Lab 9 (2%)

MongoDB: Aggregation

Learning outcomes

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

1. Use the aggregation framework to transform and combine documents in a collection.
2. Run simple aggregation commands such as count, distinct and group.
3. Group work acknowledgment

We, DIVNOOR 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- Divnoor Bhandohal 1-6

2-Sophia Huynh 7-13

3-

Data:

Restaurant-data.json

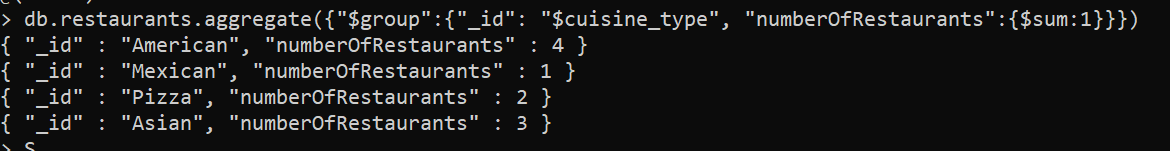
Aggregation in MongoDB

Write MongoDB queries to answer each of the following questions.

1. Show the number of restaurants per cuisine type. Hint: Use the aggregate() function and the $group operator.

Answer:

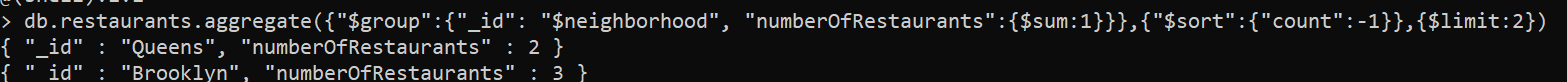
db.restaurants.aggregate({"$group":{"\_id": "$cuisine\_type", "numberOfRestaurants":{$sum:1}}})



1. Show the number of restaurants in each neighborhood. Sort the restaurants in descending order of their count and limit the results to the first 2 result documents.

Answer:

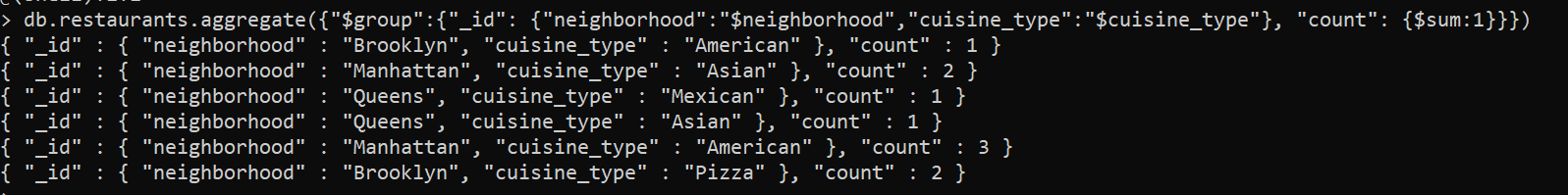
db.restaurants.aggregate({"$group":{"\_id": "$neighborhood", "numberOfRestaurants":{$sum:1}}},{"$sort":{"count":-1}},{$limit:2})



1. Show the number of restaurants per neighborhood and cuisine type. Use the aggregate() function and the $group operator.

Answer:

db.restaurants.aggregate({"$group":{"\_id": {"neighborhood":"$neighborhood","cuisine\_type":"$cuisine\_type"}, "count": {$sum:1}}})



Expected output:

{ "\_id" : { "neighborhood" : "Manhattan", "cuisine\_type" : "American" }, "count" : 3 }

{ "\_id" : { "neighborhood" : "Queens", "cuisine\_type" : "Mexican" }, "count" : 1 }

{ "\_id" : { "neighborhood" : "Manhattan", "cuisine\_type" : "Asian" }, "count" : 2 }

{ "\_id" : { "neighborhood" : "Brooklyn", "cuisine\_type" : "Pizza" }, "count" : 2 }

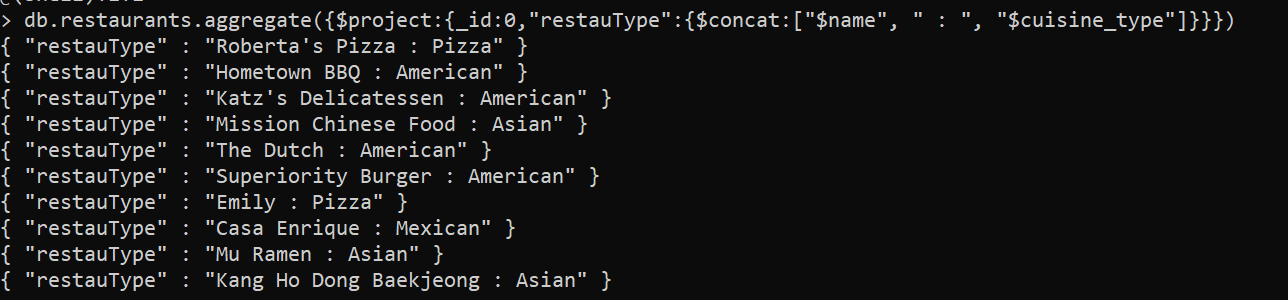
{ "\_id" : { "neighborhood" : "Queens", "cuisine\_type" : "Asian" }, "count" : 1 }

{ "\_id" : { "neighborhood" : "Brooklyn", "cuisine\_type" : "American" }, "count" : 1 }

1. Concatenate the restaurant name and the cuisine type into one string literal as restauType key. Hide \_id. Hint: Use the aggregate() function, $project operator and $concat operator.

Answer:

db.restaurants.aggregate({$project:{\_id:0,"restauType":{$concat:["$name", " : ", "$cuisine\_type"]}}})



Expected output:

{ "restautype" : "Katz's Delicatessen - American" }

{ "restautype" : "The Dutch - American" }

{ "restautype" : "Superiority Burger - American" }

{ "restautype" : "Emily - Pizza" }

{ "restautype" : "Hometown BBQ - American" }

{ "restautype" : "Kang Ho Dong Baekjeong - Asian" }

{ "restautype" : "Mission Chinese Food - Asian" }

{ "restautype" : "Mu Ramen - Asian" }

{ "restautype" : "Casa Enrique - Mexican" }

{ "restautype" : "Roberta's Pizza - Pizza" }

**count()**

1. calculate the number of documents in the collection.

Answer:

db.restaurants.count({})



**$unwind**

1. Unwind the restaurants collection to turn each review into its own document.

Answer:

db.restaurants.aggregate([{$unwind:"$reviews"},{$group:{"\_id":"$reviews"}}]).pretty()

1. Unwind the restaurants collection and show the number of reviews per restaurant.

Answer:

db.restaurants.aggregate([{ $unwind: { path: "$reviews"}}, {$group: {\_id: "$name", count: {$sum: 1}}}])

A screen shot of a computer

Description automatically generated with medium confidence

Expected output

{ "\_id" : "The Dutch", "count" : 3 }

{ "\_id" : "Casa Enrique", "count" : 3 }

{ "\_id" : "Mission Chinese Food", "count" : 3 }

{ "\_id" : "Hometown BBQ", "count" : 3 }

{ "\_id" : "Roberta's Pizza", "count" : 3 }

{ "\_id" : "Emily", "count" : 3 }

{ "\_id" : "Kang Ho Dong Baekjeong", "count" : 3 }

{ "\_id" : "Mu Ramen", "count" : 3 }

{ "\_id" : "Superiority Burger", "count" : 3 }

{ "\_id" : "Katz's Delicatessen", "count" : 3 }

1. Unwind the restaurants collection and show the number of reviews per user name.

Answer: db.restaurants.aggregate([{ $unwind: { path: "$reviews"}}, {$group: {\_id: "$reviews.name", count: {$sum: 1}}}])

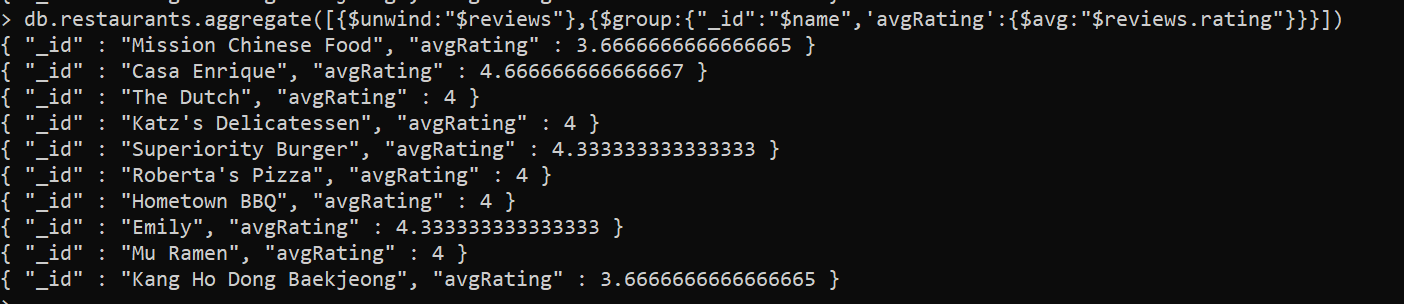
Text

Description automatically generated

1. Unwind the restaurants collection and show the average rating per restaurant.

Answer:

db.restaurants.aggregate([{$unwind:"$reviews"},{$group:{"\_id":"$name",'avgRating':{$avg:"$reviews.rating"}}}])



1. Unwind the restaurants collection and show the average rating per restaurant and sort them by avgRating in descending order.

db.restaurants.aggregate([{$unwind:"$reviews"},{$group:{"\_id":"$name",'avgRating':{$avg:"$reviews.rating"}}},{$sort:{"reviews.rating":-1}}])

Answer:

A screen shot of a computer

Description automatically generated with low confidence

1. Unwind the restaurants collection and show the average rating, min rating and max rating per restaurant and sort them by avgRating in descending order

Answer:

db.restaurants.aggregate([{$unwind:"$reviews"},{$group:{"\_id":"$name",'avgRating':{$avg:"$reviews.rating"},'minRating':{$min:"$reviews.rating"}, 'maxRating':{$max:"$reviews.rating"}}},{$sort:{"reviews.rating":-1}}])

**Text

Description automatically generated**

**Aggregation commands:**

**distinct**

1. Show the list of distinct cuisine types. Use the runCommand() function on the database.

Answer:

db.runCommand({aggregate:"restaurants",pipeline:[{$group:{\_id:null,"values":{$addToSet:"$cuisine\_type"}}},{$project:{\_id:0}}],cursor:{}})

Text

Description automatically generated

1. Show the list of distinct user names who provided reviews on the restaurants in the collection. Use the runCommand() function on the database..

Answer:

db.runCommand({aggregate:"restaurants",pipeline:[{$unwind:"$reviews"},{$group:{\_id:null,"values":{$addToSet:"$reviews.name"}}},{$project:{\_id:0}}],cursor:{}})

Text

Description automatically generated

SUBMISSION

Submit your lab9\_studentName.doc file on BB.

SUBMISSION

Submit your lab9\_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.

Submit the following files on BB:

* your lab8\_GroupX.doc: contains
  + the question as per the lab file,
  + the mongodb command **in text** and
  + a screenshot of the output.

Rubrics

Total: 65 pts