Lab 3: Indexing

Exercise 1: Optimize an index using a query

In this lab, we will create a collection with sensor data and then try to find the ideal index for a specific query:

1. Open a Mongo Shell with the according user defined functions:

```
> mongo --shell lab_03.js
```

2. Now create and initialize the data base as following:

```
> init()
```

3. The function created 20.000 documents. Please verify as following:

```
> use lab_03
> db.sensor_readings.count()
```

Now we want to create the optimal index for the following query:

```
db.sensor_readings.find( {
  tstamp : {
     $gte : ISODate("2012-08-01"),
     $lte :  ISODate("2012-09-01")
  },
  active : true
} ).limit(3)
```

At the beginning there will be only the standard index on _id and a full table scan is done:

```
> db.sensor_readings.find( {
        tstamp : {
           $gte : ISODate("2012-08-01"),
           $lte :  ISODate("2012-09-01")
        },
```

```
        active : true
      } ).limit(3).explain()

{
        "cursor" : "BasicCursor",
        "isMultiKey" : false,
        "n" : 3,
        "nscannedObjects" : 2696,
        "nscanned" : 2696,
        "nscannedObjectsAllPlans" : 2696,
        "nscannedAllPlans" : 2696,
        "scanAndOrder" : false,
        "indexOnly" : false,
        "nYields" : 0,
        "nChunkSkips" : 0,
        "millis" : 1,
        "indexBounds" : {

        }
}
```

A full table scan is never a good idea which means there is a lot of room for improvement. Please think about an ideal index, create it and check the improvement via `explain()`.

If you think you got a good one, please check as following:

```
> testIndex()
```

The result of the function keeps bothering that you still can do better? Then drop your index and try again until you found the perfect solution!

Exercise 2: Quiz Time

Assume you have a collection `quiztime` with the following indexes:

```
[
	{
		"v" : 1,
		"key" : {
			"_id" : 1
		},
		"ns" : "lab_03.quiztime",
		"name" : "_id_"
	},
	{
		"v" : 1,
		"key" : {
			"a" : 1,
			"b" : 1
		},
		"ns" : " lab_03.quiztime ",
		"name" : "a_1_b_1"
	},
	{
		"v" : 1,
		"key" : {
			"a" : 1,
			"c" : 1
		},
		"ns" : " lab_03.quiztime ",
		"name" : "a_1_c_1"
	},
	{
		"v" : 1,
		"key" : {
			"c" : 1
		},
		"ns" : " lab_03.quiztime ",
		"name" : "c_1"
	},
	{
		"v" : 1,
		"key" : {
			"a" : 1,
			"b" : 1,
			"c" : -1
		},
		"ns" : " lab_03.quiztime ",
		"name" : "a_1_b_1_c_-1"
	}
]
```

Now you want to execute the following query:

```
> db.quiztime.find({'a':{'$lt':10000}, 'b':{'$gt': 5000}},
          {'a':1, 'c':1}).sort({'c':-1})
```

Which of the following indexes can MongoDB use for this query? Mark all correct answers!

Answer 1:
_id

Answer 2:
a_1_b_1

Answer 3:
a_1_c_1

Answer 4:
c_1

Answer 5:
a_1_b_1_c_-1

Notice: Using `initQuiztime()` you can create a collection with some sample data if you prefer to find the correct answers in a more practical approach…