NOSQL_ Assignment 1

Affan Mohammed N Marikar
281911

• Create a MongoDB database named "adventure_game".

use adventure_game

```
ustdb> 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 lo
cations)
✓ Characters (name, description, location - reference to a location)
✓ Items (name, description, location - reference to a location)

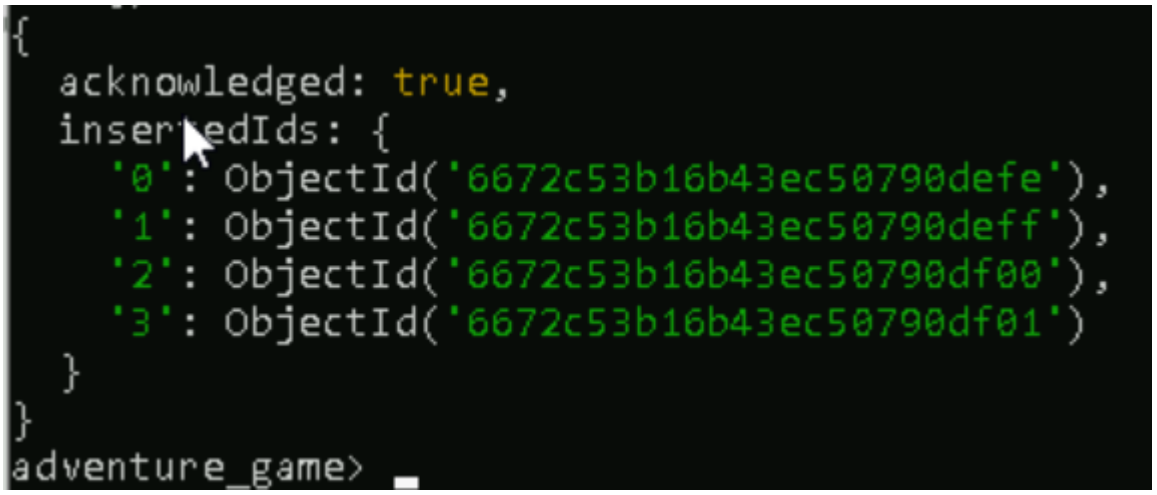
db.createCollection(‘location’)
db.createCollection('characters')
db.createCollection('items')

```
adventure_game> db.createCollection("location")
{ ok: 1 }
adventure_game> db.createCollection('characters')
{ ok: 1 }
adventure_game> db.createCollection('items')
{ ok: 1 }
adventure_game> _
```

• 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:'India',
description:'loc1',
exits:['UAE','England']
},
{
name:'UAE',
description:'loc2',
exits:['England']
},
{
name:'Africa',
```

```
description:'loc3',
exits:['India','UAE','England']
},
{
name:'England',
description:'loc4',
exits:[]
}
])
```




```
db.characters.insertMany([
{       name:'Alex',
        description:'M',
        exits:['India','USA']
},
{       name:'Ben',
        description:'M',
        exits:['England','UAE']
},
{       name:'Clair',
        description:'W',
        exits:['Africa','Australia']
}
]);
```

db.items.insertMany([
{name:'Taj',description:'Monument',location:'India'},
{name:'Burj',description:'Hotel',location:'UAE'},
{name:'BigBen',description:'Clock',location:'England'},
{name:'Sahara',description:'Desert',location:'Africa'}
])

Implement functionalities (using a MongoDB client or driver) to perform CRUD operations:

- Create new locations, characters, and items.
  db.characters.insertOne({
  name:'Diana',
  description:'W',
  exits:['England','India']

  })

```
adventure_game> db.characters.insertOne({
...     name:'Diana',
...     description:'W',
...     exits:['England','India']
... })
{
  acknowledged: true,
  insertedId: ObjectId('6673034116b43ec50790df0a')
}
```

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

  db.characters.find({exits: {$in :["India"] }});

```
adventure_game> db.characters.find({exits: {$in :["India"] }});
[
  {
    _id: ObjectId('6672fca316b43ec50790df02'),
    name: 'Alex',
    description: 'M',
    exits: [ 'India', 'USA' ]
  },
  {
    _id: ObjectId('6673034116b43ec50790df0a'),
    name: 'Diana',
    description: 'W',
    exits: [ 'England', 'India' ]
  }
]
```

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

  db.items.updateOne({name:'BigBen'},{$set:{description:'Clock Tower'}})

```
adventure_game> db.items.updateOne({name:'BigBen'},{$set:{description:'Clock Tower'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
  {
    _id: ObjectId('6672ff9516b43ec50790df07'),
    name: 'BigBen',
    description: 'Clock Tower',
    location: 'England'
  },
```

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

  db.items.deleteOne({location: ‘England’})

```
adventure_game> db.items.deleteOne({location: 'England'})
{ acknowledged: true, deletedCount: 1 }
adventure_game> db.items.find()
[
  {
    _id: ObjectId('6672ff9516b43ec50790df05'),
    name: 'Taj',
    description: 'Monument',
    location: 'India'
  },
  {
    _id: ObjectId('6672ff9516b43ec50790df06'),
    name: 'Burj',
    description: 'Hotel',
    location: 'UAE'
  },
  {
    _id: ObjectId('6672ff9516b43ec50790df08'),
    name: 'Sahara',
    description: 'Desert',
    location: 'Africa'
  }
]
```

Describe the current location based on its name or ID

db.locations.findOne({name: ‘India’})

```
adventure_game> db.locations.findOne({ name: 'India' });
{
  _id: ObjectId('6672c53b16b43ec50790defe'),
  name: 'India',
  description: 'loc1',
  exits: [ 'UAE', 'England' ]
}
```

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

db.locations.find({},{exits: 1 , name: 1, _id:0})

```
adventure_game> db.locations.find({},{exits: 1 , name: 1, _id:0})
[
  { name: 'India', exits: [ 'UAE', 'England' ] },
  { name: 'UAE', exits: [ 'England' ] },
  { name: 'Africa', exits: [ 'India', 'UAE', 'England' ] },
  { name: 'England', exits: [] }
]
adventure_game>
```

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

db.characters.find({ description: 'W' });

```
adventure_game> db.characters.find({ description: 'W' });
[
  {
    _id: ObjectId('6672fca316b43ec50790df04'),
    name: 'Clair',
    description: 'W',
    exits: [ 'Africa', 'Australia' ]
  },
  {
    _id: ObjectId('6673034116b43ec50790df0a'),
    name: 'Diana',
    description: 'W',
    exits: [ 'England', 'India' ]
  }
]
```

Utilise logical operators (AND, OR) to construct more advanced queries

```
db.characters.find({
  description: 'M',  $or: [{ name: 'Alex' }, { name: 'Ben' }]
});
```