
MONGODB PROJECT

BASIC SECTION

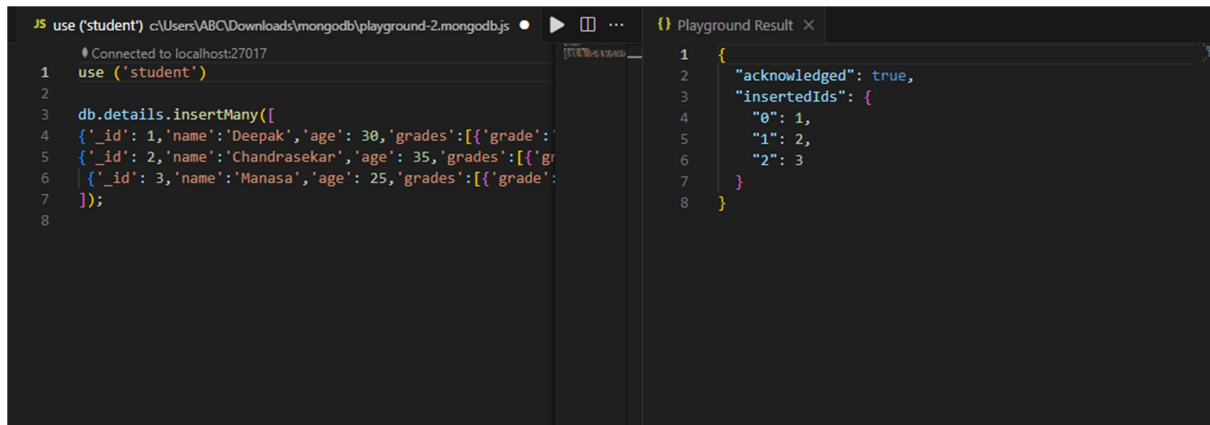
How to use \$updateOne AND updateMany operator in MongoDB

dataset

```
_id:1,
'name': "Manasa",
age: 25,
grades: [
{ grade: "Maths", score: 80 },
{ grade: "Science", score: 85 }
]
_id: 2,
'name': "Chandrasekar",
age: 35,
grades: [
{ grade: "Maths", score: 95 },
{ grade: "Science", score: 88 }
]
_id: 3,
'name': "Deepak",
age: 30,
grades: [
{ grade: "Maths", score: 85 },
{ grade: "Science", score: 90 }
]
```

```
>use('student')
>db.scorce.insertMany([
{
'_id': 1,
'name':'Deepak',
'age': 30,
'grades':[{ 'grade':'Maths','score': 85 },{'grade':'Science','score': 90 }]},
{
'_id': 2,
'name':'Chandrasekar',
```

```
'age': 35,
'grades':[{ 'grade':'Maths','score': 95 },{'grade':'Science','score': 88}],
{
'_id': 3,
'name':'Manasa',
'age': 25,
'grades':[{ 'grade':'Maths','score': 80 },{'grade':'Science','score': 85}]}
]);
```



The screenshot shows the MongoDB Playground interface. On the left, the command window contains the following JavaScript code:

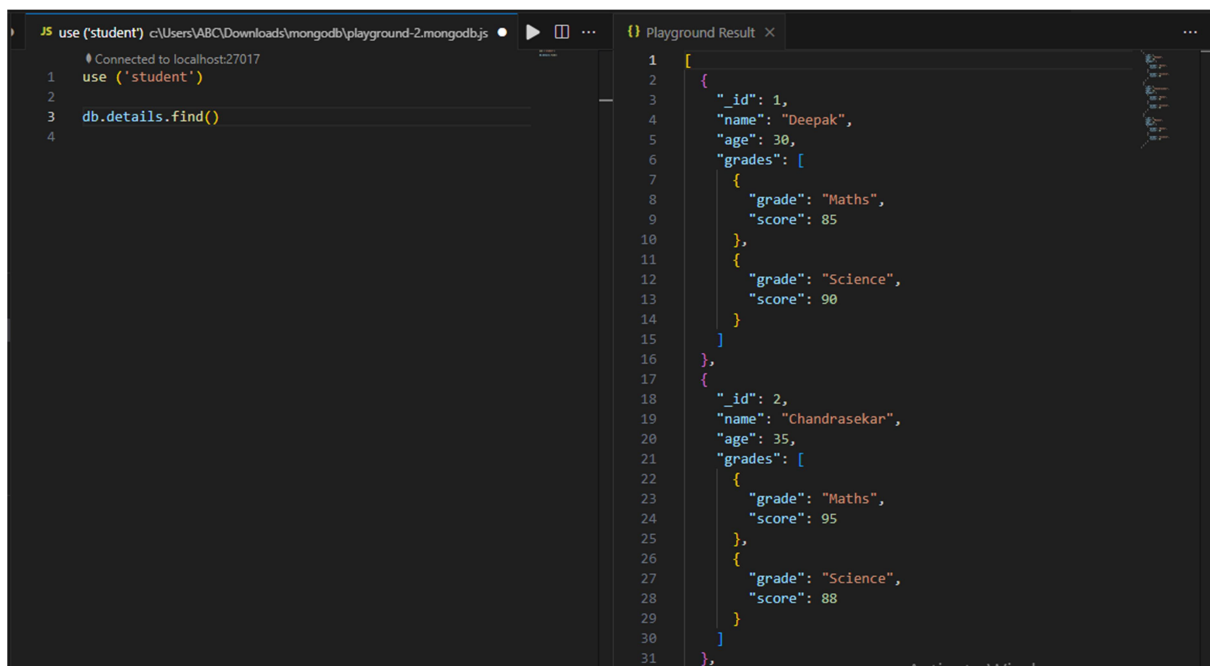
```
1 use ('student')
2
3 db.details.insertMany([
4   { '_id': 1, 'name': 'Deepak', 'age': 30, 'grades': [{ 'grade': 'Maths', 'score': 85 }, { 'grade': 'Science', 'score': 90 } ] },
5   { '_id': 2, 'name': 'Chandrasekar', 'age': 35, 'grades': [{ 'grade': 'Maths', 'score': 95 }, { 'grade': 'Science', 'score': 88 } ] },
6   { '_id': 3, 'name': 'Manasa', 'age': 25, 'grades': [{ 'grade': 'Maths', 'score': 80 }, { 'grade': 'Science', 'score': 85 } ] }
7 ]);
```

On the right, the 'Playground Result' window shows the output of the insertMany operation:

```
1 {
2   "acknowledged": true,
3   "insertedIds": {
4     "0": 1,
5     "1": 2,
6     "2": 3
7   }
8 }
```

Verifying operation

db.score.find()



The screenshot shows the MongoDB Playground interface. On the left, the command window contains the following JavaScript code:

```
1 use ('student')
2
3 db.details.find()
```

On the right, the 'Playground Result' window shows the output of the find operation:

```
1 [
2   {
3     "_id": 1,
4     "name": "Deepak",
5     "age": 30,
6     "grades": [
7       {
8         "grade": "Maths",
9         "score": 85
10      },
11      {
12        "grade": "Science",
13        "score": 90
14      }
15    ]
16  },
17  {
18    "_id": 2,
19    "name": "Chandrasekar",
20    "age": 35,
21    "grades": [
22      {
23        "grade": "Maths",
24        "score": 95
25      },
26      {
27        "grade": "Science",
28        "score": 88
29      }
30    ]
31  },
32  {
33    "_id": 3,
34    "name": "Manasa",
35    "age": 25,
36    "grades": [
37      {
38        "grade": "Maths",
39        "score": 80
40      },
41      {
42        "grade": "Science",
43        "score": 85
44      }
45    ]
46  }
47 ]
```

```
JS use ('student') c:\Users\ABC\Downloads\mongodb\playground-2.mongodb.js
1 use ('student')
2
3 db.details.find()
4

Playground Result X
17 {
31 },
32 {
33   "_id": 3,
34   "name": "Manasa",
35   "age": 25,
36   "grades": [
37     {
38       "grade": "Maths",
39       "score": 80
40     },
41     {
42       "grade": "Science",
43       "score": 85
44     }
45   ]
46 }
47 ]
```

1) Query To Update Age =26 Whoose Name Is Manasa

```
db.score.updateOne(
  { name: "Manasa" },
  { $set: { age: 36 } }
)
```

```
JS use ('student') c:\Users\ABC\Downloads\mongodb\playground-2.mongodb.js
1 use ('student')
2
3 db.details.updateOne(
4   { name: "Manasa" },
5   { $set: { age: 36 } }
6 )
7
8
9

Playground Result X
1 {
2   "acknowledged": true,
3   "insertedId": null,
4   "matchedCount": 1,
5   "modifiedCount": 1,
6   "upsertedCount": 0
7 }
```

1.1) verifying the result

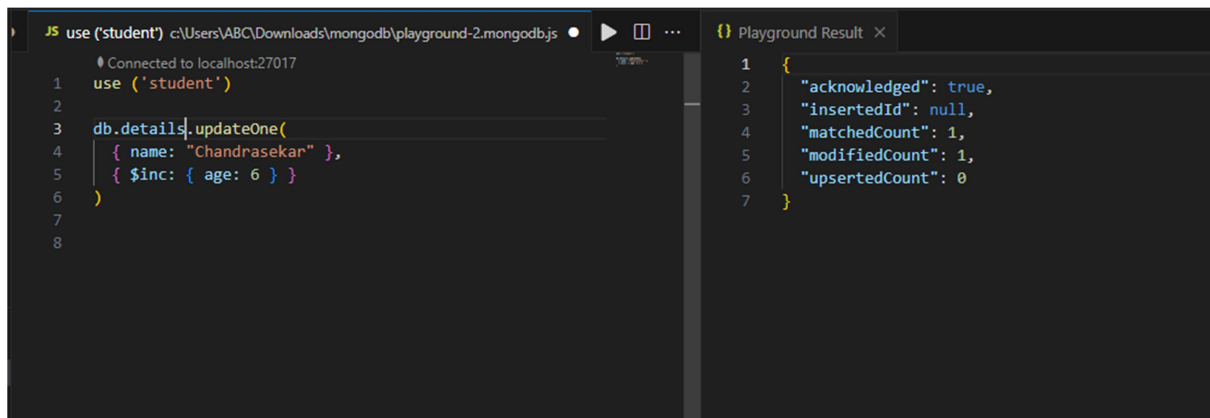
```
db.score.find({_id:3})
```

```
JS use ('student') c:\Users\ABC\Downloads\mongodb\playground-2.mongodb.js
1 use ('student')
2
3 db.details.find({_id:3})
4
5

Playground Result X
1 [
2   {
3     "_id": 3,
4     "name": "Manasa",
5     "age": 36,
6     "grades": [
7       {
8         "grade": "Maths",
9         "score": 80
10      },
11      {
12        "grade": "Science",
13        "score": 85
14      }
15    ]
16  }
17 ]
```

2) Query To Increment Age By 5 Whose Name Is Chandrasekar

```
db.score.updateOne(
  { name: "Chandrasekar" },
  { $inc: { age: 6 } }
)
```



The screenshot shows the MongoDB Playground interface. On the left, the code editor contains the following JavaScript code:

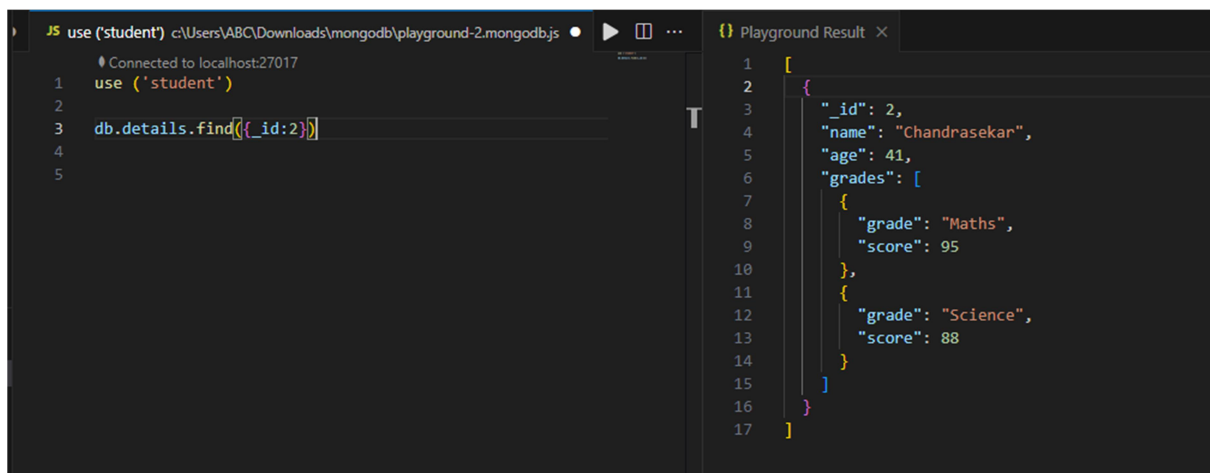
```
1 use ('student')
2
3 db.details.updateOne(
4   { name: "Chandrasekar" },
5   { $inc: { age: 6 } }
6 )
```

On the right, the 'Playground Result' tab shows the result of the update operation:

```
1 {
2   "acknowledged": true,
3   "insertedId": null,
4   "matchedCount": 1,
5   "modifiedCount": 1,
6   "upsertedCount": 0
7 }
```

2.1) verifying the result

```
db.score.find({_id:2})
```



The screenshot shows the MongoDB Playground interface. On the left, the code editor contains the following JavaScript code:

```
1 use ('student')
2
3 db.details.find({_id:2})
4
5
```

On the right, the 'Playground Result' tab shows the result of the find operation:

```
1 [
2   {
3     "_id": 2,
4     "name": "Chandrasekar",
5     "age": 41,
6     "grades": [
7       {
8         "grade": "Maths",
9         "score": 95
10      },
11      {
12        "grade": "Science",
13        "score": 88
14      }
15    ]
16   }
17 ]
```

3) Demonstrate Example of Upsert in MongoDB

```
db.score.updateOne(
  { name: "Anjali" },
  { $set: { age: 25, grades: [] } },
  { upsert: true }
)
```

```
JS use('student') c:\Users\ABC\Downloads\mongodb\playground-2.mongodb.js
Connected to localhost:27017
1 use('student')
2
3 db.details.updateOne(
4   { name: "Anjali" },
5   { $set: { age: 25, grades: [] } },
6   { upsert: true }
7 )
8

Playground Result X
1 {
2   "acknowledged": true,
3   "insertedId": {
4     "$oid": "68387a1f11443bc75ec3564c"
5   },
6   "matchedCount": 0,
7   "modifiedCount": 0,
8   "upsertedCount": 1
9 }
```

3.1)verifying output

db.score.find({age:25})

```
JS use('student') c:\Users\ABC\Downloads\mongodb\playground-2.mongodb.js
Connected to localhost:27017
1 use('student')
2
3 db.details.find({age:25})
4

Playground Result X
1 [
2   {
3     "_id": {
4       "$oid": "68387a1f11443bc75ec3564c"
5     },
6     "name": "Anjali",
7     "age": 25,
8     "grades": []
9   }
10 ]
```

4) Query Update Maths Score to 95 Using \$[Elem] And Array Filters Of Student Name Deepak

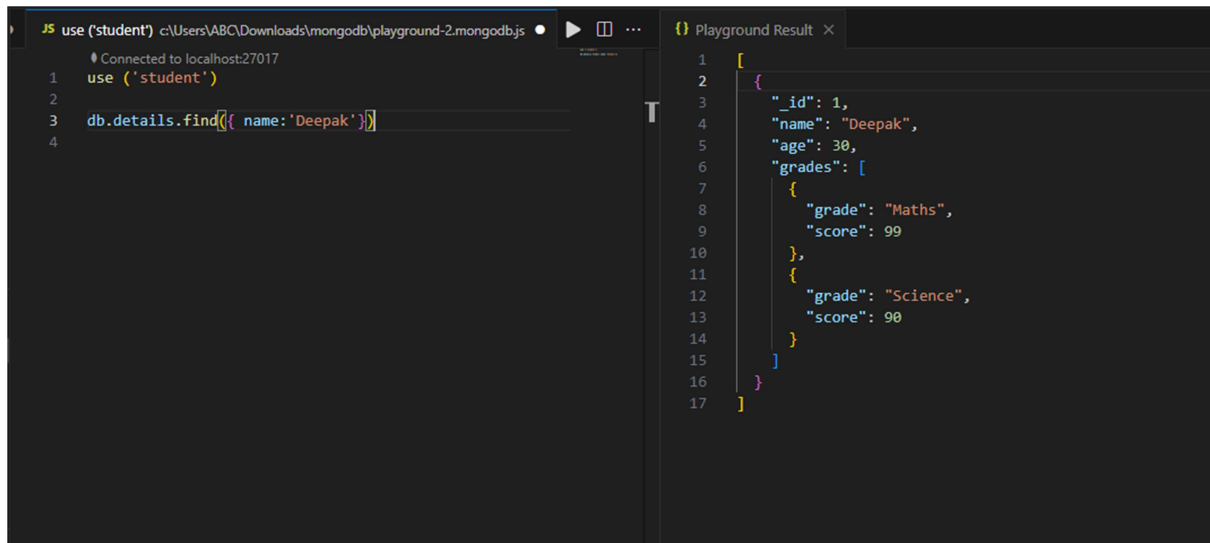
```
db.score.updateOne(
  { name: "Deepak" },
  { $set: { "grades.$[elem].score": 99 } },
  { arrayFilters: [ { "elem.grade": "Maths" } ] }
)
```

```
JS use('student') c:\Users\ABC\Downloads\mongodb\playground-2.mongodb.js
Connected to localhost:27017
1 use('student')
2
3 db.details.updateOne(
4   { name: "Deepak" },
5   { $set: { "grades.$[elem].score": 99 } },
6   { arrayFilters: [ { "elem.grade": "Maths" } ] }
7 )
8

Playground Result X
1 {
2   "acknowledged": true,
3   "insertedId": null,
4   "matchedCount": 1,
5   "modifiedCount": 1,
6   "upsertedCount": 0
7 }
```

4.1)verifying output

db.score.find({ name:'Deepak'})



The screenshot displays the MongoDB Playground interface. On the left, the code editor shows a JavaScript snippet for connecting to a local MongoDB instance and running a query. The query is `db.details.find({ name:'Deepak'})`. On the right, the 'Playground Result' pane shows the output of the query, which is a JSON array containing one document. The document has the following structure: `{ "_id": 1, "name": "Deepak", "age": 30, "grades": [{ "grade": "Maths", "score": 99 }, { "grade": "Science", "score": 90 }] }`.

```
JS use ('student') c:\Users\ABC\Downloads\mongodb\playground-2.mongodbs.js
1 use ('student')
2
3 db.details.find({ name:'Deepak'})
4
```

```
1 [
2   {
3     "_id": 1,
4     "name": "Deepak",
5     "age": 30,
6     "grades": [
7       {
8         "grade": "Maths",
9         "score": 99
10      },
11      {
12        "grade": "Science",
13        "score": 90
14      }
15    ]
16  }
17 ]
```