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

> db.restaurants.find().pretty()

{

"\_id" : ObjectId("625ee1c369570e10534a78a6"),

"address" : {

"building" : "8825",

"coord" : [

-73.8803827,

40.7643124

],

"street" : "Astoria Boulevard",

"zipcode" : "11369"

},

"borough" : "Queens",

"cuisine" : "American",

"grades" : [

{

"date" : {

"$date" : "2014-11-15T00:00:00.000Z"

},

"grade" : "Z",

"score" : 38

},

{

"date" : {

"$date" : "2014-05-02T00:00:00.000Z"

},

"grade" : "A",

"score" : 10

},

{

"date" : {

"$date" : "2013-03-02T00:00:00.000Z"

},

"grade" : "A",

"score" : 7

},

{

"date" : {

"$date" : "2012-02-10T00:00:00.000Z"

},

"grade" : "A",

"score" : 13

}

],

"name" : "Brunos On The Boulevard",

"restaurant\_id" : "40356151"

}

**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}).pretty()

{

"\_id" : ObjectId("625fc2b1066d642bc3fa96a3"),

"borough" : "Queens",

"cuisine" : "American",

"name" : "Brunos On The Boulevard",

"restaurant\_id" : "40356151"

}

**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({},{"\_id":0,"restaurant\_id":1,"name":1, "borough":1, "cuisine":1}).pretty()

{

"borough" : "Bronx",

"cuisine" : "Bakery",

"name" : "Morris Park Bake Shop",

"restaurant\_id" : "30075445"

}

{

"borough" : "Brooklyn",

"cuisine" : "Hamburgers",

"name" : "Wendy'S",

"restaurant\_id" : "30112340"

}

{

"borough" : "Manhattan",

"cuisine" : "Irish",

"name" : "Dj Reynolds Pub And Restaurant",

"restaurant\_id" : "30191841"

}

{

"borough" : "Brooklyn",

"cuisine" : "American ",

"name" : "Riviera Caterer",

"restaurant\_id" : "40356018"

}

"borough" : "Staten Island",

"cuisine" : "Jewish/Kosher",

"name" : "Kosher Island",

"restaurant\_id" : "40356442"

}

………………

**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({},{"\_id":0,"restaurant\_id":1,"name":1, "borough":1, "cuisine":1,"address":{"zipcode":1}}).pretty()

{

"address" : {

"zipcode" : "10462"

},

"borough" : "Bronx",

"cuisine" : "Bakery",

"name" : "Morris Park Bake Shop",

"restaurant\_id" : "30075445"

}

{

"address" : {

"zipcode" : "11225"

},

"borough" : "Brooklyn",

"cuisine" : "Hamburgers",

"name" : "Wendy'S",

"restaurant\_id" : "30112340"

}

{

"address" : {

"zipcode" : "10019"

},

"borough" : "Manhattan",

"cuisine" : "Irish",

"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({},{"\_id":0,"restaurant\_id":1,name:1,"borough":"Bronx"}).pretty()

{

"name" : "Morris Park Bake Shop",

"restaurant\_id" : "30075445",

"borough" : "Bronx"

}

{ "name" : "Wendy'S", "restaurant\_id" : "30112340", "borough" : "Bronx" }

{

"name" : "Dj Reynolds Pub And Restaurant",

"restaurant\_id" : "30191841",

"borough" : "Bronx"

}

{

"name" : "Riviera Caterer",

"restaurant\_id" : "40356018",

"borough" : "Bronx"

}…………………

**6. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.**

> db.rest.find({},{name:1,"borough":"Bronx"}).pretty().limit(5)

{

"\_id" : ObjectId("625fccc7612b3c0aafdde6ba"),

"name" : "Morris Park Bake Shop",

"borough" : "Bronx"

}

{

"\_id" : ObjectId("625fccc7612b3c0aafdde6bb"),

"name" : "Wendy'S",

"borough" : "Bronx"

}

{

"\_id" : ObjectId("625fccc7612b3c0aafdde6bc"),

"name" : "Dj Reynolds Pub And Restaurant",

"borough" : "Bronx"

}

{

"\_id" : ObjectId("625fccc7612b3c0aafdde6bd"),

"name" : "Riviera Caterer",

"borough" : "Bronx"

}

{

"\_id" : ObjectId("625fccc7612b3c0aafdde6be"),

"name" : "Tov Kosher Kitchen",

"borough" : "Bronx"

}

**7.Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.**

> db.rest.find({},{name:1,"borough":"Bronx"}).pretty().skip(5).limit(5)

{

"\_id" : ObjectId("625fccc7612b3c0aafdde6bf"),

"name" : "Brunos On The Boulevard",

"borough" : "Bronx"

}

{

"\_id" : ObjectId("625fccc7612b3c0aafdde6c0"),

"name" : "Kosher Island",

"borough" : "Bronx"

}

{

"\_id" : ObjectId("625fccc7612b3c0aafdde6c1"),

"name" : "Wilken'S Fine Food",

"borough" : "Bronx"

}

{

"\_id" : ObjectId("625fccc7612b3c0aafdde6c2"),

"name" : "Regina Caterers",

"borough" : "Bronx"

}

{

"\_id" : ObjectId("625fccc7612b3c0aafdde6c3"),

"name" : "Taste The Tropics Ice Cream",

"borough" : "Bronx"

}

**8. Write a MongoDB query to find the restaurants who achieved a score more than 90.**

> db.rest.find({"grades.score":{$gt:90}}).pretty()

{

"\_id" : ObjectId("625fccc7612b3c0aafdde818"),

"address" : {

"building" : "65",

"coord" : [

-73.9782725,

40.7624022

],

"street" : "West 54 Street",

"zipcode" : "10019"

},

"borough" : "Manhattan",

"cuisine" : "American ",

"grades" : [

{

"date" : ISODate("2014-08-22T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2014-03-28T00:00:00Z"),

"grade" : "C",

"score" : 131

},

{

"date" : ISODate("2013-09-25T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2013-04-08T00:00:00Z"),

"grade" : "B",

"score" : 25

},

{

"date" : ISODate("2012-10-15T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2011-10-19T00:00:00Z"),

"grade" : "A",

"score" : 13

}

],

"name" : "Murals On 54/Randolphs'S",

"restaurant\_id" : "40372466"

}………………………

**9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.**

> db.rest.find({"grades.score":{$gt:80,$lt:100}}).pretty()

{

"\_id" : ObjectId("625fccc7612b3c0aafdde818"),

"address" : {

"building" : "65",

"coord" : [

-73.9782725,

40.7624022

],

"street" : "West 54 Street",

"zipcode" : "10019"

},

"borough" : "Manhattan",

"cuisine" : "American ",

"grades" : [

{

"date" : ISODate("2014-08-22T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2014-03-28T00:00:00Z"),

"grade" : "C",

"score" : 131

},

{

"date" : ISODate("2013-09-25T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2013-04-08T00:00:00Z"),

"grade" : "B",

"score" : 25

},

{

"date" : ISODate("2012-10-15T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2011-10-19T00:00:00Z"),

"grade" : "A",

"score" : 13

}

],

"name" : "Murals On 54/Randolphs'S",

"restaurant\_id" : "40372466"

}

**10. 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()

{

"\_id" : ObjectId("625fccc7612b3c0aafdded02"),

"address" : {

"building" : "3707",

"coord" : [

-101.8945214,

33.5197474

],

"street" : "82 Street",

"zipcode" : "11372"

},

"borough" : "Queens",

"cuisine" : "American ",

"grades" : [

{

"date" : ISODate("2014-06-04T00:00:00Z"),

"grade" : "A",

"score" : 12

},

{

"date" : ISODate("2013-11-07T00:00:00Z"),

"grade" : "B",

"score" : 19

},

{

"date" : ISODate("2013-05-17T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2012-08-29T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2012-04-03T00:00:00Z"),

"grade" : "A",

"score" : 12

},

{

"date" : ISODate("2011-11-16T00:00:00Z"),

"grade" : "A",

"score" : 7

}

],

"name" : "Burger King",

"restaurant\_id" : "40534067"

}

**11. 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().limit(5)

{

"\_id" : ObjectId("625fccc7612b3c0aafdde8b9"),

"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

},

{

"date" : ISODate("2012-10-01T00:00:00Z"),

"grade" : "A",

"score" : 9

},

{

"date" : ISODate("2012-04-06T00:00:00Z"),

"grade" : "C",

"score" : 92

},

{

"date" : ISODate("2011-11-03T00:00:00Z"),

"grade" : "C",

"score" : 41

}

],

"name" : "Gandhi",

"restaurant\_id" : "40381295"

}

**12. 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.**

**Note : Do this query without using $and operator.**

> db.rest.find( {"cuisine": {$ne :"American"},"grades.score" :{$gt :70}, "address.coord" : {$lt :-65.754168} }).pretty().limit(2)

{

"\_id" : ObjectId("625fccc7612b3c0aafdde818"),

"address" : {

"building" : "65",

"coord" : [

-73.9782725,

40.7624022

],

"street" : "West 54 Street",

"zipcode" : "10019"

},

"borough" : "Manhattan",

"cuisine" : "American ",

"grades" : [

{

"date" : ISODate("2014-08-22T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2014-03-28T00:00:00Z"),

"grade" : "C",

"score" : 131

},

{

"date" : ISODate("2013-09-25T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2013-04-08T00:00:00Z"),

"grade" : "B",

"score" : 25

},

{

"date" : ISODate("2012-10-15T00:00:00Z"),

"grade" : "A",

"score" : 11

},

{

"date" : ISODate("2011-10-19T00:00:00Z"),

"grade" : "A",

"score" : 13

}

],

"name" : "Murals On 54/Randolphs'S",

"restaurant\_id" : "40372466"

}

{

"\_id" : ObjectId("625fccc7612b3c0aafdde8b9"),

"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

},

{

"date" : ISODate("2012-10-01T00:00:00Z"),

"grade" : "A",

"score" : 9

},

{

"date" : ISODate("2012-04-06T00:00:00Z"),

"grade" : "C",

"score" : 92

},

{

"date" : ISODate("2011-11-03T00:00:00Z"),

"grade" : "C",

"score" : 41

}

],

"name" : "Gandhi",

"restaurant\_id" : "40381295"

}

**13. 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"}).pretty().sort({"cuisine":-1}).limit(3)

{

"\_id" : ObjectId("625fccc8612b3c0aafddf2b4"),

"address" : {

"building" : "191",

"coord" : [

-73.95785099999999,

40.7174769

],

"street" : "Bedford Avenue",

"zipcode" : "11211"

},

"borough" : "Brooklyn",

"cuisine" : "Vegetarian",

"grades" : [

{

"date" : ISODate("2014-02-05T00:00:00Z"),

"grade" : "A",

"score" : 9

},

{

"date" : ISODate("2013-02-01T00:00:00Z"),

"grade" : "A",

"score" : 13

},

{

"date" : ISODate("2012-02-01T00:00:00Z"),

"grade" : "A",

"score" : 5

}

],

"name" : "Bliss Bakery & Cafe",

"restaurant\_id" : "40763388"

}

{

"\_id" : ObjectId("625fccc8612b3c0aafddf241"),

"address" : {

"building" : "752",

"coord" : [

-73.950456,

40.672989

],

"street" : "Nostrand Avenue",

"zipcode" : "11216"

},

"borough" : "Brooklyn",

"cuisine" : "Vegetarian",

"grades" : [

{

"date" : ISODate("2014-12-03T00:00:00Z"),

"grade" : "A",

"score" : 9

},

{

"date" : ISODate("2014-06-04T00:00:00Z"),

"grade" : "A",

"score" : 8

},

{

"date" : ISODate("2013-12-03T00:00:00Z"),

"grade" : "A",

"score" : 9

},

{

"date" : ISODate("2013-05-31T00:00:00Z"),

"grade" : "A",

"score" : 10

},

{

"date" : ISODate("2012-10-15T00:00:00Z"),

"grade" : "A",

"score" : 7

},

{

"date" : ISODate("2012-05-09T00:00:00Z"),

"grade" : "C",

"score" : 33

}

],

"name" : "Original Vegetarian Restaurant",

"restaurant\_id" : "40746628"

}………………..