

A) Σε αυτή την εργασία, θα παρουσιάσουμε δύο εναλλακτικές δομές βάσης δεδομένων MongoDB για ένα κοινωνικό δίκτυο, λαμβάνοντας υπόψη τις απαιτήσεις που τέθηκαν. Θα εστιάσουμε στη δομή των συλλογών και στα πεδία που θα περιέχουν, αιτιολογώντας τις επιλογές μας.

Απαιτήσεις

Οι βασικές απαιτήσεις που πρέπει να καλύπτει η βάση δεδομένων είναι οι εξής:

1. Αποθήκευση πληροφοριών για χρήστες, ομάδες χρηστών, αναρτήσεις και πολυμέσα.
2. Υλοποίηση λειτουργιών όπως:
3. Ανάκτηση ομάδων ενός χρήστη.
4. Ανάκτηση πολυμέσων που έχει κοινοποιήσει ένας χρήστης.
5. Ανάκτηση αναρτήσεων εντός συγκεκριμένου χρονικού διαστήματος.
6. Ανάκτηση σχολίων και απαντήσεων σε μια ανάρτηση.
7. Ανάκτηση χρηστών που σχολίασαν σε μια ανάρτηση.
8. Ανάκτηση χρηστών μιας ομάδας.
9. Ανάκτηση χρήστη που δημιούργησε μια ανάρτηση.
10. Ανάκτηση χρηστών που έκαναν like σε ένα πολυμέσο.
11. Ανάκτηση απαντήσεων σε ένα σχόλιο.

Εναλλακτική 1: Σχεδίαση με Αναφορές

Σε αυτή την προσέγγιση, θα υλοποιήσουμε αναφορές μεταξύ των συλλογών για να συνδέσουμε σχετικά δεδομένα.

Συλλογή "Χρήστες":

- `"_id"`: ObjectId(),
- `"username"`: "username",
- `"groups"`: ["group_id1", "group_id2"], // Αναφορές IDs ομάδων
- `"posts"`: ["post_id1", "post_id2"], // Αναφορές IDs αναρτήσεων
- `"media"`: ["media_id1", "media_id2"], // Αναφορές IDs πολυμέσων
- `"comments"`: ["comment_id1", "comment_id2"] // Αναφορές IDs σχολίων

Συλλογή "Αναρτήσεις":

- `"_id"`: ObjectId(),
- `"user_id"`: "user_id", // Αναφορά ID χρήστη
- `"content"`: "post_content",
- `"timestamp"`: "timestamp"

Συλλογή "Πολυμέσα":

- `"_id": ObjectId()`,
- `"user_id": "user_id",` // Αναφορά ID χρήστη
- `"type": "media_type",`
- `"url": "media_url"`

Συλλογή "Σχόλια":

- `"_id": ObjectId()`,
- `"post_id": "post_id",` // Αναφορά ID ανάρτησης
- `"user_id": "user_id",` // Αναφορά ID χρήστη
- `"content": "comment_content",`
- `"timestamp": "timestamp"`

Η χρήση αναφορών IDs στις συλλογές "Χρήστες", "Αναρτήσεις", "Πολυμέσα" και "Σχόλια" επιτρέπει την εύκολη ανάκτηση σχετικών δεδομένων.

Η δομή είναι ευέλικτη και μπορεί να επεκταθεί εύκολα για να συμπεριλάβει νέες λειτουργίες.

Εναλλακτική 2: Σχεδίαση με Ένθετα Πεδία

Σε αυτή την προσέγγιση, θα αποθηκεύσουμε ορισμένα δεδομένα απευθείας στις συλλογές, αποφεύγοντας τις αναφορές.

Συλλογή "Ομάδες":

- `"_id": ObjectId()`,
- `"group_name": "group_name",`
- `"members": [`
`{ "user_id": "user_id1", "username": "username1" },`
`{ "user_id": "user_id2", "username": "username2" }`
`]`

Συλλογή "Αναρτήσεις":

- `"_id": ObjectId()`,
- `"user_id": "user_id",`

- **"group_id"**: "group_id",
- **"content"**: "post_content",
- **"timestamp"**: "timestamp",
- **"user"**: { "username": "username" } // Ένθετο πεδίο με στοιχεία χρήστη

Συλλογή "Πολυμέσα":

- **"_id"**: ObjectId(),
- **"user_id"**: "user_id",
- **"type"**: "media_type",
- **"url"**: "media_url",
- **"likes"**: ["user_id1", "user_id2"]

Συλλογή "Σχόλια":

- **"_id"**: ObjectId(),
- **"post_id"**: "post_id",
- **"user_id"**: "user_id",
- **"content"**: "comment_content",
- **"timestamp"**: "timestamp",
- **"user"**: { "username": "username" }, // Ένθετο πεδίο με στοιχεία χρήστη
- **"replies"**: [
 - { **"user_id"**: "reply_user_id", **"content"**: "reply_content", **"timestamp"**: "timestamp" }]

Η αποθήκευση ορισμένων δεδομένων (όπως όνομα χρήστη) απευθείας στις συλλογές "Αναρτήσεις" και "Σχόλια" βελτιστοποιεί την ανάκτηση πληροφοριών για κάθε ανάρτηση/σχόλιο.

Η δομή είναι πιο απλή και κατανοητή σε σύγκριση με την πρώτη προσέγγιση.

Η επιλογή της κατάλληλης δομής εξαρτάται από τις βασικές ανάγκες και προτεραιότητες του κοινωνικού δικτύου. Η πρώτη προσέγγιση με αναφορές προσφέρει ευελιξία και δυνατότητα επέκτασης, ενώ η δεύτερη με ένθετα πεδία είναι απλούστερη και βελτιστοποιημένη για ταχύτερες ανακτήσεις. Εάν το κοινωνικό δίκτυο εστιάζει στην ευελιξία και δυνατότητα επέκτασης, η πρώτη προσέγγιση με αναφορές αποτελεί μια κατάλληλη επιλογή.

Από την άλλη πλευρά, εάν η ταχύτητα ανάκτησης δεδομένων και η απλότητα της δομής είναι κύριες προτεραιότητες, η δεύτερη προσέγγιση με ένθετα πεδία μπορεί να είναι πιο προτιμητή.

B) Task 1: Display title and year for all movies sorted by oldest release date and then in descending alphabetical order

db.movies.find({}, {"title": 1, "year": 1, "_id": 0}).sort([("year", 1), ("title", -1)])

Task 2: Display title, year, and directors for movies released after 2012 with duration less than or equal to 90 minutes

```
db.movies.find({"year": {"$gt": 2012}, "runtime": {"$lte": 90}}, {"title": 1, "year": 1, "directors": 1, "_id": 0})
```

Task 3: Display movies without "rated" field or with "rated" field value as "unrated" or "passed"

```
db.movies.find({"$or": [{"rated": {"$exists": False}}, {"rated": {"$in": ["unrated", "passed"]}]})
```

Task 4: Display title and awards for top 20 movies with most comments, sorted by num_mflix_comments in descending order

```
db.movies.find({}, {"title": 1, "awards": 1, "_id": 0}).sort([("num_mflix_comments", -1)]).limit(20)
```

Task 5: Display movie details where either "Al Pacino" or "Gary Oldman" appears in the cast

```
db.movies.find({"cast": {"$in": ["Al Pacino", "Gary Oldman"]}})
```

Task 6: Display movie details with genre "Crime" and "Drama"

```
db.movies.find({"genres": {"$all": ["Crime", "Drama"]}})
```

Task 7: Display movie details where "Tom Hanks" appears in the cast but not "Meg Ryan"

```
db.movies.find({"cast": "Tom Hanks", "cast": {"$not": {"$eq": "Meg Ryan"}}})
```

Task 8: Display count of actors, directors, and writers for each movie

```
db.movies.aggregate([
  {"$project": {"actors_count": {"$size": {"$ifNull": ["$cast", []]}}, "directors_count": {"$size": {"$ifNull": ["$directors", []]}}, "writers_count": {"$size": {"$ifNull": ["$writers", []]}}}
])
```

Task 9: Display count of movies, maximum, and minimum duration per movie rating

```
db.movies.aggregate([  
  {"$group": {"_id": "$rated", "count": {"$sum": 1}, "max_duration": {"$max":  
"$runtime"}, "min_duration": {"$min": "$runtime"}}}  
])
```

Task 10: Display count of movies with ratings ≥ 3.5 from viewers and 5 from critics per year, for years with at least 3 such movies

```
db.movies.aggregate([  
  {"$match": {"imdb.rating": {"$gte": 3.5}, "tomatoes.viewer.rating": {"$gte": 3.5},  
"tomatoes.critic.rating": {"$gte": 5}}},  
  {"$group": {"_id": "$year", "count": {"$sum": 1}}},  
  {"$match": {"count": {"$gte": 3}}}  
])
```

Task 11: Display count and list of movies by country and genre, where pairs have fewer than 20 movies

```
db.movies.aggregate([  
  {"$unwind": "$countries"},  
  {"$unwind": "$genres"},  
  {"$group": {"_id": {"country": "$countries", "genre": "$genres"}, "count": {"$sum":  
1}}},  
  {"$match": {"count": {"$lt": 20}}}  
])
```

Task 12: Display count and list of movie titles by genre for movies featuring "Edward Norton"

```
db.movies.aggregate([  
  {"$match": {"cast": "Edward Norton"}},  
  {"$unwind": "$genres"},  
  {"$group": {"_id": "$genres", "count": {"$sum": 1}, "titles": {"$push": "$title"}}}  
])
```