

Index

Sr No.	Practical	Sign
1	MongoDB Basics: <ul style="list-style-type: none"> a) Write a MongoDB query to create and drop database. b) Write a MongoDB query to create, display and drop collection 	
2	Simple Queries with MongoDB	
3	Implementing Aggregation <ul style="list-style-type: none"> a) Write a MongoDB query to use sum, avg, min and max expression. b) Write a MongoDB query to use push and addToSet expression. 	
4	Replication, Backup and Restore <ul style="list-style-type: none"> a) Write a MongoDB query to create Replica of existing database. 	
5	Programs on Basic jQuery <ul style="list-style-type: none"> a) jQuery Basic, jQuery Events b) jQuery Selectors, jQuery Hide and Show effects 	
6	jQuery Advanced <ul style="list-style-type: none"> a) jQuery Animation effects, jQuery Chaining b) jQuery Callback, jQuery Get and Set Contents 	
7	JSON <ul style="list-style-type: none"> a) Creating JSON b) Parsing JSON 	

Practical 1 MongoDB Basics:

A) Write a MongoDB query to create and drop database.

To create Database use Database_name

```
test> show dbs
admin          40.00 KiB
car_details   192.00 KiB
config         72.00 KiB
local          72.00 KiB
school         40.00 KiB
test> use pract
switched to db pract
pract> -
```

To create a collection and insert a doc and display it

```
db.collectionName.insertMany([
```

```
  { key1: "value1", key2: "value2" },
  { key1: "value3", key2: "value4" },
  { key1: "value5", key2: "value6" }
```

```
])
```

```
db.collectionName.find(query, projection)
```

```
3 | pract> db.test1.insertMany([{name:"sahil",rollno:30,dept:"bscit"},{name:"munawwar",rollno:40,dept:"arch"},{name:"pratik",rollno:22,dept:"bsccs"}])
| {
| acknowledged: true,
| insertedIds: [
|   '_id': ObjectId('68bc0f214825437a4a735189'),
|   '_id': ObjectId('68bc0f214825437a4a73518a'),
|   '_id': ObjectId('68bc0f214825437a4a73518b')
| ]
| }
pract> db.test1.find()
[ {
| _id: ObjectId('68bc0f214825437a4a735189'),
| name: 'sahil',
| rollno: 30,
| dept: 'bscit'
| },
| {
| _id: ObjectId('68bc0f214825437a4a73518a'),
| name: 'munawwar',
| rollno: 40,
| dept: 'arch'
| },
| {
| _id: ObjectId('68bc0f214825437a4a73518b'),
| name: 'pratik',
| rollno: 22,
| dept: 'bsccs'
| }
| ]
```

Dropping the database

```
use myDatabase
```

```
db.dropDatabase()
```

```
pract> show dbs
admin      40.00 KiB
car_details 192.00 KiB
config     108.00 KiB
local      72.00 KiB
pract      48.00 KiB
school     40.00 KiB
pract> use pract
already on db pract
pract> db.dropDatabase()
{ ok: 1, dropped: 'pract' }
pract> show dbs
admin      40.00 KiB
car_details 192.00 KiB
config     108.00 KiB
local      72.00 KiB
school     40.00 KiB
pract>
```

B) Write a MongoDB query to create, display and drop

collection To create collection

```
db.createCollection("collection_name")
```

```
pract> db.createCollection("test2")
{ ok: 1 }
pract> db.createCollection("test3")
{ ok: 1 }
pract> db.createCollection("test4")
{ ok: 1 }
pract> show collections
```

To show collection

Show collections

```
pract> show collections
test1
test2
test3
test4
pract> -
```

Inserting data in collection

```
db.collectionName.insertMany([  
    { key1: "value1", key2: "value2" },  
    { key1: "value3", key2: "value4" },  
    { key1: "value5", key2: "value6" }  
])  
  
MongoDB shell version: 4.0.10  
connecting to: mongodb://127.0.0.1:27017/  
MongoDB server version: 4.0.10  
WARNING: No authentication specified; you may be connecting to a shared account on a multi-user system.  
try: db.test1.insertMany([{"name": "sahil", "age: 19}, {"name": "pratik", "age: 22}, {"name": "munawwar", "age: 25}, {"name": "suchit", "age: 20}])  
{  
    acknowledged: true,  
    insertedIds: {  
        '0': ObjectId('68bc8eab833ec34027735189'),  
        '1': ObjectId('68bc8eab833ec3402773518a'),  
        '2': ObjectId('68bc8eab833ec3402773518b'),  
        '3': ObjectId('68bc8eab833ec3402773518c')  
    }  
}
```

To delete a collection

```
db.collectionName.drop()
```

```
pract> db.test1.drop()  
true  
pract> db.test1.find()  
  
pract>
```

Practical 2 Simple Queries with MongoDB:

To perform Create operation like insertOne & insertMany

```
db.collectionName.insertOne(  
    { key1: "value1", key2: "value2" },  
)
```

```
db.collectionName.insertMany([  
    { key1: "value1", key2: "value2" },  
    { key1: "value3", key2: "value4" },  
    { key1: "value5", key2: "value6" }  
)
```

```

{ key1: "value3", key2: "value4" },
{ key1: "value5", key2: "value6" }
])

```

```

pract> db.student.insertOne({name:"Sahil",dept:"bscit"})
{
  acknowledged: true,
  insertedId: ObjectId('68bc9592833ec3402773518d')
}
pract> db.student.insertMany([{name:"suchit",dept:"bms"},{name:"shivam",dept:"bfm"},{name:"pankaj",dept:"baf"},{name:"thakur",dept:"bba"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68bc9624833ec3402773518e'),
    '1': ObjectId('68bc9624833ec3402773518f'),
    '2': ObjectId('68bc9624833ec34027735190'),
    '3': ObjectId('68bc9624833ec34027735191')
  }
}
pract>

```

To perform Find operations like find & findOne

db.collectionName.find(query, projection) d

b.collectionName.findOne(query, projection)

```

pract> db.student.find()
[
  {
    _id: ObjectId('68bc9592833ec3402773518d'),
    name: 'Sahil',
    dept: 'bscit'
  },
  {
    _id: ObjectId('68bc9624833ec3402773518e'),
    name: 'suchit',
    dept: 'bms'
  },
  {
    _id: ObjectId('68bc9624833ec3402773518f'),
    name: 'shivam',
    dept: 'bfm'
  },
  {
    _id: ObjectId('68bc9624833ec34027735190'),
    name: 'pankaj',
    dept: 'baf'
  },
  {
    _id: ObjectId('68bc9624833ec34027735191'),
    name: 'thakur',
    dept: 'bba'
  }
]
pract>

```

```

]
pract> db.student.findOne()
{
  _id: ObjectId('68bc9592833ec3402773518d'),
  name: 'Sahil',
  dept: 'bscit'
}
pract>

```

```

]
pract> db.student.find({dept:"bba"})
[
  {
    _id: ObjectId('68bc9624833ec34027735191'),
    name: 'thakur',
    dept: 'bba'
  }
]
pract>

```

To perform Update operations like updateOne & updateMany

```
db.collectionName.updateOne(  
  { filterField: value }, // filter (which docs to match)  
  { $set: { field: newValue } } // update action  
)
```

```
db.collectionName.updateMany(  
  { filterField: value },  
  { $set: { field: newValue } }  
)
```

A) Suppose you want to add a new key value use \$set

```
pract> db.student.updateOne({name:"Sahil"},{$set:{attendence:"80%"}})  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
pract> db.student.find({name:"Sahil"})  
[  
  {  
    _id: ObjectId('68bc9592833ec3402773518d'),  
    name: 'Sahil',  
    dept: 'bscit',  
    attendence: '80%'  
}  
]  
pract> -
```

B) Suppose you want to remove a new key value use \$unset

```
pract> db.student.updateOne({name:"Sahil"},{$unset:{attendance:""}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pract> db.student.find({name:"Sahil"})
[
  {
    _id: ObjectId('68bc9592833ec3402773518d'),
    name: 'Sahil',
    dept: 'bscit'
  }
]
pract>
```

C) Suppose you want to add a new value in array use \$push

```
pract> db.student.updateOne({name:"Sahil"},{$push:{hobbies:"football"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pract> db.student.find({name:"Sahil"})
[
  {
    _id: ObjectId('68bc9592833ec3402773518d'),
    name: 'Sahil',
    dept: 'bscit',
    hobbies: [ 'football' ]
  }
]
pract>
```

D) Suppose you want to remove a value from array use \$pull

```
pract> db.student.updateOne({name:"Sahil"},{$pull:{hobbies:"football"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pract> db.student.find({name:"Sahil"})
[
  {
    _id: ObjectId('68bc9592833ec3402773518d'),
    name: 'Sahil',
    dept: 'bscit',
    hobbies: []
  }
]
pract>
```

B) Suppose you want to add a new key value to all document

```
pract> db.student.updateMany({}, {$set:{student:"yes"})  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 5,  
  modifiedCount: 5,  
  upsertedCount: 0  
}  
pract> db.student.find()  
[  
  {  
    _id: ObjectId('68bc9592833ec3402773518d'),  
    name: 'Sahil',  
    dept: 'bscit',  
    hobbies: [ 'football' ],  
    student: 'yes'  
  },  
  {  
    _id: ObjectId('68bc9624833ec3402773518e'),  
    name: 'suchit',  
    dept: 'bms',  
    student: 'yes'  
  },  
  {  
    _id: ObjectId('68bc9624833ec3402773518f'),  
    name: 'shivam',  
    dept: 'bfm',  
    student: 'yes'  
  },  
  {  
    _id: ObjectId('68bc9624833ec34027735190'),  
    name: 'pankaj',  
    dept: 'baf',  
    student: 'yes'  
  },  
  {  
    _id: ObjectId('68bc9624833ec34027735191'),  
    name: 'thakur',  
    dept: 'bba',  
    student: 'yes'  
  }  
]  
pract>
```

To perform Delete operations like deleteOne & deleteMany

```
db.collectionName.deleteOne({ filterField: value })
```

```
db.collectionName.deleteMany({ filterField: value })
```

```
mongosininvalidinputerror. [common-10001] miss  
pract> db.student.deleteOne({name:"thakur"})  
{ acknowledged: true, deletedCount: 1 }  
pract>
```

```
mongosininvalidinputerror. [common-10001]  
pract> db.student.deleteMany({})  
{ acknowledged: true, deletedCount: 4 }  
pract> -
```

Practical 3 Implementing Aggregation:

A) Write a MongoDB query to use sum, avg, min and max expression.

Data using for this practical

```
pract> db.shop.find()
[
  {
    _id: ObjectId('68bceb6dddc2bdf30735189'),
    id: 1,
    item: 'apple',
    qty: 10,
    price: 200
  },
  {
    _id: ObjectId('68bceb6dddc2bdf3073518a'),
    id: 2,
    item: 'banana',
    qty: 12,
    price: 60
  },
  {
    _id: ObjectId('68bceb6dddc2bdf3073518b'),
    id: 3,
    item: 'watermelon',
    qty: 20,
    price: 120
  },
  {
    _id: ObjectId('68bceba4ddc2bdf3073518c'),
    id: 5,
    item: 'kiwi',
    qty: 5,
    price: 260
  }
]
```

using sum, avg, min and max expression

```
pract> db.shop.aggregate([{$group:{_id:null,totalQuantity:{$sum:"$qty"},averagePrice:{$avg:"$price"},minPrice:{$min:"$price"},maxPrice:{$max:"$price"}}}])
[
  {
    _id: null,
    totalQuantity: 47,
    averagePrice: 160,
    minPrice: 60,
    maxPrice: 260
  }
]
pract>
```

B) Write a MongoDB query to use push and addToSet expression.

Data using for push expression

```
mongoserver1:~ [location40234] The field `customer` must be an accumulator object
pract> db.shopping.aggregate([{$group:{_id:"$customer",allitems:{$push:"$item"}}}])
[
  { _id: 'shivam', allitems: [ 'apple' ] },
  { _id: 'munawwar', allitems: [ 'orange' ] },
  { _id: 'suchit', allitems: [ 'banana' ] },
  { _id: 'sahil', allitems: [ 'apple' ] },
  { _id: 'harsh', allitems: [ 'watermelon' ] }
]
```

```
pract> db.shopping.find()
[
  {
    _id: ObjectId('68bcf6b5ddc2bdfe3073518d'),
    id: 1,
    customer: 'sahil',
    item: 'apple',
    qty: 5,
    price: 150
  },
  {
    _id: ObjectId('68bcf6b5ddc2bdfe3073518e'),
    id: 2,
    customer: 'suchit',
    item: 'banana',
    qty: 12,
    price: 60
  },
  {
    _id: ObjectId('68bcf6b5ddc2bdfe3073518f'),
    id: 3,
    customer: 'shivam',
    item: 'apple',
    qty: 10,
    price: 190
  },
  {
    _id: ObjectId('68bcf6b5ddc2bdfe30735190'),
    id: 4,
    customer: 'harsh',
    item: 'watermelon',
    qty: 12,
    price: 190
  },
  {
    _id: ObjectId('68bcf6b5ddc2bdfe30735191'),
    id: 5,
    customer: 'munawwar',
    item: 'orange',
    qty: 3,
    price: 10
  }
]
```

using addToSet expression

```
pract> db.shopping.find()
[
  {
    _id: ObjectId('68bd0e40ddc2bdfe30735192'),
    customer: 'sarthak',
    item: 'dragon'
  },
  {
    _id: ObjectId('68bd0e40ddc2bdfe30735193'),
    customer: 'Shyam',
    item: 'papaya'
  },
  {
    _id: ObjectId('68bd0e40ddc2bdfe30735194'),
    customer: 'Shyam',
    item: 'banana'
  },
  {
    _id: ObjectId('68bd0e40ddc2bdfe30735195'),
    customer: 'Ram',
    item: 'banana'
  },
  {
    _id: ObjectId('68bd0e40ddc2bdfe30735196'),
    customer: 'Ram',
    item: 'apple'
  },
  {
    _id: ObjectId('68bd0e40ddc2bdfe30735197'),
    customer: 'radhe',
    item: 'pomegranate'
  },
  {
    _id: ObjectId('68bd0e95ddc2bdfe30735198'),
    customer: 'sahil',
    item: 'apple'
  }
]
pract> db.shopping.aggregate({$group:{_id:"$customer",uniqueItems:{$addToSet:"$item"}}})
[
  { _id: 'sahil', uniqueItems: [ 'apple' ] },
  { _id: 'sarthak', uniqueItems: [ 'dragon' ] },
  { _id: 'radhe', uniqueItems: [ 'pomegranate' ] },
  { _id: 'Shyam', uniqueItems: [ 'papaya', 'banana' ] },
  { _id: 'Ram', uniqueItems: [ 'banana', 'apple' ] }
]
pract>
```

Practical 4 Replication:

A) Write a MongoDB query to create Replica of existing database.

Data for replica

```
studentTest> db.st.find().pretty()
[
  {
    _id: ObjectId('68bd1161ddc2bdfe30735199'),
    name: 'sahil',
    age: 19,
    course: 'bscit',
    subjects: [ 'Ai', 'AWP', 'NGT' ]
  },
  {
    _id: ObjectId('68bd1161ddc2bdfe3073519a'),
    name: 'munawwar',
    age: 21,
    course: 'cs',
    subjects: [ 'Ai', 'AWP', 'NGT' ]
  },
  {
    _id: ObjectId('68bd1161ddc2bdfe3073519b'),
    name: 'salman',
    age: 24,
    course: 'aviation',
    subjects: [ 'atc', 'arch', 'maths' ]
  },
  {
    _id: ObjectId('68bd1161ddc2bdfe3073519c'),
    name: 'sufi',
    age: 33,
    course: 'bsc',
    subjects: [ 'Ai', 'AWP', 'NGT' ]
  }
]
```

creating copy of current db giving its name and collection name

```
studentTest> db.st.aggregate([{$match:{}},{$out:{db:"studentTestCopy",coll:"stCopy"}}])
```

Copy of db create with collection

```
studentTest      40.00 KiB
studentTestCopy   8.00 KiB
```

```
studentTestCopy> db.stCopy.find()
[stCopy
  {
    _id: ObjectId('68bd1161ddc2bdfe30735199'),
    name: 'sahil',
    age: 19,
    course: 'bscit',
    subjects: [ 'Ai', 'AWP', 'NGT' ]
  },
  {
    _id: ObjectId('68bd1161ddc2bdfe3073519a'),
    name: 'munawwar',
    age: 21,
    course: 'cs',
    subjects: [ 'Ai', 'AWP', 'NGT' ]
  },
  {
    _id: ObjectId('68bd1161ddc2bdfe3073519b'),
    name: 'salman',
    age: 24,
    course: 'aviation',
    subjects: [ 'atc', 'arch', 'maths' ]
  },
  {
    _id: ObjectId('68bd1161ddc2bdfe3073519c'),
    name: 'sufi',
    age: 33,
    course: 'bsc',
    subjects: [ 'Ai', 'AWP', 'NGT' ]
  }
]
studentTestCopy> ■
```

Practical 5 Programs on Basic jQuery:

A) jQuery Basic, jQuery Events

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>basic j query</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <h1>hello </h1>
    <button id="bt1">click me</button>
    <script>
        $(document).ready(function(){
            $("#bt1").click(function(){
                $("h1").text("welcome to our page");
            });
        });
    </script>
</body>
</html>
```

Output:

hello

welcome to our page

[click me](#)

[click me](#)

Code:

```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>events</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <p id="para">click or hover me</p>
    <script>
        $(document).ready(function(){
            $("#para").click(function(){
                $(this).css("color","green");
            });
            $("#para").mouseenter(function(){
                $(this).css("background","red");
            });
            $("#para").mouseleave(function(){
                $(this).css("background","white");
            });
        });
    </script>
```

```
</body>
```

```
</html>
```

Output:

click or hover me

B) jQuery Selectors, jQuery Hide and Show effects **Code:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>selector</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
  <p class="cl1">this is class selector</p>
  <p id="id1">this is id selector</p>
  <p> this is tag selector</p>
  <script>
    $(document).ready(function(){
      $(".cl1").css("color","red");
      $("#id1").css("font-size","20px");
      $("p").css("font-weight","bold");
    });
  </script>
</body>
</html>
```

Output:

this is class selector

this is id selector

this is tag selector

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>hind and show</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <p id="text1">this is a para</p>
    <button id="hide">hide button</button>
    <button id="show">show button</button>
    <script>
        $(document).ready(function(){
            $("#hide").click(function(){
                $("#text1").hide();
            });
            $("#show").click(function(){
                $("#text1").show();
            });
        });
    </script>
```

```
</script>  
</body>  
</html>
```

Output:

this is a para

Practical 6 jQuery Advanced:

Code:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>animation</title>  
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  
    <style>  
        #box{ height: 100px;  
              width: 100px;  
              background-color: green;  
              position: relative;  
        }  
    </style>  
</head>  
<body>  
    <button id="bt">click here for animation</button>  
    <div id="box"></div>
```

```
<script>
$(document).ready(function(){
  $("#bt").click(function(){
    $("#box").animate({
      left:'250px',
      height:'150px',
      width:'300px',
      opacity:'0.1'
    },1000);
  });
});
</script>
</body>
</html>
```

Output:



Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device
    <title>channing</title>
    <p id="text">watch me chnage</p>
    <button id="chain">click me</button>

<script>
$(document).ready(function(){
    $("#chain").click(function(){
        $("#text").css("color","red").slideUp(1000).slideDown(1000).fadeOut(1000).fadeIn(1000);
    });
});
</script>
</script>
</body>
</html>
```

Output:

watch me chnage

click me

watch me chnage

click me

Code:

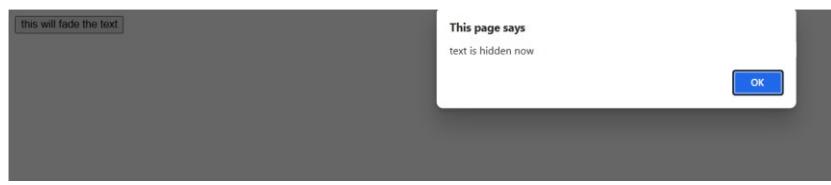
```
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device
<title>jquery callback</title>
<button id="bt">this will fade the text</button>
<p id="text">this will fade out first and then show alert</p>
<script>
$(document).ready(function(){
    $("#bt").click(function(){
        $("#text").fadeOut(1000,function(){
            alert("text is hidden now");
        });
    });
});
</script>
</body>
</html>
```

Output:

this will fade the text

this will fade out first and then show alert



Code:

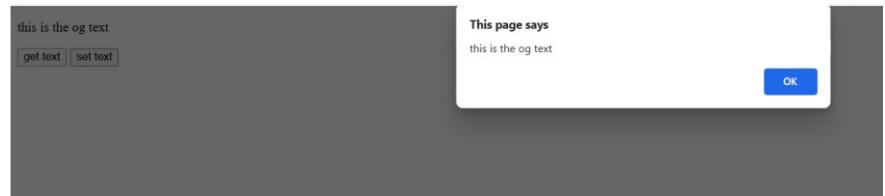
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>set and get content</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <p id="para">this is the og text</p>
    <button id="btget">get text</button>
    <button id="btset">set text</button>
    <script>
        $(document).ready(function(){
            $("#btget").click(function(){
                alert($("#para").text());
            });
            $("#btset").click(function(){
                $("#para").text("this is the new text");
            });
        });
    </script>
</body>
</html>
```

Output:

this is the og text

[get text](#)

[set text](#)



this is the new text

[get text](#)

[set text](#)

Practical 7 JSON:

A)Creating JSON

```
studentTest> db.st.insertMany([{"name": "sahil", "age": 19, "course": "bscIT", "subjects": ["Ai", "AWP", "NGT"]}, {"name": "munawwar", "age": 21, "course": "cs", "subjects": ["Ai", "AWP", "NGT"]}, {"name": "salman", "age": 24, "course": "aviation", "subjects": ["atc", "arch", "maths"]}, {"name": "sufi", "age": 33, "course": "bsc", "subjects": ["Ai", "AWP", "NGT"]}])  
{  
  acknowledged: true,  
  insertedIds: [  
    '0': ObjectId('68bd1161ddc2bdf3e0735199'),  
    '1': ObjectId('68bd1161ddc2bdf3e073519a'),  
    '2': ObjectId('68bd1161ddc2bdf3e073519b'),  
    '3': ObjectId('68bd1161ddc2bdf3e073519c')  
}
```

B)Parsing JSON

```
studentTest> db.st.find().pretty()  
[  
  {  
    _id: ObjectId('68bd1161ddc2bdf3e0735199'),  
    name: 'sahil',  
    age: 19,  
    course: 'bscIT',  
    subjects: [ 'Ai', 'AWP', 'NGT' ]  
  },  
  {  
    _id: ObjectId('68bd1161ddc2bdf3e073519a'),  
    name: 'munawwar',  
    age: 21,  
    course: 'cs',  
    subjects: [ 'Ai', 'AWP', 'NGT' ]  
  },  
  {  
    _id: ObjectId('68bd1161ddc2bdf3e073519b'),  
    name: 'salman',  
    age: 24,  
    course: 'aviation',  
    subjects: [ 'atc', 'arch', 'maths' ]  
  },  
  {  
    _id: ObjectId('68bd1161ddc2bdf3e073519c'),  
    name: 'sufi',  
    age: 33,  
    course: 'bsc',  
    subjects: [ 'Ai', 'AWP', 'NGT' ]  
}
```

```
switched to db: studentTest  
studentTest> db.st.find({name: "sahil"})  
[  
  {  
    _id: ObjectId('68bd1161ddc2bdf3e0735199'),  
    name: 'sahil',  
    age: 19,  
    course: 'bscIT',  
    subjects: [ 'Ai', 'AWP', 'NGT' ]  
}  
]  
studentTest>  
  
studentTest> db.st.find({age: {$lt: 20}})  
[  
  {  
    _id: ObjectId('68bd1161ddc2bdf3e0735199'),  
    name: 'sahil',  
    age: 19,  
    course: 'bscIT',  
    subjects: [ 'Ai', 'AWP', 'NGT' ]  
}  
]  
studentTest>
```

```
studentTest> db.st.find({subjects: "Ai"})  
[  
  {  
    _id: ObjectId('68bd1161ddc2bdf3e0735199'),  
    name: 'sahil',  
    age: 19,  
    course: 'bscIT',  
    subjects: [ 'Ai', 'AWP', 'NGT' ]  
  },  
  {  
    _id: ObjectId('68bd1161ddc2bdf3e073519a'),  
    name: 'munawwar',  
    age: 21,  
    course: 'cs',  
    subjects: [ 'Ai', 'AWP', 'NGT' ]  
  },  
  {  
    _id: ObjectId('68bd1161ddc2bdf3e073519c'),  
    name: 'sufi',  
    age: 33,  
    course: 'bsc',  
    subjects: [ 'Ai', 'AWP', 'NGT' ]  
  }  
]  
studentTest>
```