School of Computing and Information Technology
University of Wollongong

Session: **Summer 2020**
Lecturer: **Janusz R. Getta**

CSCI835 Database Systems
Assignment 4

12 July 2020

Scope

This assignment includes the tasks related to logical design of hierarchical database structures, creating and loading data into MongoDB database, and implementation of queries in MongoDB query language.

The outcomes of the assignment are due by **Saturday, 18 July, 2020, 9.00 pm (sharp).**

Please read very carefully information listed below.

This Assignment contributes to 13% of the total evaluation in a subject CSCI835 Database Systems.

A submission procedure is explained at the end of specification.

This assignment consists of 3 tasks and specification of each task starts from a new page.

It is recommended to solve the problems before attending the laboratory classes in order to efficiently use supervised laboratory time.

A submission marked by Moodle as "late" is treated as a late submission no matter how many seconds it is late.

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

A submission of compressed files (zipped, gzipped, rared, tared, 7-zipped, lhzed, … etc) is not allowed. The compressed files will not be evaluated.

All files left on Moodle in a state "`Draft(not submitted)`" will not be evaluated.

An implementation that does not compile due to one or more syntactical errors scores no marks.

It is expected that all tasks included within **Assignment 4** will be solved **individually without any cooperation** with the other students. If you have any doubts, questions, etc. please consult your lecturer or tutor during the laboratory classes. Plagiarism will result in a **FAIL** grade being recorded for the assessment task.

Prologue

If you do not have access to MongoDB database server the perform the actions listed below. Otherwise, progress to a specification of **Task 1**.

If VirtualBox is not installed yet then install it on your system first. It is explained at

https://documents.uow.edu.au/~jrg/115/cookbook/e1-1-frame.html

how to install and how to use VirtualBox.

Next, use a link provided on Moodle and located just below a link to a lecture class to download `Ubuntu18.04-64bits-MongoDB-4.2.2-08-JAN-2020.ova` file with an image of virtual machine running `MongoDB 4.2.2` on `Ubuntu 18.04`. When downloading, you must use Firefox browser. You can also use a link given below.

https://cloudstor.aarnet.edu.au/plus/s/qg8J3vE4SoxRzKo

Next, start VirtualBox and import a virtual machine `Ubuntu18.04-64bits-MongoDB-4.2.2-08-JAN-2020.ova.`

Start VirtualBox and import a virtual machine `Ubuntu18.04-64bits-MongoDB-4.2.2-08-JAN-2020.ova.`

Next, start the virtual machine and login as an Ubuntu Linux user `CSCI235` with a password `csci235`.

Next, start `Terminal` program and within a `Terminal` window start `MongoDB` server in the following way.

```
mongod –dbpath DATA –port 4000
```

A server displays a lot of messages. A successful start of `MongoDB` server is confirmed with a message like

```
…
… I NETWORK  [initandlisten] waiting for connections on port 4000…
…
```

pretty well hidden somewhere at the end of a long list of other messages issued by the starting server.

Note, that the `Terminal` window you use just now becomes a console of the running `MongoDB` server and you cannot use it any more. Do not close the `Terminal` window ! Just minimize it.

Open a new `Terminal` window. To create a `BSON` collection `orders` use a command

```
cd CSCI235
```

to move to `CSCI235` folder with the scripts.

Next, to start a command line client `mongo` process the following command.

```
mongo –port 4000
```

To make sure that client is connected to a database server process the commands `help` and `show dbs to` display available databases.

Tasks

Task 1 (3 marks)

Creating a collection, loading data, and processing simple queries

A conceptual schema of a sample database is available on Moodle in a section `SAMPLE DATABASES`. To download a conceptual schema of a sample database use a link `A conceptual schema (NoSQL database)`. A file `dbschema-nosql.bmp` contains a conceptual schema.

To download a logical schema of a sample database use a link `A logical schema (NoSQL database)`. A file `bsonschema.bmp` contains a conceptual schema. A logical schema of a sample database is located in a file `bsonschema.sql`.

The scripts `employees.js`, `suppliers.js`, and `customers.js` that create a sample database are located in the same section on Moodle.

Start a command line client `mongo` or graphical interface `Compass` and connect to MongoDB database server.

To create a collection `orders` and to load the documents into the collection, process the scripts `employees.js`, `suppliers.js`, and `customers.js` at `>` prompt in the following way.

```
load("employees.js");
load("suppliers.js");
load("customers.js");
```

Next, you can use the methods

`db.orders.find().count()` and
`db.orders.find().pretty()`

to count the total number of the documents in a collection `orders` and to list all documents in a pretty format.

Next try few simple queries.

For example, to list information about a hierarchy of customers submitting orders that consist of products process a method

```
db.orders.find({"CUSTOMER":{$exists:true}}).pretty();
```

For example, to list information about a customer who has a contact name `Maria Anders` and the orders submitted by the customer process a method

```
db.orders.find({"CUSTOMER.contact name":"Maria Anders"}).pretty();
```

For example, to list information about an order that has order id 325 process a method

```
db.orders.find({"CUSTOMER.submits.ORDER.order id":325}).pretty();
```

When finished save your implementations that include loading of sample data and the queries in a file `solution1.js`.

When ready create a report from processing of loading of sample data and the queries in file `solution1.lst`. To do so, use `gedit` editor and open a new file `solution1.lst`. Next, select the entire contents of the Terminal window and Copy&Paste it into a file `solution1.lst`. Save a file `solution1.lst`.

Deliverables

A file `solution1.lst` with a report from loading sample data and processing of the queries.

Task 1 (5 marks)

Implementation of simple data manipulations in MongoDB

Use `insert()` method to insert into a collection `orders` information about a new supplier and a new product supplied by a supplier.

Next, use `insert()` method to insert into a collection `orders` information about a new customer and a new order submitted by a customer. A new order must include one product whose description has been inserted into a collection `orders` in the previous step.

Next, use the methods `find()` and `pretty()` to list information about a new supplier, a new product supplied by a supplier, a new customer, an order submitted by a customer and the products included in a new order.

Note, that you must use meaningful data. Note, that data inserted into a collection `orders` will be used to evaluate originality of your solution.

When ready create a report from processing of simple data manipulations in file `solution2.lst`. To do so, use `gedit` editor and open a new file `solution2.lst`. Next, select the entire contents of the Terminal window with the results from processing of `insert()`, `find()`, and `pretty()` methods and Copy&Paste it into a file `solution2.lst`. Save a file `solution2.lst`.

Deliverables

A file `solution2.lst` with a report from processing of `insert()`, `find()`, and `pretty()` methods.

Task 2 (5 marks)
Implementation of simple queries in MongoDB

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

(1) Find the `total number` of suppliers.

(2) List in a pretty format all information about a supplier company `Norske Meierier`.

(3) List in a pretty format all information about the supplier companies located in `USA` in `Boston`.

(4) List in a pretty format all information about the supplier companies located in `USA` or in a city `Sandvika`.

(5) List in a pretty format all information about the supplier companies located in `USA` in `Boston` or `Los Angeles` or `Ann Arbor`.

When ready create a report from processing of the queries in file `solution3.lst`. To do so, use `gedit` editor and open a new file `solution3.lst`. Next, select the entire contents of the Terminal window with the results from processing of `find()` and `pretty()` methods that implement the queries and Copy&Paste it into a file `solution3.lst`. Save a file `solution3.lst`.

Deliverables
A file `solution3.lst` with a report from processing of `find()` and `pretty()` methods that implement the queries.

Submission

Submit the files **`solution1.lst`**, **`solution2.lst`**, and **`solution3.lst`** through Moodle in the following way:

(1) Access Moodle at **`http://moodle.uowplatform.edu.au/`**

(2) 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

(3) When logged select a site **`CSCI835 (JIS20) Database Systems`**

(4) Scroll down to a section **SUBMISSIONS**

(5) Click at a link **`In this place you can submit the outcomes of Assignment 4`**

(6) Click at a button **`Add Submission`**

(7) Move a file **`solution1.lst`** into an area **`You can drag and drop files here to add them`**. You can also use a link **Add**...

(8) Repeat a step (7) for the files **`solution2.lst`**, and **`solution3.lst`**.

(9) Click at a button **`Save changes`**

(10) Click at a button **`Submit assignment`**

(11) Click at the checkbox with a text attached: **`By checking this box, I confirm that this submission is my own work,`** ... in order to confirm the authorship of your submission.

(12) Click at a button **`Continue`**

End of specification