CSCI235 – Database Systems
2024 S2
Implementation Task 3
Due on 27 May 2024

Scope

This laboratory includes two tasks related to the implementation of queries and data manipulations on a collection of documents stored in MongoDB database.

This Implementation is due by Monday, 27 May 2024, 9:00 pm Singapore time. This task is worth 4% of the total assessment for the subject.
Only electronic submission through Moodle at: https://moodle.uowplatform.edu.au/ is accepted. All email submission will be deleted and mark 0 ("zero") will be awarded.

For all the implemented tasks, your report or output must include a listing of all JSON scripts and its output.

The submission procedure is explained at the end of this specification.

Specification

Preliminary Task

In this implementation task, you may use the virtual machine that runs MongoDB Enterprise Edition database server 3.6.5. or the actual installation of MongoDB in your computer system.

Download the file `transport.js` that is included in the section SAMPLE DATABASE on Moodle. To create a sample database, start mongo command line interface (CLI) to MongoDB database server and process the script with the command `load('transport.js')`. A collection of documents `transport` contains exactly the same information as a relational database created with a script dbcreate.sql. To verify the results, process a query `db.transport.find().pretty()`.

No report is expected from this task.

Tasks

Task 1 (2.0 marks)

Use the methods find() and pretty() to implement the following queries.

(1) Display in a pretty format information about an employee number 11. Do not list information about trips performed by the employee.

(2) Display in a pretty format information about all mechanics. For each mechanic list only employee number, name, and position.

(3) Display in a pretty format information about the trucks with the registration numbers PKR768 and PKR008 and SST005. Do not list identifiers of the documents.

(4) Find the total number of employees.

(5) Find the total number of mechanics.

(6) Display in a pretty format the names and positions of all employees;

(7) Display in a pretty format the registrations number of all trucks maintained by John Fox. there is no need to remove duplicates from a listing.

(8) Display in a pretty format the names of mechanics who maintained a car with a registration LUCY01.

(9) Display in a pretty format the names of drivers who performed no trips so far.

(10) Display in a pretty format names of drivers who started at least on trip in Wollongong.

Implementation of each query is worth 0.5 mark.

When ready create MongoDB script file solution1.js with the implementations of your queries.

Very important

Implementation of each query must consist of the invocations of two methods:

```
print('(n) db.transport.find( query )');

db.transport.find( query ).pretty();
```

where *n* is a number of a query as above, i.e. 1 or 2 or … or 10.

A method `print()` prints a text of a query into a report. A method `find()` processes a query. A method `pretty()` nicely formats the results and it includes the results into a report from the processing of a script solution1.js.

Before generation of a report it is strongly recommended to "refresh" a collection of the documents `transport` with the methods `db.transport.drop()` and `load('transport.js')`.

Finally, to create a report open a Terminal window and at `$` prompt process a script file `solution1.js` in the following way.

```
mongo –port 4000 databaseName <solution1.js >solution1.lst
```

Deliverables
A file `solution1.lst` with a report from processing of MongoDB script `solution1.js` with the implementation of the queries listed above.

A report without a listing of the processed queries in a way explained above scores no marks

---

Task 2 (2.0 marks)

Implement the following data manipulations on a collection of documents transport.

(1) Remove a mechanic John Fox.

(2) Remove information about a date of birth (dob) from a description of employee number 11.

(3) Remove information about a trip number 7 performed by an employee number 11.

(4) Add information about a new trip performed by an employee number 11. At the moment we only know a trip number 999 and registration of truck used PKR786.

(5) Change a date of a trip number 15 performed by an employee number 11 to 28-SEP-18.

Implementation of each data manipulation is worth 0.2 mark.

When ready create MongoDB script file solution2.js with the implementations of your data manipulations.

Very important

Implementation of each data manipulation must consist of the invocations of two methods:

```
print('(n) db.transport.data-manipulation-method( data
manipulation)');

db.transport.data-manipulation-method( data manipulation );
```

where n is a number of a data manipulation as above, i.e. 1 or 2 or … or 5.

A method `print()` prints a text of a data manipulation into a report. A method data-manipulation-method processes a data manipulation and it includes the results into a report from the processing of a script `solution2.js`.

Before generation of a report it is strongly recommended to "refresh" a collection of the documents `transport` with the methods `db.transport.drop()` and `load('transport.js')` .

Finally, to create a report open a Terminal window and at $ prompt process a script file `solution2.js` in the following way.

```
mongo <solution2.js >solution2.lst
```

Deliverables

A file `solution2.lst` with a report from processing of MongoDB script `solution2.js` with the implementation of the data manipulations listed above.

A report without a listing of the processed data manipulations in a way explained above scores no marks.

Submissions

This implementation task is due by 9:00 pm (2100 hours) Monday, 27 May 2024, Singapore time.

Submit the files **solution1.pdf** and **solution2.pdf** through Moodle in the following way:

1) Zip all the files (Solution1.pdf and solution2.pdf into one zipped folder. Name your zipped file as YourName-IT3)
2) Access Moodle at **http://moodle.uowplatform.edu.au/**
3) To login use a Login link located in the right upper corner the Web page or in the middle of the bottom of the Web page
4) When successfully logged in, select a site CSCI235 (SP224) Database Systems
5) Scroll down to a section Submissions of Implementation Tasks
6) Click at Submit your Implementation Task 3 here link.
7) Click at a button Add Submission
8) Move the zipped file created in Step 1 above into an area provided in Moodle. You can drag and drop files here to add them. You can also use a link *Add…*
9) Click at a button Save changes,
10) Click at check box to confirm authorship of a submission,
11) When you are satisfied, remember to click at a button Submit assignment.

A policy regarding late submissions is included in the subject outline. Only

one submission per student is accepted.

Implementation Task 3 is an individual assignment and it is expected that all its tasks will be solved individually without any cooperation with the other students. Plagiarism is treated seriously. Students involved will likely receive zero. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or over e-mail.

End of specification