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 collection in MongoDB database.
2. perform ad hoc queries on the database using the find or findOne functions and a query document.
3. query for ranges, set inclusion, inequalities, and more by using $-conditionals.
4. Group work acknowledgment

We, DIVNOOR SINGH BHANDOHAL AND SOPHIA HUYNH, 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- Sophia Huynh 1-8

2- Divnoor Bhandohal 9-15

3-

querying mongodb

Write MongoDB queries to answer each of the following questions.

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

Db.restaurants.find({“id”:1}).pretty()

Text

Description automatically generated with low confidence

**Restrict the search**

1. Find the restaurants with name “The Dutch”.

db.restaurants.find({“name”:”The Dutch”).pretty()

A picture containing text

Description automatically generated

1. Find the restaurants that have cuisine type “American”.

There should be documents 4, 6, 7, 8.

Db.restaurants.find({“cuisine\_type”:”American”}).pretty()

#8

A screenshot of a computer

Description automatically generated with medium confidence

#6

A picture containing table

Description automatically generated

#4

A picture containing text

Description automatically generated

#7

A screenshot of a computer

Description automatically generated with medium confidence

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

There should be documents 4, 6, 7, 8.

\*\*\* DOCUMENT 6 SHOULD NOT BE INCLUDED (document 6 contains neighborhood:Brooklyn \*\*\*

db.restaurants.find({"$and":[{"neighborhood":{"$in":["Manhattan"]}},{"cuisine\_type":{"$in":["American"]}}]}).pretty()

#8

A picture containing text

Description automatically generated

#4

A picture containing text

Description automatically generated

#7

A picture containing text

Description automatically generated

**Specifying which keys to return.**

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

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

Text

Description automatically generated

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

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

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

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

1. Show the restaurants whose id is between 3 and 5 inclusive. Documents 3, 4, 5 should show in the output.

db.restaurants.find({$and:[{"id":{"$gte":3}},{"id":{"$lte":5}}]}).pretty()

#5

A screenshot of a computer

Description automatically generated with medium confidence

#3

A picture containing text

Description automatically generated

#4

A picture containing text

Description automatically generated

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

**Or queries**

* "$in" can be used to query for a variety of values for a single key.
* "$or" is more general; it can be used to query for any of the given values across multiple keys.
* The opposite of "$in" is "$nin", which returns documents that don’t match any of the criteria in the array.

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

* Text

  Description automatically generated

1. Find the restaurants that have the cuisine type “Pizza” or “Mexican”. Use the "$in" operator. Show only id and cuisine\_type. The output should show document id 2, 5 and 10.

db.restaurants.find({"$or":[{"cuisine\_type":{"$in":["Pizza","Mexican"]}}]}).pretty()

#5

Text

Description automatically generated with low confidence

#2

A picture containing text

Description automatically generated

#10

Text

Description automatically generated

1. Find the restaurants that belong to "Brooklyn" neighborhood or have cuisine type "Pizza". Show only the id, name, cuisine\_type and neighborhood keys.

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

Text

Description automatically generated

1. Find the restaurants that belong to "Brooklyn" neighborhood and have cuisine type "Pizza". Show only the id, name, cuisine\_type and neighborhood keys. The output should have only restaurant id 1.

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

Text

Description automatically generated

**Querying arrays**

$all, $size

The $slice operator

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

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

Text

Description automatically generated with medium confidence

**Returning a matching array element**

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. The output should show restaurants 1 and 10.

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

A screenshot of a computer

Description automatically generated with low confidence

### **Querying on Embedded Documents**

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

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

db.restaurants.find({"reviews": {"$elemMatch": {name:"Jason",rating:4}}}, {\_id:0,id:1,name:1,neighborhood:1,adress:1,latlng:1,cuisine\_type:1,operating\_hours:1,"reviews": {"$elemMatch": {name:"Jason"}}}).pretty()

Text

Description automatically generated

1. Find the reviews provided by user “Jason” with rating equal 4 on restaurants whose cuisine\_type is Mexican. Show the id, name, cuisine\_type and only the matched review in the output.

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

A screenshot of a computer

Description automatically generated with low confidence

SUBMISSION

Each question is worth 5 pts.

Submit your lab7\_GroupX.doc file on BB. Replace X with your group number.

If a student does not contribute to the work, do not list his/her name(s) under the group section in the lab file and will get 0.