

MongoDB Insert Document

In this tutorial, we will see how to insert a document into the collection. We will understand this with the help of multiple examples.

Syntax to insert a document into the collection:

```
db.collection_name.insert()
```

Lets take an example to understand this .

MongoDB Insert Document using insert() Example

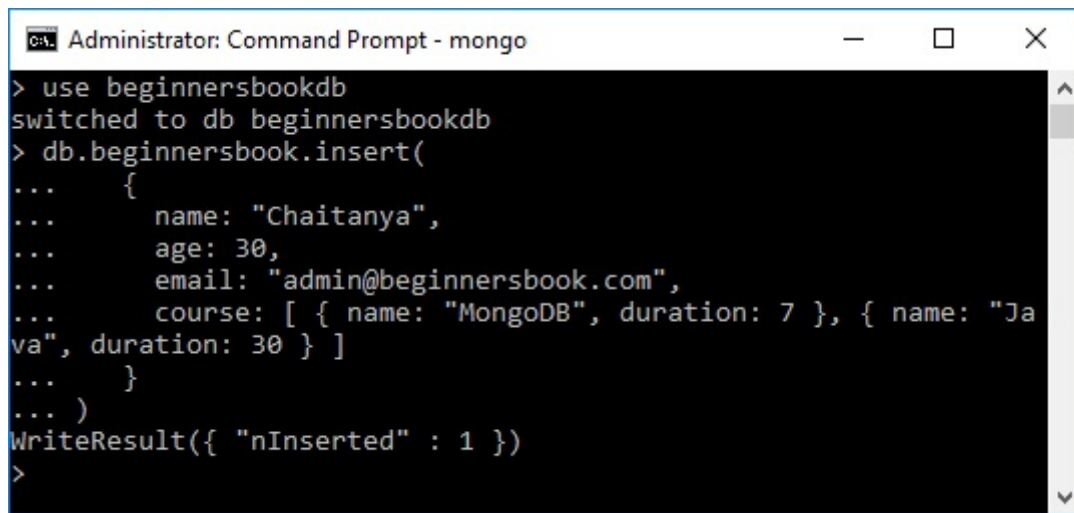
Here we are inserting a document into the collection named “beginnersbook”. The field “course” in the example below is an array that holds the several key-value pairs.

```
db.beginnersbook.insert(  
  {  
    name: "Chaitanya",  
    age: 30,  
    email: "admin@beginnersbook.com",  
    course: [ { name: "MongoDB", duration: 7 }, { name: "Java", duration: 30 } ]  
  }  
)
```

You should see a successful write message like this:

```
WriteResult({ "nInserted" : 1 })
```

The insert() method creates the collection if it doesn't exist but if the collection is present then it inserts the document into it



```
Administrator: Command Prompt - mongo
> use beginnersbookdb
switched to db beginnersbookdb
> db.beginnersbook.insert(
...   {
...     name: "Chaitanya",
...     age: 30,
...     email: "admin@beginnersbook.com",
...     course: [ { name: "MongoDB", duration: 7 }, { name: "Java", duration: 30 } ]
...   }
... )
WriteResult({ "nInserted" : 1 })
>
```

Verification:

You can also verify whether the document is successfully inserted by typing following command:

```
db.collection_name.find()
```

In the above example, we inserted the document in the collection named “beginnersbook” so the command should be:

```
> db.beginnersbook.find()
{ "_id" : ObjectId("59bce797668dcce02aaa6fec"), "name" : "Chaitanya", "age" : 30,
"email" : "admin@beginnersbook.com", "course" : [ { "name" : "MongoDB",
"duration" : 7 }, { "name" : "Java", "duration" : 30 } ] }
```

MongoDB Example: Insert Multiple Documents in collection

To insert multiple documents in collection, we define an array of documents and later we use the insert() method on the array variable as shown in the example below. Here we are inserting three documents in the collection named “students”. This command will insert the data in “students” collection, if the collection is not present then it will create the collection and insert these documents.

```
var beginners =
[
  {
    "StudentId" : 1001,
    "StudentName" : "Steve",
    "age": 30
  },
  {
    "StudentId" : 1002,
```

```

        "StudentName" : "Negan",
        "age": 42
    },
    {
        "StudentId" : 3333,
        "StudentName" : "Rick",
        "age": 35
    },
];
db.students.insert(beginners);

```

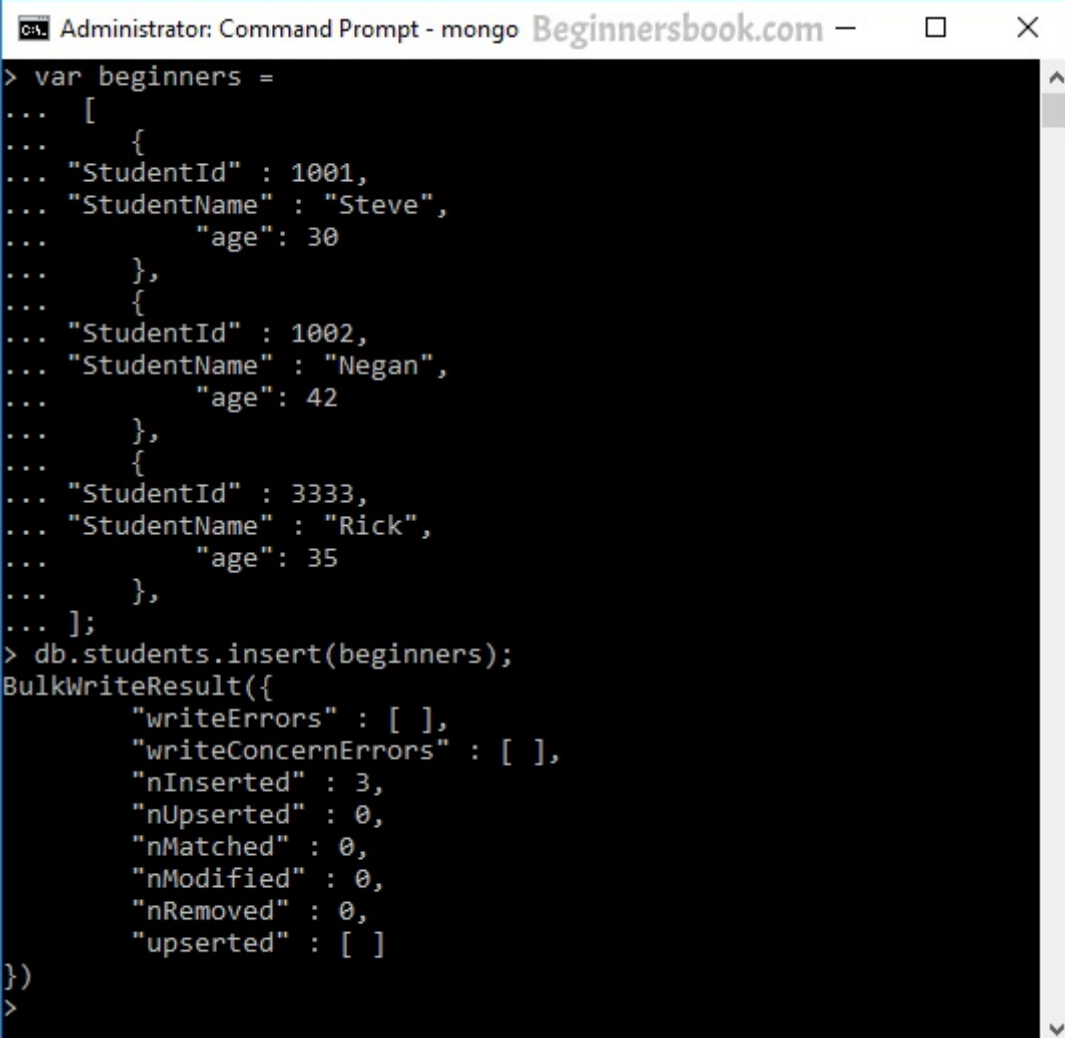
You would see this output:

```

BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 3,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})

```

As you can see that it shows number 3 in front of **nInserted**. this means that the 3 documents have been inserted by this command.



```

Administrator: Command Prompt - mongo Beginnersbook.com
> var beginners =
...   [
...     {
...       "StudentId" : 1001,
...       "StudentName" : "Steve",
...       "age": 30
...     },
...     {
...       "StudentId" : 1002,
...       "StudentName" : "Negan",
...       "age": 42
...     },
...     {
...       "StudentId" : 3333,
...       "StudentName" : "Rick",
...       "age": 35
...     },
...   ];
> db.students.insert(beginners);
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 3,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
>

```

To verify that the documents are there in collection. Run this command:

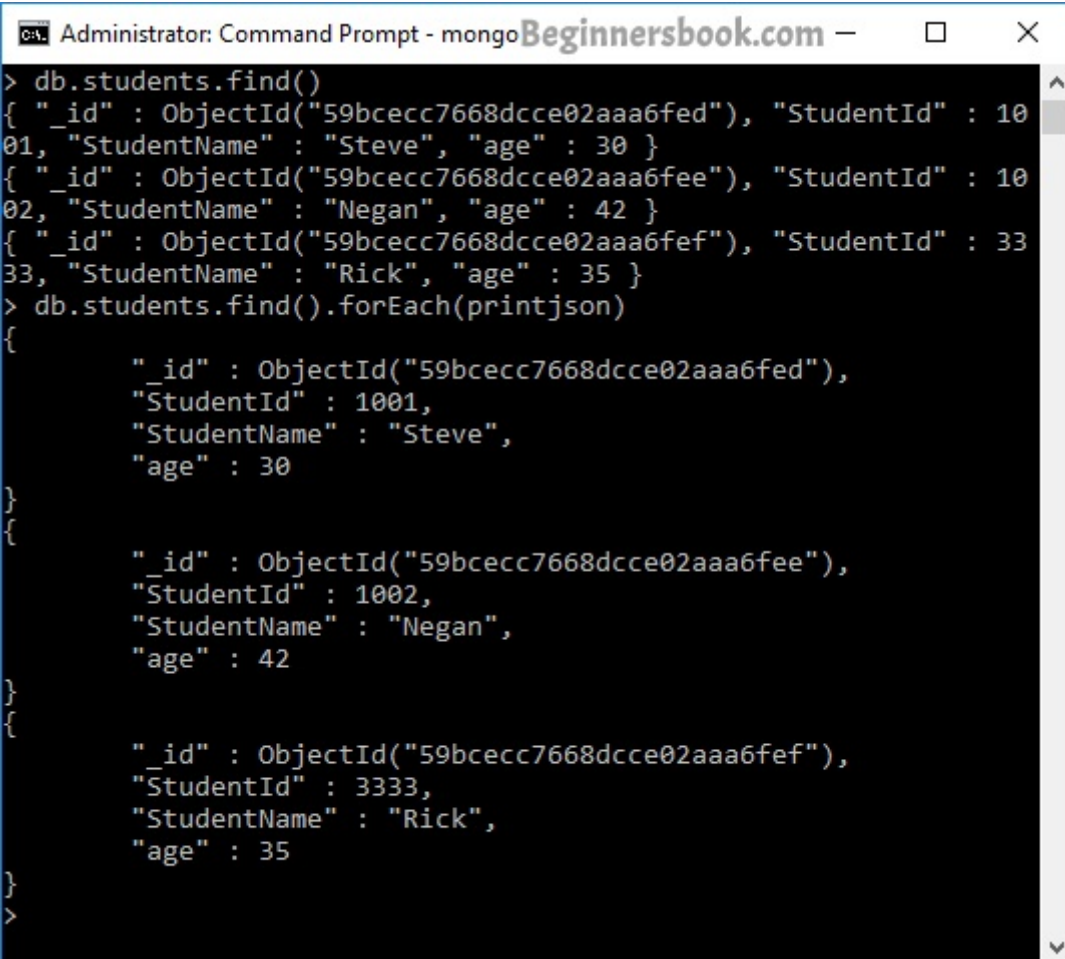
```
db.students.find()
```

Do you know? You can print the output data in a JSON format so that you can read it easily. To print the data in JSON format run the command **db.collection_name.find().forEach(printjson)**

So in our case the command would be this:

```
db.students.find().forEach(printjson)
```

In the screenshot below, you can see the difference. First we have printed the documents using normal find() method and then we printed the documents of same collection using JSON format. The documents in JSON format are neat and easy to read.



```
> db.students.find()
{ "_id" : ObjectId("59bcecc7668dcce02aaa6fed"), "StudentId" : 1001, "StudentName" : "Steve", "age" : 30 }
{ "_id" : ObjectId("59bcecc7668dcce02aaa6fee"), "StudentId" : 1002, "StudentName" : "Negan", "age" : 42 }
{ "_id" : ObjectId("59bcecc7668dcce02aaa6fef"), "StudentId" : 3333, "StudentName" : "Rick", "age" : 35 }
> db.students.find().forEach(printjson)
{
  "_id" : ObjectId("59bcecc7668dcce02aaa6fed"),
  "StudentId" : 1001,
  "StudentName" : "Steve",
  "age" : 30
}
{
  "_id" : ObjectId("59bcecc7668dcce02aaa6fee"),
  "StudentId" : 1002,
  "StudentName" : "Negan",
  "age" : 42
}
{
  "_id" : ObjectId("59bcecc7668dcce02aaa6fef"),
  "StudentId" : 3333,
  "StudentName" : "Rick",
  "age" : 35
}
>
```