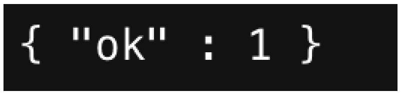
NoSQL - 3 - Restaurant Database

Question:
(Week 10)

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

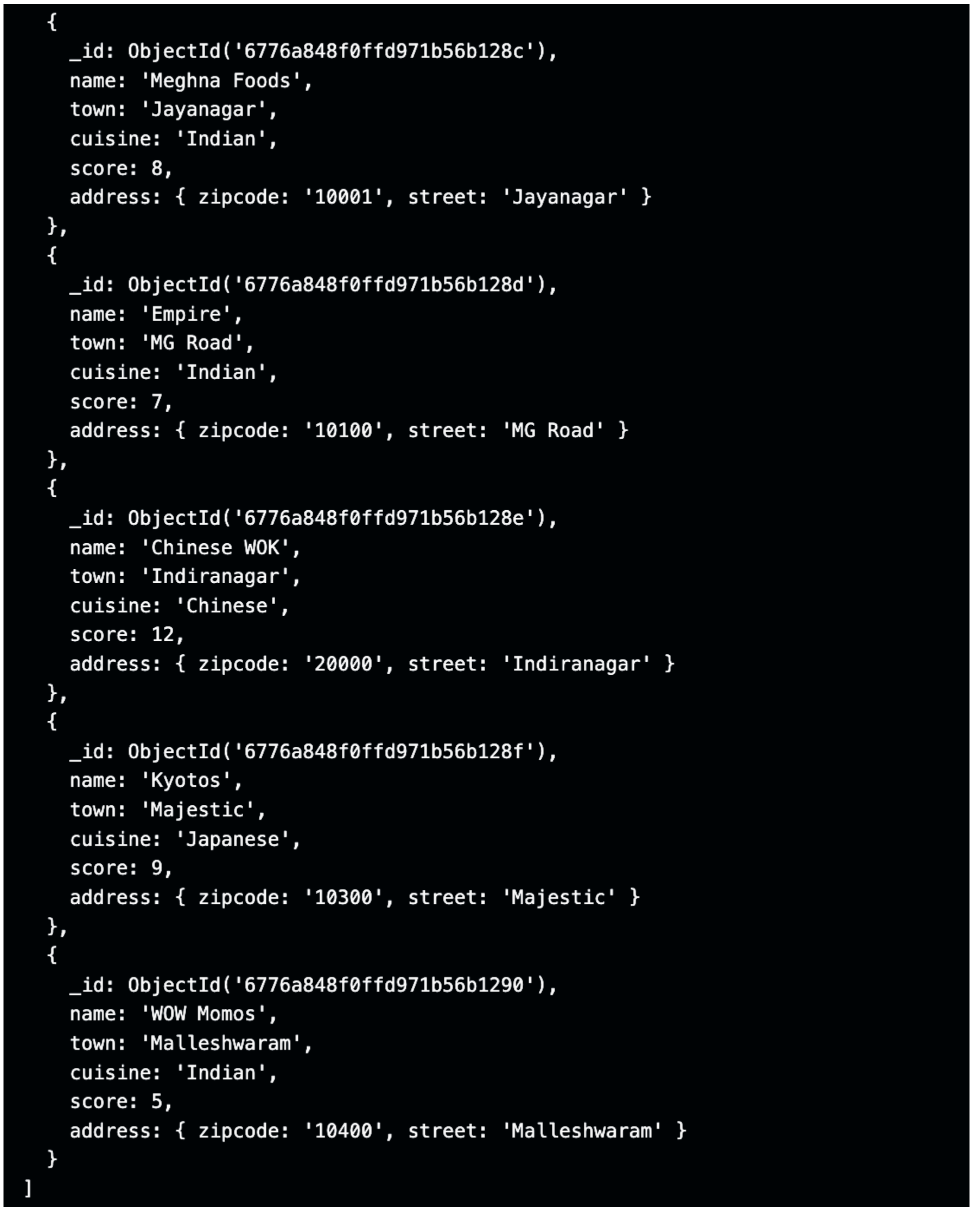
```
db.createCollection("restaurants");
```



```
{ "ok" : 1 }
```

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

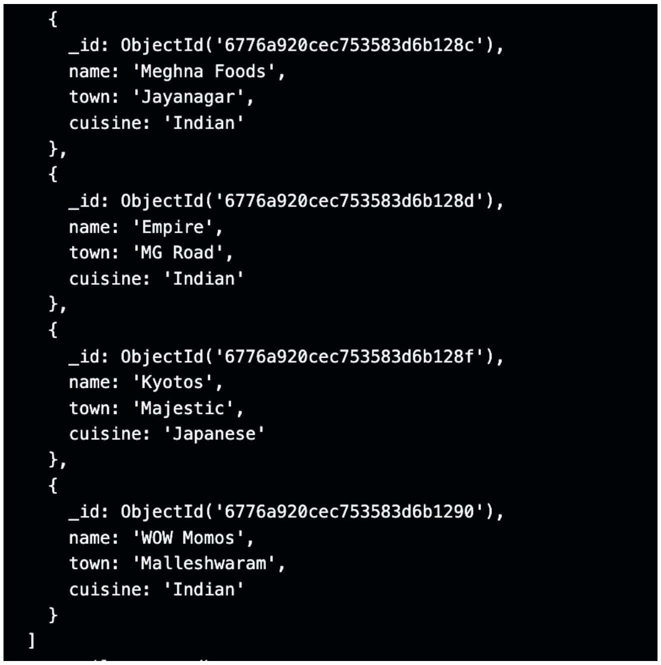
```
db.restaurants.insertMany([
{ name: "Meghna Foods", town: "Jayanagar", cuisine: "Indian", score: 8, address: {
zipcode: "10001", street: "Jayanagar“
} },
{ name: "Empire", town: "MG Road", cuisine: "Indian", score: 7, address: {
zipcode: "10100", street: "MG Road" } },
{ name: "Chinese WOK", town: "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({})
```



```
  {
    _id: ObjectId('6776a848f0ffd971b56b128c'),
    name: 'Meghna Foods',
    town: 'Jayanagar',
    cuisine: 'Indian',
    score: 8,
    address: { zipcode: '10001', street: 'Jayanagar' }
  },
  {
    _id: ObjectId('6776a848f0ffd971b56b128d'),
    name: 'Empire',
    town: 'MG Road',
    cuisine: 'Indian',
    score: 7,
    address: { zipcode: '10100', street: 'MG Road' }
  },
  {
    _id: ObjectId('6776a848f0ffd971b56b128e'),
    name: 'Chinese WOK',
    town: 'Indiranagar',
    cuisine: 'Chinese',
    score: 12,
    address: { zipcode: '20000', street: 'Indiranagar' }
  },
  {
    _id: ObjectId('6776a848f0ffd971b56b128f'),
    name: 'Kyotos',
    town: 'Majestic',
    cuisine: 'Japanese',
    score: 9,
    address: { zipcode: '10300', street: 'Majestic' }
  },
  {
    _id: ObjectId('6776a848f0ffd971b56b1290'),
    name: 'WOW Momos',
    town: 'Malleshwaram',
    cuisine: 'Indian',
    score: 5,
    address: { zipcode: '10400', street: 'Malleshwaram' }
  }
]
```

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.

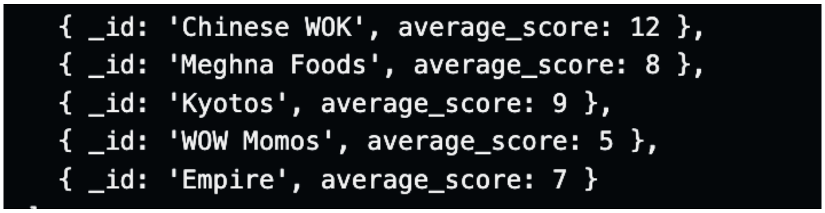
db.restaurants.find({ "score": { $lte: 10 } }, { _id: 1, name: 1, town: 1, cuisine: 1 })



```
  {
    _id: ObjectId('6776a920cec753583d6b128c'),
    name: 'Meghna Foods',
    town: 'Jayanagar',
    cuisine: 'Indian'
  },
  {
    _id: ObjectId('6776a920cec753583d6b128d'),
    name: 'Empire',
    town: 'MG Road',
    cuisine: 'Indian'
  },
  {
    _id: ObjectId('6776a920cec753583d6b128f'),
    name: 'Kyotos',
    town: 'Majestic',
    cuisine: 'Japanese'
  },
  {
    _id: ObjectId('6776a920cec753583d6b1290'),
    name: 'WOW Momos',
    town: 'Malleshwaram',
    cuisine: 'Indian'
  }
]
```

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

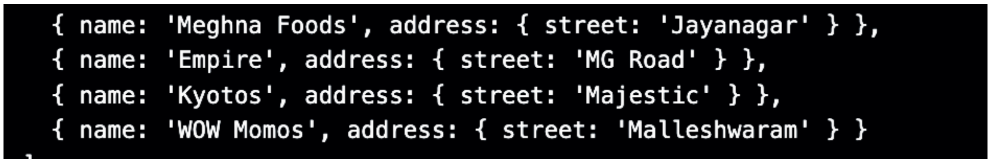
db.restaurants.aggregate([{ $group: { _id: "$name", average_score: { $avg: "$score" } } }])



```
  { _id: 'Chinese WOK', average_score: 12 },
  { _id: 'Meghna Foods', average_score: 8 },
  { _id: 'Kyotos', average_score: 9 },
  { _id: 'WOW Momos', average_score: 5 },
  { _id: 'Empire', average_score: 7 }
```

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 })



```
  { name: 'Meghna Foods', address: { street: 'Jayanagar' } },
  { name: 'Empire', address: { street: 'MG Road' } },
  { name: 'Kyotos', address: { street: 'Majestic' } },
  { name: 'WOW Momos', address: { street: 'Malleshwaram' } }
```