Fundamentals of Data Management

Credit Tasks 11.2: MongoDB

Overview

You'll learn how to add, update and query documents in the MongoDB document database.

Purpose

Gain some experience of working with non-relational databases.

Task

Download the Ubuntu virtual machine from Canvas and open it in the VMWare Player. Open a Terminal window and make a connection to MongoDB. Complete the tasks.

Time

This task should be completed in your lab class and submitted for feedback in lab 11 or at the beginning of lab 12.

Resources

- Online module (from Canvas)
- MongoDB and mongo shell on Ubuntu VM.
- MongoDB CRUD Operations manual at http://docs.mongodb.org/manual/tutorial/query-documents/

Feedback

Discuss your solutions with the tutorial instructor.

Next

That's it, you're done with the tutorials ☺. Work on D/HD level tasks if you like.

Credit Tasks 11.2 — Submission Details and Assessment Criteria

Document your solutions to the tasks using a word processor. Upload the report to Doubtfire, then discuss your results with the tutor.

Getting Started

Open the VMWare Player and start the virtual machine. The password for fdm is admin. Click on the application icon (the uppermost icon on the left in Ubuntu). Find the Terminal among the applications (Type 'Terminal' into the search field).

Type

mongo test

and <Enter> .

You have a connection to the 'test' database on MongoDB. The way you can tell is that the command prompt is now

>

Instead of

fdm@fdm-virtual-machine:~$

If you want to close the connection and return to Ubuntu, type

exit

and <Enter>

> **Note**: In MongoDB, the 'up' and 'down' arrows work just like on the command line – they bring up previously used commands. No need to type the command every time when you are working in the right syntax it.

A storage unit in MongoDB is called a collection. While you have a connection, see what collections are in the 'test' database:

show collections

You should see a 'restaurants' collection and a system collection. Query the content of the 'restaurants' collection:

db.restaurants.find()

This is the equivalent of SELECT * FROM restaurants; in SQL. You'll find that by default, MongoDB brings only the twenty first entries. If you want to list more, you have to type 'it' and <ENTER>. But how many restaurants are in the collection? Is it worth scrolling through all of them?

db.runCommand({count: 'restaurants'})

shows the number of restaurants in the collection.

Example document:


```
{
    "_id": ObjectId("55b2b4562fc7302843ef3927"),
    "address": {
        "building": "522",
        "coord": [
            -73.95171,
            40.767461
        ],
        "street": "East   74 Street",
        "zipcode": "10021"
    },
    "borough": "Manhattan",
    "cuisine": "American ",
    "grades": [
        {
            "date": ISODate("2014-09-02T00:00:00Z"),
            "grade": "A",
            "score": 12
        },
        {
            "date": ISODate("2013-12-19T00:00:00Z"),
            "grade": "B",
            "score": 16
        },
        {
            "date": ISODate("2013-05-28T00:00:00Z"),
            "grade": "A",
            "score": 9
        },

    ],
    "name": "Glorious Food",
    "restaurant_id": "40361521"
}
```


The entries have names and addresses of restaurants as well as the style of cuisine and grades received by customers.

Hint: CRUD is a relational acronym for CREATE, READ, UPDATE, DELETE; it means DDL (data definition) and DML (data manipulation).

Hint: Use a text editor to work on your scripts before running them in the terminal.

Subtask 11.2.2

Make a new collection called ‘books’ in the ‘test’ database.

ISBN	Book	Author
999-888	The secret river	Kate Grenville
777-222	Fooled by randomness	Nassim Taleb

ISBN	Reviewer	Score
999-888	Hua Tang	7
999-888	Jim Wong	8
999-888	Andrea Bauer	6
777-222	Simon Jones	5
777-222	John Messum	10

The data from the two relations shown above has to be stored in a meaningful format in MongoDB. Create the appropriate JSON documents and insert them into the books collection in MongoDB.

What message do you receive when your insert is successful?

Document the scripts used and the answer and upload.