

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'
}
```

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 }
);
```

```
> 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("6095d7f5de540138012a0b3"),
  "address": {
    "building": "1818",
    "coord": [
      -73.951,
      40.764
    ],
    "street": "5th Ave",
    "zipcode": "10028"
  },
  "borough": "Manhattan",
  "cuisine": "Chinese",
  "grades": [
    {
      "date": "2019-08-01T00:00:00.000Z",
      "grade": "A",
      "score": 9
    },
    {
      "date": "2018-06-01T00:00:00.000Z",
      "grade": "A",
      "score": 7
    }
  ],
  "name": "Golden Dragon",
  "restaurant_id": "40678921"
}
```

```
{
  "_id": ObjectId("6095d7f5de540138012a0b4"),
  "address": {
    "building": "404",
    "coord": [
      -73.999,
      40.731
    ],
    "street": "Bedford Ave",
    "zipcode": "11211"
  },
  "borough": "Brooklyn",
  "cuisine": "Caribbean",
  "grades": [
    {
      "date": "2021-03-20T00:00:00.000Z",
      "grade": "B",
      "score": 14
    },
    {
      "date": "2019-11-01T00:00:00.000Z",
      "grade": "C",
      "score": 17
    }
  ],
  "name": "Island Breeze",
  "restaurant_id": "40678563"
}
```

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("6905d7c6d6e40138012a0f7"),
  "address": {
    "building": "1200",
    "coord": [
      -73.95743,
      40.772656
    ],
    "street": "Lexington Ave",
    "zipcode": "10021"
  },
  "borough": "Manhattan",
  "cuisine": "Italian",
  "grades": [
    {
      "date": "2015-01-01T00:00:00.000Z",
      "grade": "A",
      "score": 11
    },
    {
      "date": "2014-10-01T00:00:00.000Z",
      "grade": "B",
      "score": 8
    }
  ],
  "name": "Pasta Paradise",
  "restaurant_id": "40356232"
}
```

```
{
  "_id": ObjectId("6905d7c6d6e40138012a0b3"),
  "restaurant_id": "40356081",
  "name": "Wishfire Grill",
  "borough": "Brooklyn",
  "cuisine": "Panam",
  "address": {
    "building": "1081",
    "coord": [
      -73.95,
      40.05
    ],
    "street": "Wishfire Ave",
    "zipcode": "11201"
  },
  "grades": [
    {
      "date": "2014-08-01T00:00:00.000Z",
      "grade": "A",
      "score": 11
    },
    {
      "date": "2013-01-01T00:00:00.000Z",
      "grade": "B",
      "score": 7
    }
  ]
}
```

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("6905d7f6d6e40138012a0b4"),
  "address": {
    "building": "404",
    "coord": [
      -73.995,
      40.721
    ],
    "street": "Bedford Ave",
    "zipcode": "11211"
  },
  "borough": "Brooklyn",
  "cuisine": "Caribbean",
  "grades": [
    {
      "date": "2021-03-20T00:00:00.000Z",
      "grade": "B",
      "score": 10
    },
    {
      "date": "2019-11-01T00:00:00.000Z",
      "grade": "C",
      "score": 17
    }
  ],
  "name": "Island Breeze",
  "restaurant_id": "40876543"
}
```

```
{
  "_id": ObjectId("6905d7f6d6e40138012a0b3"),
  "address": {
    "building": "1010",
    "coord": [
      -73.951,
      40.764
    ],
    "street": "5th Ave",
    "zipcode": "10020"
  },
  "borough": "Manhattan",
  "cuisine": "Chinese",
  "grades": [
    {
      "date": "2019-04-01T00:00:00.000Z",
      "grade": "A",
      "score": 9
    },
    {
      "date": "2018-06-01T00:00:00.000Z",
      "grade": "A",
      "score": 7
    }
  ],
  "name": "Golden Dragon",
  "restaurant_id": "40675921"
}
```

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("6965d7ffded40138012a82"),
  "address": {
    "building": "1200",
    "coord": [
      -73.9557413,
      40.7720266
    ],
    "street": "Lexington Ave",
    "zipcode": "10021"
  },
  "borough": "Manhattan",
  "cuisine": "Italian",
  "grades": [
    {
      "date": "2015-01-05T00:00:00.000Z",
      "grade": "A",
      "score": 11
    },
    {
      "date": "2014-10-14T00:00:00.000Z",
      "grade": "B",
      "score": 8
    }
  ],
  "name": "Pasta Paradise",
  "restaurant_id": "A6356152"
}
```

```
{
  "_id": ObjectId("6965d7c5d6e40138012a8f"),
  "address": {
    "building": "1200",
    "coord": [
      -73.9557413,
      40.7720266
    ],
    "street": "Lexington Ave",
    "zipcode": "10021"
  },
  "borough": "Manhattan",
  "cuisine": "Italian",
  "grades": [
    {
      "date": "2015-01-05T00:00:00.000Z",
      "grade": "A",
      "score": 11
    },
    {
      "date": "2014-10-14T00:00:00.000Z",
      "grade": "B",
      "score": 8
    }
  ],
  "name": "Pasta Paradise",
  "restaurant_id": "A6356152"
}
```

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": "2014-01-05T00:00:00.000Z",
      "grade": "A",
      "score": 1
    },
    {
      "date": "2013-01-05T00:00:00.000Z",
      "grade": "A",
      "score": 8
    },
    {
      "date": "2013-01-05T00:00:00.000Z",
      "grade": "A",
      "score": 10
    },
    {
      "date": "2013-01-05T00:00:00.000Z",
      "grade": "A",
      "score": 9
    },
    {
      "date": "2013-01-05T00:00:00.000Z",
      "grade": "B",
      "score": 14
    }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "10010460"
}
```

```
{
  "grades": [
    {
      "date": "2014-01-05T00:00:00.000Z",
      "grade": "A",
      "score": 1
    },
    {
      "date": "2013-01-05T00:00:00.000Z",
      "grade": "A",
      "score": 8
    },
    {
      "date": "2013-01-05T00:00:00.000Z",
      "grade": "A",
      "score": 10
    },
    {
      "date": "2013-01-05T00:00:00.000Z",
      "grade": "A",
      "score": 9
    },
    {
      "date": "2013-01-05T00:00:00.000Z",
      "grade": "B",
      "score": 14
    }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "10010460"
}
```

```
{
  "grades": [
    {
      "date": "2014-11-05T00:00:00.000Z",
      "grade": "D",
      "score": 30
    },
    {
      "date": "2014-05-02T00:00:00.000Z",
      "grade": "A",
      "score": 10
    },
    {
      "date": "2013-03-02T00:00:00.000Z",
      "grade": "A",
      "score": 7
    },
    {
      "date": "2012-02-18T00:00:00.000Z",
      "grade": "A",
      "score": 13
    }
  ],
  "name": "Breads On The Boulevard",
  "restaurant_id": "A6356151"
}
```

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 }
);
<
```

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 } }
);
```

```
> db.restaurants.find(
  { "grades.score": { $lt: 5 } }
);
{
  "_id": "505085c1-07f8-4030-b002-000000000000",
  "name": "Madame Tussauds",
  "borough": "Manhattan",
  "address.coord": [
    -74.006,
    40.712
  ],
  "cuisine": "American",
  "grades": [
    {
      "score": 4,
      "date": "2012-01-01T00:00:00Z",
      "type": "A",
      "id": "505085c1-07f8-4030-b002-000000000000"
    },
    {
      "score": 4,
      "date": "2012-01-01T00:00:00Z",
      "type": "A",
      "id": "505085c1-07f8-4030-b002-000000000000"
    },
    {
      "score": 4,
      "date": "2012-01-01T00:00:00Z",
      "type": "A",
      "id": "505085c1-07f8-4030-b002-000000000000"
    },
    {
      "score": 4,
      "date": "2012-01-01T00:00:00Z",
      "type": "A",
      "id": "505085c1-07f8-4030-b002-000000000000"
    }
  ]
}
```

```
> db.restaurants.find(
  { "grades.score": { $lt: 5 } }
);
{
  "_id": "505085c1-07f8-4030-b002-000000000000",
  "name": "Madame Tussauds",
  "borough": "Manhattan",
  "address.coord": [
    -74.006,
    40.712
  ],
  "cuisine": "American",
  "grades": [
    {
      "score": 4,
      "date": "2012-01-01T00:00:00Z",
      "type": "A",
      "id": "505085c1-07f8-4030-b002-000000000000"
    },
    {
      "score": 4,
      "date": "2012-01-01T00:00:00Z",
      "type": "A",
      "id": "505085c1-07f8-4030-b002-000000000000"
    },
    {
      "score": 4,
      "date": "2012-01-01T00:00:00Z",
      "type": "A",
      "id": "505085c1-07f8-4030-b002-000000000000"
    },
    {
      "score": 4,
      "date": "2012-01-01T00:00:00Z",
      "type": "A",
      "id": "505085c1-07f8-4030-b002-000000000000"
    }
  ]
}
```

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>
```

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('603244e306442ba3704b6970')",
  "address": {
    "building": "1007",
    "coordinates": [
      -73.850077,
      40.840447
    ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    {
      "date": "2014-03-03T00:00:00.000Z",
      "grade": "A",
      "score": 2
    },
    {
      "date": "2013-09-17T00:00:00.000Z",
      "grade": "A",
      "score": 6
    },
    {
      "date": "2013-01-24T00:00:00.000Z",
      "grade": "A",
      "score": 10
    },
    {
      "date": "2011-11-23T00:00:00.000Z",
      "grade": "A",
      "score": 9
    },
    {

```

```
{
  "_id": "ObjectID('6085d2fede3a033f7a020a')",
  "address": {
    "building": "1007",
    "coordinates": [
      -73.850077,
      40.840447
    ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    {
      "date": "2014-03-03T00:00:00.000Z",
      "grade": "A",
      "score": 2
    },
    {
      "date": "2013-09-17T00:00:00.000Z",
      "grade": "A",
      "score": 6
    },
    {
      "date": "2013-01-24T00:00:00.000Z",
      "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>
```


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("687284a206442a27b4b7e7f"),
  "address": {
    "building": "1887",
    "couch": [
      -73.856077,
      40.848447
    ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    {
      "date": "2014-03-01T00:00:00.000Z",
      "grade": "A",
      "score": 2
    },
    {
      "date": "2013-09-11T00:00:00.000Z",
      "grade": "A",
      "score": 6
    },
    {
      "date": "2013-01-24T00:00:00.000Z",
      "grade": "A",
      "score": 10
    },
    {
      "date": "2011-11-21T00:00:00.000Z",
      "grade": "A",
      "score": 9
    },
    {

```

```
{
  "_id": ObjectId("6895d2f6e4b2a227fab7ba"),
  "address": {
    "building": "1887",
    "couch": [
      -73.856077,
      40.848447
    ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    {
      "date": "2014-03-01T00:00:00.000Z",
      "grade": "A",
      "score": 2
    },
    {
      "date": "2013-09-11T00:00:00.000Z",
      "grade": "A",
      "score": 6
    },
    {
      "date": "2013-01-24T00:00:00.000Z",
      "grade": "A",
      "score": 10
    },
    {
      "date": "2011-11-21T00:00:00.000Z",
      "grade": "A",
      "score": 9
    },
    {

```

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