio Leone"` must be considered equivalent.

3 Marking Scheme

The tentative marking scheme is as follows:

Item	**Marks**
Queries	18
Code Quality	2
Total	20

All the queries, whether simple or complex, bear the same marks. As for the “code quality”, your queries must (1) be as simple as possible; for example, do not use *aggregate()* if a mere *find()* will do, (2) be properly formatted.

4 Submission

4.1 Deliverable

The deliverable only consists of the script `queries.js`, which you must rename as `LASTNAME1.FirstName1.LASTNAME2.FirstName2.LASTNAME3.FirstName3.js`, **without any space character**, to identify the team members.

4.2 Submitting

You must submit the file into the submission area directly, **without creating** an archive. You may submit it again as often as you wish until the deadline, provided you **always do so under the same Campus user**.