**MongoDB Basic Assignments**

Darren Sardonyx P

1. Insert Documents :

title : Fight Club

writer : Chuck Palahniuko

year : 1999

actors : [

Brad Pitt

Edward Norton

]

**Code: db.movies.insert({title:"Fight Club", writer: "Chuck Palahniuk", year: "1999", actors:["Brad Pitt", "Edward Norton"]})**

title : Pulp Fiction

writer : Quentin Tarantino

year : 1994

actors : [

John Travolta

Uma Thurman

]

**Code: db.movies.insert({title:”Pulp Fiction”,writer:”Quentin Tarantino,year:”1994”,actors:[“John Travolta”,”Uma Thurman”]})**

title : Inglorious Basterds

writer : Quentin Tarantino

year : 2009

actors : [

Brad Pitt

Diane Kruger

Eli Roth

]

**Code: db.movies.insert({title:”Inglorious Basterds”,writer:”Quentin Tarantino,actors:[“Brad Pitt”,”Diane Kruger”,”Eli Roth”]})**

title : The Hobbit: An Unexpected Journey

writer : J.R.R. Tolkein

year : 2012

franchise : The Hobbit

title : The Hobbit: The Desolation of Smaug

writer : J.R.R. Tolkein

**Code: db.movies.insert({title:"The Hobbit: An unexpected Journey", writer:"J.R.R. Tolkein", year:"2012",franchise:"The Hobbit"})**

title : The Hobbit: The Battle of the Five Armies

writer : J.R.R. Tolkein

year : 2012

franchise : The Hobbit

synopsis : Bilbo and Company are forced to engage in a war against an array of

combatants and keep the Lonely Mountain from falling into the hands of a rising

darkness.

**Code: db.movies.insert({title:"The Hobbit: The Battle of the Five Armies", writer:"J.R.R Tolkien", year:"2002", franchise:"The Hobbit", synopsis:"Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness."})**

title : Pee Wee Herman's Big Adventure

title : Avatar

**Code: db.movies.insert({title:"Pee Wee Herman's Big Adventures"})**

1. Query / Find documents
2. **db.movies.find()**
3. **db.movies.find({writer:"Quentin Tarantino"})**
4. **db.movies.find({actors:"Brad Pitt"})**
5. **db.movies.find({franchise:"The Hobbit"})**
6. **db.movies.find({year:{$gt:"1990", $lt:"2000"}})**
7. **db.movies.find({$or:[{year:{$gt:"2010"}},{year: {$lt:"2000"}}]})**
8. Update Documents
9. **db.movies.update({\_id:ObjectId("5c9f98e5e5c2dfe9b3729bfe")}, {$set:{synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug."}})**
10. **db.movies.update({\_id:ObjectId("5c9fa42ae5c2dfe9b3729c03")}, {$set:{synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring."}})**
11. **db.movies.update({\_id:ObjectId("5c9f983ce5c2dfe9b3729bfc")}, {$push:{actors:"Samuel L. Jackson"}})**
12. Text search

**1 db.movies.find({synopsis:{$regex:"Bilbo"}})**

**2 db.movies.find({synopsis:{$regex:"Gandalf"}})**

**3 db.movies.find({$and:[{synopsis:{$regex:"Bilbo"}}, {synopsis:{$not:/Gandalf/}}]})**

**4 db.movies.find({$or:[{synopsis:{$regex:"dwarves"}}, {synopsis:{$regex:"hobbit"}}]})**

**5 db.movies.find({$and:[{synopsis:{$regex:"gold"}}, {synopsis:{$regex:"dragon"}}]})**

1. Delete Documents

**1 db.movies.remove({\_id:ObjectId("5c9f992ae5c2dfe9b3729c00")})**

**2 db.movies.remove({\_id:ObjectId("5c9f9936e5c2dfe9b3729c01")})**

1. **Relationships**

**(Users)**

**db.users.insert({\_id:1,username:"GoodGuyGreg", first\_name:"Good Guy", last\_name:"Greg"})**

**db.users.insert({\_id:2, username:"ScumbagSteve", fullname:{first: "Scumbag", last:"Steve"}})**

**(Posts)**

**db.posts.insert({username:"GoodGuyGreg", title:"Passes out at Party", body:"Raises your credit score"})**

**db.posts.insert({ username:"GoodGuyGreg", title:"Steals your identity", body:"Raises your credit score"})**

**db.posts.insert({username:"GoodGuyGreg", title:"Reports a bug in your code", body:"Sends you a pull request"})**

**db.posts.insert({ username:"ScumbagSteve", title:"Borrows something", body:"Sells it"})**

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

**db.posts.insert({username:"ScumbagSteve", title:"Forks your repo on github", body:"Sets to private"})**

**(comments)**

**db.comments.insert({ username:"GoodGuyGreg", comment:"Hope you got a good deal!", post:ObjectId("5ca0b7e96435f98b5901f463")})**

**db.comments.insert({username:"GoodGuyGreg", comment:"What's mine is yours!", post:ObjectId("5ca0b9706435f98b5901f46a")})**

**db.comments.insert({username:"GoodGuyGreg", comment:"Don't violate the licensing agreement!", post:ObjectId("5ca0b8766435f98b5901f467")})**

**db.comments.insert({username:"ScumbagSteve", comment:"It still isn't clean", post:ObjectId("5ca0b8546435f98b5901f466")})**

**db.comments.insert({username:"ScumbagSteve", comment:"Denied your PR cause I found a hack", post:ObjectId("5ca0b9256435f98b5901f469")})**

**(Query Related collections)**

db.users.find().pretty()

db.posts.find().pretty()

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

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

db.comments.find().pretty()

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