**MONGODB ASSIGNMENTS……**

**After installing the mongodb zip file from** [**https://www.mongodb.com/try/download/community**](https://www.mongodb.com/try/download/community)**.**

-unzip the file

-create a folder in c-drive >Data > db

**This folder is created to store the data from mongo shell.**

-set the path of mongodb bin in environmental variable

-open cmd and type mongodb // to start the server

-again open the new cmd and type mongo to open mongo shell

**Queries to create database and inserting the data.**

-show dbs // to show all database

-use mongo\_practice // if it is not existing, than creates a new database and get switched to particular database.

**Inserting**

-db.movies.insertOne({title:"Fight Club",writer:"Chuck Palahnuiko",year:1999,actors:["Brad Pitt", "Edward Norton"]});

- db.movies.insertOne({title:"Pulp Fiction",writer:"Quentin Tarantino",year:1994,actors:["John Travolta","Uma Thurman"]});

-db.movies.insertOne({title:"Inglorious Basterds",writer:"Quentin Tarantino",year:2009,actors:["Brad Pitt","Diane Kruger","Eli Roth"]});

-db.movies.insertOne({title:"The Hobbit:The Desolation of Smaug",writer:"J.R.R.Tolkein",year:2013,franchise:"The Hobbit"});

-db.movies.insertOne({title:"Avatar"});

**Fechting**

-db.movies.find(); // to get all documents

-db.movies.find().pretty() //db.movies.find();

-db.movies.find({writer:"Quentin Tarantino"}).pretty(); // to get that particular actor

-db.movies.find({actors:"Brad Pitt"}).pretty();

-db.movies.find({franchise:"The Hobbit"}).pretty();

-

db.movies.find({year:{$gt:1900,$lt:2000}}).pretty(); // to get all 90’s released movies.

-db.movies.find({$or:[{year:{$lt:2000}},{year:{$gt:2010}}]}).pretty(); // to find the movies released before 2000 or after 2010.

**Updating**

- db.movies.update({title:"The Hobbit:An Unexpected Journey"},{$set:{synopsis:"A reluctant hobbit, Bilbo Baggins sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home- and the gold within it - from the dragon Smaug"}}); // to updating by adding the synopsis.

-db.movies.update({title:"The Hobbit:The Desolation of Smaug"},{$set:{synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebr, their homeland, from Smaug, Bilbo Baggins is in possession of a mysterious and magical ring"}});

- db.movies.update({title:"Pulp Fiction"},{$set:{actors:["John Travolta","Uma Thurman","Samuel L. Jackson"]}}); // by adding the actor.

**Text Search.**

- db.movies.find({$text:{$search:"Bilbo"}}).pretty(); // to find the document where Bilbo is present in synopsis

- db.movies.find({$text:{$search:"Gandalf"}}).pretty(); // to find the document where Gandalf is present in synopsis

- db.movies.find({$text:{$search:"Bilbo -Gandalf"}}).pretty(); // to find Bilbo should present but not in Gandalf “ - ” indicate that not in.

- db.movies.find({$text:{$search:"dwarves hobbit"}}).pretty(); // to find document consist of dwarves or hobbit.

- db.movies.find({$text:{$search:"gold and dragon"}}).pretty(); // to find the document which have gold and dragon.

**Deleting Documents**

**-** db.movies.remove({title:"Pee Wee Herman's Big Adventure"}); // to deleting the document,

-db.movies.remove({title:"Avatar"});

**Relationship.**

-use users // to switch to users collections.

- db.users.insertOne({username:"GoodGuyGreg",first\_name:"Good Guy",last\_name:"Greg"})

- db.users.insertOne({username:"ScumbagSteve",full\_name:[{first:"Scumbag",last:"Steve"}]})

- db.posts.insertOne({username:"GoodGuyGreg",title:"Passes out at party",body:"Wakes up early and cleans house"});

- db.posts.insertMany([{username:"GoodGuyGreg",title:"Steals your identity",body:"Raises your credit score"},{username:"GoodGuyGreg",title:"Reports a bug in your code",body:"Sends you a Pull Request"},{username:"ScumbagSteve",title:"Borrows something",body:"Sells it"},{username:"ScumbagSteve",title:"Forks your repo on github",body:"Sets to private"}])

- db.posts.insertOne({username:"ScumbagSteve",title:"Borrows everything",body:"The end"})

- db.comments.insertMany([{username:"GoodGuyGreg",comment:"Hope you got a good deal!",post: ObjectId("62cbf6165ad1717745f3be43")},{username:"GoodGuyGreg",comment:"What's mine is yours!", post: ObjectId("62cbf6655ad1717745f3be45")},{username:"GoodGuyGreg",comments:"Don't violate the licensing agreement",post: ObjectId("62cbf6165ad1717745f3be44")},{username:"ScumbagSteve",comment:"It still isn't clean",post:ObjectId("62cbf4995ad1717745f3be40")},{username:"ScumbagSteve",comment:"Denied your PR cause I found a hack",post:ObjectId("62cbf6165ad1717745f3be42")}]);

- db.users.find().pretty();

-db.posts.find().pretty();

-db.comments.find().pretty();

- db.posts.find({username:"GoodGuyGreg"}).pretty();

-db.posts.find({username:"ScumbagSteve"}).pretty();

- db.comments.find({username:"GoodGuyGreg"}).pretty();

- db.comments.find({username:"ScumbagSteve”}).pretty();