- Import Restaurant json

```
mongoimport --db rest --collection rest1 --file D:\restaurants.json
```

A) Import Restaurant.json into MongoDb and perform the following queries

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

```
db.restaurants.find().pretty()
```

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

```
db.restaurants.find({},{restaurant_id: 1,name: 1,borough: 1,cuisine: 1,_id: 0})
```

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

```
db.restaurants.find({},{restaurant_id: 1,name: 1,borough: 1,cuisine: 1,_id: 0})
```

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

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

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

```
db.restaurants.find({ borough: "Bronx" })
```

B) Import Restaurant.json into MongoDb and perform the following queries

1. 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: /mon/i }, {name: 1,borough: 1,"address.coord": 1, cuisine: 1,_id: 0})
```

2. 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: /^Mad/i },{name: 1,borough: 1,longitude: 1,latitude: 1,cuisine: 1,_id: 0})
```

3. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168

```
db.restaurants.find({ "address.coord.1": { $lt: -95.754168 } } )
```

4. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

```
db.restaurants.find({ name: /^Wil/i }, {restaurant_id: 1,name: 1,borough: 1,cuisine: 1,_id: 0})
```

5.Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

```
db.restaurants.find({ name: /ces$/i }, {restaurant_id: 1,name: 1,borough: 1,cuisine: 1,_id: 0})
```

C) Import Restaurant.json into MongoDb and perform the following queries

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

   db.restaurants.find({ name: /Reg/i }, {restaurant_id: 1,name: 1,borough: 1,cuisine: 1,_id: 0})

2. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.

   db.restaurants.find({borough: "Bronx", cuisine: { $in: ["American", "Chinese"] } })

3. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn

   db.restaurants.find({borough: { $in: ["Staten Island", "Queens", "Bronx", "Brooklyn"] } },{restaurant_id: 1,name: 1,borough: 1,cuisine: 1,_id: 0})

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

   db.restaurants.find({borough: { $nin: ["Staten Island", "Queens", "Bronx", "Brooklyn"] } },{restaurant_id: 1,name: 1,borough: 1,cuisine: 1,_id: 0})

5. 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.restaurants.find({"grades.score": { $lte: 10 } },{restaurant_id: 1,name: 1,borough: 1,cuisine: 1,_id: 0})

D) Import Restaurant.json into MongoDb and perform the following queries

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: /^Wil/i }]},{restaurant_id: 1,name: 1,borough: 1,cuisine: 1,_id: 0})
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", "grades.1.score": 9,"grades.1.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,"address.coord": 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 })

E) Import Restaurant.json into MongoDb and perform the following queries

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

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

2. 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 })

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

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

4. 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.0": { $type: 1 }, "address.coord.1": { $type: 1 }   })

5.Write a MongoDB query which will select the restaurant Id, name and gradesf or 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})