**MOVIES**

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",

]

},

{

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 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.",

},

{

title : "Pee Wee Herman's Big Adventure"

},

{

title : "Avatar"

}

])

====================================================================================

get 1

db.movies.find({})

get 2

db.movies.find({"writer":"Quentin Tarantino"}).pretty()

get 3

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

get 4

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

get 5

db.movies.find({$and:[{"year":{$lt:2000}},{"year":{$gt:1990}}]}).pretty()

get 6

db.movies.find({$or:[{"year":{$lt:2000}},{"year":{$gt:2010}}]}).pretty()

update 1

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.'}})

update 2

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.'}})

update 3

db.movies.update({'title':'Pulp Fiction'},{$addToSet:{'actors':'Samuel L. Jackson'}})

=====================================================================================

text search 1

db.movies.find({'synopsis':/Bilbo/})

text search 2

db.movies.find({'synopsis':/Gandalf/})

text search 3

db.movies.ensureIndex({synopsis:"text"})

db.movies.find({$text:{$search:"Bilbo -Gandalf"}})

text search 4

db.movies.ensureIndex({synopsis:"text"})

db.movies.find({$or:[{$text:{$search:"dwarves"},$text:{$search:"hobbit"}}]})

----------

db.movies.ensureIndex({synopsis:"text"})

db.movies.find({$text:{$search:'"dwarves hobbit"'}})

text search 5

db.movies.ensureIndex({synopsis:"text"})

db.movies.find({$and:[{$text:{$search:"gold"},$text:{$search:"dragon"}}]})

---------

db.movies.ensureIndex({synopsis:"text"})

db.movies.find({$text:{$search:'"gold" "dragon"'}})

=====================================================================================

remove1

db.movies.remove({"Pee Wee Herman's Big Adventure"})

remove 2

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

=====================================================================================

**RelationShip**

db.users.insert([

{

username : "GoodGuyGreg",

first\_name : "Good Guy",

last\_name : "Greg"

},

{

username : "ScumbagSteve",

full\_name : {

first : "Scumbag",

last : "Steve",

}

}

])

db.posts.insert([

{

\_id: 1,

username : "GoodGuyGreg",

title : "Passes out at party",

body : "Wakes up early and cleans house",

},

{

\_id: 2,

username : "GoodGuyGreg",

title : "Steals your identity",

body : "Raises your credit score",

},

{

\_id: 3,

username : "GoodGuyGreg",

title : "Reports a bug in your code",

body : "Sends you a Pull Request",

},

{

\_id: 4,

username : "ScumbagSteve",

title : "Borrows something",

body : "Sells it",

},

{

\_id: 5,

username : "ScumbagSteve",

title : "Borrows everything",

body : "The end",

},

{

\_id: 6,

username : "ScumbagSteve",

title : "Forks your repo on github",

body : "Sets to private",

},

])

db.comments.insert([

{

username : "GoodGuyGreg",

comment : "Hope you got a good deal!",

post : 4,

},

{

username : "GoodGuyGreg",

comment : "What's mine is yours!",

post : 5,

},

{

username : "GoodGuyGreg",

comment : "Don't violate the licensing agreement!",

post : 6,

},

{

username : "ScumbagSteve",

comment : "It still isn't clean",

post : 1,

},

{

username : "ScumbagSteve",

comment : "Denied your PR cause I found a hack",

post : 3

},

])

db.users.find({}).pretty()

db.posts.find({}).pretty()

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

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

db.comments.find({}).pretty()

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

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

db.comments.find({post: db.posts.findOne({title: "Reports a bug in your code"}).\_id})