

## Big Data Lab – Week 4: Working with MongoDB Database

### [solution]

#### Task 1: Basics of Working with MongoDB

8. Results of the first query include all the fields in the collection; results of the second query only include the specified 'customer' field and the '\_id' field.

10. There was one index before running this command, and now there are two indexes after the command executed.

```
> db.basket.createIndex({"customer " : 1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

11.

```
> db.basket.getIndexes()
[
  {
    "v" : 2,
    "key" : { "_id" : 1 },
    "name" : "_id_",
    "ns" : "myDatabase.basket"
  },
  {
    "v" : 2,
    "key" : { "customer " : 1 },
    "name" : "customer_1",
    "ns" : "myDatabase.basket"
  }
]
```

#### Task2: Importing Data into MongoDB

3. 25359 documents were imported into the restaurants collection in the test database. The "--drop" option drops the collection (if it already exists) before importing the data from the input.

4.

```
> use test
switched to db test
> db.restaurants.find()
```

5.

```
> db.restaurants.find({name:"Taste The Tropics Ice Cream"}, {"borough":1, "grades": 1 }).pretty()
{
  "_id" : ObjectId("59e565aa8b8edcdd5b7a43c4"),
  "borough" : "Brooklyn",
  "grades" : [
    {
      "date" : ISODate("2014-07-14T00:00:00Z"),
      "grade" : "A",
      "score" : 12
    },
    {
      "date" : ISODate("2013-07-10T00:00:00Z"),
      "grade" : "A",
      "score" : 8
    },
    {
      "date" : ISODate("2012-07-11T00:00:00Z"),
      "grade" : "A",
      "score" : 5
    },
    {
      "date" : ISODate("2012-02-23T00:00:00Z"),
      "grade" : "A",
      "score" : 8
    }
  ]
}
```

### Task 3: Querying MongoDB

2. Find how many different types of cuisine are represented in the restaurants data.

```
> db.restaurants.distinct("cuisine").length
```

```
85
```

3. List the address of all the **Hawaiian** restaurants (there are 3)

```
> db.restaurants.find({"cuisine":"Hawaiian"}, {"address":1, "cuisine":1, "_id":0}).pretty()
```

```
{
  "address" : {
    "building" : "2245",
    "coord" : [
      -73.93600099999999,
      40.795675
    ]
  },
  "cuisine" : "Hawaiian"
}
```

```

        "street" : "1 Avenue",
        "zipcode" : "10029"
      },
      "cuisine" : "Hawaiian"
    }
  {
    "address" : {
      "building" : "360",
      "coord" : [
        -73.9854351,
        40.7419304
      ],
      "street" : "Park Avenue South",
      "zipcode" : "10010"
    },
    "cuisine" : "Hawaiian"
  }
  {
    "address" : {
      "building" : "84",
      "coord" : [
        -73.9562856,
        40.7139315
      ],
      "street" : "Havemeyer Street",
      "zipcode" : "11211"
    },
    "cuisine" : "Hawaiian"
  }
}

```

4. List the title and awards of movies directed by **"Sergio Leone"** (there are 2)

```
> db.movieDetails.find({"director":"Sergio Leone"}, {"title":1, "director":1, "awards":1,
"_id":0}).pretty()
```

```

{
  "title" : "Once Upon a Time in the West",
  "director" : "Sergio Leone",
  "awards" : {
    "wins" : 4,
    "nominations" : 5,
    "text" : "4 wins & 5 nominations."
  }
}
{
  "title" : "Once Upon a Time in America",
  "director" : "Sergio Leone",
  "awards" : {
    "wins" : 11,
    "nominations" : 7,
    "text" : "Nominated for 2 Golden Globes. Another 11 wins & 7 nominations."
  }
}

```

5. Find the title and countries of movies released before the year of **1897**. (there are 2)

```
> db.movieDetails.find({year:{<1897}}, {title:1, year:1, countries:1, _id:0}).pretty()
```

```
{
  "title" : "L.L. M.M. le Tsar et la Tsarine entrant dans l'église de l'As
sompion [Moscou]",
  "year" : 1896,
  "countries" : [
    "France"
  ]
}
{
  "title" : "Monza, L.L. M.M. le Roi et la Reine d'Italie",
  "year" : 1896,
  "countries" : [
    "France"
  ]
}
```

6. Find the details of the latest movie directed by “**George Lucas**”, omit the poster information in the result

```
> db.movieDetails.find({director: "George Lucas"}, {poster:0}). sort({year:-
1}).limit(1).pretty()
```

```
{
  "_id" : ObjectId("59ee697aaa1d2db3dc4fbfa3"),
  "title" : "Star Wars: Episode III - Revenge of the Sith",
  "year" : 2005,
  "rated" : "PG-13",
  "runtime" : 140,
  "countries" : [
    "USA"
  ],
  "genres" : [
    "Action",
    "Adventure",
    "Fantasy"
  ],
  "director" : "George Lucas",
  "writers" : [
    "George Lucas"
  ],
  "actors" : [
    "Ewan McGregor",
    "Natalie Portman",
    "Hayden Christensen",
    "Ian McDiarmid"
  ],
  "plot" : "Three years after the onset of the Clone Wars; the noble Jedi
Knights are spread out across the galaxy leading a massive clone army in the war
against the Separatists. After Chancellor ...",
  "imdb" : {
    "id" : "tt0121766",
```

```

        "rating" : 7.6,
        "votes" : 483613
    },
    "tomato" : {
        "meter" : 79,
        "image" : "certified",
        "rating" : 7.3,
        "reviews" : 282,
        "fresh" : 222,
        "consensus" : "With Episode III: Revenge of the Sith, George Luc
as brings his second Star Wars trilogy to a suitably thrilling and often poignan
t -- if still a bit uneven -- conclusion.",
        "userMeter" : 65,
        "userRating" : 3.1,
        "userReviews" : 33674396
    },
    "metacritic" : 68,
    "awards" : {
        "wins" : 21,
        "nominations" : 49,
        "text" : "Nominated for 1 Oscar. Another 21 wins & 49 nomination
s."
    },
    "type" : "movie"
}

```

7. Find all Scandinavian restaurants in Manhattan, showing only the street and name of each restaurant.

```

> db.restaurants.find({"borough":"Manhattan", "cuisine":"Scandinavian"}, {"name":1,
"address.street":1, "_id":0})

{ "address" : { "street" : "Stone Street" }, "name" : "Smorgas Chef/ Crepes Du Nord" }
{ "address" : { "street" : "East 55 Street" }, "name" : "Aquavit" }
{ "address" : { "street" : "Park Avenue" }, "name" : "Smorgas Chef At Scandinavi
a House" }
{ "address" : { "street" : "Pearl Street" }, "name" : "Fika" }
{ "address" : { "street" : "Washington Street" }, "name" : "Fika" }
{ "address" : { "street" : "Laight Street" }, "name" : "Aamanns-Copenhgen" }

```

8. Find the details of the movies which have actors "**Nicolas Cage**" and "**Dennis Hopper**". (there is 1)

```

> db.movieDetails.find({actors: {$all:["Nicolas Cage","Dennis Hopper"]}}).pretty()

{
  "_id" : ObjectId("59ee697aaa1d2db3dc4fbf9b"),
  "title" : "Red Rock West",
  "year" : 1993,
  "rated" : "R",
  "runtime" : 98,
  "countries" : [
    "USA"
  ],
  "genres" : [

```

```

        "Crime",
        "Drama",
        "Thriller"
    ],
    "director" : "John Dahl",
    "writers" : [
        "John Dahl",
        "Rick Dahl"
    ],
    "actors" : [
        "Nicolas Cage",
        "Dennis Hopper",
        "Lara Flynn Boyle",
        "J.T. Walsh"
    ],
    "plot" : "When a promised job for Texan Michael fails to materialise in Wyoming, Mike is mistaken by Wayne to be the hitman he hired to kill his unfaithful wife, Suzanne. Mike takes full advantage of...",
    "poster" : "http://ia.media-imdb.com/images/M/MV5BMTk0Mjg4OTc3OF5BMT5BanBnXkFtZTYwNzg5OTg5._V1_SX300.jpg",
    "imdb" : {
        "id" : "tt0105226",
        "rating" : 7,
        "votes" : 15007
    },
    "awards" : {
        "wins" : 0,
        "nominations" : 3,
        "text" : "3 nominations."
    },
    "type" : "movie"
}

```

9. Find out how many movies include **"Jim Thomas"** or **"Flint Dille"** as one of the writers

```
> db.movieDetails.find({writers:{$in:["Jim Thomas", "Flint Dille"]}}).count()
```

```
3
```

10. List the names of all restaurants with a grade dated **2011-03-03**.  
[Hint: use ISODate("2011-03-03T00:00:00Z") to specify the date]

```
> db.restaurants.find({"grades.date":ISODate("2011-03-03T00:00:00Z")},{ "_id":0,"name":1}).pretty()
```

```

{ "name" : "The Assembly Bar" }
{ "name" : "Benny'S Famous Pizza" }
{ "name" : "Ray'S Pizza Restuarant" }
{ "name" : "La Piazza Pizzeria & Restaurant" }
{ "name" : "Kennedy Fried Chicken" }
{ "name" : "Barbara Blum Residence / Good Shepherd Services" }
{ "name" : "Le Pain Quotidien" }
{ "name" : "Blimpie" }

```

*Challenge:* use the *\$elemMatch* operator in the projection document to include the grade with that date, and no other grades. Only the first document in the desired result is shown here:

```
> db.restaurants.find({"grades.date":ISODate("2011-03-03T00:00:00Z")},{"_id":0,"name":1,"grades":{"$elemMatch":{"date":ISODate("2011-03-03T00:00:00Z")}}}).pretty()
```

```
{
  "grades" : [
    {
      "date" : ISODate("2011-03-03T00:00:00Z"),
      "grade" : "A",
      "score" : 6
    }
  ],
  "name" : "The Assembly Bar"
}
```

11. Find all Irish restaurants that have at least one for each of the grades A, B, C and Z.

```
> db.restaurants.find({"cuisine":"Irish","grades.grade":{"$all":["A","B","C","Z"]}},{"name":1,"grades.grade":1,"_id":0}).pretty()
```

```
{
  "grades" : [
    {
      "grade" : "Z"
    },
    {
      "grade" : "A"
    },
    {
      "grade" : "A"
    },
    {
      "grade" : "C"
    },
    {
      "grade" : "B"
    }
  ],
  "name" : "Niall'S On 52Nd"
}
```