20. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.

```
db.restorants.find({ 'grades.score': { $lt: 10 } }, { restaurant_id: true, name: true, borough: true, cuisine: true, _id: false })
```

21. 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.restorants.find(
  {
    $or: [
      { cuisine: { $nin: [/American/, /Chinees/] } },
      { name: { $not: /Wil/ } }
    ]
  },
  {
    restaurant_id: true,
    name: true,
    borough: true,
    cuisine: true,
    _id: false
  })
```

22. 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.restorants.find({

    'grades.grade': "A",

    'grades.score': 11,

    'grades.date': ISODate('2014-08-11T00:00:00Z')

}, {

  restaurant_id: 1,

  name: 1,

  grades: 1,

  _id: 0

})
```

23. 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.restorants.find({

    "grades.1.grade": "A",

    "grades.1.score": 9,

    "grades.1.date": ISODate('2014-08-11T00:00:00Z')

}, {

  restaurant_id: 1,

  name: 1,

  grades: 1,
  _id: 0

})
```

24. 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.restorants.find({

    "address.coord.1": { $gt: 42, $lt: 52 }

}, {

  restaurant_id: 1,

  name: 1,

  address: 1,

  _id: 0

})
```

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

```
db.restorants.find().sort({ "name": 1 })
```

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

```
db.restorants.find().sort({ "name": -1 })
```

27. 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.restorants.find().sort({ "cuisine": 1, "borough": -1 })
```

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

```
db.restorants.find({ "address.street": { $exists: true } })
```

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

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

30. 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.restorants.find({

    "grades.score": { $mod: [7, 0] }

}, {

  restaurant_id: 1,

  name: 1,

  grades: 1,

  _id: 0

})
```