MongoDB is an open-source document database, and leading NoSQL database written in c++. MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability, works on concept of collection and document.

Collection

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

Document

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

Advantages of MongoDB over RDBMS

- Schema less : MongoDB is document database in which one collection holds different different documents. Number of fields, content and size of the document can be differ from one document to another.
- Structure of a single object is clear
- No complex joins
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL
- Tuning
- Ease of scale-out: MongoDB is easy to scale
- Conversion / mapping of application objects to database objects not needed

- Uses internal memory for storing the (windowed) working set, enabling faster access of data

Some basic commands in MongoDb

1.To use MongoDB

```
->mongo
```

2.MongoDB Help

```
->db.help()
```

3.MOngoDB Statistics

```
->db.stats()
```

4.TO create Database

```
->use Database_name
switched to Database_name
```

5.Check Database lists

```
->show dbs
```

6.Drop database

```
->db.dropDatabase()
```

7.Create Collection

```
->db.createCollection(name, options)
```

where, name= name of collection

options= options about memory size and indexing

```
//in MongoDB no need to specify the options
```

8.Drop Collection

```
->db.COLLECTION_NAME.drop()
```

9.Insert Document

```
->db.COLLECTION_NAME.insert(document)

Example:

db.user.insert({user_name:"Ram",Age:"21",Address:"Kathmandu"})

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

10.To query data from MongoDB

```
->db.user.find().pretty()  //user = collection

{

"_id" : ObjectId("55e43a8cf585cca56cafc325"),

"user_name" : "Ram",

"Age" : "21",

"Address" : "Kathmandu"

}
```

11.Update Data in MongoDB

```
->db.user.update({'user_name':"Ram"},{$set:{"user_name":"Hari"}},{multi:true})

->db.user.find().pretty()
```

```
{

"_id" : ObjectId("55e43a8cf585cca56cafc325"),

"user_name" : "Hari",

"Age" : "21",

"Address" : "Kathmandu"

}
```

12. Remove document in MOngoDB

```
->db.user.remove({'user_name':"hari"}) //Delete only one column

->db.user.remove()  //same as truncate in RDBMS
```