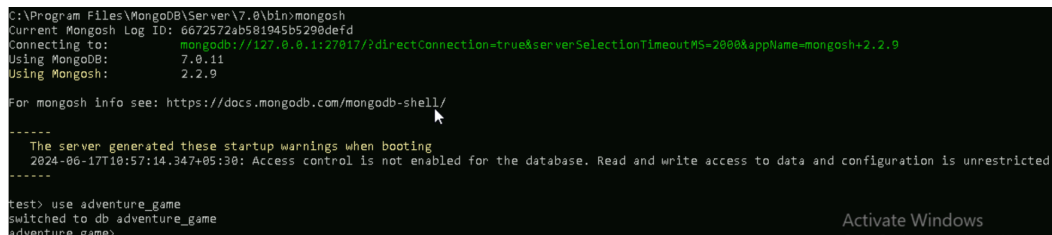Muhammad Shas K T
281775

Assignment 1 - MongoDB Essentials - A Complete MongoDB Guide

1. Building the Game World (Data Modeling & CRUD Operations)
- Create a MongoDB database named "adventure_game".

use adventure_game

```
C:\Program Files\MongoDB\Server\7.0\bin>mongosh
Current Mongosh Log ID: 6672572ab581945b5290defd
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.9
Using MongoDB:          7.0.11
Using Mongosh:          2.2.9

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2024-06-17T10:57:14.347+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> use adventure_game
switched to db adventure_game
adventure_game>
```

- Design three collections to represent the core elements of your game:
  - Locations (name, description, exits - references to other locations)

    {
        name: '',
        description: '',
        exits: ['','','...]
    }

  - Characters (name, description, location - reference to a location)

    {
        name: '',
        description: '',
        location: ''
    }

  - Items (name, description, location - reference to a location)

    {
        name: '',
        description: '',
        location: ''
    }

- Populate each collection with initial data to create your starting game world. This might include a few locations, characters, and items strategically placed.

```
db.locations.insertMany([
  {
    name: "Forest",
    description: "A dense and dark forest with towering trees.",
    exits: ["Cave", "River"]
  }
```

```
])

db.characters.insertMany([
  {
    name: "Archer",
    description: "A skilled archer with a keen eye for targets.",
    location: "Forest"
  }
])

db.items.insertMany([
  {
    name: "Bow",
    description: "A sturdy bow made from yew wood.",
    location: "Forest"
  }
])
```

```
adventure_game> db.locations.insertMany([
...  {
...    name: "Forest",
...    description: "A dense and dark forest with towering trees.",
...    exits: ["Cave", "River"]
...  },
...  {
...    name: "Cave",
...    description: "A cold, damp cave. You can hear bats in the distance.",
...    exits: ["Forest"]
...  },
...  {
...    name: "River",
...    description: "A fast-flowing river with crystal clear water.",
...    exits: ["Forest"]
...  }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6672c7e9faa79a51ca90defe'),
    '1': ObjectId('6672c7e9faa79a51ca90deff'),
    '2': ObjectId('6672c7e9faa79a51ca90df00')
  }
}
...  {
...    name: "Archer",
...    description: "A skilled archer with a keen eye for targets.",
...    location: "Forest"
...  },
...  {
...    name: "Warrior",
...    description: "A brave warrior with unmatched strength.",
...    location: "Cave"
...  }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6672c810faa79a51ca90df01'),
    '1': ObjectId('6672c810faa79a51ca90df02')
```

```
adventure_game> db.items.insertMany([
...  {
...    name: "Bow",
...    description: "A sturdy bow made from yew wood.",
...    location: "Forest"
...  },
...  {
...    name: "Sword",
...    description: "A sharp sword with a gleaming blade.",
...    location: "Cave"
...  },
...  {
...    name: "Shield",
...    description: "A large shield capable of blocking most attacks.",
...    location: "River"
...  }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6672c822faa79a51ca90df03'),
    '1': ObjectId('6672c822faa79a51ca90df04'),
    '2': ObjectId('6672c822faa79a51ca90df05')
  }
}
```
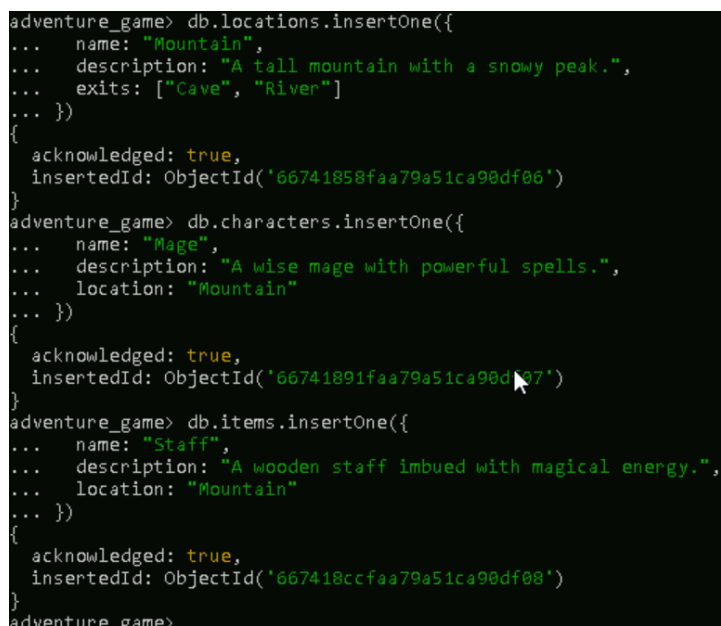
- Implement functionalities (using a MongoDB client or driver) to perform CRUD operations:

    - Create new locations, characters, and items.

        ```
        db.locations.insertOne({
          name: "Mountain",
          description: "A tall mountain with a snowy peak.",
          exits: ["Cave", "River"]
        })

        db.characters.insertOne({
          name: "Mage",
          description: "A wise mage with powerful spells.",
          location: "Mountain"
        })

        db.items.insertOne({
          name: "Staff",
          description: "A wooden staff imbued with magical energy.",
          location: "Mountain"
        })
        ```


    - Read existing data from each collection based on specific criteria (e.g., find a character by name).

        ```
        db.locations.find({ name: "Forest" })
        db.characters.findOne({ name: "Archer" })
        db.items.findOne({ name: "Sword" })
        ```

```
adventure_game> db.items.find({name:'Staff'})
[
  {
    _id: ObjectId('667418ccfaa79a51ca90df08'),
    name: 'Staff',
    description: 'A wooden staff imbued with magical energy.',
    location: 'Mountain'
  }
]
adventure_game> db.locations.findOne({name:'Forest'})
{
  _id: ObjectId('6672c7e9faa79a51ca90defe'),
  name: 'Forest',
  description: 'A dense and dark forest with towering trees.',
  exits: [ 'Cave', 'River' ]
}
adventure_game> db.characters.findOne({name:'Mage'})
{
  _id: ObjectId('66741891faa79a51ca90df07'),
  name: 'Mage',
  description: 'A wise mage with powerful spells.',
  location: 'Mountain'
}
```

- Update information about locations, characters, or items (e.g., move an item to a new location).

  db.items.updateOne(
    { name: "Sword" },
    { $set: { location: "Forest" } }
)

```
adventure_game> db.items.updateOne({name:'Sword'}, {$set: {location: 'Forest'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
adventure_game> db.items.find()
```

- Delete unnecessary data from the collections (be mindful of maintaining game world consistency).

  db.items.deleteOne({ name: "Staff" })

```
adventure_game> db.items.deleteOne({name:'Staff'})
{ acknowledged: true, deletedCount: 1 }
adventure_game>
```

2. Exploring the Game World (MongoDB Queries)
- Develop MongoDB queries to retrieve information relevant to the player's exploration:

  - Describe the current location based on its name or ID.

    db.characters.find({name:'Archer'}, {location:1})

```
adventure_game> db.characters.find({name:'Archer'},{location:1})
[ { _id: ObjectId('6672c810faa79a51ca90df01'), location: 'Forest' } ]
adventure_game>
```

- List available exits from a specific location using the references stored in the collection.

```
db.locations.find({name:'Forest'},{exits:1})
```

```
adventure_game> db.locations.find({name:'Forest'},{exits:1})
[
  {
    _id: ObjectId('6672c7e9faa79a51ca90defe'),
    exits: [ 'Cave', 'River' ]
  }
]
adventure_game>
```

- Find characters or items based on their properties (e.g., find a weapon in the current location).

```
db.characters.find({ location: "Forest" }, {name:1})
```

```
adventure_game> db.characters.find({location:'Forest'},{name:1})
[ { _id: ObjectId('6672c810faa79a51ca90df01'), name: 'Archer' } ]
adventure_game>
```

- Utilize logical operators (AND, OR) to construct more advanced queries (e.g., find a character named "Mage" located in the "Forest").

```
db.characdb.characters.findOne({$or: [{ name: "Archer" },  { location: "Cave" }] })
```

```
adventure_game> db.characters.find({$or: [{name:'Archer'},{location:'Cave'}] })
[
  {
    _id: ObjectId('6672c810faa79a51ca90df01'),
    name: 'Archer',
    description: 'A skilled archer with a keen eye for targets.',
    location: 'Forest'
  },
  {
    _id: ObjectId('6672c810faa79a51ca90df02'),
    name: 'Warrior',
    description: 'A brave warrior with unmatched strength.',
    location: 'Cave'
  }
]
adventure_game> db.characters.find({$and: [{name:'Archer'},{location:'Cave'}] })
```