NGT SLIPS

Remaining Q - 9b,11,14b,19,23

1. (A) Import Restaurant.json into MongoDb and perform the following queries:

Open cmd :

Type >mongoD

```
C:\WINDOWS\system32>mongoD
2021-10-08T12:03:30.370+0530 I CONTROL  [initandlisten] MongoDB starting : pid=45144 port=27017 dbpath=C:\data\db\ 64-
bit host=LAPTOP-KA8632CR
2021-10-08T12:03:30.373+0530 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2021-10-08T12:03:30.378+0530 I CONTROL  [initandlisten] db version v3.6.0
2021-10-08T12:03:30.378+0530 I CONTROL  [initandlisten] git version: a57d8e71e6998a2d0afde7edc11bd23e5661c915
2021-10-08T12:03:30.379+0530 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.0.1u-fips  22 Sep 2016
2021-10-08T12:03:30.379+0530 I CONTROL  [initandlisten] allocator: tcmalloc
2021-10-08T12:03:30.379+0530 I CONTROL  [initandlisten] modules: none
2021-10-08T12:03:30.379+0530 I CONTROL  [initandlisten] build environment:
2021-10-08T12:03:30.380+0530 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
2021-10-08T12:03:30.380+0530 I CONTROL  [initandlisten]     distarch: x86_64
2021-10-08T12:03:30.380+0530 I CONTROL  [initandlisten]     target_arch: x86_64
2021-10-08T12:03:30.380+0530 I CONTROL  [initandlisten] options: {}
2021-10-08T12:03:30.462+0530 I -        [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger'
 storage engine, so setting the active storage engine to 'wiredTiger'.
2021-10-08T12:03:30.465+0530 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=3541M,session_max=20
000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=jou
rnal,compressor=snappy),file_manager=(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2021-10-08T12:03:31.194+0530 I STORAGE  [initandlisten] WiredTiger message [1633674811:193632][45144:140711214273872],
 txn-recover: Main recovery loop: starting at 10/9344
2021-10-08T12:03:31.357+0530 I STORAGE  [initandlisten] WiredTiger message [1633674811:357523][45144:140711214273872],
 txn-recover: Recovering log 10 through 11
2021-10-08T12:03:31.506+0530 I STORAGE  [initandlisten] WiredTiger message [1633674811:505550][45144:140711214273872],
 txn-recover: Recovering log 11 through 11
2021-10-08T12:03:32.861+0530 I CONTROL  [initandlisten]
```

Open cmd again:

Type >mongo

```
C:\WINDOWS\system32>mongo
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
Server has startup warnings:
2021-10-08T12:03:32.861+0530 I CONTROL  [initandlisten]
2021-10-08T12:03:32.862+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the
2021-10-08T12:03:32.863+0530 I CONTROL  [initandlisten] **          Read and write access to data and cont
2021-10-08T12:03:32.863+0530 I CONTROL  [initandlisten]
2021-10-08T12:03:32.864+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2021-10-08T12:03:32.864+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to conne
2021-10-08T12:03:32.864+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <addre
2021-10-08T12:03:32.865+0530 I CONTROL  [initandlisten] **          addresses it should serve responses fr
2021-10-08T12:03:32.865+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavi
2021-10-08T12:03:32.865+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to dis
2021-10-08T12:03:32.865+0530 I CONTROL  [initandlisten]
2021-10-08T12:03:32.866+0530 I CONTROL  [initandlisten]
2021-10-08T12:03:32.866+0530 I CONTROL  [initandlisten] ** WARNING: The file system cache of this machine
poor performance.
2021-10-08T12:03:32.866+0530 I CONTROL  [initandlisten] See http://dochub.mongodb.org/core/wt-windows-syst
2021-10-08T12:03:32.867+0530 I CONTROL  [initandlisten]
```

Open cmd again:

Type >mongoimport --db tymongo --collection rest C:\mongoDB\restaurants.json

```
C:\WINDOWS\system32>mongoimport --db tymongo --collection rest C:\mongoDB\restaurants.json
2021-10-08T12:03:47.682+0530    connected to: localhost
2021-10-08T12:03:48.071+0530    imported 3772 documents

C:\WINDOWS\system32>
```

1. Write a MongoDB query to display all the documents in the collection restaurants.

use tymongo
switched to db tymongo

> db.rest.find().pretty();

```
> use tymongo
switched to db tymongo
> db.rest.find().pretty();
{
        "_id" : ObjectId("615fe64b697bf35275a76349"),
        "address" : {
                "building" : "469",
                "coord" : [
                        -73.961704,
                        40.662942
                ],
                "street" : "Flatbush Avenue",
                "zipcode" : "11225"
        },
        "borough" : "Brooklyn",
        "cuisine" : "Hamburgers",
        "grades" : [
                {
                        "date" : ISODate("2014-12-30T00:00:00Z"),
                        "grade" : "A",
                        "score" : 8
                },
                {
                        "date" : ISODate("2014-07-01T00:00:00Z"),
                        "grade" : "B",
```

2. Write a MongoDB query to display the fields , restaurant_id, name, borough and cuisine for all the documents in the collection restaurant.

>db.rest.find({},{"restaurant_id":1,"name":1,"borough":1,"cuisine" :1,"_id":0}).pretty();

```
> db.rest.find({},{"restaurant_id":1,"name":1,"borough":1,"cuisine" :1,"_id":0}).pretty();
{
        "borough" : "Brooklyn",
        "cuisine" : "Hamburgers",
        "name" : "Wendy'S",
        "restaurant_id" : "30112340"
}
{
        "borough" : "Bronx",
        "cuisine" : "Bakery",
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
}
{
        "borough" : "Manhattan",
        "cuisine" : "Irish",
        "name" : "Dj Reynolds Pub And Restaurant",
        "restaurant_id" : "30191841"
```

3. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine, but exclude the field _id for all the documents in the collection restaurant.

>db.rest.find({},{"restaurant_id":1,"name":1,"borough":1,"cuisine" :1,"_id":0}).pretty();

```
> db.rest.find({},{"restaurant_id":1,"name":1,"borough":1,"cuisine" :1,"_id":0}).pretty();
{
        "borough" : "Brooklyn",
        "cuisine" : "Hamburgers",
        "name" : "Wendy'S",
        "restaurant_id" : "30112340"
}
{
        "borough" : "Bronx",
        "cuisine" : "Bakery",
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
}
{
        "borough" : "Manhattan",
```

4. Write a MongoDB query to display the fields restaurant_id, name, borough and zip code, but exclude the field _id for all the documents in the collection restaurant

>db.rest.find({},{"restaurant_id":1,"name":1,"borough":1,"address.zipcode" :1,"_id":0}).pretty();

```
> db.rest.find({},{"restaurant_id":1,"name":1,"borough":1,"address.zipcode" :1,"_id":0}).pretty();
{
        "address" : {
                "zipcode" : "11225"
        },
        "borough" : "Brooklyn",
        "name" : "Wendy'S",
        "restaurant_id" : "30112340"
}
{
        "address" : {
                "zipcode" : "10462"
        },
        "borough" : "Bronx",
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
}
{
        "address" : {
                "zipcode" : "10019"
        },
        "borough" : "Manhattan",
        "name" : "Dj Reynolds Pub And Restaurant",
        "restaurant_id" : "30191841"
}
```

5. Write a MongoDB query to display all the restaurant which is in the borough Bronx

>db.rest.find({"borough": "Bronx"}).pretty();

```
> db.rest.find({"borough": "Bronx"}).pretty();
{
        "_id" : ObjectId("615fe64b697bf35275a7634a"),
        "address" : {
                "building" : "1007",
                "coord" : [
                        -73.856077,
                        40.848447
                ],
                "street" : "Morris Park Ave",
                "zipcode" : "10462"
        },
        "borough" : "Bronx",
        "cuisine" : "Bakery",
        "grades" : [
                {
                        "date" : ISODate("2014-03-03T00:00:00Z"),
                        "grade" : "A",
                        "score" : 2
                },
                {
```

(B) Connect Python with MongoDB and also insert and retrieve documents.

Refer P5

2. (A) Write a jQuery to animate multiple CSS properties.

Refer P8 - 11Q

(B) Write a jQuery effect method with a callback function.

Refer P8 – 13Q

3. Connect Java with MongoDB and also insert, retrieve, update and delete documents.

   Refer P6

4. Create a JSON file and persist it in any database.

   Refer P10

5.(A) Import Restaurant.json into MongoDb and perform the following queries:

1. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name

```
>db.rest.find({name :{ $regex : "mon.*",$options: "i"}},{"name":1,"borough":1,"address.coord":1,
```

“cuisine” :1}).pretty();

```
> db.rest.find({name :{ $regex : "mon.*",$options: "i"}},{"name":1,"borough":1,"address.coord":1,"cuisine":1}).pretty()
{
        "_id" : ObjectId("615fe64b697bf35275a763e4"),
        "address" : {
                "coord" : [
                        -73.98306099999999,
                        40.7441419
                ]
        },
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Desmond'S Tavern"
}
{
        "_id" : ObjectId("615fe64b697bf35275a763e9"),
        "address" : {
                "coord" : [
                        -73.8221418,
                        40.7272376
                ]
        },
        "borough" : "Queens",
        "cuisine" : "Jewish/Kosher",
        "name" : "Shimons Kosher Pizza"
```

2. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name

```
> db.rest.find( { name :{ $regex : /^Mad/i, }},
{"name":1,"borough":1,"address.coord":1,"cuisine"
:1}).pretty();
```

```
> db.rest.find( { name :{ $regex : /^Mad/i, }},      {"name":1,"borough":1,"address.coord":1,"cuisine" :1}
{
        "_id" : ObjectId("615fe64b697bf35275a76885"),
        "address" : {
                "coord" : [
                        -73.9860597,
                        40.7431194
                ]
        },
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Madison Square"
}
{
        "_id" : ObjectId("615fe64b697bf35275a76952"),
        "address" : {
                "coord" : [
                        -73.98302199999999,
                        40.742313
                ]
        },
        "borough" : "Manhattan",
        "cuisine" : "Indian",
        "name" : "Madras Mahal"
}
{
        "_id" : ObjectId("615fe64c697bf35275a76c04"),
        "address" : {
                "coord" : [
```

3. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168

db.rest.find({"address.coord":{$lt:95.754168}}). pretty();

```
> db.rest.find({"address.coord":{$lt:95.754168}}). pretty();
{
        "_id" : ObjectId("615fe64b697bf35275a76349"),
        "address" : {
                "building" : "469",
                "coord" : [
                        -73.961704,
                        40.662942
                ],
                "street" : "Flatbush Avenue",
                "zipcode" : "11225"
        },
        "borough" : "Brooklyn",
        "cuisine" : "Hamburgers",
        "grades" : [
                {
                        "date" : ISODate("2014-12-30T00:00:00Z"),
                        "grade" : "A",
                        "score" : 8
                },
                {
                        "date" : ISODate("2014-07-01T00:00:00Z"),
                        "grade" : "B",
                        "score" : 23
                },
                {
                        "date" : ISODate("2013-04-30T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
```

4. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

```
> db.rest.find( {name: /^Wil/}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
```

```
> db.rest.find( {name: /^Wil/}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
{
        "_id" : ObjectId("615fe64b697bf35275a7634f"),
        "borough" : "Brooklyn",
        "cuisine" : "Delicatessen",
        "name" : "Wilken'S Fine Food",
        "restaurant_id" : "40356483"
}
{
        "_id" : ObjectId("615fe64b697bf35275a76353"),
        "borough" : "Bronx",
        "cuisine" : "American ",
        "name" : "Wild Asia",
        "restaurant_id" : "40357217"
}
{
        "_id" : ObjectId("615fe64c697bf35275a7715b"),
        "borough" : "Bronx",
        "cuisine" : "Pizza",
        "name" : "Wilbel Pizza",
        "restaurant_id" : "40871979"
```

5. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

```
> db.rest.find( {name: /ces$/}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
```

```
> db.rest.find( {name: /ces$/}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
{
        "_id" : ObjectId("615fe64b697bf35275a767de"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Pieces",
        "restaurant_id" : "40399910"
}
{
        "_id" : ObjectId("615fe64b697bf35275a768a1"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Good Shepherd Services",
        "restaurant_id" : "40403989"
}
{
        "_id" : ObjectId("615fe64b697bf35275a768a3"),
        "borough" : "Queens",
        "cuisine" : "American ",
        "name" : "S.M.R Restaurant Services",
        "restaurant_id" : "40403857"
}
{
        "_id" : ObjectId("615fe64c697bf35275a76d54"),
        "borough" : "Queens",
        "cuisine" : "Ice Cream, Gelato, Yogurt, Ices",
```

(B) Connect Python with MongoDB and also insert and update documents.

Refer P5

6.(A) Write a jQuery to Select elements by class name,id and element name.

Refer P8 -2Q

(B)Write a jQuery to show the use of Click (), hover (), on (), trigger (), off () events.

Refer P8 -3Q

7. Connect PHP with MongoDB and also insert, retrieve, update and delete documents.

Refer P7

8. (A) Create a JSON file and parse it.

Refer P9

(B) Write a jQuery to Create slide-up, slide-down and slide-Toggle effect.

Refer P8 -8Q, 9 Q

9. (A) Import Restaurant.json into MongoDb and perform the following queries

1. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168

```
db.rest.find( {$and: [ {"cuisine" : {$ne :"American "}}, {"grades.score" : {$gt : 70}}, {"address.coord" : {$lt : -65.754168}} ] } ).pretty();
```

```
> db.rest.find( {$and: [ {"cuisine" : {$ne :"American "}}, {"grades.score" : {$gt : 70}}, {"address.coord" : {$lt : -65.754168}} ] } ).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae848b"),
        "address" : {
                "building" : "345",
                "coord" : [
                        -73.9864626,
                        40.7266739
                ],
                "street" : "East 6 Street",
                "zipcode" : "10003"
        },
        "borough" : "Manhattan",
        "cuisine" : "Indian",
        "grades" : [
                {
                        "date" : ISODate("2014-09-15T00:00:00Z"),
                        "grade" : "A",
                        "score" : 5
                },
                {
                        "date" : ISODate("2014-01-14T00:00:00Z"),
                        "grade" : "A",
                        "score" : 8
                },
                {
                        "date" : ISODate("2013-05-30T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                },
                {
```

2. Write a MongoDB query to find the restaurants which do not

prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.

```
> db.rest.find( { "cuisine" : {$ne : "American "}, "grades.score" :{$gt: 70},"address.coord" : {$lt : -65.754168} } ).pretty();
```

```
> db.rest.find( { "cuisine" : {$ne : "American "}, "grades.score" :{$gt: 70},"address.coord" : {$lt : -65.754168} } ).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae848b"),
        "address" : {
                "building" : "345",
                "coord" : [
                        -73.9864626,
                        40.7266739
                ],
                "street" : "East 6 Street",
                "zipcode" : "10003"
        },
        "borough" : "Manhattan",
        "cuisine" : "Indian",
        "grades" : [
                {
                        "date" : ISODate("2014-09-15T00:00:00Z"),
                        "grade" : "A",
                        "score" : 5
                },
                {
                        "date" : ISODate("2014-01-14T00:00:00Z"),
                        "grade" : "A",
                        "score" : 8
                },
                {
                        "date" : ISODate("2013-05-30T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                },
                {
                        "date" : ISODate("2013-04-24T00:00:00Z"),
                        "grade" : "P",
                        "score" : 2
```

3. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A'

not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order

```
db.rest.find({"cuisine" : {$ne : "American "},
 "grades.grade" :"A","borough":"Brooklyn"}
).sort({"cuisine":-1}).pretty();
```

```
> db.rest.find({"cuisine" : {$ne : "American "},
...  "grades.grade" :"A","borough":"Brooklyn"} ).sort({"cuisine":-1}).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae8a96"),
        "address" : {
                "building" : "2268",
                "coord" : [
                        -73.9564939,
                        40.650368
                ],
                "street" : "Church Avenue",
                "zipcode" : "11226"
        },
        "borough" : "Brooklyn",
        "cuisine" : "Vegetarian",
        "grades" : [
                {
                        "date" : ISODate("2014-07-28T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                },
                {
                        "date" : ISODate("2014-02-25T00:00:00Z"),
                        "grade" : "A",
                        "score" : 5
                },
                {
                        "date" : ISODate("2013-06-01T00:00:00Z"),
                        "grade" : "A",
                        "score" : 7
                },
                {
                        "date" : ISODate("2012-10-16T00:00:00Z"),
                        "grade" : "C",
```

4. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

db.rest.find({name: /^Wil/}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 }).pretty();

```
> db.rest.find( {name: /^Wil/}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae8293"),
        "borough" : "Brooklyn",
        "cuisine" : "Delicatessen",
        "name" : "Wilken'S Fine Food",
        "restaurant_id" : "40356483"
}
{
        "_id" : ObjectId("60faf62eade426b733ae829e"),
        "borough" : "Bronx",
        "cuisine" : "American ",
        "name" : "Wild Asia",
        "restaurant_id" : "40357217"
}
{
        "_id" : ObjectId("60faf62eade426b733ae909c"),
        "borough" : "Bronx",
        "cuisine" : "Pizza",
        "name" : "Wilbel Pizza",
        "restaurant_id" : "40871979"
}
{
        "_id" : ObjectId("60fba3f5f37f79b63ab711af"),
        "borough" : "Brooklyn",
        "cuisine" : "Delicatessen",
        "name" : "Wilken'S Fine Food",
        "restaurant_id" : "40356483"
}
{
        "_id" : ObjectId("60fba3f5f37f79b63ab711b4"),
        "borough" : "Bronx",
        "cuisine" : "American ",
        "name" : "Wild Asia",
        "restaurant_id" : "40357217"
```

5.Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

```
db.rest.find( {name: /ces$/}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
```

```
> db.rest.find( {name: /ces$/}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae871f"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Pieces",
        "restaurant_id" : "40399910"
}
{
        "_id" : ObjectId("60faf62eade426b733ae87de"),
        "borough" : "Queens",
        "cuisine" : "American ",
        "name" : "S.M.R Restaurant Services",
        "restaurant_id" : "40403857"
}
{
        "_id" : ObjectId("60faf62eade426b733ae87e3"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Good Shepherd Services",
        "restaurant_id" : "40403989"
}
{
        "_id" : ObjectId("60faf62eade426b733ae8c97"),
        "borough" : "Queens",
        "cuisine" : "Ice Cream, Gelato, Yogurt, Ices",
        "name" : "The Ice Box-Ralph'S Famous Italian Ices",
        "restaurant_id" : "40690899"
}
{
        "_id" : ObjectId("60faf62eade426b733ae8e9e"),
        "borough" : "Brooklyn",
        "cuisine" : "Jewish/Kosher",
        "name" : "Alices",
        "restaurant_id" : "40782042"
```

(B) Write a MongoDB query to create a backup of existing database and also create a backup of existing database.

>mongodump

>drop database in mongo cmd

>mongorestore

10. (A) . Write a jQuery to animate multiple CSS properties.

Refer P8 – 11Q

(B). Write a jQuery to Change text contents of the elements on button click.

Refer P8 -1Q

11. Write a MongoDB query to create Replica and backup of existing database.

12. (A) Write a jQuery to add and remove CSS classes from the HTML elements.

Refer P8 -23Q

(B) Write a jQuery to set the duration in slide toggle effect.

Refer P8 -24Q

13. (A) Import Restaurant.json into MongoDb and perform the following queries:

1. Write a MongoDB query to display all the documents in the collection restaurants.

2. Write a MongoDB query to display the fields, restaurant_id, name,borough and cuisine for all the documents in the collection restaurant.

3.Write a MongoDB query to display the fields restaurant_id, name,borough and cuisine, but exclude the field _id for all the documents in the collection restaurant.

4. Write a MongoDB query to display the fields restaurant_id, name, borough and zip code, but exclude the field _id for all the documents in the collection restaurant

5 Write a MongoDB query to display all the restaurant which is in the borough Bronx

Refer - 1(A)

(B) Connect Python with MongoDB and also insert and delete documents.

Refer P5

14. (A) Write a jQuery to insert multiple HTML elements at the beginning and end of the elements.

Refer P8 – 18Q

(B). Write a jQuery to create your own Customized event.

15. Connect Python with MongoDB and also insert, retrieve, update and delete documents.

Refer P5

16. Create a JSON file and persist it in any database.

Refer P10

17. (A) Import Restaurant.json into MongoDb and perform the following queries:

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name

```
> db.rest.find({"name": /.*Reg.*/}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
```

```
> db.rest.find({"name": /.*Reg.*/}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae8294"),
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "Regina Caterers",
        "restaurant_id" : "40356649"
}
{
        "_id" : ObjectId("60faf62eade426b733ae838f"),
        "borough" : "Manhattan",
        "cuisine" : "Café/Coffee/Tea",
        "name" : "Caffe Reggio",
        "restaurant_id" : "40369418"
}
{
        "_id" : ObjectId("60faf62eade426b733ae84a0"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Regency Hotel",
        "restaurant_id" : "40382679"
}
{
        "_id" : ObjectId("60faf62eade426b733ae87bd"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Regency Whist Club",
        "restaurant_id" : "40402377"
}
{
        "_id" : ObjectId("60faf62eade426b733ae88a0"),
        "borough" : "Queens",
        "cuisine" : "American ",
```

2. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.

```
> db.rest.find({"borough": "Bronx" ,$or :[{ "cuisine":"American "},{"cuisine" :"Chinese" }]}) .pretty();
```

```
> db.rest.find({"borough": "Bronx" ,$or :[{ "cuisine":"American "},{"cuisine" :"Chinese" }]}) .pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae829e"),
        "address" : {
                "building" : "2300",
                "coord" : [
                        -73.8786113,
                        40.8502883
                ],
                "street" : "Southern Boulevard",
                "zipcode" : "10460"
        },
        "borough" : "Bronx",
        "cuisine" : "American ",
        "grades" : [
                {
                        "date" : ISODate("2014-05-28T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
                {
                        "date" : ISODate("2013-06-19T00:00:00Z"),
                        "grade" : "A",
                        "score" : 4
                },
                {
                        "date" : ISODate("2012-06-15T00:00:00Z"),
                        "grade" : "A",
                        "score" : 3
                }
        ],
        "name" : "Wild Asia",
        "restaurant_id" : "40357217"
```

3. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn

```
> db.rest.find( {"borough" :{$in :["Staten
Island","Queens","Bronx","Brooklyn"]}},{
"restaurant_id" : 1, "name":1,"borough":1,
"cuisine" :1 }).pretty();
```

```
> db.rest.find( {"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}},{ "restaurant_id" : 1, "name":1,"borough":1, "cuisine
tty();
{
        "_id" : ObjectId("60faf62eade426b733ae828d"),
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "Riviera Caterer",
        "restaurant_id" : "40356018"
}
{
        "_id" : ObjectId("60faf62eade426b733ae828e"),
        "borough" : "Bronx",
        "cuisine" : "Bakery",
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
}
{
        "_id" : ObjectId("60faf62eade426b733ae828f"),
        "borough" : "Brooklyn",
        "cuisine" : "Hamburgers",
        "name" : "Wendy'S",
        "restaurant_id" : "30112340"
}
{
        "_id" : ObjectId("60faf62eade426b733ae8290"),
        "borough" : "Staten Island",
        "cuisine" : "Jewish/Kosher",
        "name" : "Kosher Island",
        "restaurant_id" : "40356442"
}
{
        "_id" : ObjectId("60faf62eade426b733ae8291"),
```

4. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which are not belonging to the borough Staten Island or Queens or Bronxor Brooklyn.

```
> db.rest.find( {"borough" :{$nin :["Staten Island","Queens","Bronx","Brooklyn"]}},{ "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
```

```
> db.rest.find( {"borough" :{$nin :["Staten Island","Queens","Bronx","Brooklyn"]}},{ "restaurant_id" : 1, "name":1,"borough":1, "cuisin
retty();
{
        "_id" : ObjectId("60faf62eade426b733ae828c"),
        "borough" : "Manhattan",
        "cuisine" : "Irish",
        "name" : "Dj Reynolds Pub And Restaurant",
        "restaurant_id" : "30191841"
}
{
        "_id" : ObjectId("60faf62eade426b733ae8297"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "1 East 66Th Street Kitchen",
        "restaurant_id" : "40359480"
}
{
        "_id" : ObjectId("60faf62eade426b733ae829d"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Glorious Food",
        "restaurant_id" : "40361521"
}
{
        "_id" : ObjectId("60faf62eade426b733ae82a1"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "P & S Deli Grocery",
        "restaurant_id" : "40362264"
}
{
        "_id" : ObjectId("60faf62eade426b733ae82a2"),
```

5. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.

```
db.rest.find( {"grades.score" : { $not: {$gt : 10} }},{
"restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1
} ).pretty();
```

```
> db.rest.find( {"grades.score" : { $not: {$gt : 10} }},{ "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae8297"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "1 East 66Th Street Kitchen",
        "restaurant_id" : "40359480"
}
{
        "_id" : ObjectId("60faf62eade426b733ae8298"),
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "C & C Catering Service",
        "restaurant_id" : "40357437"
}
{
        "_id" : ObjectId("60faf62eade426b733ae829c"),
        "borough" : "Brooklyn",
        "cuisine" : "Delicatessen",
        "name" : "Nordic Delicacies",
        "restaurant_id" : "40361390"
}
{
        "_id" : ObjectId("60faf62eade426b733ae82a5"),
        "borough" : "Brooklyn",
        "cuisine" : "Hamburgers",
        "name" : "White Castle",
        "restaurant_id" : "40362344"
}
{
        "_id" : ObjectId("60faf62eade426b733ae82b8"),
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "Sonny'S Heros",
```

(B) Connect Python with MongoDB and also insert and retrieve documents.

Refer P5

18. A. Write a jQuery Create animation effect.

Refer P8-10Q

B. Write a jQuery to perform Method chaining.

Refer P8 -12Q

19. Write a MongoDB query to create Replica of existing database. Also create the backup of existing database. Number of primary server must be one and secondary server must be 3.

20. Create a JSON file and persist it in any database.

Refer P10

21. (A) Import Restaurant.json into MongoDb and perform the following queries:

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish

except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'

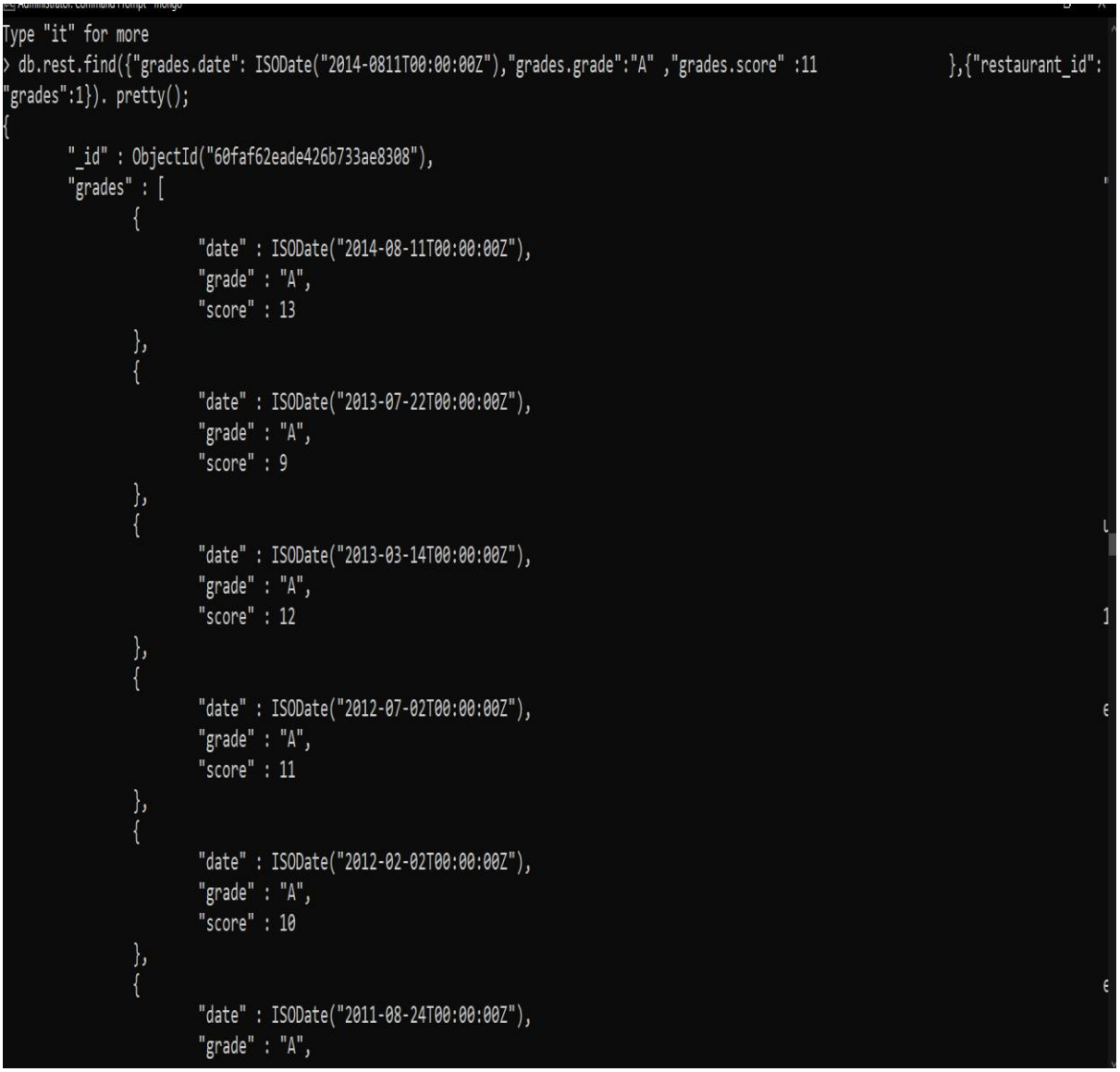
```
db.rest.find( {$or: [ {name: /^Wil/},{"$and":[
{"cuisine" : {$ne :"American "}},  {"cuisine" :{$ne
:"Chinees"}} ]} ]} ,{"restaurant_id" :
1,"name":1,"borough":1,"cuisine" :1} ).pretty();
```

```
> db.rest.find( {"grades.score" : { $not: {$gt : 10} }},{ "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae8297"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "1 East 66Th Street Kitchen",
        "restaurant_id" : "40359480"
}
{
        "_id" : ObjectId("60faf62eade426b733ae8298"),
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "C & C Catering Service",
        "restaurant_id" : "40357437"
}
{
        "_id" : ObjectId("60faf62eade426b733ae829c"),
        "borough" : "Brooklyn",
        "cuisine" : "Delicatessen",
        "name" : "Nordic Delicacies",
        "restaurant_id" : "40361390"
}
{
        "_id" : ObjectId("60faf62eade426b733ae82a5"),
        "borough" : "Brooklyn",
        "cuisine" : "Hamburgers",
        "name" : "White Castle",
        "restaurant_id" : "40362344"
}
{
        "_id" : ObjectId("60faf62eade426b733ae82b8"),
        "borough" : "Brooklyn",
        "cuisine" : "American ",
```

2. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and

scored 11 on an ISODate "2014- 08-11T00:00:00Z" among many of survey dates.

```
db.rest.find({"grades.date": ISODate("2014-0811T00:00:00Z"),"grades.grade":"A" ,"grades.score" :11 },{"restaurant_id":1,"name":1,"grades":1}). pretty();
```



```
Type "it" for more
> db.rest.find({"grades.date": ISODate("2014-0811T00:00:00Z"),"grades.grade":"A" ,"grades.score" :11 },{"restaurant_id":
"grades":1}). pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae8308"),
        "grades" : [
                {
                        "date" : ISODate("2014-08-11T00:00:00Z"),
                        "grade" : "A",
                        "score" : 13
                },
                {
                        "date" : ISODate("2013-07-22T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
                {
                        "date" : ISODate("2013-03-14T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                },
                {
                        "date" : ISODate("2012-07-02T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
                {
                        "date" : ISODate("2012-02-02T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                },
                {
                        "date" : ISODate("2011-08-24T00:00:00Z"),
                        "grade" : "A",
```

3. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades

array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z"

```
db.rest.find({"grades.1.date": ISODate("2014-0811T00:00:00Z"), "grades.1.grade":"A" ,"grades.1.score" : 9},{"restaurant_id" : 1,"name":1,"grades":1}).pretty();
```

```
> db.rest.find({"grades.1.date": ISODate("2014-0811T00:00:00Z"), "grades.1.grade":"A" ,"grades.1.score" : 9},{"restaurant_id"
: 1,"name":1,"grades":1}).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae88b8"),
        "grades" : [
                {
                        "date" : ISODate("2015-01-12T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                },
                {
                        "date" : ISODate("2014-08-11T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
                {
                        "date" : ISODate("2014-01-14T00:00:00Z"),
                        "grade" : "A",
                        "score" : 13
                },
                {
                        "date" : ISODate("2013-02-07T00:00:00Z"),
                        "grade" : "A",
                        "score" : 10
                },
                {
                        "date" : ISODate("2012-04-30T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                }
```

4. Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where

2nd element of coord array contains a value which is more than 42 and upto 52.

```
db.rest.find({"address.coord.1": {$gt : 42, $lte : 52}},{"restaurant_id" : 1,"name":1,"address":1,"coord":1}
).pretty();
```

```
> db.rest.find({"address.coord.1": {$gt : 42, $lte : 52}},{"restaurant_id" : 1,"name":1,"address":1,"coord":1}).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae852c"),
        "address" : {
                "building" : "47",
                "coord" : [
                        -78.877224,
                        42.89546199999999
                ],
                "street" : "Broadway @ Trinity Pl",
                "zipcode" : "10006"
        },
```

5. Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.

```
db.rest.find().sort({"name":1}).pretty();
```

```
> db.rest.find().sort({"name":1}).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae8f1e"),
        "address" : {
                "building" : "129",
                "coord" : [
                        -73.962943,
                        40.685007
                ],
                "street" : "Gates Avenue",
                "zipcode" : "11238"
        },
        "borough" : "Brooklyn",
        "cuisine" : "Italian",
        "grades" : [
                {
                        "date" : ISODate("2014-03-06T00:00:00Z"),
                        "grade" : "A",
                        "score" : 5
                },
                {
                        "date" : ISODate("2013-08-29T00:00:00Z"),
                        "grade" : "A",
                        "score" : 2
                },
                {
                        "date" : ISODate("2013-03-08T00:00:00Z"),
                        "grade" : "A",
                        "score" : 7
                },
                {
                        "date" : ISODate("2012-06-27T00:00:00Z"),
                        "grade" : "A",
                        "score" : 7
                },
                {
```

(B) Connect PHP with MongoDB and also insert and delete documents.

Refer P7

22. Create a JSON file and persist it in any database.

Refer P10

23. Create a Collection Employee with the following Fields

(Eid,Ename,Sal,City,hobbies) where hobbies is an array perform the Following Queries based on the collection.

A. Write a MongoDB query to use sum, avg, min and max expression.

B. Write a MongoDB query to use push and addToSet expression.

C. Write a MongoDB query to use first and last expression.

D. Perform backup and restore on the above collection.

24. (A) Create a JSON file and parse it.

Refer P10

(B) Write a jQuery to create fade-in and fade-out effect.

Refer P8 – 6Q

25. (A) Import Restaurant.json into MongoDb and perform the following queries:

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'

2. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014- 08-11T00:00:00Z" among many of survey dates.

3. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z"

4. Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52.

5. Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.

Refer 21-A

(B) Connect Java with MongoDB and also insert and delete documents.

Refer P6

26. (A) Write a jQuery to remove the parent element of an HTML element from the page.

Refer P8 – 21Q

(B). Write a jQuery to add and remove CSS classes from the HTML elements.

Refer P8 -23Q

27. Connect PHP with MongoDB and also insert, retrieve, update and delete documents.

Refer P7

28. (A) Write a jQuery to get and set text contents of the elements.

Refer P8 -14Q

(B). Write a jQuery to set the duration in slide toggle effect.

Refer P8 -24Q

29. (A) Import Restaurant.json into MongoDb and perform the following queries:

1. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns

db.rest.find().sort({"name":-1}).pretty();

```
> db.rest.find().sort( {"name":-1}).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae834b"),
        "address" : {
                "building" : "6946",
                "coord" : [
                        -73.8811834,
                        40.7017759
                ],
                "street" : "Myrtle Avenue",
                "zipcode" : "11385"
        },
        "borough" : "Queens",
        "cuisine" : "German",
        "grades" : [
                {
                        "date" : ISODate("2014-09-24T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
                {
                        "date" : ISODate("2014-04-17T00:00:00Z"),
                        "grade" : "A",
                        "score" : 7
                },
                {
                        "date" : ISODate("2013-03-12T00:00:00Z"),
                        "grade" : "A",
                        "score" : 13
                },
                {
                        "date" : ISODate("2012-10-02T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
                {
```

2. Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order

db.rest.find().sort({"cuisine":1,"borough" : -1,}).pretty();

```
> db.rest.find().sort({"cuisine":1,"borough" : -1,}).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae8978"),
        "address" : {
                "building" : "1345",
                "coord" : [
                        -73.959249,
                        40.768076
                ],
                "street" : "2 Avenue",
                "zipcode" : "10021"
        },
        "borough" : "Manhattan",
        "cuisine" : "Afghan",
        "grades" : [
                {
                        "date" : ISODate("2014-10-07T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
                {
                        "date" : ISODate("2013-10-23T00:00:00Z"),
                        "grade" : "A",
                        "score" : 8
                },
                {
                        "date" : ISODate("2012-10-26T00:00:00Z"),
                        "grade" : "A",
                        "score" : 13
                },
                {
                        "date" : ISODate("2012-04-26T00:00:00Z"),
                        "grade" : "A",
                        "score" : 7
                },
                {
```

3. Write a MongoDB query to know whether all the addresses contains the street or not.

```
db.rest.find({"address.street" :{ $exists :true }
}).pretty();
```

```
> db.rest.find({"address.street" :{ $exists :true }}).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae828c"),
        "address" : {
                "building" : "351",
                "coord" : [
                        -73.98513559999999,
                        40.7676919
                ],
                "street" : "West   57 Street",
                "zipcode" : "10019"
        },
        "borough" : "Manhattan",
        "cuisine" : "Irish",
        "grades" : [
                {
                        "date" : ISODate("2014-09-06T00:00:00Z"),
                        "grade" : "A",
                        "score" : 2
                },
                {
                        "date" : ISODate("2013-07-22T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
                {
                        "date" : ISODate("2012-07-31T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                },
                {
                        "date" : ISODate("2011-12-29T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                }
        ],
        "name" : "Dj Reynolds Pub And Restaurant",
        "restaurant_id" : "30191841"
```

4. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is double

   db.rest.find({"address.coord":{$type:1}}).pretty();

```
db.rest.find({"address.coord" :{$type :1}}).pretty();

        "_id" : ObjectId("60faf62eade426b733ae828c"),
        "address" : {
                "building" : "351",
                "coord" : [
                        -73.98513559999999,
                        40.7676919
                ],
                "street" : "West   57 Street",
                "zipcode" : "10019"
        },
        "borough" : "Manhattan",
        "cuisine" : "Irish",
        "grades" : [
                {
                        "date" : ISODate("2014-09-06T00:00:00Z"),
                        "grade" : "A",
                        "score" : 2
                },
                {
                        "date" : ISODate("2013-07-22T00:00:00Z"),
                        "grade" : "A",
                        "score" : 11
                },
                {
                        "date" : ISODate("2012-07-31T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                },
                {
                        "date" : ISODate("2011-12-29T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                }
        ],
        "name" : "Dj Reynolds Pub And Restaurant",
        "restaurant_id" : "30191841"
```

5. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.

> db.rest.find({“grades.score” :{$mod : [7,0]}}, {“restaurant_id” : 1,” name” :1,” grades” :1}).pretty();

```
Type "it" for more
> db.rest.find( {"grades.score" :{$mod : [7,0]}}, {"restaurant_id" : 1,"name":1,"grades":1}).pretty();
{
        "_id" : ObjectId("60faf62eade426b733ae828d"),
        "grades" : [
                {
                        "date" : ISODate("2014-06-10T00:00:00Z"),
                        "grade" : "A",
                        "score" : 5
                },
                {
                        "date" : ISODate("2013-06-05T00:00:00Z"),
                        "grade" : "A",
                        "score" : 7
                },
                {
                        "date" : ISODate("2012-04-13T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                },
                {
                        "date" : ISODate("2011-10-12T00:00:00Z"),
                        "grade" : "A",
                        "score" : 12
                }
        ],
        "name" : "Riviera Caterer",
        "restaurant_id" : "40356018"
}
{
        "_id" : ObjectId("60faf62eade426b733ae828e"),
        "grades" : [
                {
                        "date" : ISODate("2014-03-03T00:00:00Z"),
                        "grade" : "A",
                        "score" : 2
                },
```

(B) Write a MongoDB query to create Replica of existing database.

30.(A) Write a jQuery to animate multiple CSS properties.

Refer P8 -11Q

(B). Create a JSON file and parse it.

Refer P10

31. Connect Java with MongoDB and also insert, retrieve, update and delete documents.

Refer P6

32. (A) Write a Jquery to get and set text contents of the elements.

Refer P8 – 14Q

(B) Write a Jquery to insert HTML elements at the beginning and end of the elements

Refer P8 – 17Q