

Assignment 2 - MongoDB Essentials - A Complete MongoDB Guide

1. Game Insights with Aggregation (Aggregation Framework):

- Utilize the Aggregation Framework to perform data manipulation and analysis within your game
- Count the total number of locations in your game world.

```
db.locations.aggregate({$count: "totalLocations"})
```

```
adventure_game> db.locations.aggregate({$count: 'totalLocations'})  
[ { totalLocations: 4 } ]  
adventure_game> db.locations.find()
```

- Calculate the average number of exits per location.

```
db.locations.aggregate([ { $project: { totalNumberOfExits: { $size: "$exits" } } }, {  
$group: { _id: null, averageExits: { $avg: "$totalNumberOfExits" } } } ] )
```

```
adventure_game> db.locations.aggregate([ { $project: { totalNumberOfExits: { $size: "$exits" } } }, { $group: { _id: null, averageExits: { $avg: "$totalNumberOfExits" } } } ] )  
[ { _id: null, averageExits: 1.5 } ]  
adventure_game> _
```

- Identify the most prevalent item type (e.g., weapons, potions) using aggregation pipelines.

```
db.items.aggregate([{$group: {_id:"$type", count: {$sum: 1}}}, {$sort: {count: -1}},  
{$limit: 1}])
```

```
[ { _id: "weapon", count: 2 }, { _id: "potion", count: 1 } ]  
adventure_game> db.items.aggregate([{$group: {_id:"$type", count: {$sum: 1}}}, {$sort: {count: -1}}, {$limit: 1}])  
[ { _id: "weapon", count: 2 } ]  
adventure_game> _
```

2. Speedy Navigation with Indexing:

- Identify frequently used query fields in your game (e.g., location names, item types).
 - i). location in the characters collection
 - ii). name in the locations collection
 - iii). type in the items collection
- Create indexes on these fields within the relevant collections.

```

db.characters.createIndex({ location: 1 });
db.locations.createIndex({ name: 1 });
db.items.createIndex({ type: 1 });

```

```

// To exit, press Ctrl+C again or Ctrl+D or type :exit)
adventure_game> db.characters.createIndex({location:1})
location_1
adventure_game> db.locations.createIndex({name:1})
name_1
adventure_game> db.items.createIndex({type:1})
type_1
adventure_game> db.items.find()

```

- Test the impact of indexes on query speed by comparing performance before and after indexing.

```

// Query to find characters in 'Forest' location without index
db.characters.find({ location: "Forest" }).explain("executionStats")

```

```

adventure_game> db.characters.find({ location: "Forest" }).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'adventure_game.characters',
    indexFilterSet: false,
    parsedQuery: { location: { '$eq': 'Forest' } },
    queryHash: '102282E8',
    planCacheKey: '102282E8',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'COLLSCAN',
      filter: { location: { '$eq': 'Forest' } },
      direction: 'forward'
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 3,
    executionStages: {
      stage: 'COLLSCAN',
      filter: { location: { '$eq': 'Forest' } },
      nReturned: 1,
      executionTimeMillisEstimate: 0,
      works: 4,
      advanced: 1,
      needTime: 2,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      direction: 'forward',
      docsExamined: 3
    }
  }
}

```

```

// Create an index on the 'location' field in the 'characters' collection
db.characters.createIndex({ location: 1 })

```

```

adventure_game> db.characters.createIndex({location:1})
location_1
adventure_game> // Query to find items by type with index

```

```
// Query to find characters in 'Forest' location with index
db.characters.find({ location: "Forest" }).explain("executionStats")
```

```
adventure_game> db.items.find({ type: "weapon" }).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: "adventure_game.items",
    indexFilterSet: false,
    parsedQuery: { type: { '$eq': 'Weapon' } },
    queryHash: "0C730C7C",
    planCacheKey: "024d5900",
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { type: 1 },
        indexName: 'type_1',
        isMultikey: false,
        multikeyPaths: { type: [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { type: [ '['"Weapon"', '"Weapon"]' ] }
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 2,
    executionTimeMillis: 20,
    totalKeysExamined: 2,
    totalDocsExamined: 2,
    executionStages: {
      stage: 'FETCH',
      nReturned: 2,
      executionTimeMillisEstimate: 12,
      work: 3
    }
  }
}
```

Activ
Go to