MongoDB Exercise

Query / Find Documents

1. Get all documents : db.movies.find()

2. Get all documents with writer set to "Quentin Tarantino"

Query : db.movies.aggregate([{$match:{writer:"Quentin Tarantino"}}])

3. Get all documents where actors include "Brad Pitt"

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

4. Get all documents with franchise set to "The Hobbit"

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

5. Get all movies released in the 90s

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

6. Get all movies released before the year 2000 or after 2010

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

Update Documents

1. Add a synopsis to "The Hobbit: An Unexpected Journey" : "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."

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

2. Add a synopsis to "The Hobbit: The Desolation of Smaug" : "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."

Query: 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."}})

3. Add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"
Query: db.movies.update({title:"Pulp Fiction"},{$set:{"actors":"Samuel L.Jackson"}})

Text Search

1. find all movies that have a synopsis that contains the word "Bilbo"

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

2. find all movies that have a synopsis that contains the word "Gandalf"

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

3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"

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

4. find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

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

5. find all movies that have a synopsis that contains the word "gold" and "dragon"

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

Delete Documents

1. delete the movie "Pee Wee Herman's Big Adventure"

   Query: db.movies.deleteOne({title:"Pee Wee Herman's Big Adventure"})

2. delete the movie "Avatar"

   Query : db.movies.deleteOne({title:"Avatar"})

Querying related collections

1. find all users

   Query: db.users.find()

2. find all posts

   Query: db.posts.find()

3. find all posts that was authored by "GoodGuyGreg"

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

4. find all posts that was authored by "ScumbagSteve"

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

5. find all comments

   Query: db.comments.find()

6. find all comments that was authored by "GoodGuyGreg"

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

7. find all comments that was authored by "ScumbagSteve"

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

8. find all comments belonging to the post "Reports a bug in your code"

   Query: db.comments.find({post:”Reports a bug in your code”})