MongoDB Assignment-1

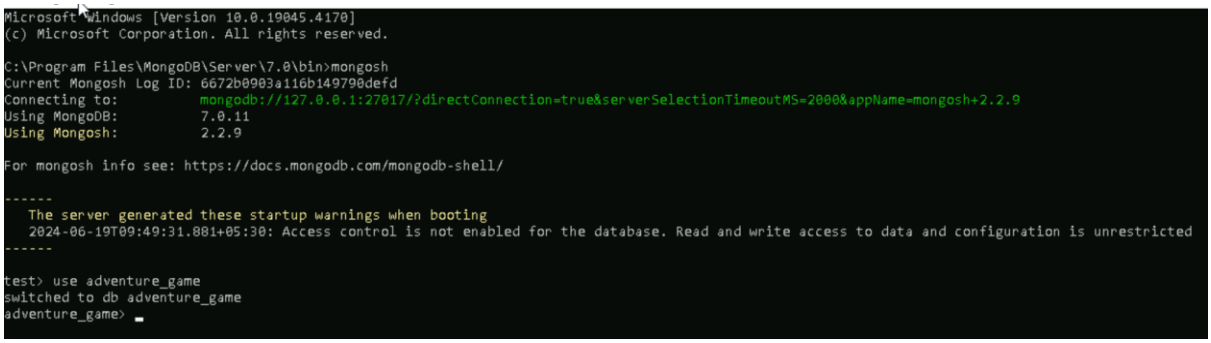
Problem Statement 1: Building the Game World (Data Modeling & CRUD Operations):

Objective: Design your game's data model in MongoDB and establish CRUD operations for data manipulation.

Task:

• Create a MongoDB database named "adventure_game" :

use adventure_game



• Design three collections to represent the core elements of your game:

✓ Locations (name, description, exits - references to other locations)

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

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

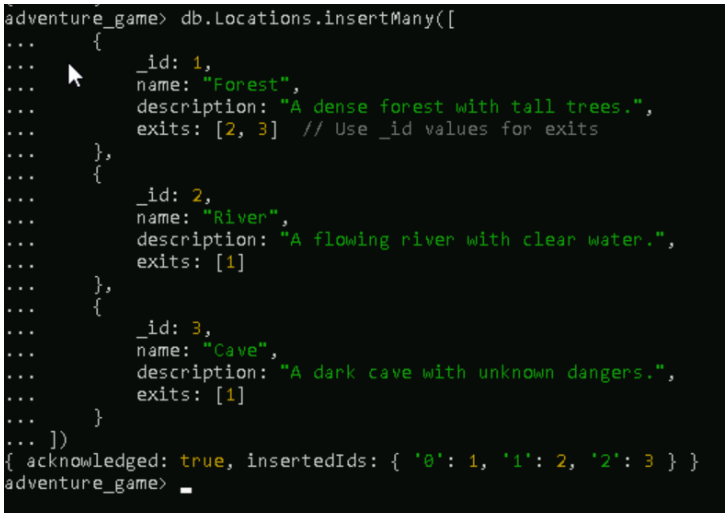
db.createCollections(“Locations”);

db.createCollections(“Characters”);

db.createCollections(“Items”);

• 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([
  {     _id: 1,
    name: "Forest",
    description: "A dense forest with tall trees.",
    exits: [2, 3]  // Use _id values for exits
  },
  {     _id: 2,
    name: "River",
    description: "A flowing river with clear water.",
    exits: [1]
  },
  {     _id: 3,
    name: "Cave",
    description: "A dark cave with unknown dangers.",
    exits: [1]
  }
])
```



```
db.Characters.insertMany([
  {
    name: "Alice",
    description: "A brave explorer.",
    location: 1
  },
  {
    name: "Bob",
    description: "A seasoned warrior.",
    location: 3
  }
])

db.Items.insertMany([
  {
    name: "Sword",
    description: "A sharp blade.",
    location: 3
  },
  {
    name: "Shield",
    description: "A sturdy shield.",
    location: 1
  }
])
```

```
adventure_game> db.Characters.insertMany([
...     {
...         name: "Alice",
...         description: "A brave explorer.",
...         location: 1
...     },
...     {
...         name: "Bob",
...         description: "A seasoned warrior.",
...         location: 3
...     }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6672bd793a116b149790df01'),
    '1': ObjectId('6672bd793a116b149790df02')
  }
}
adventure_game>
```

```
adventure_game> db.Items.insertMany([
...     {
...         name: "Sword",
...         description: "A sharp blade.",
...         location: 3
...     },
...     {
...         name: "Shield",
...         description: "A sturdy shield.",
...         location: 1
...     }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6672bdb73a116b149790df03'),
    '1': ObjectId('6672bdb73a116b149790df04')
  }
}
adventure_game>
```

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

db.Characters.drop()

db.Items.drop()

```
}
adventure_game> db.Characters.drop()
true
adventure_game> db.Items.drop()
true
adventure_game>
```

• Create new locations, characters, and items.

```
adventure_game> db.locations.insertOne({
...     name: "Mountain",
...     description: "Gigantic heights that may fill in the fear",
...     exits: {
...         east: "Cave"
...     }
... })
{
  acknowledged: true,
  insertedId: ObjectId('6672c33f3a116b149790df09')
}
adventure_game> db.characters.insertOne({
...     name: "jim",
...     description: "A mindful magician.",
...     location: "Mountain"
... })
{
  acknowledged: true,
  insertedId: ObjectId('6672c34c3a116b149790df0a')
}
adventure_game> db.items.insertOne({
...     name: "potion",
...     description: "Magical power inducing potion.",
...     location: "Mountain"
... })
{
  acknowledged: true,
  insertedId: ObjectId('6672c35b3a116b149790df0b')
}
adventure_game>
```

```
// Create a new location
db.locations.insertOne({
  name: "Mountain",
  description: "Gigantic heights that may fill in the fear",
  exits: {
    east: "Cave"
  }
})

// Create a new character
db.characters.insertOne({
  name: "jim",
  description: "A mindful magician.",
  location: "Mountain"
})

// Create a new item
db.items.insertOne({
  name: "potion",
  description: "Magical power inducing potion.",
  location: "Mountain"
})
```

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

```
// Find location by name
db.Locations.find({ name: "Cave" }).pretty()

// Find character by name
db.Characters.find({ name: "Bob" }).pretty()
```

// Find items in a specific location

db.Items.find({ name: "Sword" }).pretty()

```
adventure_game> db.Locations.find({ name: "Cave" }).pretty()
[
  {
    _id: 3,
    name: 'Cave',
    description: 'A dark cave with unknown dangers.',
    exits: [ 1 ]
  }
]
adventure_game> db.Characters.find({ name: "Bob" }).pretty()
[
  {
    _id: ObjectId('6672c2003a116b149790df08'),
    name: 'Bob',
    description: 'A seasoned warrior.',
    location: 3
  }
]
adventure_game> db.Items.find({ name: "Sword" }).pretty()
[
  {
    _id: ObjectId('6672c1fa3a116b149790df05'),
    name: 'Sword',
    description: 'A sharp blade.',
    location: 3
  }
]
adventure_game>
```

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

```
adventure_game> db.Locations.updateOne( { name: "Forest" }, { $set: { description: "A dense and dark forest with towering trees." } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
adventure_game> db.Items.updateOne( { name: "Sword" }, { $set: { location: 4 } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
adventure_game> db.Characters.updateOne( { name: "Bob" }, { $set: { location:4 } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
adventure_game>
```

// Update a location's description

db.Locations.updateOne(

{ name: "Forest" },

{ $set: { description: "A dense and dark forest with towering trees." } }

)

// Move an item to a new location

db.Items.updateOne(

{ name: "Sword" },

{ $set: { location:4 } }

)

// Change a character's location

db.Characters.updateOne(

{ name: "Bob" },

{ $set: { location:4 } }

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

```
adventure_game> db.Locations.deleteOne({ name: "Cave" })
{ acknowledged: true, deletedCount: 1 }
adventure_game> db.Characters.deleteOne({ name: "Bob" })
{ acknowledged: true, deletedCount: 1 }
adventure_game> db.Items.deleteOne({ name: "potion" })
{ acknowledged: true, deletedCount: 1 }
adventure_game>
```

// Delete a location by name

db.Locations.deleteOne({ name: "Cave" })

// Delete a character by name

db.Characters.deleteOne({ name: "Bob" })

// Delete an item by name

db.Items.deleteOne({ name: "potion" })

TASK 2

Develop MongoDB queries to retrieve information relevant to the player's exploration:

- **Describe the current location based on its name or ID.**
  db.Characters.aggregate([{ $lookup: { from: "Locations", localField: "location", foreignField: "_id", as:
  "location_details" } }, { $unwind: "$location_details" }, { $project: { _id: 1, name: 1, description: 1,
  "location_name": "$location_details.name" } }]);

```
adventure_game> db.Characters.aggregate([ { $lookup: { from: "Locations", localField: "location", foreignField: "_id", as:
... "location_details" } }, { $unwind: "$location_details" }, { $project: { _id: 1, name: 1, description: 1,
... "location_name": "$location_details.name" } }] );
[
  {
    _id: ObjectId('6672c2003a116b149790df07'),
    name: 'Alice',
    description: 'A brave explorer.',
    location_name: 'Forest'
  },
  {
    _id: ObjectId('6672c9ff3a116b149790df13'),
    name: 'jim',
    description: 'A mindful magician.',
    location_name: 'River'
  }
]
```

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

  db.Locations.aggregate([{ $lookup: { from: "Locations", localField: "exits", foreignField: "_id", as: "location_details" } }, { $unwind: "$location_details" }, { $project: { _id: 1, name: 1, description: 1, "location_name": "$location_details.name" } }]);

```
adventure_game> db.Locations.aggregate([ { $lookup: { from: "Locations", localField: "exits", foreignField: "_id", as:
... "location_details" } }, { $unwind: "$location_details" }, { $project: { _id: 1, name: 1, description: 1,
... "location_name": "$location_details.name" } }] );
[
  {
    _id: 1,
    name: 'Forest',
    description: 'A dense and dark forest with towering trees.',
    location_name: 'River'
  },
  {
    _id: 2,
    name: 'River',
    description: 'A flowing river with clear water.',
    location_name: 'Forest'
  }
]
adventure_game>
```

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

  db.Locations.aggregate([{ $lookup: { from: "Locations", localField: "exits", foreignField: "_id", as: "location_details" } }, { $unwind: "$location_details" }, { $project: { _id: 1, name: 1, description: 1, "location_name": "$location_details.name" } }]);

```
adventure_game> db.Locations.aggregate([ { $lookup: { from: "Locations", localField: "exits", foreignField: "_id", as:
... "location_details" } }, { $unwind: "$location_details" }, { $project: { _id: 1, name: 1, description: 1,
... "location_name": "$location_details.name" } }] );
[
  {
    _id: 1,
    name: 'Forest',
    description: 'A dense and dark forest with towering trees.',
    location_name: 'River'
  },
  {
    _id: 2,
    name: 'River',
    description: 'A flowing river with clear water.',
    location_name: 'Forest'
  }
]
adventure_game>
```

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

```
db.Characters.find({
 "$and": [
{ "name": "Alice" },
{ "$or": [ { "location": 1},
{ "location": 2 }]}
 ]});
```