MongoDb

Also read below link

<https://www.tutorialspoint.com/mongodb/index.htm>

# Installation

Download MongoDb from

<https://www.mongodb.com/try/download/community>

For Mac : <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/>

For Windows :

# **Setup & Start**

* After Installation Set the Environment path

**C:\Apps\MongoDB\Server\4.4\bin** to Path Variable

As per your installation folder

* Start MongoDB Instance
  + By default after installation MongoDb Instance is Started
* Run MongoDB Shell
  + **CMD - mongo**

Note : Default configuration of mongodb instance is located on directory

**C:\Program Files\MongoDB\Server\4.4\bin\mongod.cfg**

Where we can change port and data path manually in this file,

Default Port of MongoDB is **27017**

* However, We can install Mongodb Software in Any Directory

Choosing Custom installation Settings while installing Mongodb

* Start MongoDB Instance on Different port and Different Database Directory
  + **CMD - mongod --port 9000 --dbpath="D:\data\db"**
* Run MongoDB Shell on Different Port
  + **CMD - mongo --port 9000**

# **Basic Commands**

| Commands | Use |
| --- | --- |
| **use** DATABASE\_NAME | Creating Database |
| db.**dropDatabase**() | Deleting Database |
| db | Show Current Use-in Database |
| show **dbs** | Show All Database list |
| db.**createCollection**(“collectionName”) | Create Collection or Table in Database |
| db.COLLECTION\_NAME.**drop**() | Deleting Collection from Database |
| show **collections** | For show list of Collections or Table |
| db.COLLECTION\_NAME.**insert**({  name : ‘Jemish’,  age : ‘19’,  Mobile: 9512183895  }) | For Insert data into Collection (Table) |
| db.COLLECTION\_NAME.**insert**([  { },  { }  ]) | For Insert Multiple documents into Collection Name |
| db.COLLECTION\_NAME.**insertOne**(  {  Fisrtname: ‘Jemish’,  Lastname: ‘Dhameliya’  }  ) | For inserting one document into Collection |
| db.COLLECTION\_NAME.**insertMany**([  {  first\_name : ‘Jemish’,  last\_name : ‘Dhameliya’  },  {  first\_name : ‘Maya’,  age : 20  },  ]) | For Inserting Many Records or documents at a time |
| db.COLLECTION\_NAME.**find**() | For Get all documents Inside Collection |
| db.COLLECTION\_NAME.**find**().**pretty**() | For get all documents in Prettier format |
| db.COLLECTIONNAME.**findOne**() | For Retrieve one document from the collections |

# **CRUD Operations**

| db.acc\_transaction.**find**(  {AMOUNT : '890'}  ) | Select Data where Amount = 890 |
| --- | --- |
| db.customer.**find**(  {CITY : 'Salem'}  ) | Where City **is** “Salem” |
| db.avengers.**find**({hobby : 'Chess'}) | Hobby : [“Chess”,”Abc”,”Pqr”] |
| db.customer.find(  {**$and**:  [{CITY: 'Salem'},{CUST\_TYPE\_CD: 'B'}]}  ) | Where City = Salem **AND** Cust\_type\_cd = B |
| db.customer.find({**$or** : [{CUST\_ID : '13'},{CITY : 'Wilmington'}]}) | Where Cust\_id = 12 **OR** City is = 'Wilmington' |
| db.customer.find(  {CUST\_ID : {**$gt** : 6}}  ) | Where Cust\_id **>** 6 |
| db.customer.find(  {CUST\_ID : {**$gte** : 8}}  ) | Where Cust\_id **>=** 8 |
| db.customer.find(  {CUST\_ID : {**$lt** : 5}}  ) | Where Cust\_id **<** 5 |
| db.customer.find(  {CUST\_ID : {**$lte** : 5}}  ) | Where Cust\_id **<=** 5 |
| db.customer.find(  {EMP\_ID : {**$eq**: 5}}) | Where Emp\_id **=** 5 |
| db.account.find(  STATUS : {**$ne** : 'Deactive'}}  ) | Where Status **!=** ‘Deactive’ |
| db.customer.find(  {TITLE : 'Teller'  ,**$or** : [{LAST\_NAME : 'Tucker'},{FIRST\_NAME : 'Sarah'}]  }) | Where Title = ‘Teller’ AND  (Last\_name = ‘Tucker’ or  First\_name = ‘Sarah’) |
| db.customer.find(  {AVAIL\_BALANCE : {**$in** : ["2000","3000","5000","10000"]}}  ) | Where Avail\_balance in (“2000”,”3000”,”5000”,” 10000“) |
| db.customer.find(  {AVAIL\_BALANCE : {**$nin** : ["2000","3000","5000","10000"]}}  ) | Where Avail\_balance not in (“2000”,”3000”,”5000”,” 10000“) |
| db.avengers.find({name : **/Cap/**})  db.avengers.find({hobby : {**$regex** : **/gh/**}}) | Where name like = ‘%Cap%’  Where hobby like = ‘%gh%’ |
| db.avengers.find({btctrans : {**$exists** : true}}) | Shows data which exists specified Key  Here : btctrans |
| db.account.find({**$expr** : {$gt : ["**$PENDING\_BALANCE**","**$AVAIL\_BALANCE"**]}}) | Expression  Where field a is compared to field b  With comparative operator |
| db.avengers.**updateOne**({name : 'Captain America'},{**$set** : {age : 94}}) | Update avengers **SET** age = 94 **where** name = ‘Captain America’  Only 1 document Will Update |
| db.avengers.**updateOne**({},{**$set** : {age : 94}}) | Here Where not Defined then it will update only 1st document becoz of **updateOne** |
| db.avengers.**updateMany**({},{**$set** : {designation : 'Avengers'}}) | If Field which is going to update is **not Available** then it will Insert on documents |
| db.avengers.**updateMany**({},{**$set** : {designation : 'Avengers'},**$currentDate**: {**modify\_date** : **true**} }) | $currentDate is used for insert or update current DateTime in the documents  If modify\_date is not Exist then, It will create field |
| db.avengers.**update**({designation : 'Avenger'} , {**$set** : {designation : 'Avengers'}}, {**multi** : true}) | Update command also can be used for update operation setting multi property to true or false according to Update Single document or Multiple document |
| db.avengers.**findOneAndUpdate**({name : 'Jemish'},{**$set** : {age : 15}} | It will return original document before update |
| db.account.**deleteOne**({STATUS : ‘Deactive’},1) | Deletes 1 document where status is Deactive |
| db.account.**deleteMany**({STATUS : ‘Deactive’},1) | Deletes all Document where status is Deactive |
| db.acc\_transaction.**remove**({AMOUNT : '250'}) | Delete from acc\_transaction where Amount = ‘250’ |
| db.acc\_transaction.**remove**({}) | It will remove all documents inside Collection |
| db.account.**remove**({STATUS : ‘Deactive’},1) | Second parameter is of justOne  Also {justOne : true} can be use  But not working |
| db.avengers.**find**({}**,{\_id : 0,age:0}**) | Where \_id and age is hidden |
| db.avengers.**find**(  {},  **{\_id:0,name:1,age:1,designation:1}**  ) | Where only id is hidden name,age,designation is visible |
| db.avengers.find(  {},  {\_id:0 ,name:1,age:1,designation:1})  .**limit**(4) | Limit 4 |
| db.avengers.**find**(  {},  {\_id:0 ,name:1,age:1,designation:1})  .**limit**(4)  .**skip**(2) | Skip function is used for skipping Documents |
| db.avengers.**find**(  {},  {\_id:0 ,name:1,age:1,designation:1})  .**sort**({name : 1}) | sort() for sorting fields  -1 for Desc  1 for Asc |
| db.avengers.**createIndex**({name:1}) | It Will Create an Index for Field name  On Ascending Order  1 for Ascending  -Descending |
| db.avengers.**dropIndex**({name:1}) | For drop or Remove Created index |
| db.avengers.**dropIndexes**() | Remove all Created Index except \_id |
|  |  |
|  |  |

# **User Authorization (Optional, required for production becoz of Security)**

Know More About Access Control

<https://docs.mongodb.com/manual/tutorial/enable-authentication/>

Make Admin or Root user

**use admin**

**db.createUser(**

**{**

**user: "tom",**

**pwd: passwordPrompt(), // or cleartext password**

**roles: ["root"]**

**}**

**)**

Stop Mongodb Instance by

**net stop mongodb**

Restart Mongodb Instance with enabled Authorization

**mongod --port 9000 --dbpath "D:\data" --auth**

Connect that authenticated Database

**mongo --port 9000 --authenticationDatabase "admin" -u "root" -p**

Create user and Permit to Database

**use marvel**

**db.createUser(**

**{**

**user: "powerball",**

**pwd: passwordPrompt(), // or cleartext password**

**roles: [ { role: "readWrite", db: "powerball" }]**

**}**

**)**

Connect MongoDB Shell using authorization

**mongo --port 9000 -u "Jemish" --authenticationDatabase "marvel" -p**

Authorization in Mongo shell

**db.auth("Jemish")**

Enter Password to Authenticate

As we know we can create multiple user and assign multiple role per Database

For example

**Database A** =>

**User A** : **readWrite** Access of **Database A** ,

Also **readWrite** Access of **Database B**

**Database B** =>

**User B** : readWrite Access of **Database B**

Then if **User A** wants to Connect **Database B**

Need to define **authSource=database\_name** in Connection String

Now Connect MongoDB Compass using Connection string

**mongodb://tom:jerry@localhost:27017 =>** where tom is user from admin db

Note : In Mongodb compass there is not need to pass authSource in Connection String

But in Mongoose or Another 3rd Party Library need to Define authSource in Connection String

**mongodb://Jemish:password@localhost:27017/?authSource=marvel**