Mongo DB Basic Assignment

By Harsh Pradhan

Query / Find Documents

query the **movies** collection to

1. get all documents

Sol:

db.movies.find().pretty()

- *The cursor.pretty() method beautify the JSON documents or the collections within in the Mongo shell.*

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

Sol:

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

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

Sol:

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

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

Sol:

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

5. get all movies released in the 90s

Sol:

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

- *$and performs a logical AND operation on an array of* one or more *expressions*

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

Sol:

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

- *The $or operator performs a logical OR operation on an array of* two or more *<expressions> and selects the documents that satisfy* at least *one of the <expressions>.*

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."

Sol:
db.movies.updateOne({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 a single document in a collection based on a query filter.*
- *The `$set` operator replaces the value of a field with the specified value.*

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."

Sol:

db.movies.updateOne({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"

Sol:

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

Text Search

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

Sol:

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

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

Sol:

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

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

Sol:

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

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

Sol:

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

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

Sol:

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

Delete Document

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

Sol:

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

2. delete the movie "Avatar"

Sol:

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

Querying related collections

1. find all users

Sol:

db.users.find().pretty()

2. find all posts

Sol:

db.posts.find().pretty()

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

Sol:

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

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

Sol:

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

5. find all comments

Sol:

db.comments.find().pretty()

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

Sol:

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

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

Sol:

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

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

Sol:

db.comments.find({post:{$regex:"Reports a bug in your code"}})