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

```
db.rest.find();
```

2. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine for all the documents in the collection restaurant.
   ```
   db.rest.find({},{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1});
   ```

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.rest.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.rest.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.rest.find({"borough": "Bronx"});
```

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

```
db.rest.find({"borough": "Bronx"}).limit(5);
```

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

```
db.rest.find({"borough": "Bronx"}).skip(5).limit(5);
```

8. Write a MongoDB query to find the restaurants who achieved a score more than 90.

```
db.rest.find({grades : { $elemMatch:{"score":{$gt : 90}}}});
```

9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.

```
db.rest.find({grades : { $elemMatch:{"score":{$gt : 80 , $lt
:100}}}});
```

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

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