MONGODB

The insert() Method

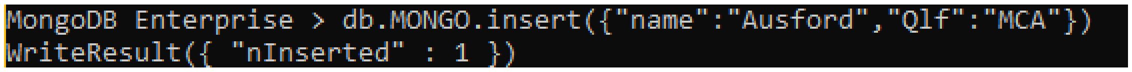
To insert data into MongoDB collection, you need to use MongoDB's **insert()** or **save()** method.

Syntax

The basic syntax of **insert()** command is as follows −

>db. COLLECTION_NAME.insert(document).

Eg:



```
MongoDB Enterprise > db.MONGO.insert({"name":"Ausford","Qlf":"MCA"})
WriteResult({ "nInserted" : 1 })
```

The Update () Method

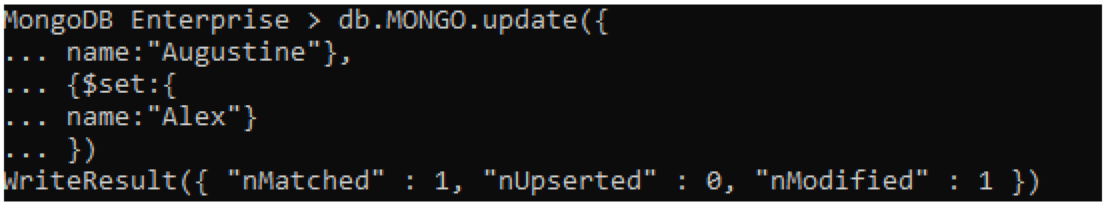
The update() method updates the values in the existing document.

Syntax

The basic syntax of **update ()** method is as follows −

>db. COLLECTION_NAME.update(SELECTION_CRITERIA, UPDATED_DATA)

Eg:



```
MongoDB Enterprise > db.MONGO.update({
... name:"Augustine"},
... {$set:{
... name:"Alex"}
... })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

The remove () Method

MongoDB's **remove ()** method is used to remove a document from the collection. remove() method accepts two parameters. One is deletion criteria and second is justOne flag.

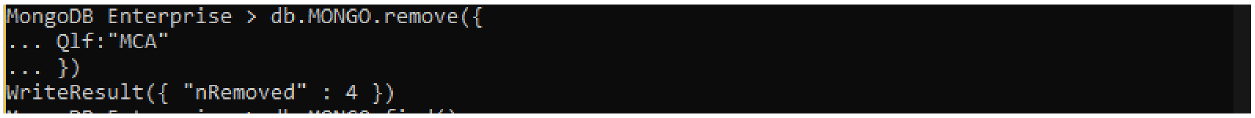
- **deletion criteria** − (Optional) deletion criteria according to documents will be removed.
- **justOne** − (Optional) if set to true or 1, then remove only one document.

Syntax

Basic syntax of **remove ()** method is as follows –

>db. COLLECTION_NAME.remove(DELLETION_CRITTERIA)

Eg:



```
MongoDB Enterprise > db.MONGO.remove({
... Qlf:"MCA"
... })
WriteResult({ "nRemoved" : 4 })
```

Insert multiple documents with Bulk

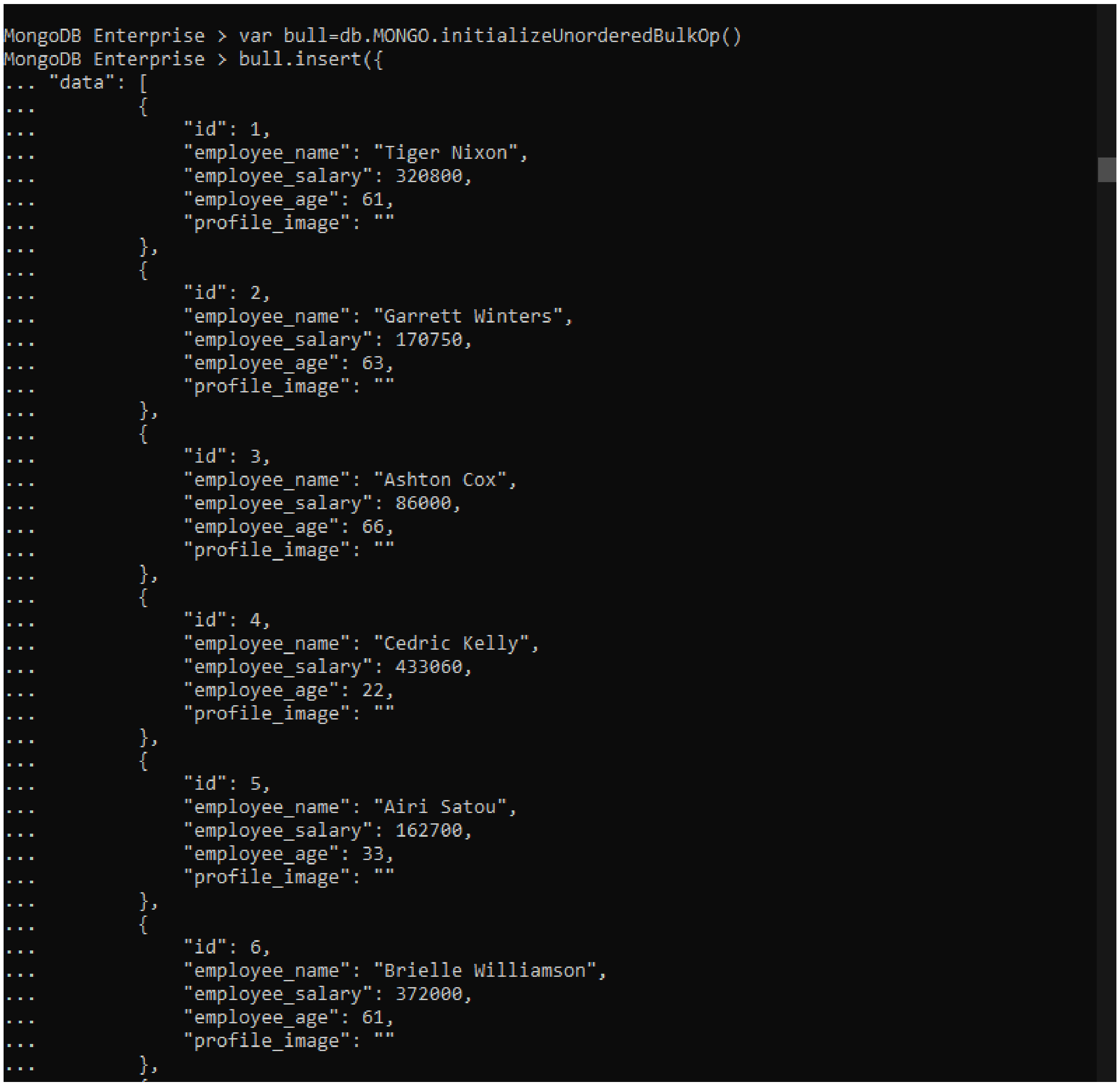
It can be used to perform multiple write operations in bulk.

Initialize a bulk operation builder

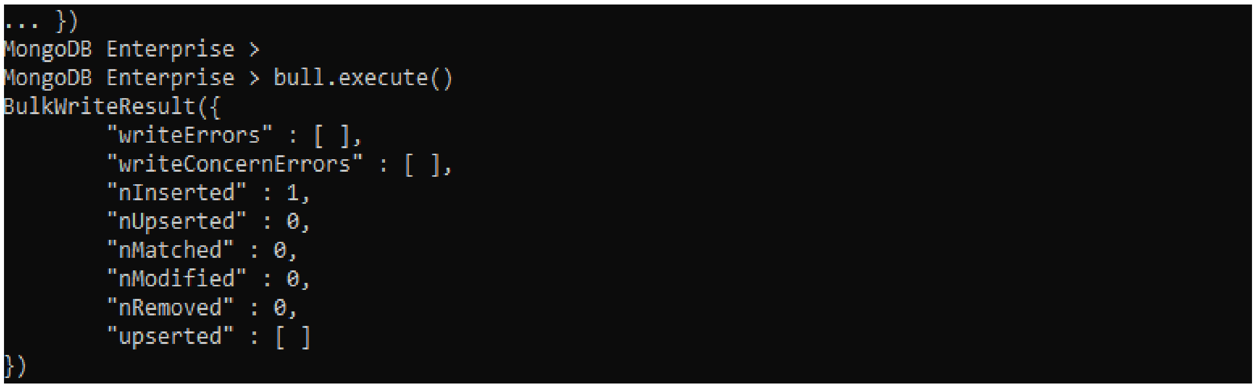
First initialize a bulk operation builder for the collection.

- var bulk = db. COLLECTION_NAME.initializeUnorderedBulkOp();

Eg:



```
MongoDB Enterprise > var bull=db.MONGO.initializeUnorderedBulkOp()
MongoDB Enterprise > bull.insert({
... "data": [
...         {
...             "id": 1,
...             "employee_name": "Tiger Nixon",
...             "employee_salary": 320800,
...             "employee_age": 61,
...             "profile_image": ""
...         },
...         {
...             "id": 2,
...             "employee_name": "Garrett Winters",
...             "employee_salary": 170750,
...             "employee_age": 63,
...             "profile_image": ""
...         },
...         {
...             "id": 3,
...             "employee_name": "Ashton Cox",
...             "employee_salary": 86000,
...             "employee_age": 66,
...             "profile_image": ""
...         },
...         {
...             "id": 4,
...             "employee_name": "Cedric Kelly",
...             "employee_salary": 433060,
...             "employee_age": 22,
...             "profile_image": ""
...         },
...         {
...             "id": 5,
...             "employee_name": "Airi Satou",
...             "employee_salary": 162700,
...             "employee_age": 33,
...             "profile_image": ""
...         },
...         {
...             "id": 6,
...             "employee_name": "Brielle Williamson",
...             "employee_salary": 372000,
...             "employee_age": 61,
...             "profile_image": ""
...         },
```



```
... })
MongoDB Enterprise >
MongoDB Enterprise > bull.execute()
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 1,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
```

The Find () Method

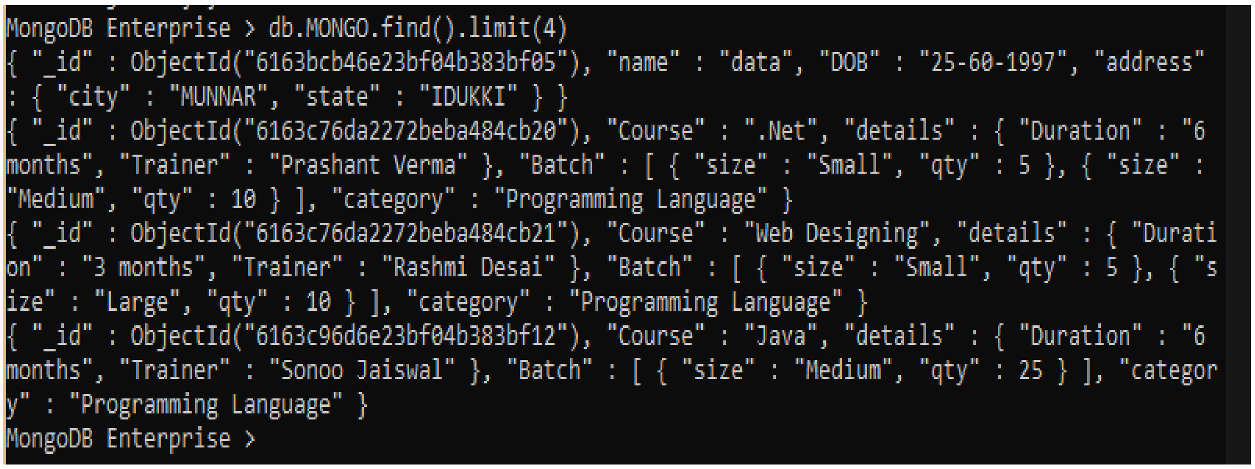
MongoDB's find () method, when you execute find () method, then it displays all fields of a document. To limit this, you need to set a list of fields with value 1 or 0. 1 is used to show the field while 0 is used to hide the fields.

Syntax

The basic syntax of find () method with projection is as follows −

>db. COLLECTION_NAME.find({}, {KEY:1})

Eg:



```
MongoDB Enterprise > db.MONGO.find().limit(4)
{ "_id" : ObjectId("6163bcb46e23bf04b383bf05"), "name" : "data", "DOB" : "25-60-1997", "address"
: { "city" : "MUNNAR", "state" : "IDUKKI" } }
{ "_id" : ObjectId("6163c76da2272beba484cb20"), "Course" : ".Net", "details" : { "Duration" : "6
months", "Trainer" : "Prashant Verma" }, "Batch" : [ { "size" : "Small", "qty" : 5 }, { "size" :
"Medium", "qty" : 10 } ], "category" : "Programming Language" }
{ "_id" : ObjectId("6163c76da2272beba484cb21"), "Course" : "Web Designing", "details" : { "Durati
on" : "3 months", "Trainer" : "Rashmi Desai" }, "Batch" : [ { "size" : "Small", "qty" : 5 }, { "s
ize" : "Large", "qty" : 10 } ], "category" : "Programming Language" }
{ "_id" : ObjectId("6163c96d6e23bf04b383bf12"), "Course" : "Java", "details" : { "Duration" : "6
months", "Trainer" : "Sonoo Jaiswal" }, "Batch" : [ { "size" : "Medium", "qty" : 25 } ], "categor
y" : "Programming Language" }
MongoDB Enterprise >
```

The sort () Method

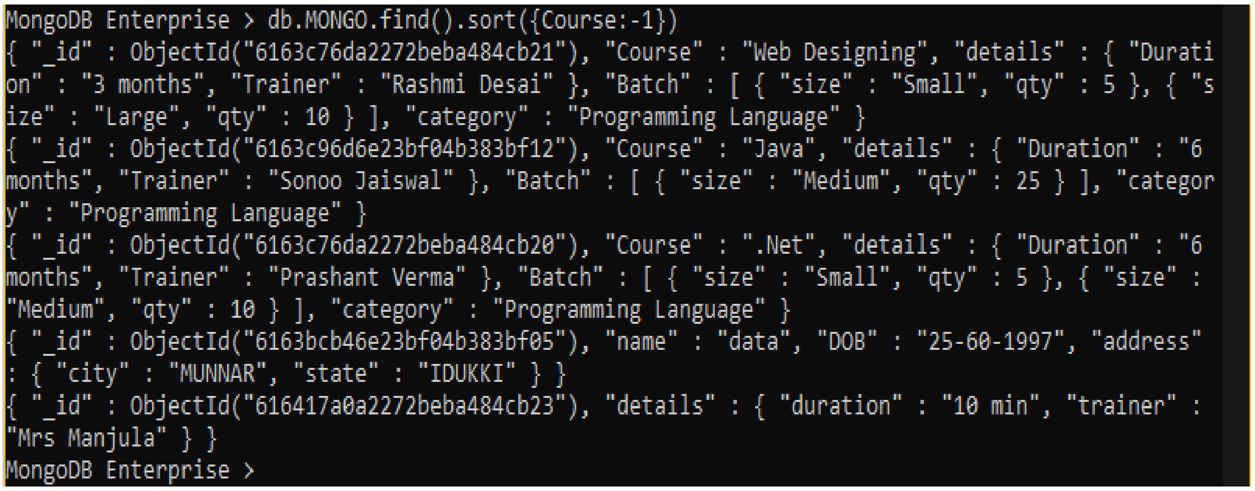
To sort documents in MongoDB, you need to use **sort ()** method. The method accepts a document containing a list of fields along with their sorting order. To specify sorting order 1 and -1 are used. 1 is used for ascending order while -1 is used for descending order.

Syntax

The basic syntax of **sort ()** method is as follows −

>db. COLLECTION_NAME.find(). sort ({KEY:1})

Eg:



```
MongoDB Enterprise > db.MONGO.find().sort({Course:-1})
{ "_id" : ObjectId("6163c76da2272beba484cb21"), "Course" : "Web Designing", "details" : { "Durati
on" : "3 months", "Trainer" : "Rashmi Desai" }, "Batch" : [ { "size" : "Small", "qty" : 5 }, { "s
ize" : "Large", "qty" : 10 } ], "category" : "Programming Language" }
{ "_id" : ObjectId("6163c96d6e23bf04b383bf12"), "Course" : "Java", "details" : { "Duration" : "6
months", "Trainer" : "Sonoo Jaiswal" }, "Batch" : [ { "size" : "Medium", "qty" : 25 } ], "categor
y" : "Programming Language" }
{ "_id" : ObjectId("6163c76da2272beba484cb20"), "Course" : ".Net", "details" : { "Duration" : "6
months", "Trainer" : "Prashant Verma" }, "Batch" : [ { "size" : "Small", "qty" : 5 }, { "size" :
"Medium", "qty" : 10 } ], "category" : "Programming Language" }
{ "_id" : ObjectId("6163bcb46e23bf04b383bf05"), "name" : "data", "DOB" : "25-60-1997", "address"
: { "city" : "MUNNAR", "state" : "IDUKKI" } }
{ "_id" : ObjectId("616417a0a2272beba484cb23"), "details" : { "duration" : "10 min", "trainer" :
"Mrs Manjula" } }
MongoDB Enterprise >
```

The aggregate () Method

For the aggregation in MongoDB, you should use aggregate () method.

Syntax

Basic syntax of aggregate () method is as follows −

>db. COLLECTION_NAME.aggregate(AGGREGATE_OPERATION)

$match

The $match stage can use an index to filter documents if it occurs at the beginning of a pipeline.

$sort

The $sort stage can use an index as long as it is not preceded by a $project, $unwind, or $group stage.

$group

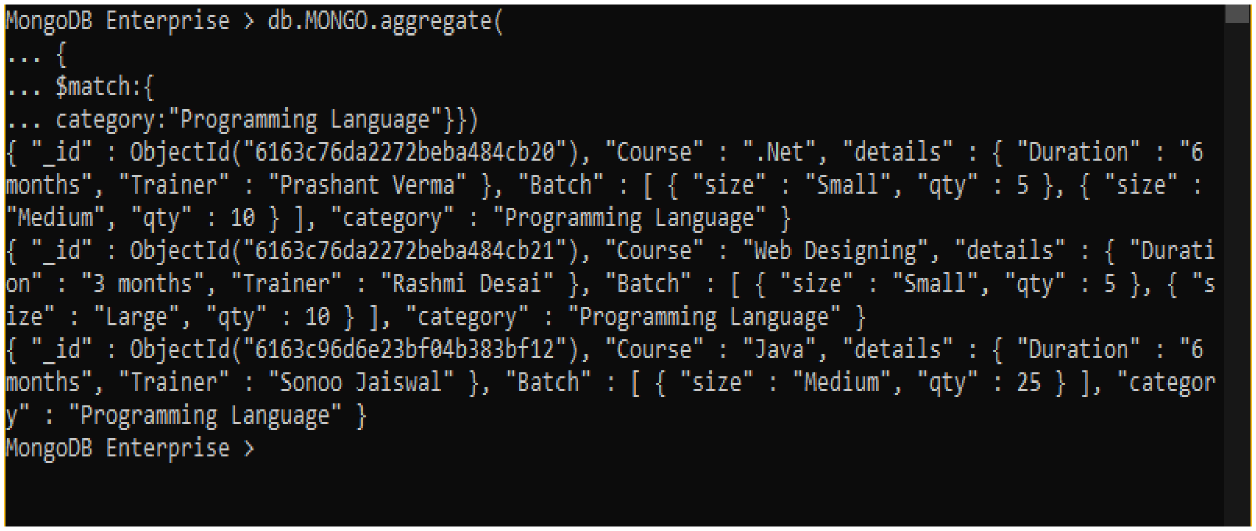
The $group stage can sometimes use an index to find the first document in each group if all of the following criteria are met:

The $group stage is preceded by a $sort stage that sorts the field to group by,

There is an index on the grouped field which matches the sort order and

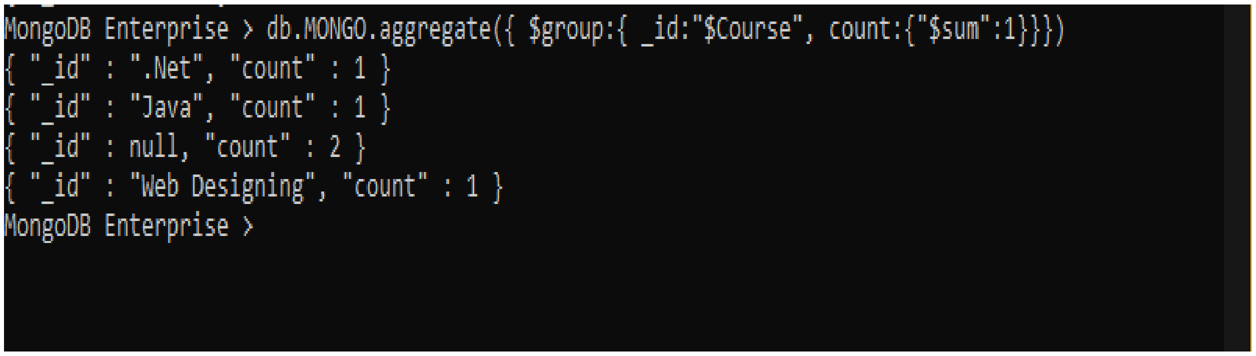
The only accumulator used in the $group stage is $first.

Eg for $match:



```
MongoDB Enterprise > db.MONGO.aggregate(
... {
... $match:{
... category:"Programming Language"}})
{ "_id" : ObjectId("6163c76da2272beba484cb20"), "Course" : ".Net", "details" : { "Duration" : "6
months", "Trainer" : "Prashant Verma" }, "Batch" : [ { "size" : "Small", "qty" : 5 }, { "size" :
"Medium", "qty" : 10 } ], "category" : "Programming Language" }
{ "_id" : ObjectId("6163c76da2272beba484cb21"), "Course" : "Web Designing", "details" : { "Durati
on" : "3 months", "Trainer" : "Rashmi Desai" }, "Batch" : [ { "size" : "Small", "qty" : 5 }, { "s
ize" : "Large", "qty" : 10 } ], "category" : "Programming Language" }
{ "_id" : ObjectId("6163c96d6e23bf04b383bf12"), "Course" : "Java", "details" : { "Duration" : "6
months", "Trainer" : "Sonoo Jaiswal" }, "Batch" : [ { "size" : "Medium", "qty" : 25 } ], "categor
y" : "Programming Language" }
MongoDB Enterprise >
```

Eg for $group:



```
MongoDB Enterprise > db.MONGO.aggregate({ $group:{ _id:"$Course", count:{"$sum":1}}})
{ "_id" : ".Net", "count" : 1 }
{ "_id" : "Java", "count" : 1 }
{ "_id" : null, "count" : 2 }
{ "_id" : "Web Designing", "count" : 1 }
MongoDB Enterprise >
```