MongoDB Lab Assignment-1

# MongoDB Exercise in mongo shell

Connect to a running mongo instance, use a database named mongo\_practice. Document all your queries in a javascript file to use as a reference.

## Insert Documents

ANSWER-1---db.movies.insertMany([{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",]},{title : "Inglorious Basterds",writer : "Quentin Tarantino",year : 2009,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",year : 2013,franchise : "The Hobbit",},{title : "The Hobbit: The Battle of the FiveArmies",writer : "J.R.R. Tolkein",year : 2012,franchise : "The Hobbit",synopsis : "Bilbo and Company are forced toengage in a war against an array of combatants and keep theLonely Mountain from falling into the hands of a risingdarkness.",},{title : "Pee Wee Herman's Big Adventure"},{title : "Avatar"}])

## Query / Find Documents

ANS-1-- db.movies.find()

ANS-2-- db.movies.find({writer:"Quentin Tarantino"})

ANS-3-- db.movies.find({actors: "Brad Pitt"})

ANS-4--db.movies.find({franchise: "The Hobbit"})

ANS-5-- db.movies.find({$and: [{year: {$gt: 1900}}, {year: {$lt: 2000}}]})

Ans-6-- db.movies.find({ $or: [{ year: { $lt: 2000 } }, { year: { $gt: 2010 } }] })

# Update Documents

ANS1-- 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." } })

ANS2-- 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 Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring." } })

# Text Search

ANS1-- db.movies.find({ synopsis: /Bilbo/g })

ANS2-- db.movies.find({ synopsis: /Gandalf/g })

ANS3-- db.movies.find({ $and: [{ synopsis: /Bilbo/g }, { synopsis: { $not: /Gandalf/g } }] })

ANS4-- db.movies.find({ synopsis: /(dwarves|hobbit)/g })

ANS5-- db.movies.find({ $and: [{ synopsis: /gold/g }, { synopsis: /dragon/g }] })

# Delete Documents

ANS1-- db.movies.deleteMany({ title: "Pee Wee Herman's Big Adventure" })

ANS2-- db.movies.deleteMany({ title: "Avatar" });

# Relationships

## Insert the following documents into a users collection

ANS-- db.users.insertMany([  
{  
username: "GoodGuyGreg",  
first\_name: "Good Guy",  
last\_name: "Greg"  
},  
{  
username: "ScumbagSteve",  
full\_name: {  
first: "Scumbag",  
last: "Steve"  
}  
}  
])

## Insert the following documents into a posts collection

ANS-- db.posts.insertMany([  
{  
username: "GoodGuyGreg",  
title: "Passes out at party",  
body: "Wakes up early and cleans house"  
},  
{  
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: "Borrows everything",  
body: "The end"  
},  
{  
username: "ScumbagSteve",  
title: "Forks your repo on github",  
body: "Sets to private"  
}  
])

## Insert the following documents into a comments collection

ANS-- db.comments.insertMany([  
{  
username: "GoodGuyGreg",  
comment: "Hope you got a good deal!",  
post: ObjectId("5f44d3a148197d7749864def")  
},  
{  
username: "GoodGuyGreg",  
comment: "Don't violate the licensing agreement!",  
post: ObjectId("5f44d3a148197d7749864df0")  
},  
{  
username: "GoodGuyGreg",  
comment: "Don't violate the licensing agreement!",  
post: ObjectId("5f44d3a148197d7749864df1")  
},  
{  
username: "ScumbagSteve",  
comment: "It still isn't clean",  
post: ObjectId("5f44d3a148197d7749864dec")  
},  
{  
username: "ScumbagSteve",  
comment: "Denied your PR cause I found a hack",  
post: ObjectId("5f44d3a148197d7749864dee")  
}  
])

## Querying related collections

ANS1-- db.users.find().pretty();

ANS2-- db.posts.find().pretty();

ANS3-- db.posts.find({ username: 'GoodGuyGreg' }).pretty();

ANS4-- db.posts.find({ username: 'ScumbagSteve' }).pretty();

ANS5-- db.comments.find().pretty();

ANS6— db.comments.find({ username: 'GoodGuyGreg' }).pretty();

ANS7— db.comments.find({ username: 'ScumbagSteve' }).pretty();

ANS8-- db.posts.aggregate([  
{  
$match: { title: 'Reports a bug in your code' }  
},  
{  
$lookup: {  
from: 'comments',  
localField: '\_id',  
foreignField: 'post',  
as: 'comments'  
}  
}  
]).pretty();