

PROGRAMING LAB 3

ASSIGNMENT NO. – 7

NAME : AKASH RAKESH METKARI.

PRN : 22520012.

BATCH : T3.

Study of MongoDB

Problem Statement 1:

Connecting to MongoDB :

```
C:\Users\ametak>mongosh
Current Mongosh Log ID: 65476799494ea08e61149c54
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2
.0.2
Using MongoDB:      7.0.2
Using Mongosh:       2.0.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
  The server generated these startup warnings when booting
  2023-11-04T16:37:40.802+05:30: Access control is not enabled for the database. Read and write access to data and conf
  igation is unrestricted
  -----

test> |
```

Creating the games collection in assignment7 database :

```
test> use assignment7
switched to db assignment7
assignment7> db.createCollection("games")
{ ok: 1 }
assignment7> |
```

Inserting 5 games :

db.games.insertMany([

```
{ name: 'chess', genre: 'puzzle', rating: 9 },
{ name: 'cricket', genre: 'sport', rating: 8 },
{ name: 'boating', genre: 'Adventure', rating: 7 },
```

```
{ name: 'crossword', genre: 'Puzzle', rating: 5 },  
{ name: 'football', genre: 'Sport', rating: 6 },  
]);
```

Inserted one game :

```
assignment7> db.games.insertOne({name:"chess", genre:"puzzle",rating:9})  
{  
  acknowledged: true,  
  insertedId: ObjectId("65476b11494ea08e61149c55")  
}
```

Inserting many :

```
assignment7> db.games.insertMany([{game:"cricket",genre:"sport",rating :10},{game:"football",genre:"sport",rating :6},{game:"kabaddi",genre:"sport",rating :5},])  
{  
  acknowledged: true,  
  insertedIds: {  
    '0': ObjectId("65476bf0494ea08e61149c56"),  
    '1': ObjectId("65476bf0494ea08e61149c57"),  
    '2': ObjectId("65476bf0494ea08e61149c58"),  
    '3': ObjectId("65476bf0494ea08e61149c59")  
  }  
}
```

Returning all data : db.games.find()

```
assignment7> db.games.find()
[
  {
    _id: ObjectId("65476b11494ea08e61149c55"),
    name: 'chess',
    genre: 'puzzle',
    rating: 9
  },
  {
    _id: ObjectId("65476bf0494ea08e61149c56"),
    game: 'cricket',
    genre: 'sport',
    rating: 10
  },
  {
    _id: ObjectId("65476bf0494ea08e61149c57"),
    game: 'football',
    genre: 'sport',
    rating: 8
  },
  {
    _id: ObjectId("65476bf0494ea08e61149c58"),
    game: 'crossword',
    genre: 'puzzle',
    rating: 6
  },
  {
    _id: ObjectId("65476bf0494ea08e61149c59"),
    game: 'kabaddi',
    genre: 'sport',
    rating: 5
  }
]
```

Searching game by name :

```
assignment7> db.games.findOne({game:"cricket"})
{
  _id: ObjectId("65476bf0494ea08e61149c56"),
  game: 'cricket',
  genre: 'sport',
  rating: 10
}
```

Searching 3 highest rating games :

```
assignment7> db.games.find().sort({rating:-1}).limit(3)
[
  {
    _id: ObjectId("65476bf0494ea08e61149c56"),
    game: 'cricket',
    genre: 'sport',
    rating: 10
  },
  {
    _id: ObjectId("65476b11494ea08e61149c55"),
    name: 'chess',
    genre: 'puzzle',
    rating: 9
  },
  {
    _id: ObjectId("65476bf0494ea08e61149c57"),
    game: 'football',
    genre: 'sport',
    rating: 8
  }
]
```

Updated one game by using update():

```
assignment7> db.games.update({name:'cricket'},{$set:{achievement:['gamemaster','daemon']}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

After update :

```
assignment7> db.games.find()
[
  {
    _id: ObjectId("65476b11494ea08e61149c55"),
    name: 'chess',
    genre: 'puzzle',
    rating: 9,
    achievements: [ 'Game Master', 'Speed Demon' ]
  },
  {
    _id: ObjectId("65476bf0494ea08e61149c56"),
    game: 'cricket',
    genre: 'sport',
    rating: 10
  },
  {
    _id: ObjectId("65476bf0494ea08e61149c57"),
    game: 'football',
    genre: 'sport',
    rating: 8,
    achievements: [ 'Game Master', 'Speed Demon' ]
  },
  {
    _id: ObjectId("65476bf0494ea08e61149c58"),
    game: 'crossword',
    genre: 'puzzle',
    rating: 6
  },
  {
    _id: ObjectId("65476bf0494ea08e61149c59"),
    game: 'kabaddi',
    genre: 'sport',
    rating: 5
  }
]
```

Query that return games having achievements :

```
assignment7> db.games.find({ achievements: { $all: ['Game Master', 'Speed Demon'] } });
[
  {
    _id: ObjectId("65476b11494ea08e61149c55"),
    name: 'chess',
    genre: 'puzzle',
    rating: 9,
    achievements: [ 'Game Master', 'Speed Demon' ]
  },
  {
    _id: ObjectId("65476bf0494ea08e61149c57"),
    game: 'football',
    genre: 'sport',
    rating: 8,
    achievements: [ 'Game Master', 'Speed Demon' ]
  }
]
```

Problem statement 2: Mapreduce

Created a collection :

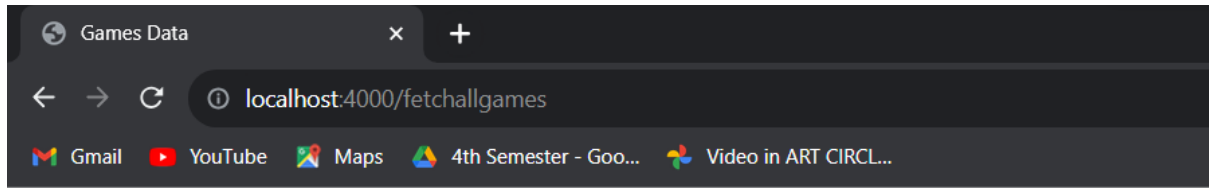
```
orders> db.orders.find()
[
  {
    _id: 1,
    cust_id: 'Ant O. Knee',
    ord_date: ISODate("2020-03-01T00:00:00.000Z"),
    price: 25,
    items: [
      { sku: 'oranges', qty: 5, price: 2.5 },
      { sku: 'apples', qty: 5, price: 2.5 }
    ],
    status: 'A'
  },
  {
    _id: 2,
    cust_id: 'Ant O. Knee',
    ord_date: ISODate("2020-03-08T00:00:00.000Z"),
    price: 70,
    items: [
      { sku: 'oranges', qty: 8, price: 2.5 },
      { sku: 'chocolates', qty: 5, price: 10 }
    ],
    status: 'A'
  },
]
```

mapReduce Function :

```
orders> var mapFunction1 = function () { emit(this.cust_id, this.price) };
orders> var reduceFunction1 = function (key, values) { return Array.sum(values) };
orders> db.orders.mapReduce(mapFunction1, reduceFunction1, { out: "result_mapReduce" });
{ result: 'result_mapReduce', ok: 1 }
orders> db.result_mapReduce.find();
[
  { _id: 'Cam Elot', value: 60 },
  { _id: 'Ant O. Knee', value: 95 },
  { _id: 'Busby Bee', value: 125 },
  { _id: 'Don Quis', value: 155 }
]
```

Problem statement 3 : Rest API

Showing the game data in HTML table format



Games Data

Name	Genre	Rating	Achievements
	puzzle	9	Game Master, Speed Demon
	sport	10	
	sport	8	Game Master, Speed Demon
	puzzle	6	
	sport	5	