05 Practice

MongoDB Practice

Setup MongoDB Exercise in mongo shell

Connect to a running mongo instance, use a database named `mongo_practice`.

Insert Document

Task Insert the following data into movies collection

```
[
    {
        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"}

]
```

Answer

```
db.movies.insertMany(givenArray)
```

Query / Find Documents

Query Tasks Query the `movies` collection to:

1. get all documents
2. get all documents with `writer` set to "Quentin Tarantino"
3. get all documents where `actors` include "Brad Pitt"
4. get all documents with `franchise` set to "The Hobbit"
5. get all movies released in the 90s
6. get all movies released before the year 2000 or after 2010

Query / Find Documents Answer

1. Get all documents

```
db.movies.find()
```

2. Get documents with writer "Quentin Tarantino"

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

3. Get documents where actors include "Brad Pitt"

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

4. Get documents with franchise "The Hobbit"

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

5. Get movies released in the 90s

```
db.movies.find({year: {$gte: 1990, $lt: 2000}})
```

6. Get movies released before 2000 or after 2010

```
db.movies.find(
    {
```

```
        $or: [
            {year: {$lt: 2000}},
            {year: {$gt: 2010}}
        ]
    }
)
```

Update Documents

Update Tasks

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

Update Documents Answer

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

```
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."}}
)
```

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

```
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 actor "Samuel L. Jackson" to "Pulp Fiction"

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

Text Search

Text Search Tasks

1. find all movies that have a synopsis that contains the word "Bilbo"
2. find all movies that have a synopsis that contains the word "Gandalf"
3. find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf"
4. find all movies that have a synopsis that contains the word "dwarves" or "hobbit"

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

Delete Documents

Delete Tasks

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

Delete Documents Answer

1. Delete "Pee Wee Herman's Big Adventure"

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

2. Delete "Avatar"

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

Insert Documents - Persons Collection

Task Insert the following data into `persons` collection

```
[
    {
        name: "Sue Ramsey",
        age: 35,
        country: "Algeria"
    },
    {
        name: "Leah McLaughlin",
        age: 29,
        country: "Japan"
    },
    {
        name: "Luke Chandler",
        age: 20,
        country: "Argentina"
    },
    {
        name: "Douglas Harrington",
        age: 29,
        country: "United Kingdom"
    },
    {
        name: "Esther Lamb",
        age: 30,
        country: "Italy"
    },
    {
        name: "Viola Gomez",
        age: 34,
        country: "United States"
    },
```

```
    {
        name: "Beatrice Webster",
        age: 17,
        country: "India"
    },
    {
        name: "Steve Webster",
        age: 33,
        country: "Japan"
    },
    {
        name: "Mollie Estrada",
        age: 24,
        country: "Russia"
    },
    {
        name: "Susie Fisher",
        age: 28,
        country: "Canada"
    },
    {
        name: "Loretta Reynolds",
        age: 27,
        country: "United States"
    },
    {
        name: "Lou Campbell",
        age: 34,
        country: "France"
    },
    {
        name: "Marion Henry",
        age: 33,
        country: "India"
    },
    {
        name: "Daniel Thomas",
        age: 20,
        country: "Italy"
    },
    {
        name: "Sarah Palmer",
        age: 18,
        country: "India"
    },
    {
        name: "Emilie Stevenson",
        age: 35,
        country: "Japan"
    },
    {
        name: "Manuel Knight",
        age: 33,
        country: "United Kingdom"
    },
    {
        name: "Nathan Howell",
        age: 29,
```

```
        country: "Germany"
    },
    {
        name: "Brent Thornton",
        age: 21,
        country: "United States"
    },

]
```

Answer

```
db.persons.insertMany(givenData)
```

Questions - Persons Collection

Practice Questions

1. Find all persons between the age between 20 and 30.
2. Find all persons except those whose name isn't in ["Nathan Howell", "Marion Henry", "Emilie Stevenson"]
3. Find all persons in one of these countries ["Canada", "United States", "France"]
4. Set every person's gender to male.
5. Remove the field gender from every document.
6. Rename the name field name to fullName.
7. Rename the fullName field to name.
8. Find the count of documents.
9. How many people live in each country? The output should only contain the country and count.
10. How many people live in India.
11. Find the average age of each country. The output should only contain country and averageAge
12. Find the people belonging to each country. The output should be like {country: "name of the country", persons: [names of people]}

Answer

1. Find persons between age 20 and 30

```
db.persons.find({age: {$gt: 20, $lt: 30}})
```

2. Find persons except specified names

```
db.persons.find({name: {$nin: ["Nathan Howell", "Marion Henry", "Emilie Stevenson"]}})
```

3. Find persons in specified countries

```
db.persons.find({country: {$in: ["Canada", "United States", "France"]}})
```

4. Set gender to male for all persons

```
db.persons.updateMany(
    {},
```

```
    {$set: {gender: "male"}}
)
```

5. Remove gender field from all documents

```
db.persons.updateMany(
    {},
    {$unset: {gender: true}}
)
```

6. Rename name field to fullName

```
db.persons.updateMany(
    {},
    {$rename: {name: "fullName"}}
)
```

7. Rename fullName field back to name

```
db.persons.updateMany(
    {},
    {$rename: {fullName: "name"}}
)
```

8. Count all documents

```
db.persons.countDocuments()
```

9. Count people by country

```
db.persons.aggregate([
    {
        $group:{
            _id: "$country",
            count: {$count: {}}
        }
    }
])
```

10. Count people in India

```
db.persons.find({country: "India"}).count()
```

11. Average age by country

```
db.persons.aggregate([
    {
        $group: {
            _id: "$country",
            averageAge: {$avg: "$age"}
        }
    }
])
```

12. Group people by country

```
db.persons.aggregate([
    {
        $group: {
            _id: "$country",
        persons: {$push: "$name"}
        }
    }
])
```

Relationships

Insert Users Collection

Task Insert the following documents into a `users` collection

```
[
    {   username : "GoodGuyGreg",
        first_name : "Good Guy",
        last_name : "Greg"
    },
    {
        username : "ScumbagSteve",
        first_name : "Scumbag",
        last_name : "Steve"
    }
]
```

Answer

```
db.users.insertMany(givenData)
```

Insert Posts Collection

Task Insert the following array into a `posts` collection

```
[
    {
        username : "GoodGuyGreg",
        title : "Passes out at party",
        body : "Wakes up early and cleans house"
    },
    {
        username : "GoodGuyGreg",
        title : "Steals your identity",
        body : "Raises your credit score"
    },
    {
        username : "GoodGuyGreg",
        title : "Reports a bug in your code",
        body : "Sends you a Pull Request"
    },
```

```
    {
        username : "ScumbagSteve",
        title : "Borrows something",
        body : "Sells it"
    },
    {
        username : "ScumbagSteve",
        title : "Borrows everything",
        body : "The end"
    },
    {
        username : "ScumbagSteve",
        title : "Forks your repo on github",
        body : "Sets to private"
    }
]
```

Answer

```
db.posts.insertMany(givenData)
```

Insert Comments Collection

Task Insert the following documents into a `comments` collection

Comment 1:

```
{
    username : "GoodGuyGreg",
    comment : "Hope you got a good deal!",
    post : [post_obj_id]
}
```

where [post_obj_id] is the ObjectId of the `posts` document: "Borrows something"

Comment 2:

```
{
    username : "GoodGuyGreg",
    comment : "What's mine is yours!",
    post : [post_obj_id]
}
```

where [post_obj_id] is the ObjectId of the `posts` document: "Borrows everything"

Comment 3:

```
{
    username : "GoodGuyGreg",
    comment : "Don't violate the licensing agreement!",
    post : [post_obj_id]
}
```

where [post_obj_id] is the ObjectId of the `posts` document: "Forks your repo on github"

Comment 4:

```
{
    username : "ScumbagSteve",
    comment : "It still isn't clean",
    post : [post_obj_id],
}
```

where [post_obj_id] is the ObjectId of the `posts` document: "Passes out at party"

Comment 5:

```
{
    username : "ScumbagSteve",
    comment : "Denied your PR cause I found a hack",
    post : [post_obj_id]
}
```

where [post_obj_id] is the ObjectId of the `posts` document: "Reports a bug in your code"

Important Don't forget to copy the post_obj_id from the posts collection

Answer

```
db.posts.insertMany(givenData)
```

Querying Related Collections

Query Tasks

1. find all users
2. find all posts
3. find all posts that was authored by "GoodGuyGreg"
4. find all posts that was authored by "ScumbagSteve"
5. find all comments
6. find all comments that was authored by "GoodGuyGreg"
7. find all comments that was authored by "ScumbagSteve"
8. find all comments belonging to the post "Reports a bug in your code"
9. Set the gender to male for every user.
10. Find the documents and add a new field called full_name to the documents.
11. Find all articles and comments related to a user.

Querying Related Collections Answer

1. Find all users

```
db.users.find()
```

2. Find all posts

```
db.posts.find()
```

3. Find posts by "GoodGuyGreg"

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

4. Find posts by "ScumbagSteve"

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

5. Find all comments

```
db.comments.find()
```

6. Find comments by "GoodGuyGreg"

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

7. Find comments by "ScumbagSteve"

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

8. Find comments for "Reports a bug in your code" post

```
db.posts.aggregate([
    {
        $match: {title: "Reports a bug in your code"}
    },
    {
        $project: {title: 1}
    },
    {
        $lookup: {
            from: "comments",
            localField: "_id",
            foreignField: "post",
            as: "comments"
        }
    },
    {
        $project: {_id: 0}
    }
])
```

9. Set gender to male for all users

```
db.users.updateMany(
    {},
    {$set: {gender: "male"}}
)
```

10. Add full_name field to documents

```
db.users.aggregate([
    {
        /**
         * newField: The new field name.
```

```
         * expression: The new field expression.
         */
        $addFields: {
          full_name: {$concat: ["$first_name", " ", "$last_name"]}
        }
    }
])
```

11. Find all articles and comments related to a user

```
db.users.aggregate([
    {
        $lookup: {
          from: "posts",
          localField: "username",
          foreignField: "username",
          as: "posts"
        }
    },
    {
        $lookup: {
          from: "comments",
          localField: "username",
          foreignField: "username",
          as: "comments"
        }
    },
    {
        $project: {
            username: 1,
            "posts.title": 1,
            "posts.body": 1,
            "comments.comment": 1,
            "comments.post": 1
        }
    }
])
```