

## Ukistu02 - MongoDB Ex2

1. Create a Database called **student**?

```
1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connection");
7      const url = "mongodb://localhost:27017/student";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         res.write("Database created !");
12         db.close();
13         res.end();
14       });
15     })
16     .listen(3000);
```

Database created !

2. Create a collection called studentmarks?

```
1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connection");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         dbo.createCollection("studentmarks", (err, response) => {
13             if (err) throw err;
14             res.write("Collection created!");
15             db.close();
16             res.end();
17         });
18       });
19     })
20     .listen(3000);
```

Collection created!

3. Create the documents listed in above table.

```
1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connetion");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         var myobj = [
13           { name: "Vanu", maths_marks: 80, english_marks: 75, science_marks: 85 },
14           { name: "Kala", maths_marks: 32, english_marks: 46, science_marks: 53 },
15           { name: "Aruli", maths_marks: 78, english_marks: 85, science_marks: 80 },
16           { name: "Shayu", maths_marks: 80, english_marks: 76, science_marks: 65 },
17           { name: "Kumaran", maths_marks: 32, english_marks: 73, science_marks: 84 },
18           { name: "Lucky", maths_marks: 66, english_marks: 90, science_marks: 45 },
19           { name: "Gva", maths_marks: 71, english_marks: 75, science_marks: 56 },
20           { name: "Raam", maths_marks: 41, english_marks: 65, science_marks: 88 }
21         ];
22
23         dbo.collection("studentmarks").insertMany(myobj, (err, response) => {
24           if (err) throw err;
25           res.write("Inserted successfully !");
26           db.close();
27           res.end();
28         });
29       });
30     })
31     .listen(3000);
```

Inserted successfully !

4. Increase the maths marks of Mala by 6 mark?

```
1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connetion");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         var query = { name: "Mala", maths_marks: 45 };
13         var newquery = { $inc: { maths_marks: 6 } };
14
15         dbo
16           .collection("studentmarks")
17           .updateOne(query, newquery, (err, response) => {
18             if (err) throw err;
19             res.write("Marks updated");
20             db.close();
21             res.end();
22           });
23       });
24     })
25     .listen(3000);
```

Marks updated

5. List the names of students who got more than 50 marks in Maths Subject?

```
1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connection");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         var query = { maths_marks: { $gt: 50 } };
13
14         dbo
15           .collection("studentmarks")
16           .find(query, { projection: { _id: 0, name: 1 } })
17           .toArray((err, result) => {
18             if (err) throw err;
19             console.log(result);
20             db.close();
21             res.end();
22           });
23       });
24     })
25     .listen(3000);
```

```
[
  { name: 'Mala' },
  { name: 'Vanu' },
  { name: 'Aruli' },
  { name: 'Shayu' },
  { name: 'Lucky' },
  { name: 'Gva' }
]
```

6. Add a new column(field) for Average for all students.

```

1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connection");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         dbo
13           .collection("studentmarks")
14           .updateMany({}, { $set: { average: 1 } }, (err, response) => {
15             if (err) throw err;
16
17             console.log(response);
18             db.close();
19           });
20         res.write("Updated");
21         res.end();
22       });
23     })
24     .listen(3000);

```

```

{
  acknowledged: true,
  modifiedCount: 9,
  upsertedId: null,
  upsertedCount: 0,
  matchedCount: 9
}

```

7. Update Marks\_Science=75 to Lucky

```

1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connetion");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         dbo
13           .collection("studentmarks")
14           .updateOne(
15             { name: "Lucky" },
16             { $rename: { science_marks: "marks_science" } },
17             (err, response) => {
18               if (err) throw err;
19
20               console.log(response);
21               db.close();
22             }
19           );
23         res.write("Renamed and updated data");
24         res.end();
25       });
26     })
27   .listen(3000);

```

```

1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connetion");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         dbo
13           .collection("studentmarks")
14           .updateOne({ 'name': 'Lucky' }, { $set : { marks_science: 75 } }, (err, response) => {
15             if (err) throw err;
16
17             console.log(response);
18             db.close();
19           });
20         res.write("Renamed and updated data");
21         res.end();
22       });
23     })
24   .listen(3000);

```

```
{
  acknowledged: true,
  modifiedCount: 1,
  upsertedId: null,
  upsertedCount: 0,
  matchedCount: 1
}
```

8. List the names who got more than 50 marks in all subjects.

```
1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connection");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         var query = {
13           $and: [
14             { maths_marks: { $gt: 50 } },
15             { english_marks: { $gt: 50 } },
16             { science_marks: { $gt: 50 } },
17           ],
18         };
19         dbo
20           .collection("studentmarks")
21           .find(query, { projection: { _id: 0, name: 1 } })
22           .toArray((err, response) => {
23             if (err) throw err;
24             console.log(response);
25             db.close();
26           });
27       });
28       res.write("dd");
29       res.end();
30     })
31     .listen(3000);
```

```
Got a connection
[
  { name: 'Mala' },
  { name: 'Vanu' },
  { name: 'Aruli' },
  { name: 'Shayu' },
  { name: 'Gva' }
]
```

9. List the names who got less than 50 marks in Maths subject and more than 50 marks in English

```
1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connection");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         var query = {
13           $and: [
14             { maths_marks: { $lt: 50 } },
15             { english_marks: { $gt: 50 } },
16           ],
17         };
18         dbo
19           .collection("studentmarks")
20           .find(query, { projection: { _id: 0, name: 1 } })
21           .toArray((err, response) => {
22             if (err) throw err;
23             console.log(response);
24             db.close();
25           });
26       });
27       res.write("Find it !");
28       res.end();
29     })
30     .listen(3000);
```

Got a connection

```
[ { name: 'Kumaran' }, { name: 'Raam' } ]
```

10. List the names who got less than 40 in both Maths and Science.

```

1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connetion");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         var query = {
13             $and: [
14                 { maths_marks: { $lt: 40 } },
15                 { science_marks: { $lt: 40 } },
16             ],
17         };
18         dbo
19             .collection("studentmarks")
20             .find(query, { projection: { _id: 0, name: 1 } })
21             .toArray((err, response) => {
22                 if (err) throw err;
23                 console.log(response);
24                 db.close();
25             });
26     });
27     res.write("Find it !");
28     res.end();
29 })
30 .listen(3000);

```

```
Got a connetion
```

```
[]
```

11. List the names who got less than 40 in both Maths and Science.



```

1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connection");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10        if (err) throw err;
11        var dbo = db.db("student");
12        dbo
13          .collection("studentmarks")
14          .updateOne({name:"Raam"},{$unset:{science_marks:1}}, (err, response) => {
15            if (err) throw err;
16
17            console.log(response);
18            db.close();
19          });
20        res.write("Remove science column data");
21        res.end();
22      });
23    })
24    .listen(3000);

```

```

Got a connection
{
  acknowledged: true,
  modifiedCount: 1,
  upsertedId: null,
  upsertedCount: 0,
  matchedCount: 1
}

```

12. Update John's Math mark as 87 and English mark as 23, if john not available upsert.

```

1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connetion");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         var query = { $set: { maths_marks: 87, english_marks: 23 } };
13
14         dbo
15           .collection("studentmarks")
16           .updateOne(
17             { name: "John" },
18             query,
19             { upsert: true },
20             (err, response) => {
21               if (err) throw err;
22               console.log(response);
23               db.close();
24             }
25           );
26       });
27       res.write("Upsert data");
28       res.end();
29     })
30     .listen(3000);

```

```

Got a connetion
{
  acknowledged: true,
  modifiedCount: 0,
  upsertedId: new ObjectId("62c04e7dcc59af4fe0ee0c10"),
  upsertedCount: 1,
  matchedCount: 0
}
Got a connetion
{
  acknowledged: true,
  modifiedCount: 0,
  upsertedId: null,
  upsertedCount: 0,
  matchedCount: 1
}

```

13. Rename the english\_marks column/field for John to science\_marks

```

1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5    .createServer((req, res) => {
6      console.log("Got a connetion");
7      const url = "mongodb://localhost:27017/";
8
9      MongoClient.connect(url, (err, db) => {
10         if (err) throw err;
11         var dbo = db.db("student");
12         var query = { $rename: { english_marks: "science_marks" } };
13
14         dbo
15           .collection("studentmarks")
16           .updateOne({ name: "John" }, query, (err, response) => {
17             if (err) throw err;
18             console.log(response);
19             db.close();
20           });
21       });
22       res.write("Field updated");
23       res.end();
24     })
25     .listen(3000);

```

```

Got a connetion
{
  acknowledged: true,
  modifiedCount: 1,
  upsertedId: null,
  upsertedCount: 0,
  matchedCount: 1
}

```

14. Remove Kumaran's document from collection

```

1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5  ✓ .createServer((req, res) => {
6    console.log("Got a connetion");
7    const url = "mongodb://localhost:27017/";
8
9  ✓   MongoClient.connect(url, (err, db) => {
10     if (err) throw err;
11     var dbo = db.db("student");
12     var query = { name: "Kumaran" };
13  ✓   dbo.collection("studentmarks").deleteOne(query, (err, response) => {
14     if (err) throw err;
15     console.log(response);
16     db.close();
17   });
18 });
19 res.write("Data deleted!");
20 res.end();
21 })
22 .listen(3000);

```

```

15 C:\Users\ignor\Desktop\2020\Node App\js>
Got a connetion
{ acknowledged: true, deletedCount: 1 }

```

15. Find Kala's or Aruli's math\_marks and science\_marks

```

1  const http = require("http");
2  const MongoClient = require("mongodb").MongoClient;
3
4  http
5  ✓ .createServer((req, res) => {
6      console.log("Got a connetion");
7      const url = "mongodb://localhost:27017/";
8
9      try {
10     MongoClient.connect(url, (err, db) => {
11         if (err) throw err;
12         var dbo = db.db("student");
13         var query = { $or: [{ name: "Kala" }, { name: "Aruli" }] };
14
15         dbo
16             .collection("studentmarks")
17             .find(query, {
18                 projection: { maths_marks: 1, science_marks: 1, _id: 0, name: 1 },
19             })
20             .toArray((err, response) => {
21                 if (err) throw err;
22                 console.log(response);
23                 db.close();
24             });
25     });
26     } catch (error) {
27         console.log(error);
28     }
29     res.write("Find it!");
30     res.end();
31 }
32 .listen(3000);

```

Got a connetion

```

[
  { name: 'Kala', maths_marks: 32, science_marks: 53 },
  { name: 'Aruli', maths_marks: 78, science_marks: 80 }
]

```