1. **Write a MongoDB query to display all the documents in the collection restaurants.**

db.restaurants.find()

2. **Write a MongoDB query to display the fields restaurant_id, name, district and cuisine for all the documents in the collection restaurant.**

db.restaurants.find({},{name:1, district:1, cuisine: 1})

3. **Write a MongoDB query to display the fields restaurant_id, name, district and cuisine, but exclude the field _id for all the documents in the collection restaurant.**

db.restaurants.find({},{name:1, district:1, cuisine: 1, _id:0})

4. **Write a MongoDB query to display the fields restaurant_id, name, district and zipcode, but exclude the field _id for all the documents in the collection restaurant.**

db.restaurants.find({},{name:1, district:1, "address.zipcode": 1, _id:0})

5. **Write a MongoDB query to display all the restaurant which is in the district Bronx.**

db.restaurants.find({district:"Bronx"}).pretty()

6. **Write a MongoDB query to display the first 5 restaurant which is in the district Bronx.**

db.restaurants.find({district:"Bronx"}).limit(5).pretty()

7. **Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the district Bronx.**

db.restaurants.find({district:"Bronx"}).skip(5).limit(5).pretty()

8. **Write a MongoDB query to find the restaurants which locates in latitude value less than -95.754168.**

db.restaurants.find({"address.coord[0]": {$lt: -95.754168}})

9. **Write a MongoDB query to find the restaurants that does not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.**

db.restaurants.find({$and: [{cuisine: {$ne: "American "}}, {"grades.score": {$gt: 70}}, {"address.coord[0]": {$lte: 0}}]})

10. **Write a MongoDB query to find the restaurant Id, name, district and cuisine for those restaurants which contains 'Wil' as first three letters for its name.**

db.restaurants.find({name: {$regex: "^Wil"}},{name: 1, district: 1, cuisine: 1})

11. **Write a MongoDB query to find the restaurant Id, name, district and cuisine for those restaurants which contains 'ces' as last three letters for its name.**

db.restaurants.find({name: {$regex: "ces$"}}, {name:1, district:1, cuisine:1})

12. Write a MongoDB query to find the restaurant Id, name, district and cuisine for those restaurants which contains 'Reg' as three letters somewhere in its name.

db.restaurants.find({name: {$regex: "Reg"}}, {name:1, district:1, cuisine:1})

13. Write a MongoDB query to find the restaurants which belongs to the district Bronx and prepared either American or Chinese dish.

db.restaurants.find({$and: [{district: "Bronx"}, {$or: [{cuisine: "American "}, {cuisine: "Chinese"}]}]}).pretty()

14. Write a MongoDB query to find the restaurant Id, name, district and cuisine for those restaurants which belongs to the district Staten Island or Queens or Bronx or Brooklyn.

db.restaurants.find({district: {$in: ["Bronx", "Queens", "Staten Island", "Brooklyn"]}}, {name:1, district: 1, cuisine: 1})

15. Write a MongoDB query to find the restaurant Id, name, district and cuisine for those restaurants which are not belonging to the district Staten Island or Queens or Bronx or Brooklyn.

db.restaurants.find({district: {$nin: ["Bronx", "Queens", "Staten Island"]}}, {name:1, district: 1, cuisine: 1})

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

db.restaurants.find({"grades.score": {$lte: 10}}, {restaurant_id:1, name:1, district:1, cuisine:1})

17. 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 up to 52.

db.restaurants.find({"address.coord[1]": {$lte: 52, $gt:42}}, {restaurant_id:1, name:1, address:1})

18. 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}).pretty()

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

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

20. Write a MongoDB query to arrange the name of the cuisine in ascending order and for those same cuisine district should be in descending order.

db.restaurants.find().sort({cuisine:1, district: -1})