

Assessment I

1. Restore the dump 1 from training folder into your running mongod. Perform a findOne on the collection called *question1* in the database *springboard2018*. That will return one document. Please provide the value corresponding to the "answer" key from the document returned.

Ans:

2. Which of the following are valid JSON documents? Please choose all that apply. (Multiple Answers)

- A. { "name" : "Fred Flinstone" ; "occupation": "Miner" ; "wife" : "Wilma" }
- B. { "title" : "Star Wars", "quotes" : ["Use the Force", "These are not the droids you are looking for"], "director" : "George Lucas" }
- C. {}
- D. { "city" = "New York", "population" = 7999034, "boroughs" = ["queens", "manhattan", "staten island", "the bronx", "brooklyn"] }
- E. { "a" : 1, "b" : { "b" : 1, "c" : "foo", "d" : "bar", "e" : [1, 2, 4] } }

3. Import the *grades.json* into your local mongo database *springboard2018*. The dataset contains 4 scores for 200 students. To confirm the successful import, the number of documents in grades collection should be 800.

Find all exam scores greater than or equal to 65 and sort those scores from lowest to highest. What is the *student_id* of the lowest exam score above 65?

- A. 115
- B. 22
- C. 48
- D. 67
- E. 87
- F. 114

4. Which of the choices below is the title of a movie from the year 2013 that is rated PG-13 and won no awards? Please query the video.*movieDetails* collection to find the answer.

- A. Journey to the West
- B. Saving Mr. Banks
- C. Escape from Planet Earth
- D. World War Z
- E. Man of Steel
- F. Iron Man 3
- G. Evil Dead
- H. A Tribute to J.J. Abrams
- I. Thor: The Dark World
- J. A Decade of Decadence, Pt. 2: Legacy of Dreams

5. Which optimization will typically have the greatest impact on the performance of a database?

- A. Adding more memory so that the working set fits in memory.
- B. Adding a faster drive so that operations that hit disk will happen more quickly.
- C. Replacing your CPU with a faster one (say one 2x as fast)
- D. Adding appropriate indexes on large collections so that only a small percentage of queries need to scan the collection.

6. Given the following output from explain, what is the best description of what happened during the query?

```
1 > exp = db.example.explain("executionStats")
2 Explainable(test.example)
3 > exp.find( { a : 7 } )
4 {
5   "queryPlanner" : {
6     "plannerVersion" : 1,
7     "namespace" : "test.example",
8     "indexFilterSet" : false,
9     "parsedQuery" : {
10      "a" : {
11        "$eq" : 7
12      }
13    },
14    "winningPlan" : {
15      "stage" : "COLLSCAN",
16      "filter" : {
17        "a" : {
18          "$eq" : 7
19        }
20      },
21      "direction" : "forward"
22    },
23    "rejectedPlans" : [ ]
24  },
25  "executionStats" : {
26    "executionSuccess" : true,
27    "nReturned" : 10000,
28    "executionTimeMillis" : 619,
29    "totalKeysExamined" : 0,
30    "totalDocsExamined" : 999999,
31    "executionStages" : {
32      "stage" : "COLLSCAN",
33      "filter" : {
34        "a" : {
35          "$eq" : 7
36        }
37      },
38      "nReturned" : 10000,
39      "executionTimeMillisEstimate" : 520,
40      "works" : 1000001,
41      "advanced" : 10000,
42      "needTime" : 990000,
43      "needFetch" : 0,
44      "saveState" : 7812,
45      "restoreState" : 7812,
46      "isEOF" : 1,
47      "invalidates" : 0,
48      "direction" : "forward",
49      "docsExamined" : 999999
50    }
51  },
52  "serverInfo" : {
53    "host" : "cross-mb-air.local",
```

```

54         "port" : 27017,
55         "version" : "3.0.1",
56         "gitVersion" : "534b5a3f9d10f00cd27737fbcd951032248b
57         5952"
58     },
59     "ok" : 1
60 }

```

Choose the best answer:

- A. The query used an index called BasicCursor and returned its results in 619 milliseconds.
- B. The query scanned 999,999 documents, returning 10,000 in 619 milliseconds.
- C. The query scanned 10,000 documents and return in 520 milliseconds.
- D. The query's explain plan is performed on a cursor.

7. You would like to perform a covered query on the example collection. You have the following indexes:

```

{ name : 1, dob : 1 }

{ _id : 1 }

{ hair : 1, name : 1 }

```

Which of the following is likely to be a covered query? Check all that apply.

- A. `db.example.find({ _id : 1117008 }, { _id : 0, name : 1, dob : 1 })`
- B. `db.example.find({ name : { $in : ["Alfred", "Bruce"] } }, { name : 1, hair : 1 })`
- C. `db.example.find({ name : { $in : ["Bart", "Homer"] } }, { _id : 0, hair : 1, name : 1 })`
- D. `db.example.find({ name : { $in : ["Bart", "Homer"] } }, { _id : 0, dob : 1, name : 1 })`

8. Suppose you have a collection with the following indexes:

```

1 > db.products.getIndexes()
2 [
3   {
4     "v" : 1,
5     "key" : {
6       "_id" : 1
7     },
8     "ns" : "store.products",
9     "name" : "_id_"
10  },
11  {
12    "v" : 1,
13    "key" : {
14      "sku" : 1
15    },

```

```

16         "unique" : true,
17         "ns" : "store.products",
18         "name" : "sku_1"
19     },
20     {
21         "v" : 1,
22         "key" : {
23             "price" : -1
24         },
25         "ns" : "store.products",
26         "name" : "price_-1"
27     },
28     {
29         "v" : 1,
30         "key" : {
31             "description" : 1
32         },
33         "ns" : "store.products",
34         "name" : "description_1"
35     },
36     {
37         "v" : 1,
38         "key" : {
39             "category" : 1,
40             "brand" : 1
41         },
42         "ns" : "store.products",
43         "name" : "category_1_brand_1"
44     },
45     {
46         "v" : 1,
47         "key" : {
48             "reviews.author" : 1
49         },
50         "ns" : "store.products",
51         "name" : "reviews.author_1"
52     }
53 ]

```

Which of the following queries can utilize at least one index to find all matching documents, or to sort? Check all that apply. **(Multiple Answers)**

- A. `db.products.find({ 'brand' : "GE" })`
- B. `db.products.find({ 'brand' : "GE" }).sort({ price : 1 })`
- C. `db.products.find({ $and : [{ price : { $gt : 30 } }, { price : { $lt : 50 } }] }).sort({ brand : 1 })`
- D. `db.products.find({ brand : 'GE' }).sort({ category : 1, brand : -1 })`

9. Finding the most frequent author of comments on your blog

In this problem you will use the aggregation framework to find the name of author who has added the most number of comments.

Start by importing the *posts* collection from *posts.json* into the *blog* database.

To help you verify your work before submitting, the author with the fewest comments is Mariela Sherer and she commented 387 times.

Please choose your answer below for the most prolific comment author:

Choose the answer:

- A. Gwyneth Garling
- B. Milan McGavock
- C. Elizabeth Kleine
- D. Dusti Lemmond
- E. Leonida Lafond

10. Which types of nodes can participate in elections of a new primary? (Multiple Answers)

- A. Regular replica set members
- B. Hidden Members
- C. Arbiters
- D. Lawyers

11. You have a sharded system with three shards and have sharded the collections "students" in the "school" database across those shards. The output of sh.status() when connected to mongos looks like this:

```
1  mongos> sh.status()
2  --- Sharding Status ---
3    sharding version: {
4      "_id" : 1,
5      "minCompatibleVersion" : 5,
6      "currentVersion" : 6,
7      "clusterId" : ObjectId("5531512ac723271f602db407")
8    }
9    shards:
10     { "_id" : "s0", "host" : "s0/localhost:37017,localhost
:37018,localhost:37019" }
11     { "_id" : "s1", "host" : "s1/localhost:47017,localhost
:47018,localhost:47019" }
12     { "_id" : "s2", "host" : "s2/localhost:57017,localhost
:57018,localhost:57019" }
13    balancer:
14      Currently enabled:  yes
15      Currently running:  yes
16      Balancer lock taken at Fri Apr 17 2015 14:32:02 GMT-
0400 (EDT) by education-iMac-2.local:27017:1429295401:16807:
Balancer:1622650073
17      Collections with active migrations:
18        school.students started at Fri Apr 17 2015 14:32:03
GMT-0400 (EDT)
19      Failed balancer rounds in last 5 attempts:  0
20      Migration Results for the last 24 hours:
21        2 : Success
22        1 : Failed with error 'migration already in progress
', from s0 to s1
23    databases:
24     { "_id" : "admin", "partitioned" : false, "primary" :
"config" }
25     { "_id" : "school", "partitioned" : true, "primary" :
"s0" }
26     school.students
27       shard key: { "student_id" : 1 }
28     chunks:
29       s0 1
```

```

30             s1 3
31             s2 1
32             { "student_id" : { "$minKey" : 1 } } -->> { "stu
dent_id" : 0 } on : s2 Timestamp(3, 0)
33             { "student_id" : 0 } -->> { "student_id" : 2 } o
n : s0 Timestamp(3, 1)
34             { "student_id" : 2 } -->> { "student_id" : 3497
} on : s1 Timestamp(3, 2)
35             { "student_id" : 3497 } -->> { "student_id" : 77
78 } on : s1 Timestamp(3, 3)
36             { "student_id" : 7778 } -->> { "student_id" : {
"$maxKey" : 1 } } on : s1 Timestamp(3, 4)

```

If you ran the below query on students collection:

```

use school

db.students.find({'student_id':2000})

```

Which shards would be involved in answering the query?

- A. s0, s1, and s2
- B. s0
- C. s1
- D. s2