MongoDB – Update Document in a Collection

In MongoDB, we have two ways to update a document in a collection. 1) update() method 2) save() method. Although both the methods update an existing document, they are being used in different scenarios. The update() method is used when we need to update the values of an existing document while save() method is used to replace the existing document with the document that has been passed in it.

To update a document in MongoDB, we provide a criteria in command and the document that matches that criteria is updated. To understand how criteria works in the MongoDB commands, refer the last tutorial: **MongoDB Query Document**

Updating Document using update() method

Syntax:

```
db.collection_name.update(criteria, update_data)
```

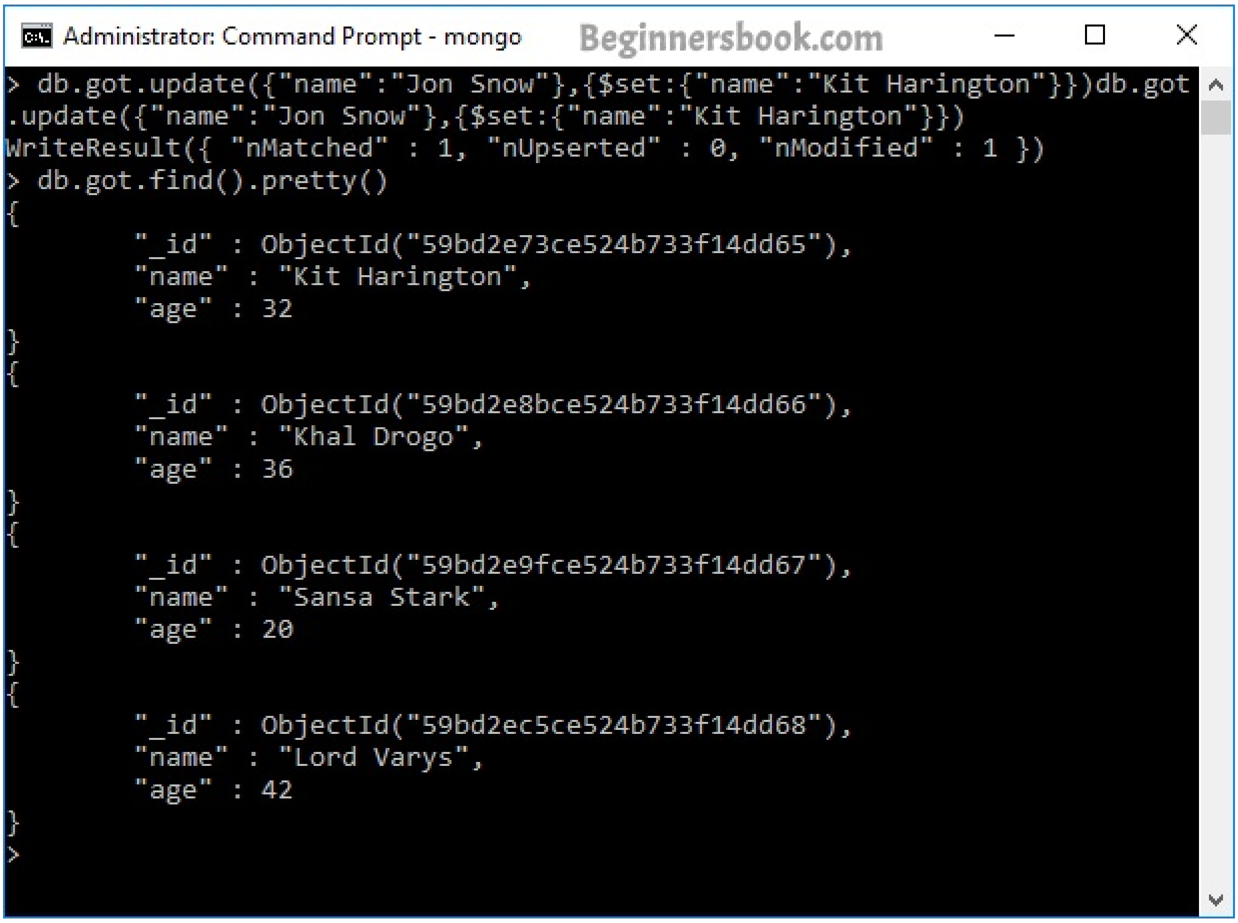
Example:
For example: Lets say I have a collection named “got” in the database “beginnersbookdb”. The documents inside “got” are:

```
> db.got.find().pretty()
{
        "_id" : ObjectId("59bd2e73ce524b733f14dd65"),
        "name" : "Jon Snow",
        "age" : 32
}
{
        "_id" : ObjectId("59bd2e8bce524b733f14dd66"),
        "name" : "Khal Drogo",
        "age" : 36
}
{
        "_id" : ObjectId("59bd2e9fce524b733f14dd67"),
        "name" : "Sansa Stark",
        "age" : 20
}
{
        "_id" : ObjectId("59bd2ec5ce524b733f14dd68"),
        "name" : "Lord Varys",
        "age" : 42
}
```

Now suppose if I want to update the name of Jon Snow with the name “Kit Harington”. The command for this would be:

```
db.got.update({"name":"Jon Snow"},{$set:{"name":"Kit Harington"}})
```

Output:



```
> db.got.update({"name":"Jon Snow"},{$set:{"name":"Kit Harington"}})db.got
.update({"name":"Jon Snow"},{$set:{"name":"Kit Harington"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.got.find().pretty()
{
        "_id" : ObjectId("59bd2e73ce524b733f14dd65"),
        "name" : "Kit Harington",
        "age" : 32
}
{
        "_id" : ObjectId("59bd2e8bce524b733f14dd66"),
        "name" : "Khal Drogo",
        "age" : 36
}
{
        "_id" : ObjectId("59bd2e9fce524b733f14dd67"),
        "name" : "Sansa Stark",
        "age" : 20
}
{
        "_id" : ObjectId("59bd2ec5ce524b733f14dd68"),
        "name" : "Lord Varys",
        "age" : 42
}
>
```

As you can see that the document has been updated.

> **Do you know?** By default the update method updates a single document. In the above example we had only one document matching with the criteria, however if there were more then also only one document would have been updated. To enable update() method to update multiple documents you have to set “multi” parameter of this method to true as shown below.

To update multiple documents with the update() method:

```
db.got.update({"name":"Jon Snow"},
 {$set:{"name":"Kit Harington"}},{multi:true})
```

Updating Document using save() method

Syntax:

```
db.collection_name.save( {_id:ObjectId(), new_document} )
```

Lets take the same example that we have seen above. Now we want to update the name of “Kit Harington” to “Jon Snow”. To work with save() method you should know the unique _id field of that document.

> A very important **point to note** is that when you do not provide the **_id** field while using save()
> method, it calls insert() method and the passed document is inserted into the collection as a new document

To get the _id of a document, you can either type this command:

```
db.got.find().pretty()
```

Here got is a collection name. This method of finding unique _id is only useful when you have few documents, otherwise scrolling and searching for that _id in huge number of documents is tedious.

If you have huge number of documents then, to get the _id of a particular document use the criteria in find() method. For example: To get the _id of the document that we want to update using save() method, type this command:

```
> db.got.find({"name": "Kit Harington"}).pretty()
{
        "_id" : ObjectId("59bd2e73ce524b733f14dd65"),
        "name" : "Kit Harington",
        "age" : 32
}
```

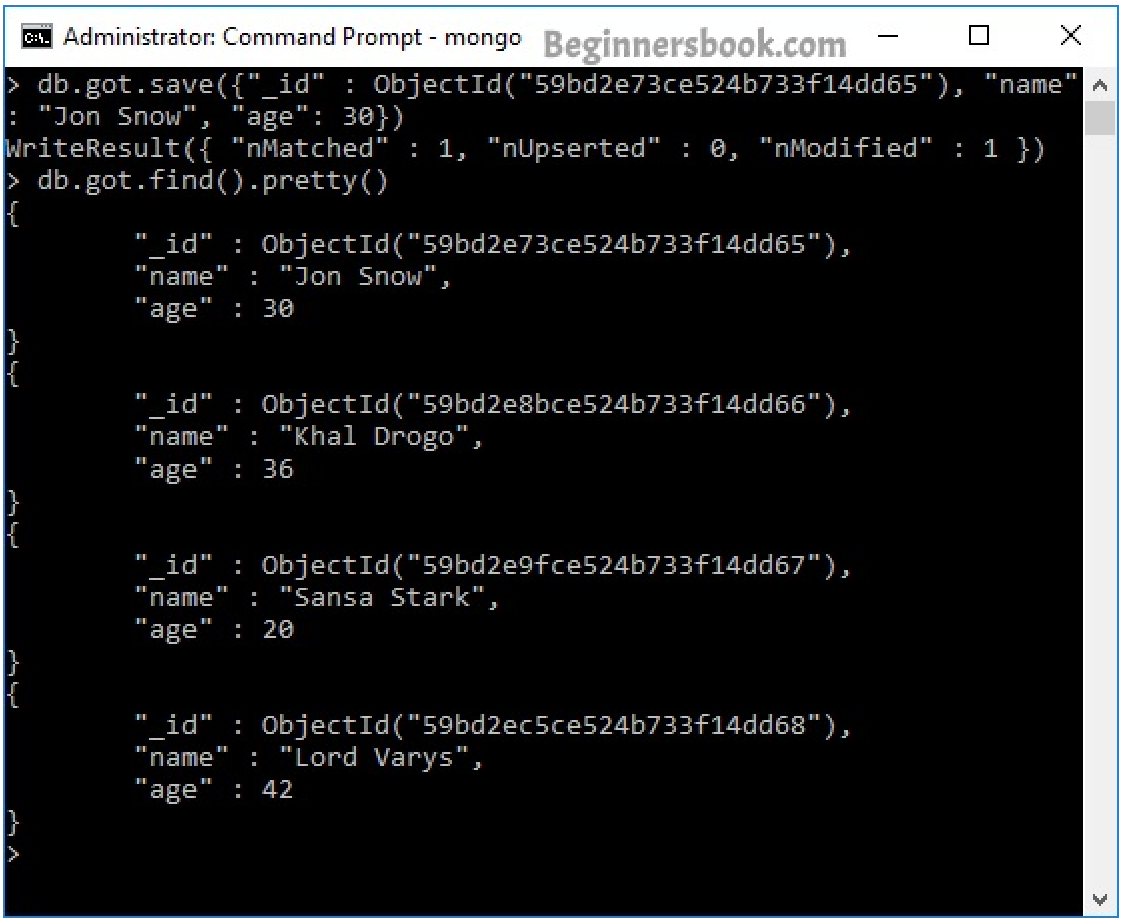
Now we know the unique _id field of that document. Lets write the command using save() method.

```
> db.got.save({"_id" : ObjectId("59bd2e73ce524b733f14dd65"), "name":
 "Jon Snow", "age": 30})
>
```

You should see this output:

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Verify the update:



```
> db.got.save({"_id" : ObjectId("59bd2e73ce524b733f14dd65"), "name"
: "Jon Snow", "age": 30})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.got.find().pretty()
{
        "_id" : ObjectId("59bd2e73ce524b733f14dd65"),
        "name" : "Jon Snow",
        "age" : 30
}
{
        "_id" : ObjectId("59bd2e8bce524b733f14dd66"),
        "name" : "Khal Drogo",
        "age" : 36
}
{
        "_id" : ObjectId("59bd2e9fce524b733f14dd67"),
        "name" : "Sansa Stark",
        "age" : 20
}
{
        "_id" : ObjectId("59bd2ec5ce524b733f14dd68"),
        "name" : "Lord Varys",
        "age" : 42
}
>
```