MongoDB Update() Document with Example

Details

Last Updated: 03 January 2020

Basic document updates

MongoDB provides the update() command to update the documents of a collection. To update only the documents you want to update, you can add a criteria to the update statement so that only selected documents are updated.

The basic parameters in the command is a condition for which document needs to be updated, and the next is the modification which needs to be performed.

The following example shows how this can be done.

Step 1) Issue the update command

Step 2) Choose the condition which you want to use to decide which document needs to be updated. In our example, we want to update the document which has the Employee id 22.

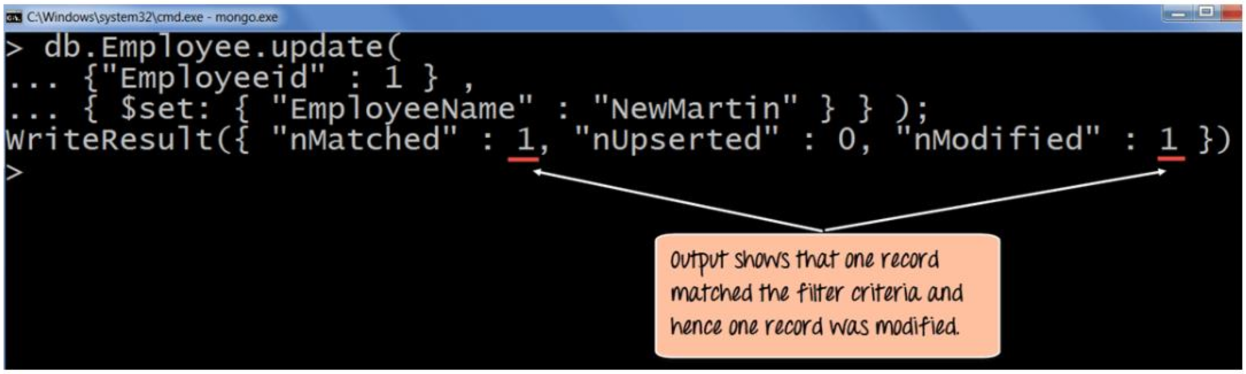
Step 3) Use the set command to modify the Field Name

Step 4) Choose which Field Name you want to modify and enter the new value accordingly.

```
db.Employee.update(
{"Employeeid" : 1},
{$set: { "EmployeeName" : "NewMartin"}});
```

If the command is executed successfully, the following Output will be shown

Output:



```
> db.Employee.update(
... {"Employeeid" : 1 } ,
... { $set: { "EmployeeName" : "NewMartin" } } );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

Output shows that one record matched the filter criteria and hence one record was modified.

The output clearly shows that one record matched the condition and hence the relevant field value was modified.

Updating Multiple Values

To ensure that multiple/bulk documents are updated at the same time in MongoDB you need to use the multi option because otherwise by default only one document is modified at a time.

The following example shows how to update many documents.

In this example, we are going to first find the document which has the Employee id as "1" and change the Employee name from "Martin" to "NewMartin"

Step 1) Issue the update command

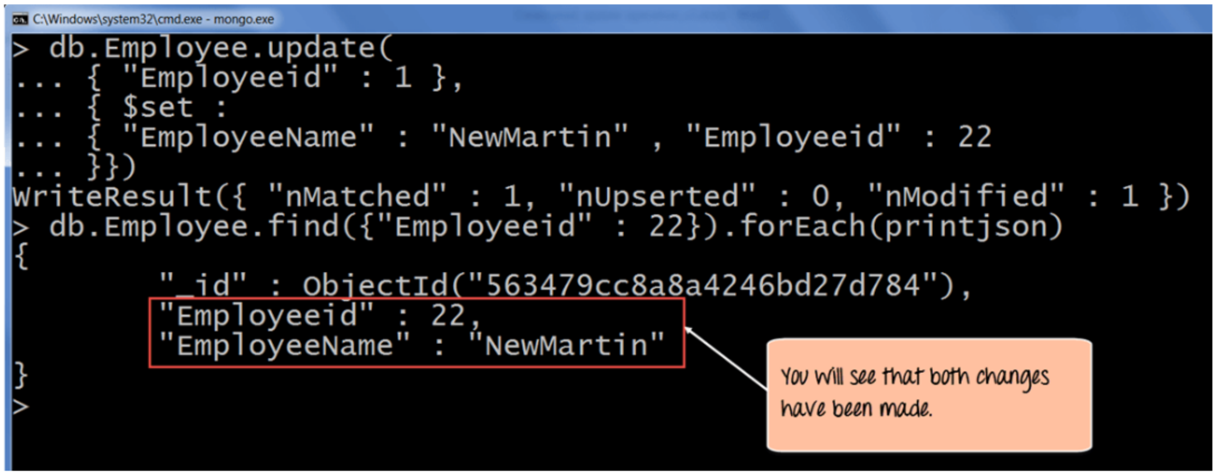
Step 2) Choose the condition which you want to use to decide which document needs to be updated. In our example, we want the document which has the Employee id of "1" to be updated.

Step 3) Choose which Field Name's you want to modify and enter their new value accordingly.

```
db.Employee.update
(
        {
                Employeeid : 1
        },
        {
                $set :
                {
                        "EmployeeName" : "NewMartin"
                        "Employeeid" : 22
                }
        }
)
```

If the command is executed successfully and if you run the "**find**" command to search for the document with Employee id as 22 you will see the following Output will be shown

Output:



```
> db.Employee.update(
... { "Employeeid" : 1 },
... { $set :
... { "EmployeeName" : "NewMartin" , "Employeeid" : 22
... }})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Employee.find({"Employeeid" : 22}).forEach(printjson)
{
        "_id" : ObjectId("563479cc8a8a4246bd27d784"),
        "Employeeid" : 22,
        "EmployeeName" : "NewMartin"
}
>
```

You will see that both changes have been made.

The output clearly shows that one record matched the condition and hence the relevant field value was modified.