**DrawBacks Of RDBMS**

🡪It becomes complex when the amount of data grows as it is relational.

🡪As is it isolated sharing of database is difficult from one system to other

🡪Cost is more

🡪Structured Limits, some names or queries are shorter than the actual and this can lead to data loss

**MongoDb**

🡪Mongodb is a nosql DB

🡪It is document based

🡪IT doesnot require any structuring

🡪It is faster in most of operations

Installation

Install that from mogodb.com site

After installation open the command prompts

Cmd1: open bin path

C:\Softwares\mongodb-win32-x86\_64-2012plus-4.2.1\mongodb-win32-x86\_64-2012plus-4.2.1\bin: mongod –port 27017 –dbpath “C:\Softwares\mongodb-win32-x86\_64-2012plus-4.2.1\mongodb-win32-x86\_64-2012plus-4.2.1\data”

Cmd2: open binpath

C:\Softwares\mongodb-win32-x86\_64-2012plus-4.2.1\mongodb-win32-x86\_64-2012plus-4.2.1\bin:mongo

**Commands:**

To show list of dabases:

>Show dbs

To use any database

>use databseName

>use myCustomers

Then it creates and switch to that database

To Delete/remove any database

* db.dropDatabase()

In this we give the data in JSON object(JavaScript Object Notation)

//create user

db.createUser({user:"khairu",pwd:"1234",roles:["readWrite","dbAdmin"]})

🡺Next create a collection

db.createCollection(collectionName)

db.createCollection('customers');

🡺Next to show collections

> Show collections o/p:customers

🡺To Insert values

>db.customers.insert({first\_name:"Khairunnisa",last\_name:"Begum"})

🡪If you want to insert multiple values at a time ,we will give this in array

> db.customers.insert([{first\_name:"Sree Lekha",last\_name:"Mula"},{first\_name:"Rahul",last\_name:"Konde"}])

🡺To get that values

> db.customers.find()

🡺To get that values in nyc way

> db.customers.find().pretty()

🡺To update any field(for this we can use any field or id number for identification)

>db.customers.update({first\_name:"Khairunnisa"},{first\_name:"Khairunnisa",last\_name:"Begum",gender:"female"})

🡪Another way to update using $set

>db.customers.update({\_id:ObjectId("5ddcd57f25f867cb36c85401")},{$set:{gender:"male"}})

🡺To increment values {$inc:{field:value (how much to be incremented

)}}

>db.customers.update({\_id:ObjectId("5ddcd57f25f867cb36c85400")},{$inc:{age:5}})

🡺BY using update query we can add new value also {upsert:true}

>db.customers.update({first\_name:"Mahesh"},{first\_name:"Mahesh",last\_name:"Pandugula",gender:"male"},{upsert:true})

🡺How to rename the fields{$rename:{"gender":"sex"}}

>db.customers.update({first\_name:"Mahesh"},{$rename:{"gender":"sex"}})

🡺 How to remove/delete any value it removes complete data

db.customers.remove({first\_name:"Rahul"})

🡺To unset/remove field it removes that field

db.customers.update({\_id:ObjectId("5ddcd57f25f867cb36c85400")},{$unset:{age:1}})

🡺To get the value which greater than

db.customers.find({age:{$gt:13}}).pretty()

db.customers.find({age:{$gte:13}}).pretty()

db.customers.find({age:{$lt:13}}).pretty()

db.customers.find({age:{$lte:13}}).pretty()

🡺To get the value from below address field

{

"\_id" : ObjectId("5ddcecd93b077caea9e8de5a"),

"first\_name" : "Mahesh",

"last\_name" : "Pandugula",

"sex" : "male",

"age" : 13,

"gender" : "male",

"address" : {

"street" : "x Road",

"city" : "LB Nagar",

"district" : "RangaReddy"

}

db.customers.find({"address.city":"LB Nagar"}).pretty()

🡺Using Or clause

db.customers.find({$or:[{gender:"female"},{age:23}]}).pretty()

🡺For sorting we use

//ascending A-Z

db.customers.find().sort({last\_name:1}).pretty()

//descending Z-A

db.customers.find().sort({last\_name:-1}).pretty()

🡺To get the count of records

db.customers.find().count()

db.customers.find({gender:"male"}).count()

🡺To get only limited records

db.customers.find({gender:"male"}).limit(2).pretty()

🡺To get only limited records in sorted way

db.customers.find({gender:"male"}).limit(2).sort({last\_name:1}).pretty()

🡺To iterate using forEach

db.customers.find().forEach(function(doc){print("Customer Name is"+doc.first\_name)})

🡺TO find one specific row

db.customers.findOne({gender:"female"})

🡺To search

db.customers.createIndex({first\_name:'text'})

db.customers.find( {$text:{ $search:"\"Kha\"" } }

🡺To get matched docs

db.customers.find({address:{$elemMatch:{street:"x Road"}}})

Ex:

{

first\_name:”john”,

last\_name:”mark’s”,

address:{

street:”4 main street”,

roadNo:5,

}

}