1. **Batch insert or Inserting Multiple documents into a collection**

To insert multiple documents or Batch insert into a collection `insertMany()` method is available to do this task.

**Syntax:**

```javascript
db.collection.insertMany(
    [ <document 1> , <document 2>, ... ],
)
```

Where

- Each of these document should be separated by a `, comma`
- `[<document 1>, <document 2>, ...]` is an array of the various s documents which needs to be inserted into the collection.

**On Successful execution, it returns a**

- A boolean acknowledged as true if the operation is successful or false if uncessful.
- An array of `_id` for each successfully inserted documents

**Example:**

```javascript
> db.collection.insertMany([{
    "name": "star",
    "course": "python",
    "fees": "14000"
},
{
    "name": "best",
    "course": "c++",
    "fees": "15000"
}])
{
    "acknowledged": true,
    "insertedIds": [
        ObjectId("5eb1fcb0d424bc63930ebc00"),
        ObjectId("5eb1fcb0d424bc63930ebc00"")
    ]
}

> db.collection.find()
[{
   "_id": ObjectId("5eb1fcb0d424bc63930ebc00"),
   "name": "star",
   "course": "python",
   "fees": "14000"
},
{
   "_id": ObjectId("5eb1fcb0d424bc63930ebc00"),
   "name": "best",
   "course": "c++",
   "fees": "15000"
}]
```

2. **`remove()` method to remove documents from collection**

`remove()` method is used to remove a document from collection.

**Syntax**

```javascript
db.COLLECTION_NAME.remove(DELLETION_CRITTERIA, justOne)
```

this method has two parameters One is deletion criteria and second is `justOne` flag.

- **DELLETION_CRITTERIA** – (Optional) deletion criteria of the documents to be removed. To delete all documents in a collection, pass an empty document `{}`.
• justOne – (Optional) if set to true or 1, then remove only one document i.e. the 1\textsuperscript{st} document. Optional. If not specified i.e. the default value of false and it will delete all documents matching the deletion criteria.

**Example 1: Remove only one document**

```javascript
db.school.remove({"class":"12th"},1)
```

will remove 1\textsuperscript{st} record of class 12\textsuperscript{th}.

**Example 2: Remove all records of the matching criteria**

```javascript
db.school.remove({"class":"12th"})
```

will remove all the record of class 12\textsuperscript{th}.

```javascript
WriteResult({ "nRemoved" : 2 })
```

Here it has removed all the 2 records of matching criteria.

**Example 3: Remove all document if no criteria is specified**

```javascript
db.school.remove({})
```

will remove all the record from the collection.
WriteResult({ "nRemoved" : 2 })
Here it has removed all the 2 records of matching criteria

```
db.school.find()
{ "id" : ObjectId("5e0b90effe73fbd802h1es"), "class" : "10th", "section" : "A section", "student count" : "30" }
{ "id" : ObjectId("5e0b90effe73fbd802h1es"), "class" : "10th", "section" : "B section", "student count" : "20" }
{ "id" : ObjectId("5e0b90effe73fbd802h1es"), "class" : "10th", "section" : "C section", "student count" : "37" }
```

WriteResult({ "nRemoved" : 4 })
Here it has removed all the 4 records from collection and now the collection is empty.

3. update() Method : Updating the values in document
The update() method is used to update the value in the existing document of specified collection.

**Syntax**
```
db.COLLECTION_NAME.update(SELECTION_CRITERIA, UPDATED_DATA)
```

where
- **SELECTION_CRITERIA** -- The selection criteria for the updating the document
- **UPDATED_DATA** -- The modifications to apply to the document. It may be simply given with $set: parameter.

**Example 1**
```
db.school.update({"class":"10th"}, {$set:{class:"11th"}})
```
after this command, record with class 10th will be updated to class 11th

```
{ "id" : ObjectId("5e0b90effe73fbd802h1es"), "class" : "10th", "section" : "A section", "student count" : "30" }
{ "id" : ObjectId("5e0b90effe73fbd802h1es"), "class" : "10th", "section" : "B section", "student count" : "20" }
{ "id" : ObjectId("5e0b90effe73fbd802h1es"), "class" : "10th", "section" : "C section", "student count" : "37" }
```

```
{ "id" : ObjectId("5e0b90effe73fbd802h1es"), "class" : "11th", "section" : "A section", "student count" : "30" }
```

**Example 2** : In case of multiple matching criteria, only 1st document will be updated
```
db.school.update({"class":"12th"}, {$set:{class:"9th"}})
```
after this command, 1st record with class 12th will be updated to class 9th
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 }) means 1 record matched and updated one record.

4. updateMany() Method:

The updateMany() method is used to update all the matching records of collection. All other parameter and system is similar to update() method.

Syntax

db.COLLECTION_NAME.updateMany(SELECTION_CRITERIA, UPDATED_DATA)

Example:

Lets update all the records of the matching criteria,

db.school.updateMany({"class":"9th"}, {$set:{class:"12th"}})

It results as:

{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }

Its shows that two records are matched and 2 of them has been modified.

Assignment

1. How to batch insert documents in collection?
2. How to remove documents from collection?
3. What is the difference between update and updateMany?