

EXERCISE 18

Structure of 'restaurants' collection:

```
{  
  "address": {  
    "building": "1007",  
    "coord": [ -73.856077, 40.848447 ],  
    "street": "Morris Park Ave",  
    "zipcode": "10462"  
  },  
  "borough": "Bronx",  
  "cuisine": "Bakery",  
  "grades": [  
    { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },  
    { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },  
    { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },  
    { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },  
    { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }  
  ],  
  "name": "Morris Park Bake Shop",  
  "restaurant_id": "30075445"  
}
```

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.restaurants.find(  
  {  
    $or: [  
      { cuisine: { $nin: ['American', 'Chinese'] } },  
      { name: { $regex: /^Wil/ } }  
    ]  
  },  
  { restaurant_id: 1, name: 1, borough: 1, cuisine: 1, _id: 0 }  
);
```

```
{  
  "borough": "Bronx",  
  "cuisine": "Bakery",  
  "name": "Morris Park Bake Shop",  
  "restaurant_id": "30075445"  
}  
{  
  "borough": "Bronx",  
  "cuisine": "Bakery",  
  "name": "Morris Park Bake Shop",  
  "restaurant_id": "30075445"
```

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.restaurants.find(  
  {  
    grades: {  
      $elemMatch: {  
        grade: "A",  
        score: 11,  
        date: ISODate("2014-08-11T00:00:00Z")  
      }  
    },  
    { restaurant_id: 1, name: 1, grades: 1, _id: 0 }  
  );
```

```
{  
  restaurant_id: 'sample0001',  
  name: 'Wilshire Grille',  
  grades: [  
    {  
      date: 2014-08-11T00:00:00.000Z,  
      grade: 'A',  
      score: 11  
    },  
    {  
      date: 2015-01-21T00:00:00.000Z,  
      grade: 'B',  
      score: 7  
    }  
  ]  
}
```

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.restaurants.find(  
  {  
    "grades.1": {  
      grade: "A",  
      score: 9,  
      date: ISODate("2014-08-11T00:00:00Z")  
    }  
  },  
  { restaurant_id: 1, name: 1, grades: 1, _id: 0 }  
);
```

```
> db.restaurants.find(  
  {  
    "grades.1": {  
      grade: "A",  
      score: 9,  
      date: ISODate("2014-08-11T00:00:00Z")  
    }  
  },  
  { restaurant_id: 1, name: 1, grades: 1, _id: 0 }  
);  
<  
test>
```

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.restaurants.find(  
  { "address.coord.1": { $gt: 42, $lte: 52 } },  
  { restaurant_id: 1, name: 1, address: 1, _id: 0 }  
);
```

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

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

```
({"_id": ObjectId("6965d7ffde54e0138021aaeb3"),  
address: {  
building: "100B",  
coord: {  
-73.951,  
40.764  
},  
street: "5th Ave.",  
zipcode: "10028"},  
borough: "Manhattan",  
cuisine: "Chinese",  
grades: [  
{  
date: 2013-08-01T00:00:00.000Z,  
grade: "A",  
score: 9  
},  
,  
{  
date: 2018-06-11T00:00:00.000Z,  
grade: "A",  
score: 7  
}  
],  
name: "Golden Dragon",  
restaurant_id: "46678921"  
},  
{  
_id: ObjectId("6965d7ffde54e0138021aaeb4"),  

```

NEIL DANIEL A - 240701356 - DBMS RECORD

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

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

```
[{"_id": ObjectId("6085d7cddde540138812aabf"),  
address: {  
building: "1200",  
coord: [  
-71.057413,  
40.772926  
],  
street: "Lexington Ave",  
zipcode: "10023"},  
borough: "Manhattan",  
cuisine: "Italian",  
grades: [  
{  
date: 2015-01-08T00:00:00Z,  
grade: "A",  
score: 13  
},  
{  
date: 2014-10-14T00:00:00Z,  
grade: "B",  
score: 8  
}],  
name: "Pasta Paradise",  
restaurant_id: "40356152"},  
{"_id": ObjectId("6085d7ffddde540138812aabf"),  
address: {  
building: "1010",  
coord: [  
-71.951,  
40.764  
],  
street: "Bedford Ave",  
zipcode: "11211"},  
borough: "Brooklyn",  
cuisine: "Caribbean",  
grades: [  
{  
date: 2021-03-20T00:00:00Z,  
grade: "B",  
score: 14  
},  
{  
date: 2019-11-15T00:00:00Z,  
grade: "C",  
score: 17  
}],  
name: "Island Breeze",  
restaurant_id: "48876549"}]
```

7. 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.restaurants.find().sort(  
{ cuisine: 1, borough: -1 });
```

```
[{"_id": ObjectId("6085d7ffddde540138812aabf"),  
address: {  
building: "1010",  
coord: [  
-71.951,  
40.764  
],  
street: "Bedford Ave",  
zipcode: "11202"},  
borough: "Brooklyn",  
cuisine: "Chinese",  
grades: [  
{  
date: 2019-08-01T00:00:00Z,  
grade: "A",  
score: 9  
},  
{  
date: 2018-11-11T00:00:00Z,  
grade: "A",  
score: 7  
}],  
name: "Golden Dragon",  
restaurant_id: "46678921"}]
```

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

```
db.restaurants.find({ "address.street": { $exists: false } });
```

```
> db.restaurants.find({ "address.street": { $exists: false } });  
<
```

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

```
db.restaurants.find(
  { "address.coord": 
    { $type: "double" } 
  });

```

```
{
  "_id": ObjectId("6985d7ffddc540138812aab2"),
  "address": {
    "building": "1280",
    "coord": [
      -73.9557413,
      40.7720266
    ],
    "street": "Lexington Ave",
    "zipcode": "10021"
  },
  "borough": "Manhattan",
  "cuisine": "Italian",
  "grades": [
    {
      "date": 2015-01-08T00:00:00Z,
      "grade": "A",
      "score": 11
    },
    {
      "date": 2014-10-14T00:00:00Z,
      "grade": "B",
      "score": 8
    }
  ],
  "name": "Pasta Paradise",
  "restaurant_id": "40356152"
},
{
  "_id": ObjectId("6985d7c6ddc540138812aaaf"),
  "address": {
    "building": "1280",
    "coord": [
      -73.9557413,
      40.7720266
    ],
    "street": "Lexington Ave",
    "zipcode": "10021"
  },
  "borough": "Manhattan",
  "cuisine": "Italian",
  "grades": [
    {
      "date": 2015-01-06T00:00:00Z,
      "grade": "A",
      "score": 11
    },
    {
      "date": 2014-10-14T00:00:00Z,
      "grade": "B",
      "score": 8
    }
  ],
  "name": "Pasta Paradise",
  "restaurant_id": "40356152"
}
```

10. 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.restaurants.find(
  { "grades.score": { $mod: [7, 0] } },
  { restaurant_id: 1,
    name: 1, grades: 1, _id: 0 }
);

```

```
{
  "grades": [
    {
      "date": 2014-03-01T00:00:00Z,
      "grade": "A",
      "score": 2
    },
    {
      "date": 2013-09-11T00:00:00Z,
      "grade": "A",
      "score": 6
    },
    {
      "date": 2013-03-21T00:00:00Z,
      "grade": "A",
      "score": 10
    },
    {
      "date": 2013-11-21T00:00:00Z,
      "grade": "A",
      "score": 9
    },
    {
      "date": 2013-03-01T00:00:00Z,
      "grade": "B",
      "score": 14
    }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "140356152"
},
{
  "grades": [
    {
      "date": 2014-03-01T00:00:00Z,
      "grade": "E",
      "score": 26
    },
    {
      "date": 2013-09-11T00:00:00Z,
      "grade": "A",
      "score": 6
    },
    {
      "date": 2013-03-21T00:00:00Z,
      "grade": "A",
      "score": 10
    },
    {
      "date": 2013-11-21T00:00:00Z,
      "grade": "A",
      "score": 9
    },
    {
      "date": 2013-03-01T00:00:00Z,
      "grade": "B",
      "score": 14
    }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "140356152"
},
{
  "grades": [
    {
      "date": 2014-11-15T00:00:00Z,
      "grade": "E",
      "score": 36
    },
    {
      "date": 2014-05-02T00:00:00Z,
      "grade": "A",
      "score": 18
    },
    {
      "date": 2013-03-02T00:00:00Z,
      "grade": "A",
      "score": 17
    },
    {
      "date": 2012-02-18T00:00:00Z,
      "grade": "A",
      "score": 13
    }
  ],
  "name": "Brunch On The Boulevard",
  "restaurant_id": "140356152"
}
```

11. 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.restaurants.find(
  { name: { $regex: "mon" } },
  { name: 1, borough: 1, "address.coord": 1, cuisine: 1, _id: 0 }
);
> db.restaurants.find(
  { name: { $regex: "mon" } },
  { name: 1, borough: 1, "address.coord": 1, cuisine: 1, _id: 0 }
);
<
```

NEIL DANIEL A - 240701356 - DBMS RECORD

12. 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.restaurants.find(  
  { name: { $regex: "^Mad" } },  
  { name: 1, borough: 1, "address.coord": 1, cuisine: 1, _id: 0 }  
)  
  > db.restaurants.find(  
    { name: { $regex: "^Mad" } },  
    { name: 1, borough: 1, "address.coord": 1, cuisine: 1, _id: 0 }  
  );  
<  
test>
```

13. Write a MongoDB 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 } }  
)
```

```
10:05:21 [1] {MongoDB shell version v3.2.10  
connecting to: test  
test> db.restaurants.find({ "grades.score": { $lt: 5 } })  
[  
  {  
    "_id": "55a191c17f98000000000000",  
    "borough": "Bronx",  
    "cuisine": "American",  
    "name": "The Bronx Zoo",  
    "address": "Bronx Park East",  
    "lat": 40.846447,  
    "lon": -73.856571,  
    "grades": [ {  
      "date": "2013-01-11T00:00:00Z",  
      "grade": "A",  
      "score": 5  
    }, {  
      "date": "2013-01-11T00:00:00Z",  
      "grade": "A",  
      "score": 5  
    }, {  
      "date": "2013-01-11T00:00:00Z",  
      "grade": "A",  
      "score": 5  
    }, {  
      "date": "2013-01-11T00:00:00Z",  
      "grade": "A",  
      "score": 5  
    } ]  
  },  
  {  
    "_id": "55a191c17f98000000000001",  
    "borough": "Bronx",  
    "cuisine": "American",  
    "name": "The Bronx Zoo",  
    "address": "Bronx Park East",  
    "lat": 40.846447,  
    "lon": -73.856571,  
    "grades": [ {  
      "date": "2013-01-11T00:00:00Z",  
      "grade": "A",  
      "score": 5  
    }, {  
      "date": "2013-01-11T00:00:00Z",  
      "grade": "A",  
      "score": 5  
    }, {  
      "date": "2013-01-11T00:00:00Z",  
      "grade": "A",  
      "score": 5  
    }, {  
      "date": "2013-01-11T00:00:00Z",  
      "grade": "A",  
      "score": 5  
    } ]  
  }]  
]
```

14. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan.

```
db.restaurants.find(  
  { borough: "Manhattan", "grades.score": { $lt: 5 } }  
)  
  > db.restaurants.find(  
    { borough: "Manhattan", "grades.score": { $lt: 5 } }  
  );  
<  
test>
```

NEIL DANIEL A - 240701356 - DBMS RECORD

15. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn.

```
db.restaurants.find(  
  { borough: { $in: ["Manhattan", "Brooklyn"] }, "grades.score": { $lt:  
5 } }  
);  
  > db.restaurants.find(  
    { borough: { $in: ["Manhattan", "Brooklyn"] }, "grades.score": { $lt: 5 } }  
  );  
  <  
  test> |
```

16. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn, and their cuisine is not American.

```
db.restaurants.find(  
{  
  borough: { $in: ["Manhattan", "Brooklyn"] },  
  "grades.score": { $lt: 5 },  
  cuisine: { $ne: "American" }  
}  
);
```

> db.restaurants.find(
{
 borough: { \$in: ["Manhattan", "Brooklyn"] },
 "grades.score": { \$lt: 5 },
 cuisine: { \$ne: "American" }
}
);
<
test>|

17. Write a MongoDB query to find the restaurants that have at least one grade with a score of less than 5 and that are located in the borough of Manhattan or Brooklyn, and their cuisine is not American or Chinese.

```
db.restaurants.find({  
  grades: {  
    $all: [  
      { $elemMatch: { score: 2 } },  
      { $elemMatch: { score: 6 } }  
    ]  
  }  
});
```

{
 _id: ObjectId('60f9334c20d6433e27b45ef7'),
 address: {
 building: '10097',
 coord: [
 -73.856077,
 40.68447
],
 street: 'Morris Park Ave',
 zipcode: '10462'
 },
 borough: 'Bronx',
 cuisine: 'Bakery',
 grades: [
 {
 date: 2014-03-03T00:00:00.000Z,
 grade: 'A',
 score: 2
 },
 {
 date: 2013-09-11T00:00:00.000Z,
 grade: 'A',
 score: 6
 },
 {
 date: 2013-01-27T00:00:00.000Z,
 grade: 'A',
 score: 10
 },
 {
 date: 2011-11-23T00:00:00.000Z,
 grade: 'A',
 score: 5
 },
 {
 date: 2014-03-03T00:00:00.000Z,
 grade: 'A',
 score: 2
 },
 {
 date: 2013-09-11T00:00:00.000Z,
 grade: 'A',
 score: 6
 },
 {
 date: 2013-01-27T00:00:00.000Z,
 grade: 'A',
 score: 10
 },
 {
 date: 2011-11-23T00:00:00.000Z,
 grade: 'A',
 score: 9
 }
]
}

18. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6.

```
db.restaurants.find(  
{  
    borough: "Manhattan",  
    grades: {  
        $all: [  
            { $elemMatch: { score: 2 } },  
            { $elemMatch: { score: 6 } }  
        ]  
    }  
};
```

```
> db.restaurants.find(  
{  
    borough: "Manhattan",  
    grades: {  
        $all: [  
            { $elemMatch: { score: 2 } },  
            { $elemMatch: { score: 6 } }  
        ]  
    }  
};  
<  
test>|
```

19. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan.

```
db.restaurants.find(  
{  
    borough: { $in: ["Manhattan", "Brooklyn"] },  
    grades: {  
        $all: [  
            { $elemMatch: { score: 2 } },  
            { $elemMatch: { score: 6 } }  
        ]  
    }  
};
```

```
> db.restaurants.find(  
{  
    borough: { $in: ["Manhattan", "Brooklyn"] },  
    grades: {  
        $all: [  
            { $elemMatch: { score: 2 } },  
            { $elemMatch: { score: 6 } }  
        ]  
    }  
};  
<  
test>|
```

NEIL DANIEL A - 240701356 - DBMS RECORD

20. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan or Brooklyn.

```
db.restaurants.find(  
  {  
    borough: { $in: ["Manhattan", "Brooklyn"] },  
    grades: {  
      $all: [  
        { $elemMatch: { score: 2 } },  
        { $elemMatch: { score: 6 } }  
      ]  
    },  
    cuisine: { $ne: "American" }  
  }  
);
```

```
> db.restaurants.find(  
  {  
    borough: { $in: ["Manhattan", "Brooklyn"] },  
    grades: {  
      $all: [  
        { $elemMatch: { score: 2 } },  
        { $elemMatch: { score: 6 } }  
      ]  
    },  
    cuisine: { $ne: "American" }  
  }  
);  
<  
test>
```

21. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan or Brooklyn, and their cuisine is not American.

```
db.restaurants.find({  
  borough: { $in: ["Manhattan", "Brooklyn"] },  
  cuisine: { $ne: "American" },  
  grades: {  
    $all: [  
      { $elemMatch: { score: 2 } },  
      { $elemMatch: { score: 6 } }  
    ]  
  }  
});
```

```
> db.restaurants.find({  
  borough: { $in: ["Manhattan", "Brooklyn"] },  
  cuisine: { $ne: "American" },  
  grades: {  
    $all: [  
      { $elemMatch: { score: 2 } },  
      { $elemMatch: { score: 6 } }  
    ]  
  }  
});  
<  
test>
```

22. Write a MongoDB query to find the restaurants that have a grade with a score of 2 and a grade with a score of 6 and are located in the borough of Manhattan or Brooklyn, and their cuisine is not American or Chinese.

```
db.restaurants.find(  
{  
    borough: { $in: ["Manhattan", "Brooklyn"] },  
    grades: {  
        $all: [  
            { $elemMatch: { score: 2 } },  
            { $elemMatch: { score: 6 } }  
        ]  
    },  
    cuisine: { $nin: ["American", "Chinese"] }  
}  
);
```

```
> db.restaurants.find({  
    
    borough: { $in: ["Manhattan", "Brooklyn"] },  
    grades: {  
      $all: [  
        { $elemMatch: { score: 2 } },  
        { $elemMatch: { score: 6 } }  
      ]  
    },  
    cuisine: { $nin: ["American", "Chinese"] }  
  }  
);  
<  
test>
```

23. Write a MongoDB query to find the restaurants that have a grade with a score of 2 or a grade with a score of 6.

```
db.restaurants.find(  
  {  
    grades: {  
      $elemMatch: { score: { $in: [2, 6] } }  
    }  
  }  
);
```

```
  {
    "_id": ObjectId("0ef934cc2086442e2b705e7d"),
    "address": {
      "building": "18807",
      "corner": "73-856077",
      "cross": "48.866447"
    },
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  {
    "borough": "Bronx",
    "cuisine": "Bakery",
    "grades": [
      {
        "date": "2014-03-03T00:00:00Z",
        "grade": "A",
        "score": 10
      },
      {
        "date": "2013-09-11T00:00:00Z",
        "grade": "A",
        "score": 10
      },
      {
        "date": "2013-01-24T00:00:00Z",
        "grade": "A",
        "score": 10
      },
      {
        "date": "2011-11-23T00:00:00Z",
        "grade": "A",
        "score": 9
      }
    ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  {
    "_id": ObjectId("0f955d2fedde2e232ffbb39a"),
    "address": {
      "building": "18807",
      "corner": "73-856077",
      "cross": "48.866447"
    },
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  {
    "borough": "Bronx",
    "cuisine": "Bakery",
    "grades": [
      {
        "date": "2014-03-03T00:00:00Z",
        "grade": "A",
        "score": 10
      },
      {
        "date": "2013-09-11T00:00:00Z",
        "grade": "A",
        "score": 6
      },
      {
        "date": "2013-01-24T00:00:00Z",
        "grade": "A",
        "score": 10
      },
      {
        "date": "2011-11-23T00:00:00Z",
        "grade": "A",
        "score": 9
      }
    ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  }
]
```

NEIL DANIEL A - 240701356 - DBMS RECORD

Evaluation Procedure	Marks awarded
MONGODB Procedure(5)	
Program/Execution (5)	
Viva(5)	
Total (15)	
Faculty Signature	