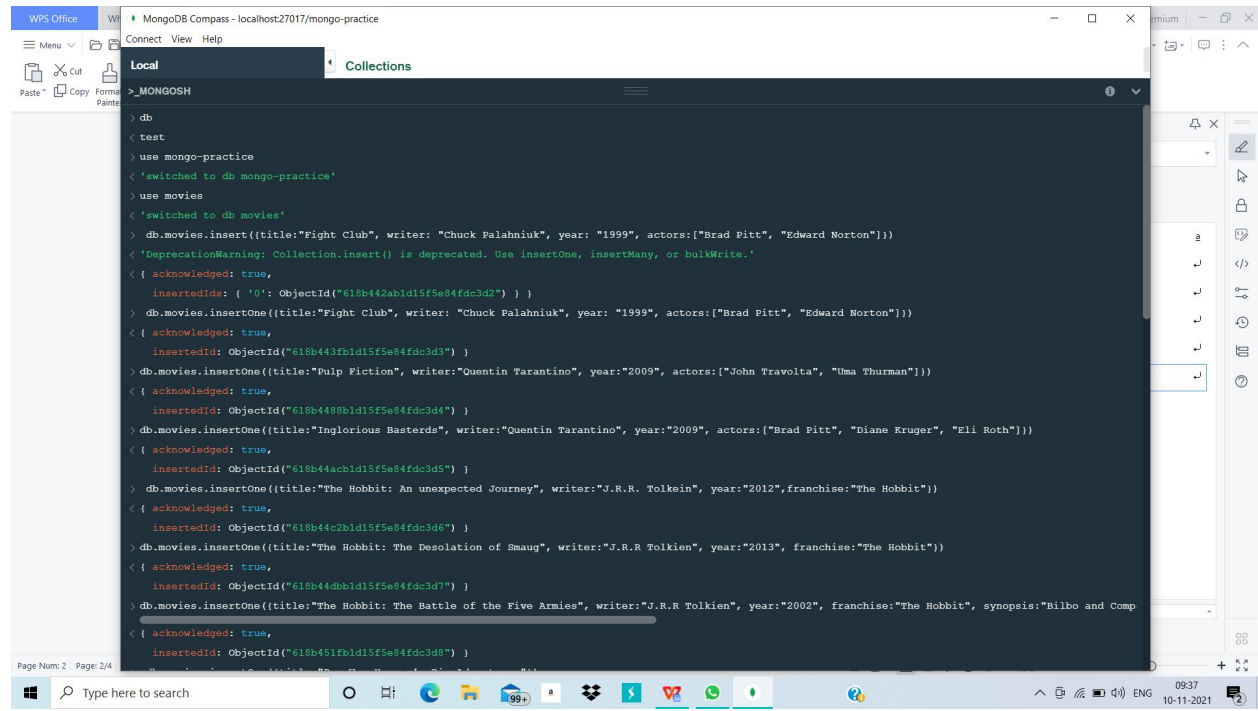


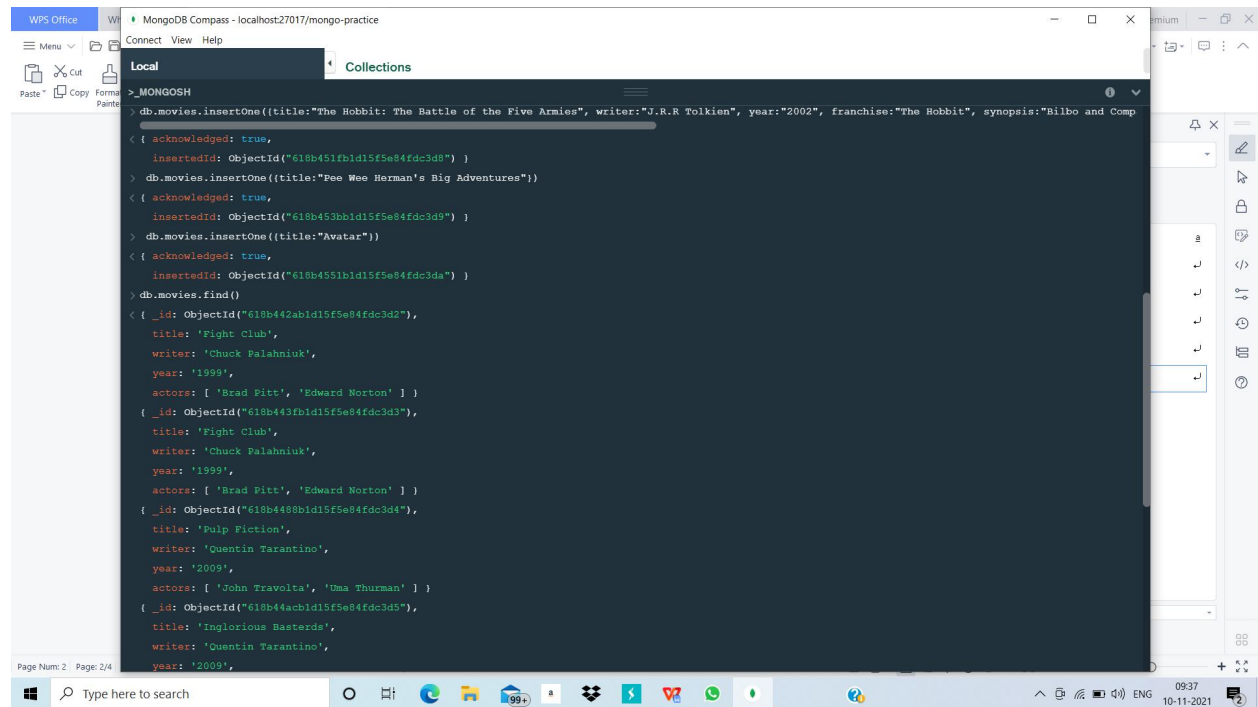
MONGODB ASSIGNMENT

Insert Documents:



The screenshot shows the MongoDB Compass interface with the 'movies' collection selected. The command window displays the following commands and their results:

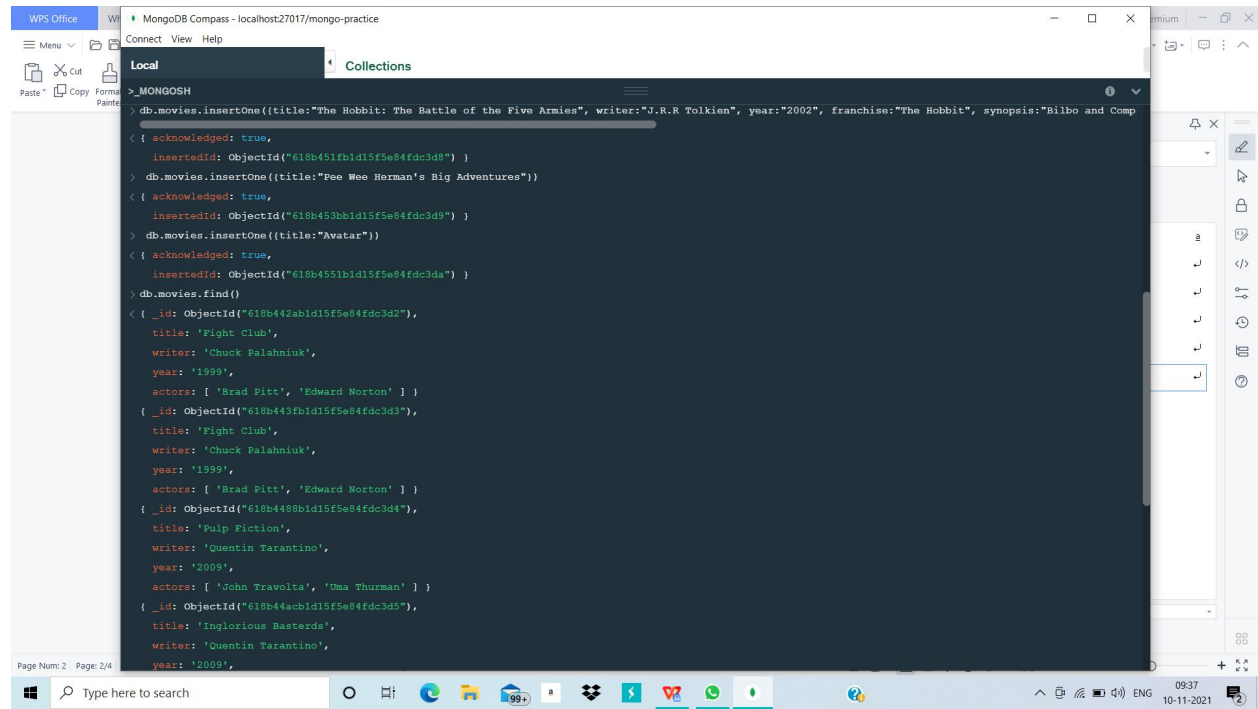
```
> use mongo-practice
'switched to db mongo-practice'
> use movies
'switched to db movies'
> db.movies.insert({title:'Fight Club', writer: 'Chuck Palahniuk', year: '1999', actors:['Brad Pitt', 'Edward Norton']})
'DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.'
{ acknowledged: true,
  insertedIds: { '0': ObjectId('618b442ab1d15f5e84fdc3d2') } }
> db.movies.insertOne({title:'Fight Club', writer: 'Chuck Palahniuk', year: '1999', actors:['Brad Pitt', 'Edward Norton']})
{ acknowledged: true,
  insertedId: ObjectId('618b443fb1d15f5e84fdc3d3') }
> db.movies.insertOne({title:'Pulp Fiction', writer:'Quentin Tarantino', year:'2009', actors:['John Travolta', 'Uma Thurman']})
{ acknowledged: true,
  insertedId: ObjectId('618b448b1d15f5e84fdc3d4') }
> db.movies.insertOne({title:'Inglorious Basterds', writer:'Quentin Tarantino', year:'2009', actors:['Brad Pitt', 'Diane Kruger', 'Eli Roth']})
{ acknowledged: true,
  insertedId: ObjectId('618b44acb1d15f5e84fdc3d5') }
> db.movies.insertOne({title:'The Hobbit: An unexpected Journey', writer:'J.R.R. Tolkien', year:'2012', franchise:'The Hobbit'})
{ acknowledged: true,
  insertedId: ObjectId('618b44c2b1d15f5e84fdc3d6') }
> db.movies.insertOne({title:'The Hobbit: The Desolation of Smaug', writer:'J.R.R. Tolkien', year:'2013', franchise:'The Hobbit'})
{ acknowledged: true,
  insertedId: ObjectId('618b44dbb1d15f5e84fdc3d7') }
> db.movies.insertOne({title:'The Hobbit: The Battle of the Five Armies', writer:'J.R.R. Tolkien', year:'2012', franchise:'The Hobbit', synopsis:'Bilbo and Comp
{ acknowledged: true,
  insertedId: ObjectId('618b451fb1d15f5e84fdc3d8') }
```



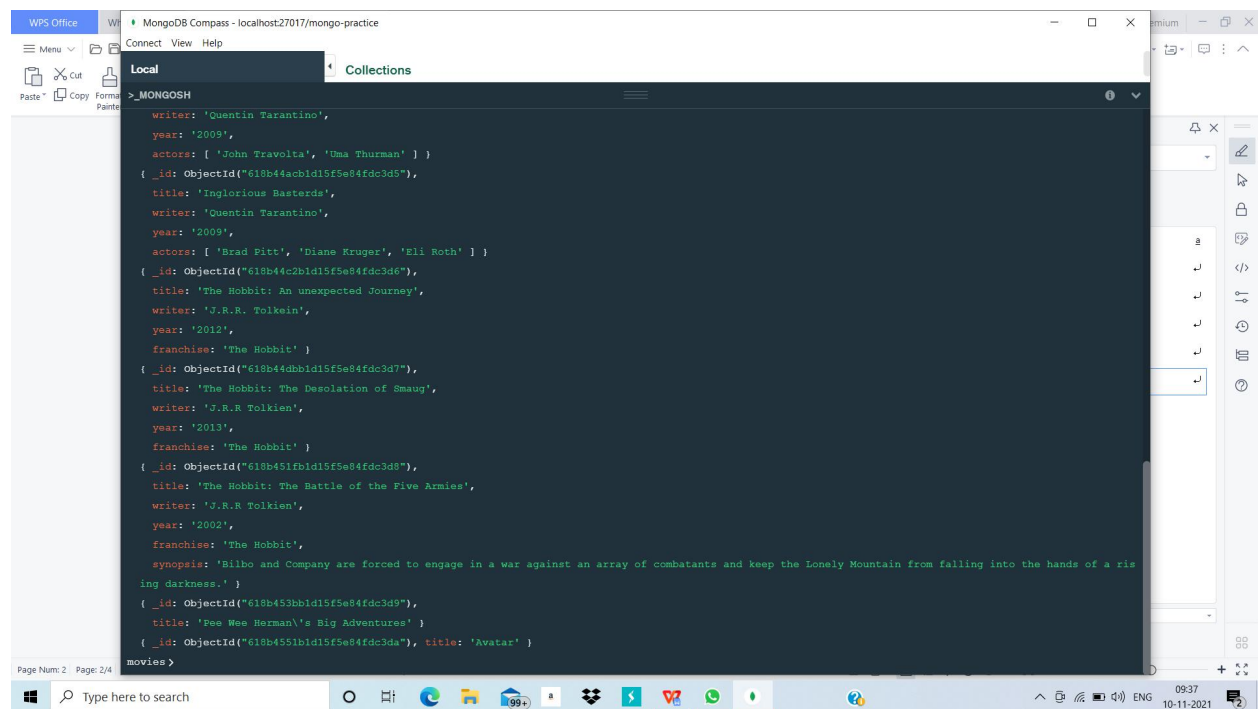
The screenshot shows the MongoDB Compass interface with the 'movies' collection selected. The command window displays the following commands and their results:

```
> db.movies.insertOne({title:'The Hobbit: The Battle of the Five Armies', writer:'J.R.R. Tolkien', year:'2012', franchise:'The Hobbit', synopsis:'Bilbo and Comp
{ acknowledged: true,
  insertedId: ObjectId('618b451fb1d15f5e84fdc3d8') }
> db.movies.insertOne({title:'Pee Wee Herman's Big Adventures'})
{ acknowledged: true,
  insertedId: ObjectId('618b453bb1d15f5e84fdc3d9') }
> db.movies.insertOne({title:'Avatar'})
{ acknowledged: true,
  insertedId: ObjectId('618b4551b1d15f5e84fdc3da') }
> db.movies.find()
{ '_id': ObjectId('618b442ab1d15f5e84fdc3d2'),
  title: 'Fight Club',
  writer: 'Chuck Palahniuk',
  year: '1999',
  actors: [ 'Brad Pitt', 'Edward Norton' ] }
{ '_id': ObjectId('618b443fb1d15f5e84fdc3d3'),
  title: 'Fight Club',
  writer: 'Chuck Palahniuk',
  year: '1999',
  actors: [ 'Brad Pitt', 'Edward Norton' ] }
{ '_id': ObjectId('618b448b1d15f5e84fdc3d4'),
  title: 'Pulp Fiction',
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'John Travolta', 'Uma Thurman' ] }
{ '_id': ObjectId('618b44acb1d15f5e84fdc3d5'),
  title: 'Inglorious Basterds',
  writer: 'Quentin Tarantino',
  year: '2009',
```

Find Documents



```
>_MONGOSH
> db.movies.insertOne({title:"The Hobbit: The Battle of the Five Armies", writer:"J.R.R Tolkien", year:"2002", franchise:"The Hobbit", synopsis:"Bilbo and Comp
< { acknowledged: true,
  insertedId: ObjectId("618b451fb1d15f5e84fdc3d8") }
> db.movies.insertOne({title:"Foe Wee Herman's Big Adventures"})
< { acknowledged: true,
  insertedId: ObjectId("618b453bb1d15f5e84fdc3d9") }
> db.movies.insertOne({title:"Avatar"})
< { acknowledged: true,
  insertedId: ObjectId("618b4551b1d15f5e84fdc3da") }
> db.movies.find()
< { _id: ObjectId("618b442ab1d15f5e84fdc3d2"),
  title: 'Fight Club',
  writer: 'Chuck Palahniuk',
  year: '1999',
  actors: [ 'Brad Pitt', 'Edward Norton' ] }
{ _id: ObjectId("618b443fb1d15f5e84fdc3d3"),
  title: 'Fight Club',
  writer: 'Chuck Palahniuk',
  year: '1999',
  actors: [ 'Brad Pitt', 'Edward Norton' ] }
{ _id: ObjectId("618b4488b1d15f5e84fdc3d4"),
  title: 'Pulp Fiction',
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'John Travolta', 'Uma Thurman' ] }
{ _id: ObjectId("618b44acb1d15f5e84fdc3d5"),
  title: 'Inglorious Basterds',
  writer: 'Quentin Tarantino',
  year: '2009',
```

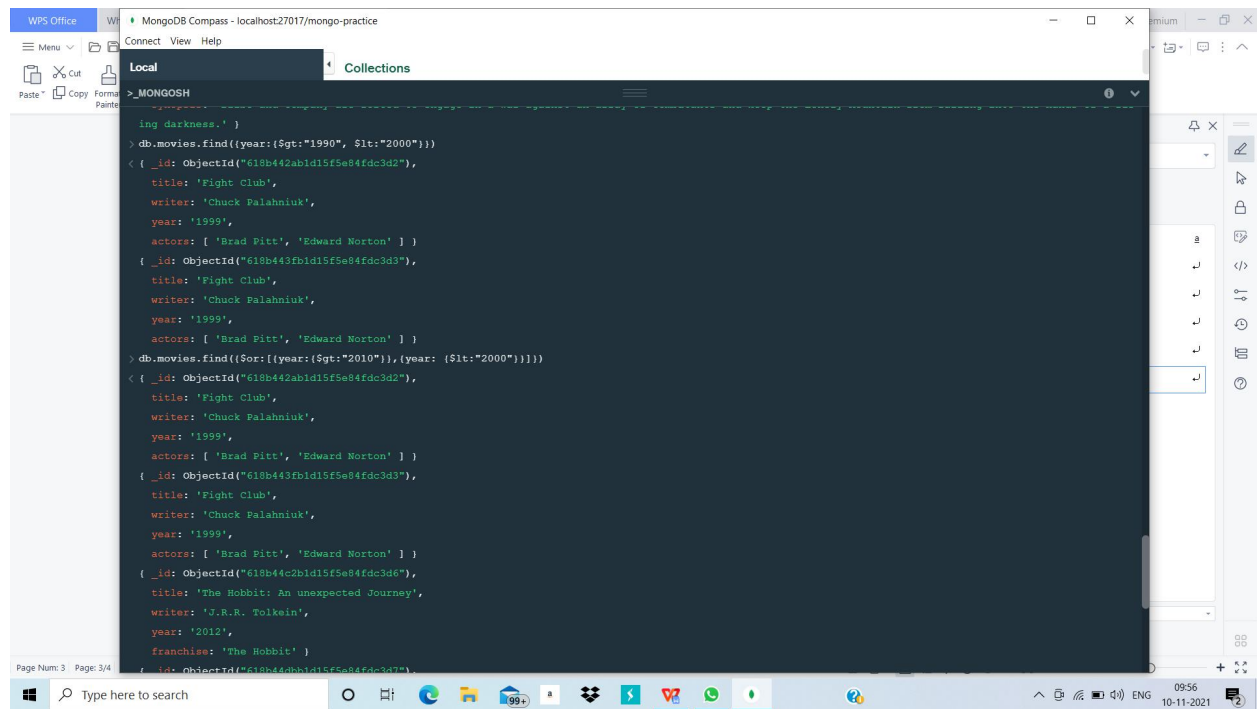
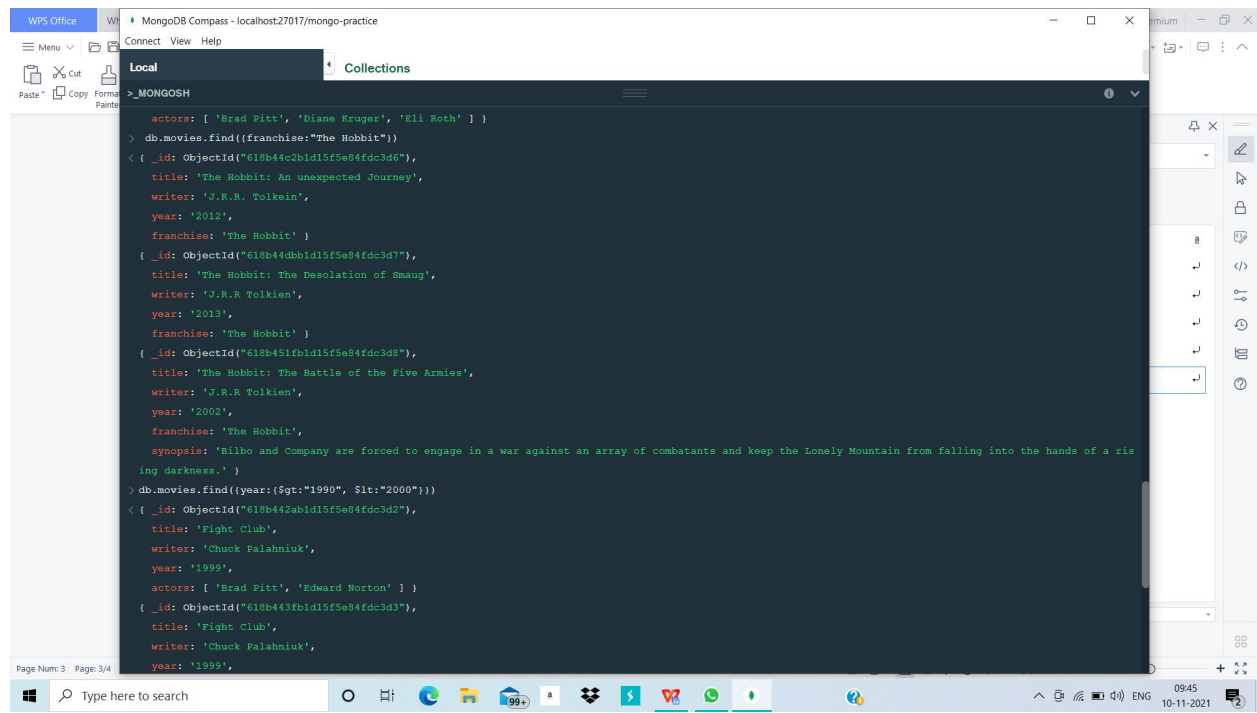


```

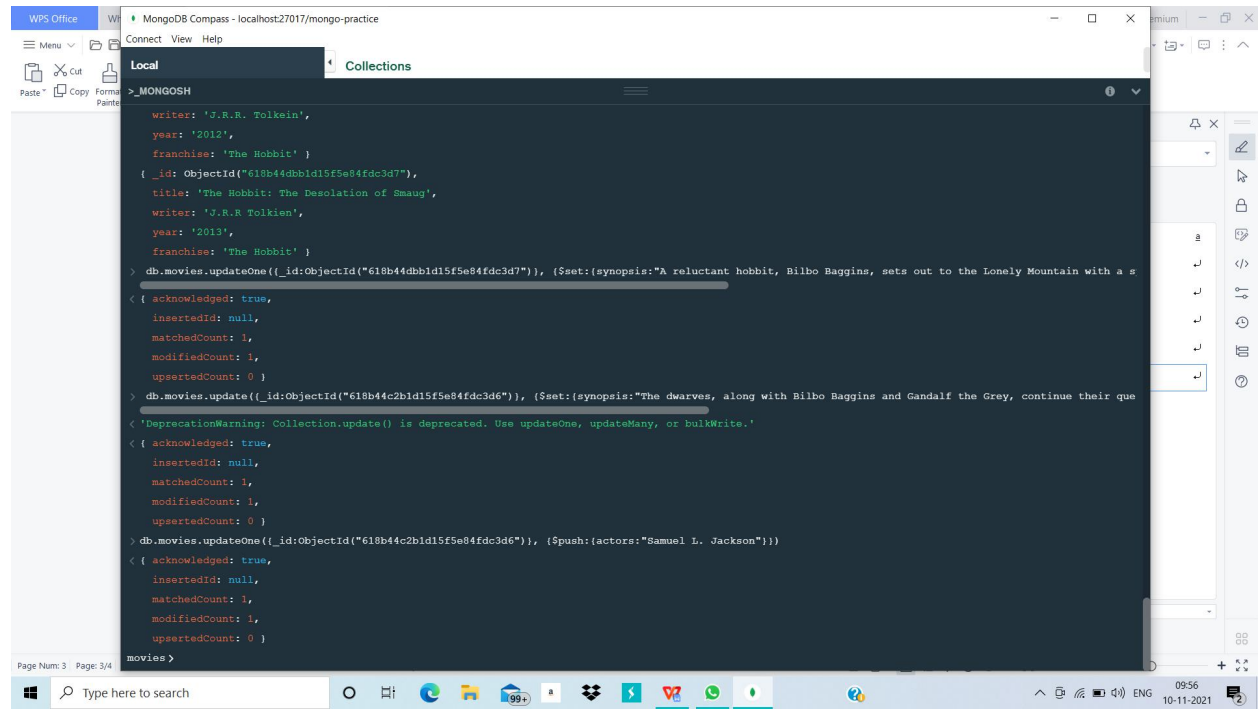
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'John Travolta', 'Uma Thurman' ] }
{ _id: ObjectId("618b44acb1d15f5e84fdc3d5"),
  title: 'Inglorious Basterds',
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
{ _id: ObjectId("618b44c2b1d15f5e84fdc3d6"),
  title: 'The Hobbit: An unexpected Journey',
  writer: 'J.R.R. Tolkein',
  year: '2012',
  franchise: 'The Hobbit' }
{ _id: ObjectId("618b44db1d15f5e84fdc3d7"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkien',
  year: '2013',
  franchise: 'The Hobbit' }
{ _id: ObjectId("618b451fb1d15f5e84fdc3d8"),
  title: 'The Hobbit: The Battle of the Five Armies',
  writer: 'J.R.R. Tolkien',
  year: '2002',
  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 ris
ing darkness.' }
{ _id: ObjectId("618b453bb1d15f5e84fdc3d9"),
  title: 'Foe Wee Herman\'s Big Adventures' }
{ _id: ObjectId("618b4551b1d15f5e84fdc3da"), title: 'Avatar' }
movies >
```

```
WPS Office | MongoDB Compass - localhost27017/mongo-practice
Connect View Help
Local Collections
>_MONGOSH
{ "_id": ObjectId("610b44acbd15f5e84fdc3d5"),
  title: 'Inglorious Basterds',
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
{ "_id": ObjectId("610b44c2bd15f5e84fdc3d6"),
  title: 'The Hobbit: An unexpected Journey',
  writer: 'J.R.R. Tolkien',
  year: '2012',
  franchise: 'The Hobbit' }
{ "_id": ObjectId("610b44dbbd15f5e84fdc3d7"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkien',
  year: '2013',
  franchise: 'The Hobbit' }
{ "_id": ObjectId("610b451fbd15f5e84fdc3d8"),
  title: 'The Hobbit: The Battle of the Five Armies',
  writer: 'J.R.R. Tolkien',
  year: '2002',
  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.' }
{ "_id": ObjectId("610b453bbd15f5e84fdc3d9"),
  title: 'Pee Wee Herman's Big Adventures' }
{ "_id": ObjectId("610b4551bd15f5e84fdc3da"), title: 'Avatar' }
> db.movies.find(writer:"Quentin Tarantino")
< { "_id": ObjectId("610b4408bd15f5e84fdc3d4"),
  title: 'Pulp Fiction',
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'John Travolta', 'Uma Thurman' ] }
{ "_id": ObjectId("610b44acbd15f5e84fdc3d5"),
  title: 'Inglorious Basterds',
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
> db.movies.find(actors:"Brad Pitt")
< { "_id": ObjectId("610b442abd15f5e84fdc3d2"),
  title: 'Fight Club',
  writer: 'Chuck Palahniuk',
  year: '1999',
  actors: [ 'Brad Pitt', 'Edward Norton' ] }
{ "_id": ObjectId("610b443fbd15f5e84fdc3d3"),
  title: 'Fight Club',
  writer: 'Chuck Palahniuk',
  year: '1999',
  actors: [ 'Brad Pitt', 'Edward Norton' ] }
{ "_id": ObjectId("610b44acbd15f5e84fdc3d5"),
  title: 'Inglorious Basterds',
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
> db.movies.find(franchise:"The Hobbit")
< { "_id": ObjectId("610b44c2bd15f5e84fdc3d6"),
  title: 'The Hobbit: An unexpected Journey',
  writer: 'J.R.R. Tolkien',
  year: '2012',
  franchise: 'The Hobbit' }
{ "_id": ObjectId("610b44dbbd15f5e84fdc3d7"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkien',
  year: '2013',
  franchise: 'The Hobbit' }
{ "_id": ObjectId("610b451fbd15f5e84fdc3d8"),
  title: 'The Hobbit: The Battle of the Five Armies',
  writer: 'J.R.R. Tolkien',
  year: '2002',
  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.' }
Page Num: 3 Page: 3/4
Type here to search
09:45 10-11-2021
```

```
WPS Office | MongoDB Compass - localhost27017/mongo-practice
Connect View Help
Local Collections
>_MONGOSH
{ "_id": ObjectId("610b4408bd15f5e84fdc3d4"),
  title: 'Pulp Fiction',
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'John Travolta', 'Uma Thurman' ] }
{ "_id": ObjectId("610b44acbd15f5e84fdc3d5"),
  title: 'Inglorious Basterds',
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
> db.movies.find(actors:"Brad Pitt")
< { "_id": ObjectId("610b442abd15f5e84fdc3d2"),
  title: 'Fight Club',
  writer: 'Chuck Palahniuk',
  year: '1999',
  actors: [ 'Brad Pitt', 'Edward Norton' ] }
{ "_id": ObjectId("610b443fbd15f5e84fdc3d3"),
  title: 'Fight Club',
  writer: 'Chuck Palahniuk',
  year: '1999',
  actors: [ 'Brad Pitt', 'Edward Norton' ] }
{ "_id": ObjectId("610b44acbd15f5e84fdc3d5"),
  title: 'Inglorious Basterds',
  writer: 'Quentin Tarantino',
  year: '2009',
  actors: [ 'Brad Pitt', 'Diane Kruger', 'Eli Roth' ] }
> db.movies.find(franchise:"The Hobbit")
< { "_id": ObjectId("610b44c2bd15f5e84fdc3d6"),
  title: 'The Hobbit: An unexpected Journey',
  writer: 'J.R.R. Tolkien',
  year: '2012',
  franchise: 'The Hobbit' }
{ "_id": ObjectId("610b44dbbd15f5e84fdc3d7"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkien',
  year: '2013',
  franchise: 'The Hobbit' }
{ "_id": ObjectId("610b451fbd15f5e84fdc3d8"),
  title: 'The Hobbit: The Battle of the Five Armies',
  writer: 'J.R.R. Tolkien',
  year: '2002',
  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.' }
Page Num: 3 Page: 3/4
Type here to search
09:45 10-11-2021
```



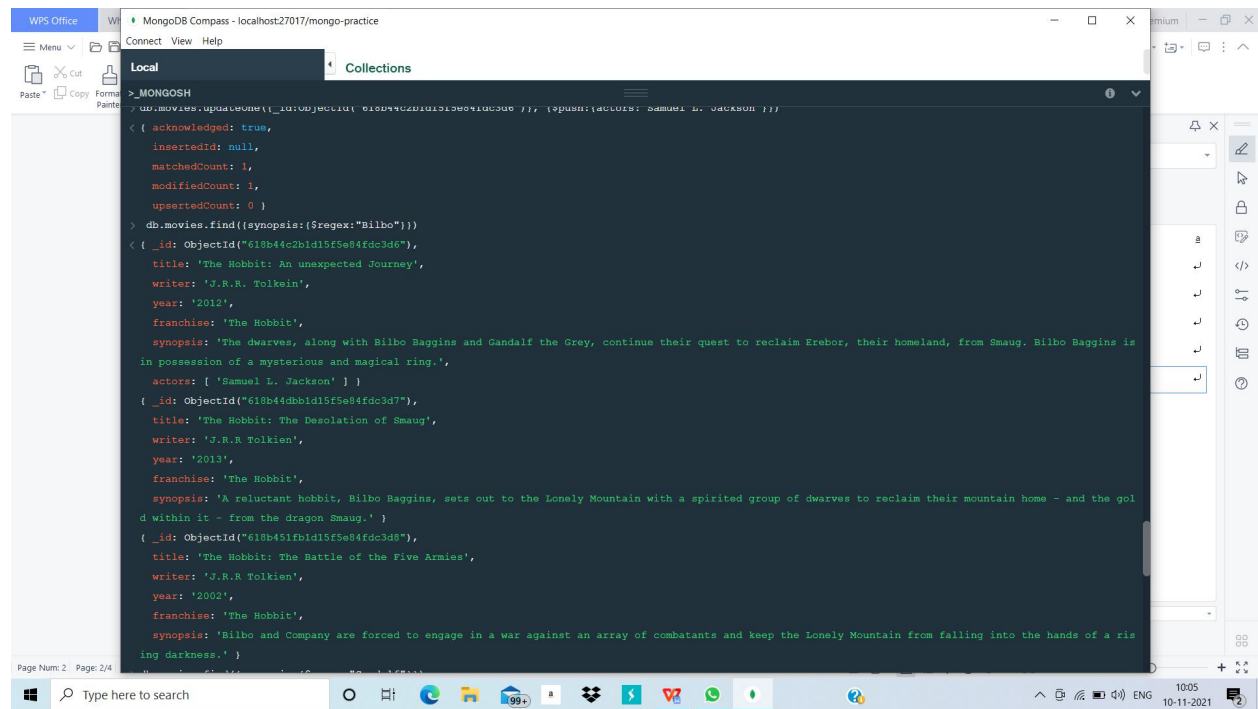
Update Documents:



The screenshot shows the MongoDB Compass interface with the 'movies' collection selected. The terminal window displays the following commands and results:

```
>_MONGOSH
writer: 'J.R.R. Tolkien',
year: '2012',
franchise: 'The Hobbit' }
{ _id: ObjectId("618b44dbb1d15f5e84fdc3d7"),
title: 'The Hobbit: The Desolation of Smaug',
writer: 'J.R.R. Tolkien',
year: '2013',
franchise: 'The Hobbit' }
> db.movies.updateOne({_id:ObjectId("618b44dbb1d15f5e84fdc3d7")}, {$set:{synopsis:"A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a s
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.movies.updateOne({_id:ObjectId("618b44c2b1d15f5e84fdc3d6")}, {$set:{synopsis:"The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their que
< 'DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.'
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.movies.updateOne({_id:ObjectId("618b44c2b1d15f5e84fdc3d6")}, {$push:{actors:"Samuel L. Jackson"}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
movies >
```

Text Search:



The screenshot shows the MongoDB Compass interface with the 'movies' collection selected. The terminal window displays the following commands and results:

```
>_MONGOSH
> db.movies.updateOne({_id:ObjectId("618b44c2b1d15f5e84fdc3d6")}, {$push:{actors:"Samuel L. Jackson"}})
< { acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
> db.movies.find({$regex:"Bilbo"})
< { _id: ObjectId("618b44c2b1d15f5e84fdc3d6"),
  title: 'The Hobbit: An unexpected Journey',
  writer: 'J.R.R. Tolkien',
  year: '2012',
  franchise: 'The Hobbit',
  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.',
  actors: [ 'Samuel L. Jackson' ] }
{ _id: ObjectId("618b44dbb1d15f5e84fdc3d7"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkien',
  year: '2013',
  franchise: 'The Hobbit',
  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 gol
d within it - from the dragon Smaug.' }
{ _id: ObjectId("618b451fb1d15f5e84fdc3d8"),
  title: 'The Hobbit: The Battle of the Five Armies',
  writer: 'J.R.R. Tolkien',
  year: '2002',
  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 ris
ing darkness.' }
```

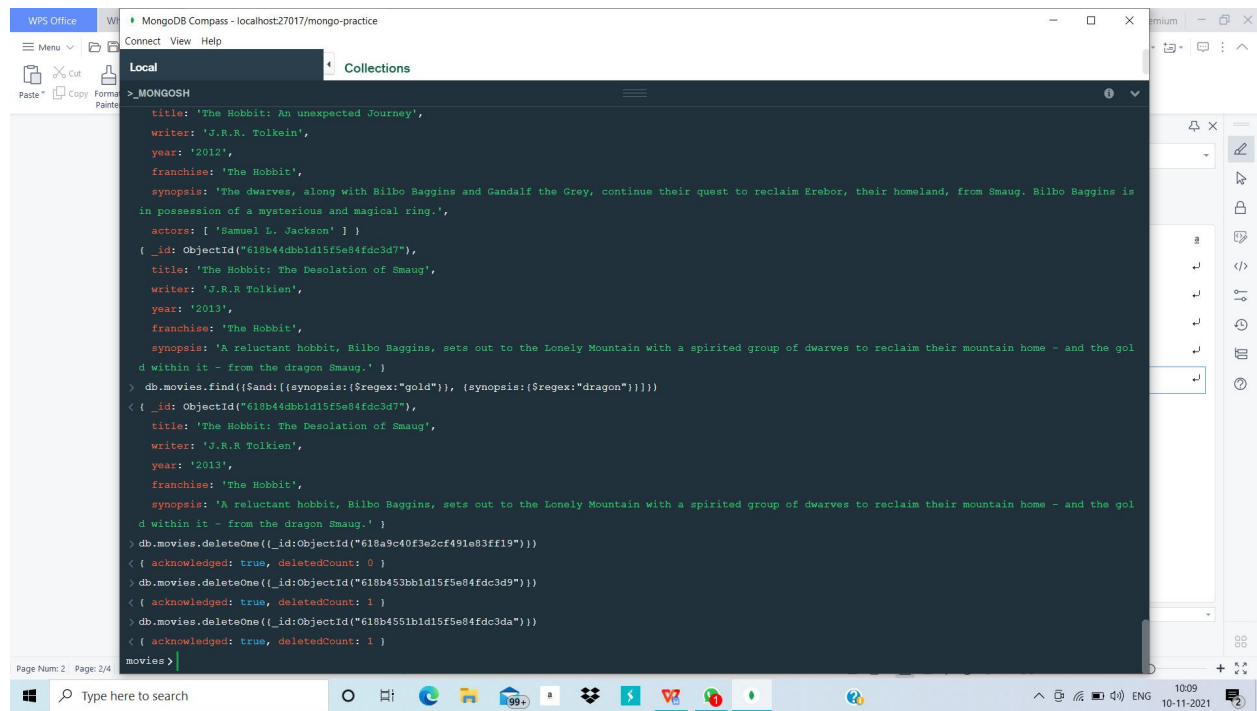

The screenshot shows the MongoDB Compass interface with a query in the MONGOSH console. The query uses the `find()` method to search for documents in the `movies` collection where the `synopsis` field contains the word "Gandalf" (using a regular expression). The results show three documents, all from the franchise "The Hobbit".

```
> MONGOSH
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.' }
> db.movies.find({$regex:{$regex:"Gandalf"}})
{ _id: ObjectId("618b44c2b1d15f5e84fdc3d6"),
  title: 'The Hobbit: An unexpected Journey',
  writer: 'J.R.R. Tolkien',
  year: '2012',
  franchise: 'The Hobbit',
  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.',
  actors: [ 'Samuel L. Jackson' ] }
> db.movies.find({$and:[{$synopsis:{$regex:"Bilbo"}}, {$synopsis:{$not:/Gandalf/}}]})
{ _id: ObjectId("618b44dbb1d15f5e84fdc3d7"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkien',
  year: '2013',
  franchise: 'The Hobbit',
  synopsis: 'A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug.' }
{ _id: ObjectId("618b451fb1d15f5e84fdc3d8"),
  title: 'The Hobbit: The Battle of the Five Armies',
  writer: 'J.R.R. Tolkien',
  year: '2014',
  franchise: 'The Hobbit',
  synopsis: 'Bilbo and Company are forced to engage in a war against an array of combatants and keep the Lonely Mountain from falling into the hands of a rising darkness.' }
> db.movies.find({$or:[{$synopsis:{$regex:"dwarves"}}, {$synopsis:{$regex:"hobbit"}}]})
{ _id: ObjectId("618b44c2b1d15f5e84fdc3d6"),
  title: 'The Hobbit: An unexpected Journey',
  writer: 'J.R.R. Tolkien',
  year: '2012',
  franchise: 'The Hobbit',
  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.',
  actors: [ 'Samuel L. Jackson' ] }
{ _id: ObjectId("618b44dbb1d15f5e84fdc3d7"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkien',
  year: '2013',
  franchise: 'The Hobbit',
  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.' }
{ _id: ObjectId("618b451fb1d15f5e84fdc3d8"),
  title: 'The Hobbit: The Battle of the Five Armies',
  writer: 'J.R.R. Tolkien',
  year: '2014',
  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.' }
```

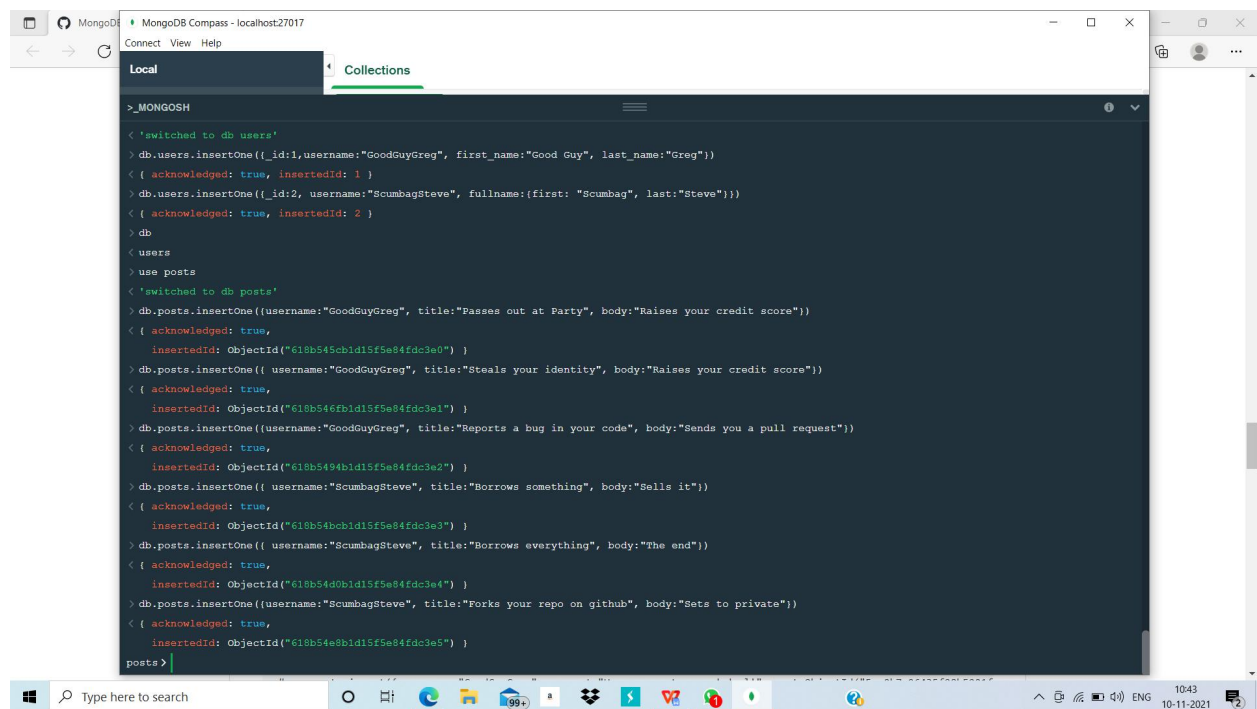
The screenshot shows the MongoDB Compass interface with a query and a delete operation. The query uses the `find()` method to search for documents in the `movies` collection where the `synopsis` field contains the word "gold" (using a regular expression). The results show one document, "The Hobbit: The Desolation of Smaug". Below the query, a `deleteOne()` operation is shown, which successfully deletes the document with `_id: ObjectId("618a9c40f3e2cf491e83ff19")`.

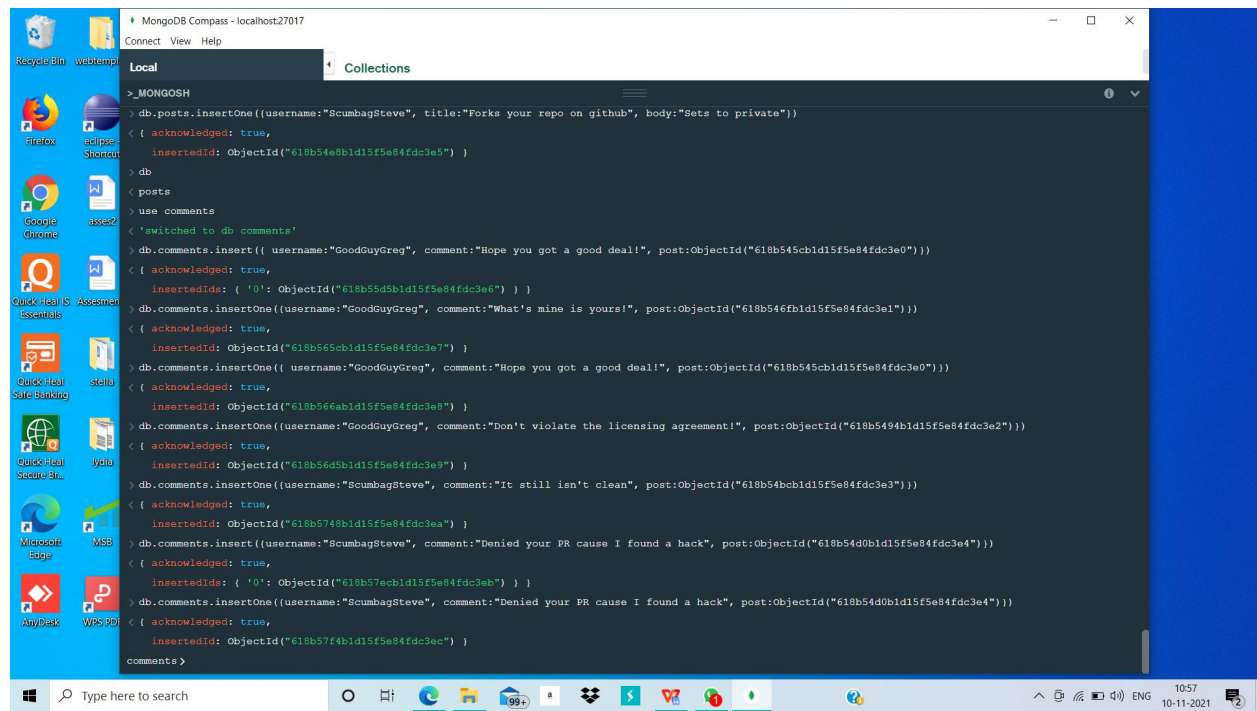
```
> db.movies.find({$or:[{$synopsis:{$regex:"dwarves"}}, {$synopsis:{$regex:"hobbit"}}]})
{ _id: ObjectId("618b44c2b1d15f5e84fdc3d6"),
  title: 'The Hobbit: An unexpected Journey',
  writer: 'J.R.R. Tolkien',
  year: '2012',
  franchise: 'The Hobbit',
  synopsis: 'The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring.',
  actors: [ 'Samuel L. Jackson' ] }
{ _id: ObjectId("618b44dbb1d15f5e84fdc3d7"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkien',
  year: '2013',
  franchise: 'The Hobbit',
  synopsis: 'A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug.' }
> db.movies.find({$and:[{$synopsis:{$regex:"gold"}}, {$synopsis:{$regex:"dragon"}}]})
{ _id: ObjectId("618b44dbb1d15f5e84fdc3d7"),
  title: 'The Hobbit: The Desolation of Smaug',
  writer: 'J.R.R. Tolkien',
  year: '2013',
  franchise: 'The Hobbit',
  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.' }
> db.movies.deleteOne({_id:ObjectId("618a9c40f3e2cf491e83ff19")})
{ acknowledged: true, deletedCount: 1 }
> db.movies.deleteOne({_id:ObjectId("")})
```

Delete Documents:

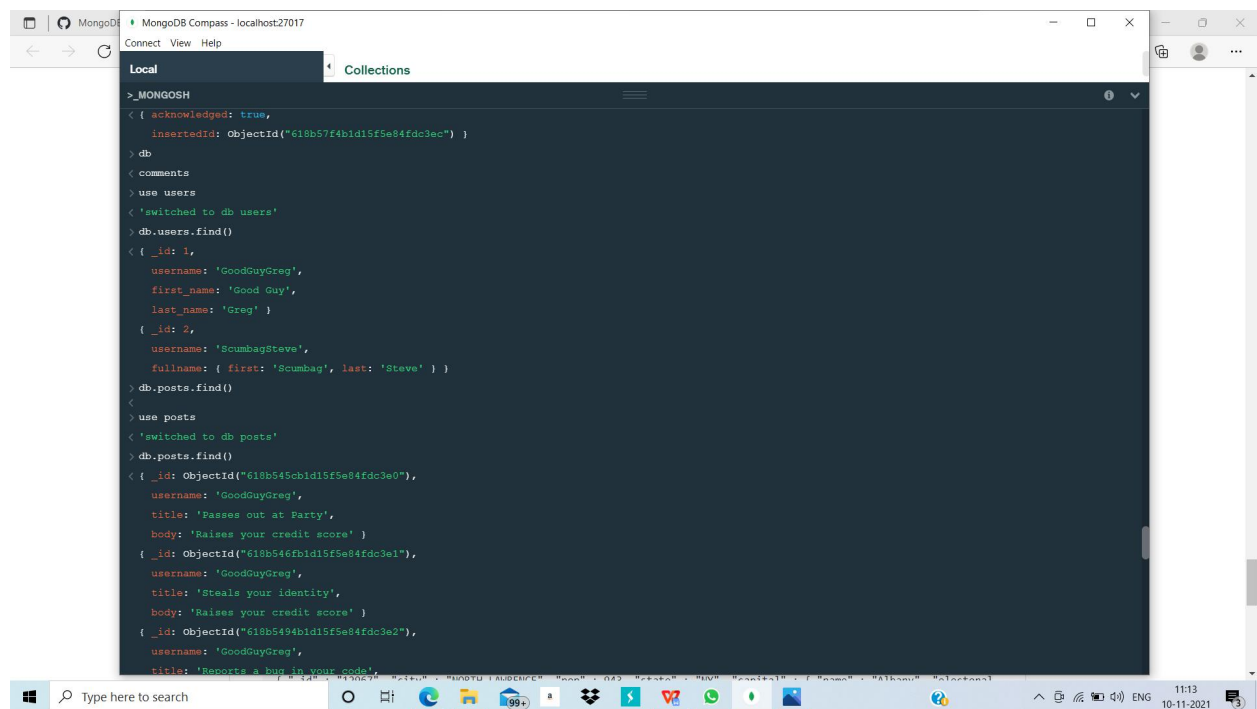


Relationships:





Querying Related Collections:




```
MongoDB Compass - localhost:27017
Connect View Help

Local Collections

>_MONGOSH
body: 'Raises your credit score' }
{ _id: ObjectId("618b5494b1d15f5e84fdc3e2"),
  username: 'GoodGuyGreg',
  title: 'Reports a bug in your code',
  body: 'Sends you a pull request' }
{ _id: ObjectId("618b54bcb1d15f5e84fdc3e3"),
  username: 'ScumbagSteve',
  title: 'Borrows something',
  body: 'Sells it' }
{ _id: ObjectId("618b54d0b1d15f5e84fdc3e4"),
  username: 'ScumbagSteve',
  title: 'Borrows everything',
  body: 'The end' }
{ _id: ObjectId("618b54e8b1d15f5e84fdc3e5"),
  username: 'ScumbagSteve',
  title: 'Forks your repo on github',
  body: 'Sets to private' }
> db.posts.find({username:"GoodGuyGreg"})
< { _id: ObjectId("618b545cb1d15f5e84fdc3e0"),
  username: 'GoodGuyGreg',
  title: 'Passes out at Party',
  body: 'Raises your credit score' }
{ _id: ObjectId("618b546fb1d15f5e84fdc3e1"),
  username: 'GoodGuyGreg',
  title: 'Steals your identity',
  body: 'Raises your credit score' }
{ _id: ObjectId("618b5494b1d15f5e84fdc3e2"),
  username: 'GoodGuyGreg',
  title: 'Reports a bug in your code',
```

```
MongoDB Compass - localhost:27017
Connect View Help

Local Collections

>_MONGOSH
body: 'Raises your credit score' }
{ _id: ObjectId("618b5494b1d15f5e84fdc3e2"),
  username: 'GoodGuyGreg',
  title: 'Reports a bug in your code',
  body: 'Sends you a pull request' }
> db.posts.find({username:"ScumbagSteve"})
< { _id: ObjectId("618b54bcb1d15f5e84fdc3e3"),
  username: 'ScumbagSteve',
  title: 'Borrows something',
  body: 'Sells it' }
{ _id: ObjectId("618b54d0b1d15f5e84fdc3e4"),
  username: 'ScumbagSteve',
  title: 'Borrows everything',
  body: 'The end' }
{ _id: ObjectId("618b54e8b1d15f5e84fdc3e5"),
  username: 'ScumbagSteve',
  title: 'Forks your repo on github',
  body: 'Sets to private' }
> use comments
< 'switched to db comments'
> db.comments.find()
< { _id: ObjectId("618b55d5b1d15f5e84fdc3e6"),
  username: 'GoodGuyGreg',
  comment: 'Hope you got a good deal!',
  post: ObjectId("618b545cb1d15f5e84fdc3e0") }
{ _id: ObjectId("618b565cb1d15f5e84fdc3e7"),
  username: 'GoodGuyGreg',
  comment: 'What's mine is yours!',
  post: ObjectId("618b546fb1d15f5e84fdc3e1") }
```

```
MongoDB Compass - localhost:27017
Connect View Help

Local Collections

>_MONGOSH
{
  username: 'GoodGuyGreg',
  comment: 'Hope you got a good deal!',
  post: ObjectId("618b545cb1d15f5e84fdc3e0")
}
{
  _id: ObjectId("618b565cb1d15f5e84fdc3e7"),
  username: 'GoodGuyGreg',
  comment: 'What\'s mine is yours!',
  post: ObjectId("618b546fb1d15f5e84fdc3e1")
}
{
  _id: ObjectId("618b56ab1d15f5e84fdc3e8"),
  username: 'GoodGuyGreg',
  comment: 'Hope you got a good deal!',
  post: ObjectId("618b545cb1d15f5e84fdc3e0")
}
{
  _id: ObjectId("618b56d5b1d15f5e84fdc3e9"),
  username: 'GoodGuyGreg',
  comment: 'Don\'t violate the licensing agreement!',
  post: ObjectId("618b5494b1d15f5e84fdc3e2")
}
> db.comments.find({username:'ScumbagSteve'})
{
  _id: ObjectId("618b5748b1d15f5e84fdc3ea"),
  username: 'ScumbagSteve',
  comment: 'It still isn\'t clean',
  post: ObjectId("618b54bcb1d15f5e84fdc3e3")
}
{
  _id: ObjectId("618b57ecb1d15f5e84fdc3eb"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found a hack',
  post: ObjectId("618b54d0b1d15f5e84fdc3e4")
}
{
  _id: ObjectId("618b57f4b1d15f5e84fdc3ec"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found a hack',
  post: ObjectId("618b54d0b1d15f5e84fdc3e4")
}
comments>
```

```
MongoDB Compass - localhost:27017
Connect View Help

Local Collections

>_MONGOSH
{
  _id: ObjectId("618b56d5b1d15f5e84fdc3e9"),
  username: 'GoodGuyGreg',
  comment: 'Don\'t violate the licensing agreement!',
  post: ObjectId("618b5494b1d15f5e84fdc3e2")
}
> db.comments.find({username:'ScumbagSteve'})
{
  _id: ObjectId("618b5748b1d15f5e84fdc3ea"),
  username: 'ScumbagSteve',
  comment: 'It still isn\'t clean',
  post: ObjectId("618b54bcb1d15f5e84fdc3e3")
}
{
  _id: ObjectId("618b57ecb1d15f5e84fdc3eb"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found a hack',
  post: ObjectId("618b54d0b1d15f5e84fdc3e4")
}
{
  _id: ObjectId("618b57f4b1d15f5e84fdc3ec"),
  username: 'ScumbagSteve',
  comment: 'Denied your PR cause I found a hack',
  post: ObjectId("618b54d0b1d15f5e84fdc3e4")
}
> use posts
< 'switched to db posts'
> db.posts.find({title:'Reports a bug in your code'})
<
> db
< posts
> db.posts.find({title:'Reports a bug in your code'})
{
  _id: ObjectId("618b5494b1d15f5e84fdc3e2"),
  username: 'GoodGuyGreg',
  title: 'Reports a bug in your code',
  body: 'Sends you a pull request'
}
posts>
```