**Query / Find Documents:**

query the movies collection to

1. get all documents

//db.movies.find().pretty();

2. get all documents with writer set to "Quentin Tarantino" //db.movies.find({writer:"Quentin Tarantino"});

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

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

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

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

5. get all movies released in the 90s

// db.movies.find({year:{$gt:1989,$lt:2000}})

6. get all movies released before the year 2000 or after 2010 //db.movie.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."

**////** 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."}},

{upsert:true});

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."

////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."}},{upsert:true});

3. add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction"

////.movie.updateOne({title:"Pulp Fiction"},{$set:{actors:"Samuel L. Jackson"}})

**Text Search**

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

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

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

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

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

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

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

///db.movies.find({$and:[{synopsis:/dwarves/},{synopsis:/hobbit/}]});

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

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

**Delete Documents**

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

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

2.delete the movie "Avatar"

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

**Querying related collections**

1.find all users

///db.users.find().pretty();

2.find all posts

///db.posts.find().pretty();

3.find all posts that was authored by "GoodGuyGreg" ///db.posts.find({username:"GoodGuyGreg"}).pretty();

4.find all posts that was authored by "ScumbagSteve" ///db.posts.find({username:"ScumbagSteve"}).pretty();

5.find all comments

///db.comments.find().pretty();

6.find all comments that was authored by "GoodGuyGreg" ///db.comments.find({username:"GoodGuyGreg"}).pretty();

7.find all comments that was authored by "ScumbagSteve" ///db.comments.find({username:"ScumbagSteve"}).pretty();

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

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