Lab 7 (2%)

MongoDB: Querying for Data

Learning outcomes

Upon successful completion of this workshop, you will have demonstrated the abilities to:

1. Import data from a json file into a MongoDB database.
2. Perform ad hoc queries on the database using the find function.
3. Query for ranges, set inclusion, inequalities, and more by using $-conditionals.

Group work acknowledgment

We, Mansoor Ahmad Zafar declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) **or distributed to other students.**

Specify below what each member has done towards the completion of this work:

Name Task(s)

1. Mansoor Ahmad Zafar Everything

2-

3-

Data:

Restaurant-data.json

Importing data

Use MongoDB compass to import the restaurant data. Use database name restaudb and collection name restaurants.

Querying mongodb [40pts]

Write MongoDB queries to answer each of the following questions.

Q1-6: 2pts each, total: 12pts

**Restrict the search**

1. Show one document of the restaurants collection in lab7 database.

Command:

db.restaurants.findOne()

Screenshot:

A blue and green background with lines

Description automatically generated

1. Find the restaurants with name “The Dutch”. Show only the id field and the name field.

Command:

db.restaurants.find({"name" : "The Dutch"}, {"id" : 1, "name" : 1, "\_id" : 0})

Screenshot:

A screen shot of a computer

Description automatically generated

1. Find the restaurants that have cuisine type “American”. Show only the id field.

Output: documents 4, 6, 7, 8.

Command:

db.restaurants.find({"cuisine\_type" : "American"}, {"id" : 1, "\_id" : 0})

Screenshot:

A computer screen shot of a blue screen

Description automatically generated

1. Show the restaurants that are located in the “Manhattan” neighborhood and have American cuisine type. Show only the id field.

Output: documents 4, 7, 8.

Command:

db.restaurants.find({"neighborhood" : "Manhattan", "cuisine\_type" : "American"}, {"id" : 1, "\_id" : 0})

Screenshot:

A blue background with text on it

Description automatically generated

**Specifying which keys to return.**

1. Show the id and cuisine type and neighborhood of all restaurants.

Command:

db.restaurants.find({}, {"id" : 1, "cuisine\_type" : 1, "neighborhood" : 1, "\_id" : 0})

Screenshot:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

1. Show all restaurants with all their key/value pairs except the reviews and the \_id.

Command:

db.restaurants.find({}, {"reviews" : 0, "\_id" : 0})

Screenshot:

A screen shot of a computer program

Description automatically generated A screenshot of a computer program

Description automatically generated A screenshot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated A screenshot of a computer program

Description automatically generated

Q7-13: 4pts each, total: 28pts

**Query conditionals:** "$lt", "$lte", "$gt", and "$gte", "$ne"

1. show the restaurants whose id is between 3 and 5 inclusive. Show only the id field.

Output: Documents 3, 4, 5.

Command:

db.restaurants.find({"id" : {"$gte" : 3, "$lte" : 5}}, {"id" : 1, "\_id" : 0})

Screenshot:

A screen shot of a computer

Description automatically generated

1. Find the restaurants that do not belong to the American cuisine type. Show only the id field.

Output: Documents 1, 2, 3, 5, 9, 10

Command:

db.restaurants.find({"cuisine\_type" : {"$ne" : "American"}}, {"id" : 1, "\_id" : 0})

Screenshot:

A screen shot of a computer

Description automatically generated

1. Find the restaurants that have the cuisine type “Pizza” or “Mexican”. Use the "$in" operator. Show only the id field.

Output: documents 2, 5 and 10.

Command:

db.restaurants.find({"cuisine\_type" : {"$in" : ["Pizza", "Mexican"]}}, {"id" : 1, "\_id" : 0})

Screenshot:

A screen shot of a computer

Description automatically generated

1. Find the restaurants that belong to "Brooklyn" neighborhood or have cuisine type "Pizza". Show only the id field.

Output: document 2, 5, 6.

Command:

db.restaurants.find({"$or" : [{"neighborhood" : "Brooklyn"}, {"cuisine\_type" : "Pizza"}] } , {"id" : 1, "\_id" : 0})

Screenshot:

A screen shot of a computer

Description automatically generated

**Querying arrays**

1. Use the $slice operator to show the last review added to restaurant id 1.

**Command:**

db.restaurants.find( {"id" : 1}, {"reviews" : {$slice : -1}, "\_id" : 0} )

**Screenshot:**

**A computer screen shot of a computer screen

Description automatically generated**

1. Find the restaurants that user Jason provided a review for. Show the restaurant id, name and the matched review element in the reviews array for that particular user.

Output: documents 1 and 10.

**Querying on Embedded Documents**

Use "$elemMatch" when you have more than one key you want to match on in an embedded document.

Command:

db.restaurants.find({"reviews" : {"$elemMatch" : {"name" : "Jason"}}}, {"id" : 1, "name" : 1, "reviews.$": 1, "\_id" : 0})

Screenshot:

A screen shot of a computer screen

Description automatically generated

1. Find the reviews provided by user “Jason” with rating equal 4. Show the restaurant document and only the review that matches the criteria.

Output: document 10

Command:

db.restaurants.find({"reviews" : {"$elemMatch" : {"name" : "Jason", "rating" : 4}}}, {"id" : 1, "name" : 1, "reviews.$": 1, "\_id" : 0})

Screenshot:

A screen shot of a computer

Description automatically generated

SUBMISSION

Submit the following files on BB:

* your lab7\_GroupXX.doc: contains
  + the question as per the lab file,
  + the mongodb command **in text** and
  + a screenshot of command and the output