**MongoDB Exercise2 - Solution**

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

db.restaurants.find().pretty()

1. Write a 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.restaurants.find(

{},{ \_id: 0, restaurant\_id: 1, name: 1, borough: 1, cuisine: 1 })

1. Write a query to display the first 5 restaurant which is in the borough London

db.restaurants.find({ borough: "London" }).limit(5)

1. Write a query to find the restaurants who achieved a score more than 90.

db.restaurants.find({ borough: "London" }).limit(5).pretty()

1. Write a query to find the restaurants who achieved a score more than 10.

db.restaurants.find({ "grades.score": { $gt: 10 } })

1. Write a query to find the restaurants that do not prepare any 'American' cuisine and their grade score is more than 5.

db.restaurants.find({

cuisine: { $ne: "American" },

"grades.score": { $gt: 5 }

})

1. Write a query to find the restaurants that prepare 'Indian' cuisine and their grade score more than 5.

db.restaurants.find({

cuisine: "Indian",

"grades.score": { $gt: 5 }

})

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

db.restaurants.find({}).sort({ name: -1 })

1. Write a query to find the count of restaurants in each borough.

db.restaurants.aggregate([{

$group: {

\_id: "$borough",

count: {

$sum: 1}}}])

1. Write a query to find the lowest score for each borough.

db.restaurants.aggregate([

{ $unwind: "$grades" },

{ $group: {

\_id: { borough: "$borough" },

lowestScore: { $min: "$grades.score" }

}

}])

1. Write a query to find the name and address of the restaurants that received a grade of 'A'

db.restaurants.find(

{ "grades.grade": "A" },

{ name: 1, address: 1, \_id: 0 }

)

1. Write a query to find the name and address of the restaurants that have at least one 'A' grade and one 'B' grade.

db.restaurants.find({

$and: [

{ "grades.grade": "A" },

{ "grades.grade": "B" }

]

},{ name: 1, address: 1, \_id: 0 })

1. Write a  query to find the name and address of the restaurants that have at least one 'A' grade and no 'B' grades.

db.restaurants.find({

$and: [

{ "grades.grade": "A" },

{ "grades.grade": { $not: { $eq: "B" } } }

]

},{ name: 1, address: 1, \_id: 0 })

1. Write a query to find the restaurants that have a grade with a score of 2 or a grade with a score of 6 and are located in the borough of London or Bristol.

db.restaurants.find({

$or: [

{ grades: { $elemMatch: { score: 2 } } },

{ grades: { $elemMatch: { score: 6 } } }

],

$or: [

{ borough: "London" },

{ borough: "Bristol" }

]})

1. Write a query to find the restaurants that have at least one grade with a score of less than 5.

db.restaurants.find({

"grades.score": { $lt: 5 }

})