Lab 12: NoSQL using MongoDB

CS355/CE373 Database Systems
Fall 2023

Dhanani School of Science and Engineering
Habib University

Copyright © 2023 Habib University

Contents

[1] from MongoDB's docs
[2] from W3Schools.com
[3] from W3Schools.com
[4] from W3Schools.com

1 Instructions

- This lab will contribute 1% towards the final grade.
- The deadline to submit this lab is the end of your lab.
- The lab must be submitted online via CANVAS.
- The pdf file should be named as *Lab_12_aa01234.pdf* where *aa1234* will be replaced your student id. ***Files which don't follow the appropriate naming convention will not be graded***.

.

1.1 Marking scheme

This lab will be marked out of 100.

- 60 Marks are for the completion of the lab.
- 40 Marks are for progress and attendance during the lab.

2 Objective

The objective of this lab is to provide hands-on practice on MongoDB commands to enhance the fundamental understanding of CRUD operations on MongoDB.

3 MongoDB Guide

3.1 Methods[5]

- **db.collection.insert()**: Insert one or more documents into a collection
- **db.collection.insertOne()**: Insert one document into a collection
- **db.collection.insertMany()**: Inserts multiple documents into a collection.
- **db.collection.find({query}, {projection})**: Returns documents for the given criteria in the query. Only those attributes will be returned which are mentioned in the projection. If there are no projection attributes, then return the entire document.
- **db.collection.updateOne()**: Updates at most a single document that match a specified filter even though multiple documents may match the specified filter.
- **db.collection.updateMany()**: Update all documents that match a specified filter.
- **db.collection.deleteOne()**: Delete at most a single document that match a specified filter even though multiple documents may match the specified filter.
- **db.collection.deleteMany()**: Delete all documents that match a specified filter.

3.2 Query operators[6]

- $eq: Values are equal
- $ne: Values are not equal

[5] from MongoDB's docs
[6] from W3Schools.com

- $gt: Value is greater than another value
- $gte: Value is greater than or equal to another value
- $lt: Value is less than another value
- $lte: Value is less than or equal to another value
- $in: Value is matched within an array

3.3 Update Operators[7]

- $currentDate: Sets the field value to the current date
- $inc: Increments the field value
- $rename: Renames the field
- $set: Sets the value of a field
- $unset: Removes the field from the document

3.4 Array operators[8]

- $addToSet: Adds distinct elements to an array
- $pop: Removes the first or last element of an array
- $pull: Removes all elements from an array that match the query
- $push: Adds an element to an array

3.5 Example queries

- **db.collection.find({})**: returns all documents in collection
- **db.collection.find({"attribute":"value"})**: returns all documents having specified attribute with the given value
- **db.collection.find({},{"attribute1":1,"attribute2":0})**: returns all documents with attribute1 but without attribute2 (along with *_objectid*)
- **db.collection.find({ status: { $in: ["A", "D"] }})**: returns all documents having status either A or D
- **db.collection.find($or: [{ status: "A" }, { qty: { $lt: 30 } }])** : returns all documents either having status A *OR* qty less than 30
- **db.collection.find({$and: [{ status: "A" }, { qty: { $lt: 30 } }])}**: returns all documents either having status A *AND* qty less than 30
- **db.collection.find({ tags: "red" })**: returns all documents where tags is an array that contains the string "red" as one of its elements
- **db.collection.find({ dim_cm: { $elemMatch: { $gt: 22, $lt: 30 } } })**: returns documents where the dim_cm array contains at least one element that is greater than 22 AND less than 30

[7] from W3Schools.com
[8] from W3Schools.com

- **db.collection.find({"instock.qty": { $lte: 20 } })** : returns all documents where the instock array has at least one embedded document that contains the field qty whose value is less than or equal to 20
- **db.collection.find({ "instock.warehouse": "A" })**: returns all records having embedded attribute warehouse (inside instock document) equal to A
- **db.collection.find({ "attribute": { $exists: 1 } })**: returns all those records having attribute in the document (0 for not having the attribute in the document)
- **db.collection.insert({ })**: inserts a new document into collection
- **db.collection.insertMany([{},{}])**: insert more than one documents into collection
- **db.collection.deleteMany({})**: delete all documents that fulfill the given criteria
- **db.collection.updateOne({ "item": "paper" },{$set: { "size.uom": "cm" } })**: update the first document where item equals "paper" and set the value of attribute *size.om* to *cm*
- **db.collection.updateMany({},{$inc: {"attribute":1}})**: increment the attribute value by 1 for all documents matching the given criteria
- **db.collection.updateMany({},**
  { $push: { "array_name":"element" }}): push a new element in array for all documents matching the given criteria

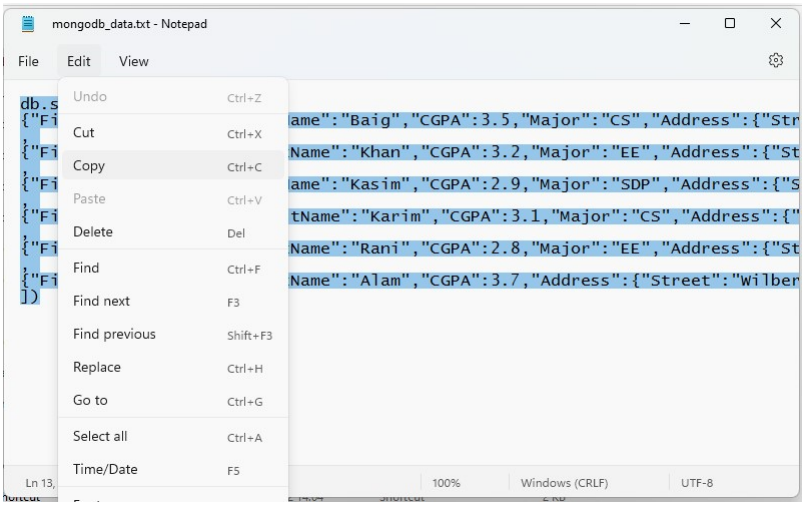
3.6 References:

- MongoDB Docs:
  `https://www.mongodb.com/docs/manual/crud/`

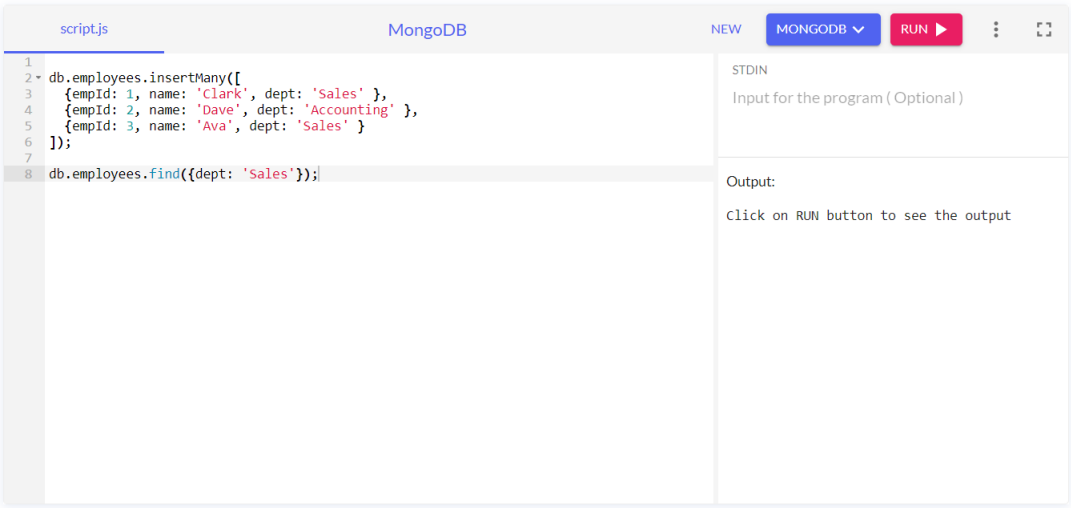
4 Exercises

4.1 Setup

- Download mongdb_data.txt from the LMS.
- Open the file and copy the content of the file.



- Open OneCompiler or JDoodle on your browser. The following screen should appear.



- Paste the file content on the editor.

- After pasting the file content in the editor, click on run. The output should look as follows:

4.2 Questions

Note: For each Question, you are required to write the Query as well as attach a screenshot of the output.

1. Display all students.
2. Display all the students with **CS** major.

3. Display all the students with **SDP** major.
4. Display all the students from Karachi City.
5. Display all the students **not** from Karachi City.
6. Display all the students having CGPA more than 3.
7. Display all the students having CGPA equal to 3.5
8. Display all the students from Karachi with a CS Major.
9. Display all the students for the **SDP** and **EE** majors.
10. Display only the first and last name of all students
    Result: FirstName, LastName
11. Display only the first name, last name, and city for all students.
    Result: FirstName, LastName, City
12. Display only the first name, last name for all the students with CGPA greater than or equal to 3.5 (without Object IDs)
13. Insert a student record with the following information
    - **First Name**: Fatima
    - **Last Name**: Tariq
    - **Phone**: 03124567891
    - **Address**
      - **Street**: Zainab Street Saddar
      - **City**: Karachi
14. Display all the student who do not have major property.
15. Insert the following students using insertMany
    - **First Name**: Zahid , **Last Name**: Hakim, **Major**: CS
    - **First Name**: Ayesha , **Last Name**: Asif
    - **First Name**: Faheem , **Last Name**: Faiz, **Phone**: 0213456789
16. Display all the students have taken the **CS355** course.
17. Display all the students have not taken the **CS355** course.
18. Increment the CGPA of all students by 0.1.
19. Set the CGPA of all students to 0.1
20. Introduce a new attribute **Minor** for the students with the value **CH**
21. Add a new course all students with the following information
    - **course_id**: CH101
    - **course_name**: Liberal Core
    - **credits**: 1
22. Delete all the students with CGPA less than 3.0