**AIM: Performing queries based on AND, OR, Limit, Sort and Projection and apply some queries to get specified output.**

1. **AND , OR, SORT, LIMIT, SKIP, count, projections SYNTAX**
   - $and

     *Syntax*: { $and: [ { <expression1> }, { <expression2> } , ... , { <expressionN> } ] }

     **db.inventory.find( { $and: [ { price: { $ne: 1.99 } }, { price: 10} ] } )**

   - $or

     { $or: [ { <expression1> }, { <expression2> }, ... , { <expressionN> } ] }

     **db.inventory.find( { $or: [ { quantity: { $lt: 20 } }, { price: 10 } ] } )**

   - skip

     If we want to fetch two documents after the first two documents from the collection 'userdetails', the following mongodb command can be used :

     >db.userdetails.find().skip(2).pretty();

   - limit()

     To limit the records in MongoDB, you need to use **limit()** method, which is the number of documents that you want to be displayed.

     >db.COLLECTION_NAME.find().limit(NUMBER)

   - count
     The db.collection.count() method is used to return the count of documents that would match a find() query. Also counts number of records.

     db.restaurants.count();
     db.restaurants.find({"cuisine" : "American "}).count()

   - projection

Consider the collection mycol has the following data −

```
{ "_id" : ObjectId(5983548781331adf45ec5), "title":"MongoDB Overview"}
{ "_id" : ObjectId(5983548781331adf45ec6), "title":"NoSQL Overview"}
{ "_id" : ObjectId(5983548781331adf45ec7), "title":"Tutorials Point Overview"}
```

Following example will display the title of the document while querying the document.

```
>db.mycol.find({},{"title":1,_id:0})
{"title":"MongoDB Overview"}
{"title":"NoSQL Overview"}
{"title":"Tutorials Point Overview"}
>
```

Please note _id field is always displayed while executing find() method, if you don't want this field, then you need to set it as 0.

## PRACTICE QUESTIONS:

**Create collection name as "restaurants". Also insert 10 more documents in following format which satisfy all requirement of query.**

```
{
 "address": {
    "building": "1007",
    "coord": [ -73.856077, 40.848447 ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
 },
 "borough": "Bronx",
 "cuisine": "Bakery",
 "grades": [
            {"grade": "A", "score": 2 },
            {"grade": "A", "score": 6 },
            {"grade": "A", "score": 10 },
            { "grade": "A", "score": 9 },
            { "grade": "B", "score": 14 }
             ],
 "name": "Morris Park Bake Shop",
 "restaurant_id": "30075445"
}
```

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 zip code, but exclude the field _id for all the documents in the collection restaurant. (USING PROJECTION)

4. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx. (USING LIMIT)

5. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx. (USING SKIP)

6. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168. (USING AND)

7. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish. (USING OR)

8. Write a MongoDB query to arrange the name of the restaurants in ascending / descending order along with all the columns. (USING SORT)