Consider the Structure of 'restaurants' collection given below:

{

"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"

}

Tasks

1. Write a MongoDB query to display all the documents in the collection restaurants.
2. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant.
3. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine, but exclude the field \_id for all the documents in the collection restaurant.
4. Write a MongoDB query to display the fields restaurant\_id, name, borough and zip code, but exclude the field \_id for all the documents in the collection restaurant.
5. Write a MongoDB query to display all the restaurant which is in the borough Bronx.
6. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.
7. Write a MongoDB query to find the restaurants who achieved a score more than 90.
8. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.
9. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168.
10. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.