Alexander Powers
ECE:5995 Modern Databases
Prof. Guadalupe Canahuate
October 20th, 2020

The first two questions of this homework are answered by the code and corresponding outputs in the appendix.

3. Consider the need to identify the employee that performed the inspection. Employees have an employee number, first name, last name, address, phone number(s), preferred schedule (day and time for the week), and license plate. What changes/additions would you make to the database to store this information? Provide the rationale and implications (strengths and weaknesses) of your approach.

Two approaches immediately come to mind. 1) add a field to the city_inspections collection model that is a reference to a city_inspector document (which would be in it's own collection. 2) nest the city_inspection documents that correspond to each city inspector into a list of inspections.

If I were to implement this, I would choose to nest the city_inspections as a list inside of a new city_inspectors collection. The main benefit is that when querying all inspections done by one inspector, we only need one query instead of two. We also don't loose too much efficiency by nesting, as we can still have an index on the "city_inspector.inspections._id" attribute.

1) This is what the first option (adding a reference) might look like as a model

```
city_inspections
{
   "_id" : ObjectId,
   "id" : text,
   "inspector": RefId
   "certificate_number" : int,
   "business_name" : text,
   "date" : text,
   "result" : text,
   "sector" : text,
   "address" : {
       "city" : text,
       "zip" : int,
       "street" : text,
       "number" : int
   }
}

city_inspectors
{
   "_id" : ObjectId,
   "employee_number" : int,
```

```
   "first_name": text
   "last_name" : text,
   "phone_numbers" : [
       {
           "name": text,
           "number": text
       }
   ],
   "lisence_plate" : text,
   "schedule" : [
       {
           "day": text,
           "time": timestamp/text
       }
   ],
   "address" : {
       "city" : text,
       "zip" : int,
       "street" : text,
       "number" : int
   }
}
```

2) This is what the second option (nesting city_inspections) might look like as a model

```
city_inspectors
{
   "_id" : ObjectId,
   "employee_number" : int,
   "first_name": text
   "last_name" : text,
   "phone_numbers" : [
       {
           "name": text,
           "number": text
       }
   ],
   "lisence_plate" : text,
   "schedule" : [
       {
           "day": text,
           "time": timestamp/text
       }
   ],
   "address" : {
       "city" : text,
       "zip" : int,
       "street" : text,
       "number" : int
```

```
    },
    "inspections": [
        {
            "_id" : ObjectId,
            "id" : text,
            "inspector": RefId
            "certificate_number" : int,
            "business_name" : text,
            "date" : text,
            "result" : text,
            "sector" : text,
            "address" : {
                "city" : text,
                "zip" : int,
                "street" : text,
                "number" : int
            }
        }
    ]
}
```

Appendix

Q1

```
// The number of zipcodes in the state of New York
db.zips.find({ 'city': /^new york$/i }).count()
// 40
```

Q2

```
// The number of cities in  the state of New York.
db.zips.distinct('city', { 'state': /^NY$/i }).length
// 1370
```

Q3

```
// List  the  cities  in  the state of New York with more than 10 zip  codes in
alphabetical order.
// The result should include the name of the city, the state, and the number of zip codes.
db.zips.aggregate([
   { $match: { "state": { $eq: "NY" } } },
   {
       $group: {
           _id: "$city",
           state: { $first: "$state" },
           count: { $sum: 1 },
       }
   },
   { $match: { "count": { $gt: 10 } } },
   { $sort: { _id: 1 } }
])
// { "_id" : "BRONX", "state" : "NY", "count" : 25 }
// { "_id" : "BROOKLYN", "state" : "NY", "count" : 37 }
// { "_id" : "BUFFALO", "state" : "NY", "count" : 18 }
// { "_id" : "NEW YORK", "state" : "NY", "count" : 40 }
// { "_id" : "ROCHESTER", "state" : "NY", "count" : 19 }
// { "_id" : "STATEN ISLAND", "state" : "NY", "count" : 12 }
```

Q4

```
// List the cities in the state of New York with over 100K  people in descending order of
population.
db.zips.aggregate([
   { $match: { "state": { $eq: "NY" } } },
   {
       $group: {
           _id: "$city",
           state: { $first: "$state" },
           pop: { $sum: "$pop" },
       }
   },
   { $match: { "pop": { $gt: 100000 } } },
   { $sort: { pop: 1 } }
```

```
])
// { "_id" : "ASTORIA", "state" : "NY", "pop" : 165629 }
// { "_id" : "BRONX", "state" : "NY", "pop" : 1209548 }
// { "_id" : "BROOKLYN", "state" : "NY", "pop" : 2300504 }
// { "_id" : "BUFFALO", "state" : "NY", "pop" : 375479 }
// { "_id" : "FAR ROCKAWAY", "state" : "NY", "pop" : 100646 }
// { "_id" : "FLUSHING", "state" : "NY", "pop" : 224162 }
// { "_id" : "JACKSON HEIGHTS", "state" : "NY", "pop" : 145967 }
// { "_id" : "JAMAICA", "state" : "NY", "pop" : 195205 }
// { "_id" : "NEW YORK", "state" : "NY", "pop" : 1476790 }
// { "_id" : "ROCHESTER", "state" : "NY", "pop" : 396013 }
// { "_id" : "STATEN ISLAND", "state" : "NY", "pop" : 378977 }
// { "_id" : "SYRACUSE", "state" : "NY", "pop" : 184963 }
// { "_id" : "YONKERS", "state" : "NY", "pop" : 172131 }
```

Q5

```
// For the ten most popular cities in NY (the ones with the most population), get the
minimum and maximum latitude and longitude.
db.zips.aggregate([
   { $match: { "state": { $eq: "NY" } } },
   {
       $group: {
           _id: "$city",
           state: { $first: "$state" },
           pop: { $sum: "$pop" },
           min_lat_long: { $min: "$loc" },
           max_lat_long: { $max: "$loc" }
       }
   },
   { $sort: { pop: -1 } },
   { $limit: 10 }
])
// { "_id" : "BROOKLYN", "state" : "NY", "pop" : 2300504, "min_lat_long" : [ -74.030304,
40.625106 ], "max_lat_long" : [ -73.873649, 40.676191 ] }
// { "_id" : "NEW YORK", "state" : "NY", "pop" : 1476790, "min_lat_long" : [ -74.016323,
40.710537 ], "max_lat_long" : [ -73.922077, 40.866222 ] }
// { "_id" : "BRONX", "state" : "NY", "pop" : 1209548, "min_lat_long" : [ -73.921735,
40.8222 ], "max_lat_long" : [ -73.787436, 40.846941 ] }
// { "_id" : "ROCHESTER", "state" : "NY", "pop" : 396013, "min_lat_long" : [ -77.703996,
43.21257 ], "max_lat_long" : [ -77.549501, 43.14524 ] }
// { "_id" : "STATEN ISLAND", "state" : "NY", "pop" : 378977, "min_lat_long" : [
-74.244482, 40.508452 ], "max_lat_long" : [ -74.076795, 40.597296 ] }
// { "_id" : "BUFFALO", "state" : "NY", "pop" : 375479, "min_lat_long" : [ -78.897815,
42.949062 ], "max_lat_long" : [ -78.810375, 42.881132 ] }
// { "_id" : "FLUSHING", "state" : "NY", "pop" : 224162, "min_lat_long" : [ -73.873535,
40.772117 ], "max_lat_long" : [ -73.758646, 40.745847 ] }
// { "_id" : "JAMAICA", "state" : "NY", "pop" : 195205, "min_lat_long" : [ -73.811121,
40.702934 ], "max_lat_long" : [ -73.77584, 40.677483 ] }
```

```
// { "_id" : "SYRACUSE", "state" : "NY", "pop" : 184963, "min_lat_long" : [ -76.226159,
43.040943 ], "max_lat_long" : [ -76.104609, 43.042134 ] }
// { "_id" : "YONKERS", "state" : "NY", "pop" : 172131, "min_lat_long" : [ -73.895041,
40.917665 ], "max_lat_long" : [ -73.843435, 40.965574 ] }
```

Q6

```
// Count the number of inspections performed in the cities of Brooklyn, New York, or Bronx.
db.city_inspections.find({ "address.city": /(^bronx$)|(^new\syork$)|(^brooklyn$)/i }).count()
// 57088
```

Q7

```
// List the cities with inspections containing the keyword “bro” (case insensitive).
db.city_inspections.createIndex({ id: "text", business_name: "text", result: "text", sector:
"text" })
db.city_inspections.find({ $text: { $search: "bro" } })
// { "_id" : ObjectId("56d61034a378eccde8a90267"), "id" : "67922-2015-ENFO",
"certificate_number" : 9311307, "business_name" : "KAL-BRO, INC.", "date" : "Dec  1 2015",
"result" : "No Violation Issued", "sector" : "Secondhand Dealer [General] - 006", "address"
: { "city" : "COLLEGE POINT", "zip" : 11356, "street" : "14TH RD", "number" : 11414 } }
// { "_id" : ObjectId("56d61034a378eccde8a91947"), "id" : "55831-2015-ENFO",
"certificate_number" : 9315506, "business_name" : "JOHN'S BRO'S DELI & GROCERY INC.",
"date" : "Sep 25 2015", "result" : "No Violation Issued", "sector" : "Grocery-Retail -
808", "address" : { "city" : "BROOKLYN", "zip" : 11226, "street" : "CHURCH AVE", "number" :
1720 } }
```

Q8

```
// List the businesses that contain the keywords (case insensitive):  “gourmet”, “food”,
“corp” in the business name
// and whose inspection resulted in an issued violation.
db.city_inspections.createIndex({ business_name: "text", result: "text" })
db.city_inspections.find({ $text: { $search: "gourmet food corp" }, result:
/^(violation\sissued)$/i }, { _id: 0, business_name: 1, result: 1 })
// mongo hw4 --quiet --eval 'DBQuery.shellBatchSize = 10000; db.city_inspections.find({
$text: { $search: "gourmet food corp" }, result: /^(violation\sissued)$/i }, {_id: 0,
business_name: 1, result: 1})' > output.json
////////////////////////////
// OUTPUT at Q8_output.json
////////////////////////////
```

Q9

```
// Count the number of inspections per result performed in Feb 2015.
db.city_inspections.aggregate([
   { $match: { "date": /^(feb\s*\d{0,2}\s*2015)$/i } },
   {
       $group: {
           _id: "$result",
```

```
            count: { $sum: 1 },
        }
    }
])
// { "_id" : "Samples Obtained", "count" : 3 }
// { "_id" : "No Violation Issued", "count" : 2571 }
// { "_id" : "Unable to Locate", "count" : 13 }
// { "_id" : "No Evidence of Activity", "count" : 214 }
// { "_id" : "Licensed", "count" : 15 }
// { "_id" : "License Confiscated", "count" : 2 }
// { "_id" : "Re-inspection", "count" : 10 }
// { "_id" : "Warning", "count" : 46 }
// { "_id" : "Fail", "count" : 46 }
// { "_id" : "Pass", "count" : 894 }
// { "_id" : "Violation Issued", "count" : 1018 }
// { "_id" : "Posting Order Served", "count" : 12 }
// { "_id" : "Closed", "count" : 74 }
// { "_id" : "Out of Business", "count" : 463 }
// { "_id" : "NOH Withdrawn", "count" : 10 }
// { "_id" : "Unable to Complete Inspection", "count" : 3 }
```

Q10

```
// Which is the city with the most inspections in 2015?
db.city_inspections.aggregate([
    { $match: { "date": /^(.*2015)$/i } },
    {
        $group: {
            _id: "$address.city",
            count: { $sum: 1 },
        }
    },
    { $sort: { count: -1 } },
    { $limit: 1 }
])
// { "_id" : "BROOKLYN", "count" : 25024 }
```