**MongoDB**

**->** it is a document oriented database that provides high performance, high availability, and easy scalability.

-> It works on concept of collection and document. Collection is a group of MongoDB documents. Document is a set of key-value pairs.

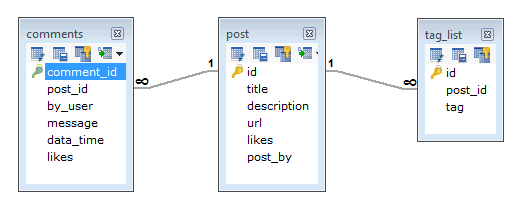
->Data is stored in the form of JSON style documents.

->it is used for Big Data, Datahub, User data management etc.

-> RDBMS Vs MongoDB

|  |  |
| --- | --- |
| RDBMS | MongoDB |
| Database | Database |
| Table | Collection |
| Row | Document |
| column | field |
| Table join | Embedded Documents |
| Primary Key | Primary Key(Default key \_id provided by mongodb itself) |

->Ex: Suppose a clients needs a database design for his website that is post,taglist and comments. So in RDMS schema design of this requirement will have minimum three tables.



-> While in MongoDB schema, design will have only one collection post.

{

\_id: POST\_ID

title: TITLE\_OF\_POST,

description: POST\_DESCRIPTION,

by: POST\_BY,

url: URL\_OF\_POST,

tags: [TAG1, TAG2, TAG3],

likes: TOTAL\_LIKES,

comments: [

{

user:'COMMENT\_BY',

message: TEXT,

dateCreated: DATE\_TIME,

like: LIKES

},

{

user:'COMMENT\_BY',

message: TEXT,

dateCreated: DATE\_TIME,

like: LIKES

}

]

}

-> So while showing the data, in RDBMS you need to join three tables and in MongoDB, data will be shown from one collection only.

* Document Structure
* {
* \_id: ObjectId(7df78ad8902c)
* title: 'MongoDB Overview',
* description: 'MongoDB is no sql database',
* by: 'tutorials point',
* url: 'http://www.tutorialspoint.com',
* tags: ['mongodb', 'database', 'NoSQL'],
* likes: 100,
* comments: [
* {
* user:'user1',
* message: 'My first comment',
* dateCreated: new Date(2011,1,20,2,15),
* like: 0
* },
* {
* user:'user2',
* message: 'My second comments',
* dateCreated: new Date(2011,1,25,7,45),
* like: 5
* }
* ]
* }

->Installation

1)Download the msi file and install.

2) create directory D:\data\db.

2)Run the command > mongod.exe --dbpath "D:\data"

3)Open another cmd and go bin and run the command >mongod.exe.

4)We get terminal to fire document based query.

5)command to check currently selected database is

>db

6)Command to check our database list is

>show dbs

-> collections will present inside database and documents will present inside collections.

-> db->collections->documents

>show collections

>use collection

If our created database is not present in the list it means database does not contain any document.

>db.dropDatabase()

>db.createCollection(“mycollection”)

> db. COLLECTION\_NAME.drop()

7)To insert data into MonoDB collection.for multiple document we have to pass array of documents.

>db.COLLECTION\_NAME.insert({“name”:”vikash”})

> db.COLLECTION\_NAME.insert([{“name”:”vikash”},{“sex”:”male”}])

>db.post.save(document)

8)if we are not passing any id the no difference between save() and insert(). But if we are passing id and that id is already available the it will update the contents.

9)for finding the documents we have to use command.

> db. COLLECTION\_NAME.find()

If we want to display in formatted way then we have to use pretty() method.

> db. COLLECTION\_NAME.find().pretty()

> db.love.find({"sal":1000})//where sal=2000

> db.love.find({"sal":{$lt:2000}})//where sal<2000

> db.love.find({"sal":{$lte:2000}})//where sal<=2000

> db.love.find({"sal":{$ne:2000}})//sal!=2000

> db.love.find({ $and :[ {"sal":{$lt:2000}} , {key2:value2} ] })

>db.love.find({ $or :[ {"sal":{$lt:2000}} , {key2:value2} ] })

> db.mycol.find({"likes": {$gt:10}, $or: [{"by": "tutorials point"},

{"title": "MongoDB Overview"}]}).pretty()//where like>10 and (by=”” or title=””)

> db.COLLECTION\_NAME.update(SELECTION\_CRITERIA, UPDATED\_DATA)

> db.mycol.update({'title':'MongoDB Overview'},{$set:{'title':'New MongoDB Tutorial'}})

> db.COLLECTION\_NAME.remove(DELLETION\_CRITTERIA)

> db.mycol.remove({'title':'MongoDB Overview'})

> db.COLLECTION\_NAME.remove(DELETION\_CRITERIA,1)//delete one in multiple record

> db.mycol.remove({'title':'MongoDB Overview'},1)

10) Projection : use to show part of the field rather than all fields.

> db.mycol.find({},{"title":1,\_id:0})

11)Limiting document: Use to limit the document.

> db.mycol.find({},{"title":1,\_id:0}).limit(2)

12)Skip document: Use to skip the number of documents.

> db.mycol.find({},{"title":1,\_id:0}).limit(1).skip(1)

13)Sort document : Use to sort the documents.

> db.mycol.find({},{"title":1,\_id:0}).sort({"title":-1})//desc

> db.mycol.find({},{"title":1,\_id:0}).sort({"title":1})//asc

14)Create index : used to select the documents very fast.

> db.mycol.ensureIndex({"title":1})

15)Aggregations Operation : use to group values from multiple documents together like sum(sal).

> db.mycol.aggregate([{$group : {\_id : "$by\_user", num\_tutorial : {$sum : 1}}}])

16)Dump data : use to dump the entire data of server into the dump directory **/bin/dump/**.

>mongodump

17)Replication : use to create copy of the data for backup.

> mongod --port 27017 --dbpath "D:\set up\mongodb\data" --replSet rs0

> rs.add("mongod1.net:27017")