MongoDB

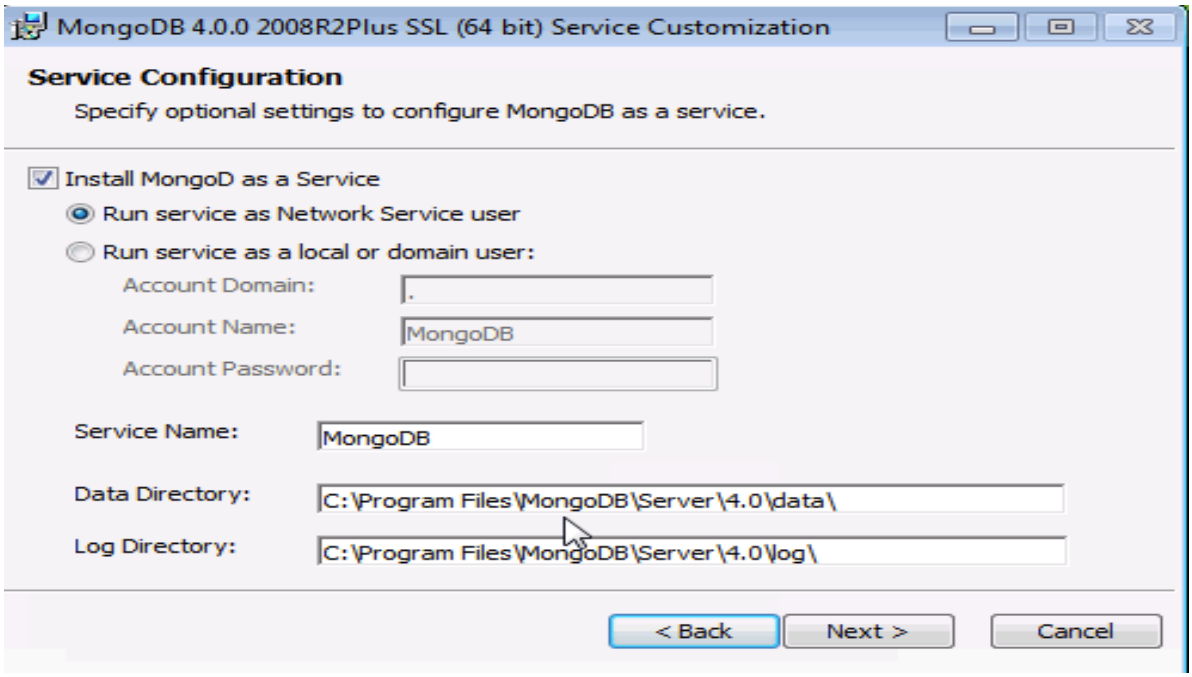
- MongoDB is a document-oriented database.
- schema-less database.

Installation of MongoDB

1. Download the installer MongoDB community
   (a) In the Version dropdown, select the version of MongoDB to download.
   (b) In the Platform dropdown, select Windows.
   (c) In the Package dropdown, select .msi installer.
   (d) Click Download.
2. Run the MongoDB installer

   (a)Go to the directory where you downloaded the MongoDB installer (.msi file). By default, this is your Downloads directory.
   (b)Double-click the .msi file.

3. Follow the MongoDB Community Edition installation wizard



(A)Select Install MongoDB as a Service MongoDB as a service.

Select either:Run the service as Network Service user (Default)This is a Windows user account that is built-in to Windows or Run the service as a local or domain user

For an existing local user account, specify a period (i.e. .) for the Account Domain and specify the Account Name and the Account Password for the user.

For an existing domain user, specify the Account Domain, the Account Name and the Account Password for that user

4. Install MongoDB Compass

   To have the wizard install MongoDB Compass and click Install

5. If You Installed MongoDB as a Windows Service

The MongoDB service starts upon successful installation. Configure the MongoDB instance with the configuration file <install directory>\bin\mongod.cfg.

6. If You Did Not Install MongoDB as a Windows Service
   If you only installed the executables and did not install MongoDB as a Windows service, you must manually start the MongoDB instance.

Insert() Method

Used to add or insert new documents into a collection in your database.

Syntax: db.COLLECTION_NAME.**insert**(document)

- To check the inserted document

>db.javatpoint.find()

```
{ "_id" : ObjectId("56482d3e27e53d2dbc93cef8"), "course" : "java", "details" :
{ "duration" : "6 months", "Trainer" : "Sonoo jaiswal" }, "Batch" :
[ {"size" : "Small", "qty" : 15 }, { "size" : "Medium", "qty" : 25 } ],
 "category" : "Programming language" }
```

Upsert

Upsert is an operation that performs either an update of existing document or an insert of new document if the document to modify does not exist.

➢ Create an array of document

```
MongoDB Enterprise > var Allcourses =
...     [
...       {
...         Course: "Java",
...         details: { Duration: "6 months", Trainer: "Sonoo Jaiswal" },
...         Batch: [ { size: "Medium", qty: 25 } ],
...         category: "Programming Language"
...       },
...       {
...         Course: ".Net",
...         details: { Duration: "6 months", Trainer: "Prashant Verma" },
...         Batch: [ { size: "Small", qty: 5 }, { size: "Medium", qty: 10 }, ],
...         category: "Programming Language"
...       },
...       {
...         Course: "Web Designing",
...         details: { Duration: "3 months", Trainer: "Rashmi Desai" },
...         Batch: [ { size: "Small", qty: 5 }, { size: "Large", qty: 10 } ],
...         category: "Programming Language"
...       }
...     ];
MongoDB Enterprise > db.second.insert(allcourse)
uncaught exception: ReferenceError: allcourse is not defined :
@(shell):1:1
MongoDB Enterprise > db.second.insert(Allcourse)
uncaught exception: ReferenceError: Allcourse is not defined :
@(shell):1:1
MongoDB Enterprise > db.second.insert(Allcourses)
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 3,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
```

Update

In MongoDB, update() method is used to update or modify the existing documents of a collection.

Syntax: db.COLLECTION_NAME.**update**(SELECTIOIN_CRITERIA, UPDATED_DATA)

Input

```
MongoDB Enterprise > show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
mydb    0.000GB
test    0.000GB
MongoDB Enterprise > db.student.update({'course':'java'},{$set:{'course':'android'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Output

```
MongoDB Enterprise > show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
mydb    0.000GB
test    0.000GB
MongoDB Enterprise > db.student.update({'course':'java'},{$set:{'course':'android'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

Delete

Syntax: db.collection_name.remove (DELETION_CRITERIA)

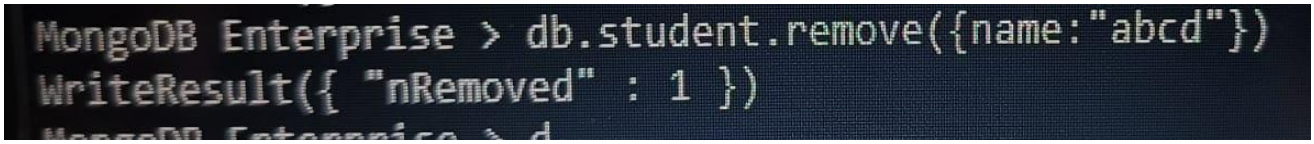
Used to delete documents from a collection. The remove() method works on two parameters.

1. Deletion criteria: With the use of its syntax you can remove the documents from the collection.

2. Just One**:** It removes only one document when set to true or 1.

- To Remove all documents

db.javatpoint.remove({})



Query

The db.collection.find() method is used to retrieve documents from a collection. This method returns a cursor to the retrieved documents.
Syntax:
db.COLLECTION_NAME.find()

AGGREGATION

.aggregate()

>db.COLLECTION_NAME.aggregate(AGGREGATE_OPERATION)

```
_id: ObjectId("6164270061fc5b504b706f31")
name: "ash"
Age: 24
mark: 99
```

```
_id: ObjectId("6164293cc46cdb60a6f93889")
name: "zamna"
Age: 24
mark: 100
```

```
_id: ObjectId("61642983c46cdb60a6f9388a")
name: "jems"
Age: 24
mark: 89
```

```
_id: ObjectId("616429e6c46cdb60a6f9388b")
name: "neeeth"
mark: 91
Age: 24
```

1. $group

Used to group **data** from the collection, we can achieve group by clause using aggregate function and group method in Mongo DB. While using aggregate function with group by clause query operations is faster as normal query, basically aggregate function is used in multiple condition.

$avg

Calculates the average of all given values from all documents in the collection.

$Sum

Sums up the defined value from all documents in the collection.

$max

Gets the maximum of the corresponding values from all documents in the collection.

$min

Gets the minimum of the corresponding values from all documents in the collection.

2. $match

This is a filtering operation and thus this can reduce the amount of documents that are given as input to the next stage.


$match
/**
* query: The query in MQL.
*/
{
"name":"zamna"
}
Output after $match stage (Sample of 1 document)
_id: ObjectId("6164293cc46cdb60a6f93889")
name: "zamna"
Age: 24
mark: 100