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 }
);
```

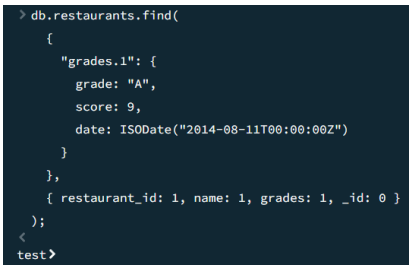

```
{
  borough: 'Bronx',
  cuisine: 'Bakery',
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
}
{
  borough: 'Bronx',
  cuisine: 'Bakery',
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
}
```

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 }
);
```

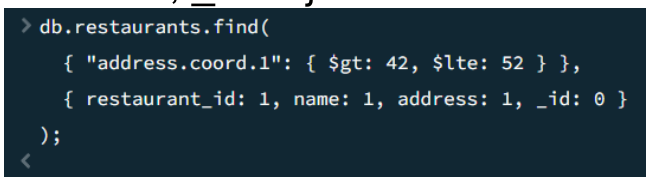
3. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z".

```
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 }
);
```



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 });
```

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

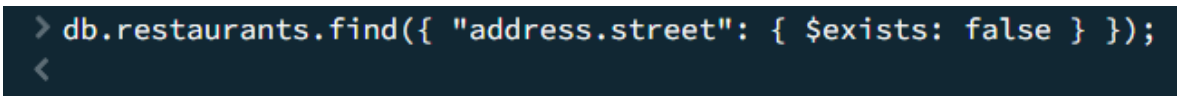
```
db.restaurants.find().sort({ name: -1 });
```

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 });
```

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

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



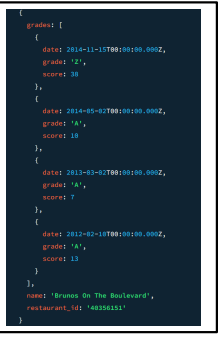
NEIL DANIEL A - 240701356 - DBMS RECORD

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" } }
);
```

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 }
);
```



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 }
);
```

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 } }
);
```

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 } }
);
```

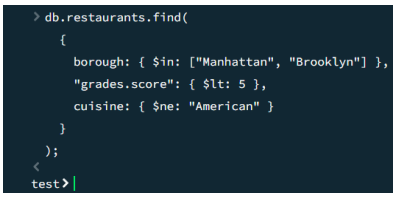
```
> db.restaurants.find(
    { borough: "Manhattan", "grades.score": { $lt: 5 } }
  );
<
test>
```

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 } }
);
```

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" }
  }
);
```



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 } }
      ]
    }
  }
);
```

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 } }
      ]
    }
  }
);
```

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 } }
      ]
    }
  }
);
```

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" }
  }
);
```

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 } }
    ]
  }
});
```

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"] }
  }
);
```

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] } }
    }
  }
);
```

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