

Modifiers and operators cont'd

Insert values into Array

Lets insert the multiple values into an array. The arrays are represented in []. For example an array of marks can be [45, 44, 43].

The values in array can be inserted in [] using simple **db.collectionname.insert()** method.

Example:

C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe

```
> db.result.find()
>
> db.result.insert({"name": "amit", "marks":[45, 48, 41]})
WriteResult({"nInserted" : 1 })
>
> db.result.insert({"name": "indu", "marks":[42, 44]})
WriteResult({"nInserted" : 1 })
>
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 45, 48, 41 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44 ] }
>
```

Examples of Update Operator & Modifiers

1. **\$each modifier:** Let update the Marks array and add elements to it. This is being done using **\$each** modifier and **\$push** operator. With this all the elements specified in the update, will be appended to the array.
In the following example, 3 elements has been appended to the marks where name is indu.

C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe

```
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 45, 48, 41 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44 ] }
>
> db.result.update({"name":"indu"}, {$push: {"marks":{"$each: [43,47,51]}}})
WriteResult({"nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 45, 48, 41 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44, 43, 47, 51 ] }
>
```

2. **\$addToSet operator with \$each modifier:** Let update the Marks array and add elements to it. This is being done using **\$each** modifier and **\$addToSet** operator. With this only elements specified in the update, will be appended to the array, if they do not exist in the array, if for existing elements, do nothing.

In the below example, we are adding 3 elements to the array marks only if the specified elements are not existing in the array for amit.

```
C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 45, 48, 41 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44, 43, 47, 51 ] }
>
> db.result.update({"name":"amit"}, {$addToSet: {"marks":{$each: [42,41,48]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 45, 48, 41, 42 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44, 43, 47, 51 ] }
>
```

Here, only one element from the update command out of 3, has been inserted as remaining 2 elements, 41 and 48 already exists in the array Marks.

3. \$slice and \$sort modifier:

We may use sort and slice modifier with Update also. Sort may sort the specified array in ascending or descending order and slice will limit the given no of elements in the array. In the following example we are appending some elements to marks array, then sorting it in descending order and slicing top 5 records in the collection.

```
C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 45, 48, 41 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44, 43, 47, 51 ] }
>
> db.result.update({name:"amit"},{$push:{marks:{$each:[40,78,89],$sort:-1,$slice:5}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 89, 78, 48, 45, 41 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44, 43, 47, 51 ] }
>
```

4. \$position modifier:

Using the **\$position** modifier, we may specify the element location in the array at which the **\$push** operator inserts elements. **\$position** modifier is always used with **\$each** modifier.

In the following example, we are adding elements at the position 2 in the array marks for indu.

```

C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 89, 78, 48, 45, 41 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44, 43, 47, 51 ] }
>
> db.result.update({name:"indu"},{$push:{marks:{$each:[91,79],$position:2}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 89, 78, 48, 45, 41 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44, 91, 79, 43, 47, 51 ] }
>

```

If we specify a negative number in \$position, it will add the element in the array counting the **\$position** from the last element of the array. Example:

```

C:\Program Files\MongoDB\Server\4.2\bin\mongo.exe
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 89, 78, 48, 45, 41 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44, 91, 79, 43, 47, 51 ] }
>
> db.result.update({name:"indu"},{$push:{marks:{$each:[77, 78],$position:-2}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
> db.result.find()
{ "_id" : ObjectId("5eb8e6c6c92747cab0598379"), "name" : "amit", "marks" : [ 89, 78, 48, 45, 41 ] }
{ "_id" : ObjectId("5eb8e6d0c92747cab059837a"), "name" : "indu", "marks" : [ 42, 44, 91, 79, 43, 77, 78, 47, 51 ] }
>

```

Assignment

- 1.What is the use of position modifier explain with example?
2. Explain usage of Sort, Slice, each modifiers with example.