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

collection restaurants.

db.createCollection("restaurants");

2.Write a MongoDB query to arrange the name of the restaurants in

descending along with all the columns.

db.restaurants.insertMany([

{ name: "Meghna Foods", town: "Jayanagar", cuisine: "Indian", score: 8, address: { zipcode: "10001", street: "Jayanagar“

} },

{ name: "Empire", town: "MG Road", cuisine: "Indian", score: 7, address: { zipcode: "10100", street: "MG Road" } },

{ name: "Chinese WOK", town: "Indiranagar", cuisine: "Chinese", score: 12, address: { zipcode: "20000", street: "Indiranagar" } },

{ name: "Kyotos", town: "Majestic", cuisine: "Japanese", score: 9, address: { zipcode: "10300", street: "Majestic" } },

{ name: "WOW Momos", town: "Malleshwaram", cuisine: "Indian", score: 5, address: { zipcode: "10400", street: "Malleshwaram" }

} ])

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



3.Write a MongoDB query to find the restaurant Id, name, town and

cuisine for those restaurants which achieved a score which is not

more than 10.

db.restaurants.find({ "score": { $lte: 10 } }, { \_id: 1, name: 1, town: 1, cuisine: 1 })

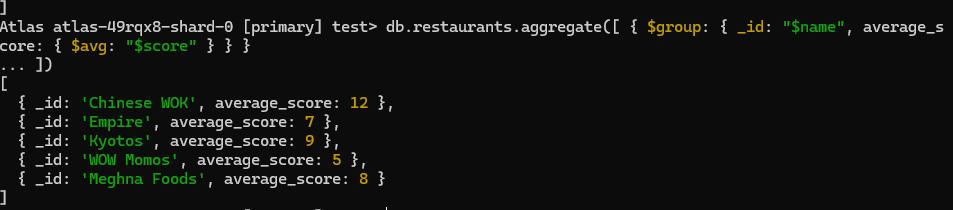


4.Write a MongoDB query to find the average score for each

restaurant.

db.restaurants.aggregate([ { $group: { \_id: "$name", average\_score: { $avg: "$score" } } }

])



5.Write a MongoDB query to find the name and address of the

restaurants that have a zipcode that starts with '10'.

db.restaurants.find({ "address.zipcode": /^10/ }, { name: 1, "address.street": 1, \_id: 0 })

