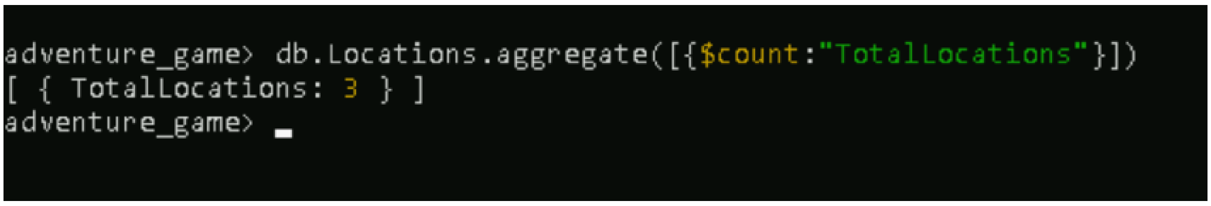
MongoDB Assignment 2

Task 1 :

• 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: 3 } ]
adventure_game> 
```

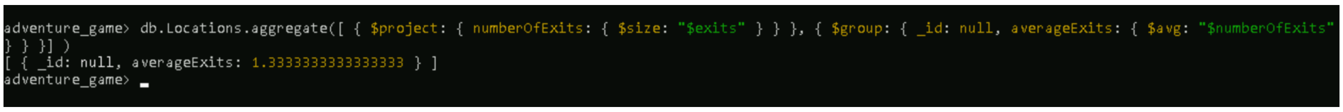
✓ Calculate the average number of exits per location.

db.Locations.aggregate([

{$project:{numberOfExits: {$size: "$exits"}}},

{$group:{_id: null, averageExits: {$avg: "$numberOfExits"}}}

])



```
adventure_game> db.Locations.aggregate([ { $project: { numberOfExits: { $size: "$exits" } } }, { $group: { _id: null, averageExits: { $avg: "$numberOfExits"
} } }] )
[ { _id: null, averageExits: 1.3333333333333333 } ]
adventure_game> 
```

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

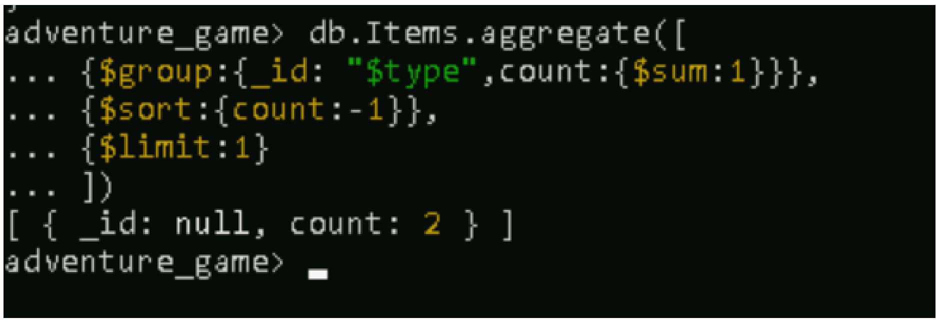
db.Items.aggregate([

{$group:{_id: "$type",count:{$sum:1}}},

{$sort:{count:-1}},

{$limit:1}

])



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

Task 2 :

• Identify frequently used query fields in your game (e.g., location names, item types).

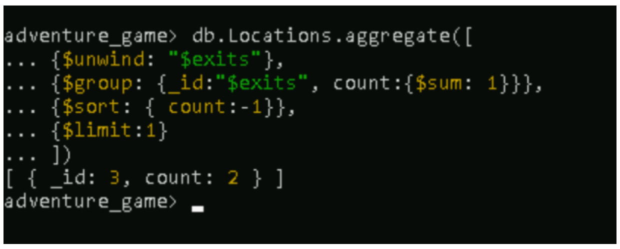
db.Locations.aggregate([

{$unwind: "$exits"},

{$group: {_id:"$exits", count:{$sum: 1}}},

{$sort: { count:-1}},

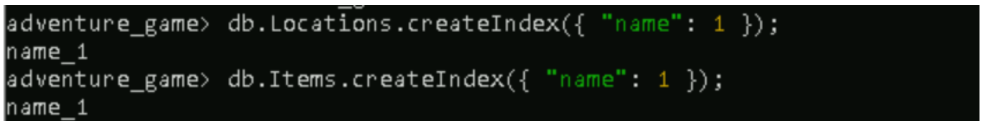
{$limit:1}])



```
adventure_game> db.Locations.aggregate([
... {$unwind: "$exits"},
... {$group: {_id:"$exits", count:{$sum: 1}}},
... {$sort: { count:-1}},
... {$limit:1}
... ])
[ { _id: 3, count: 2 } ]
adventure_game> 
```

• Create indexes on these fields within the relevant collections.

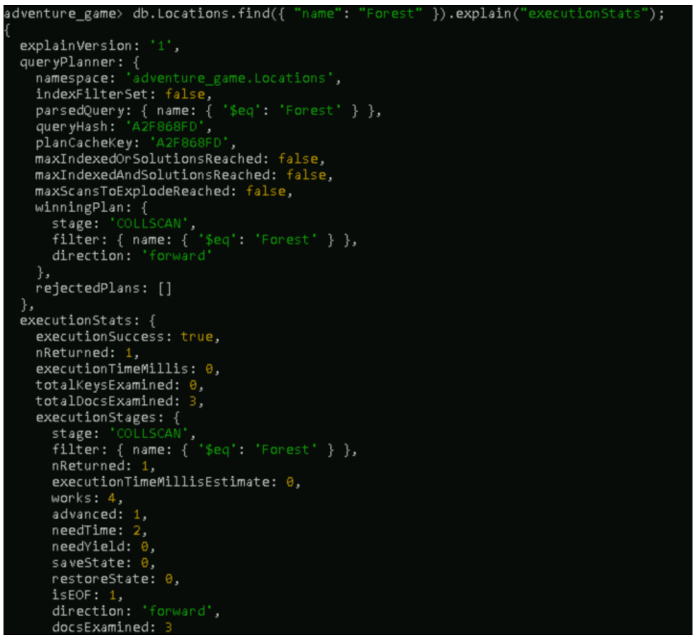
db.Locations.createIndex({ "name": 1 });



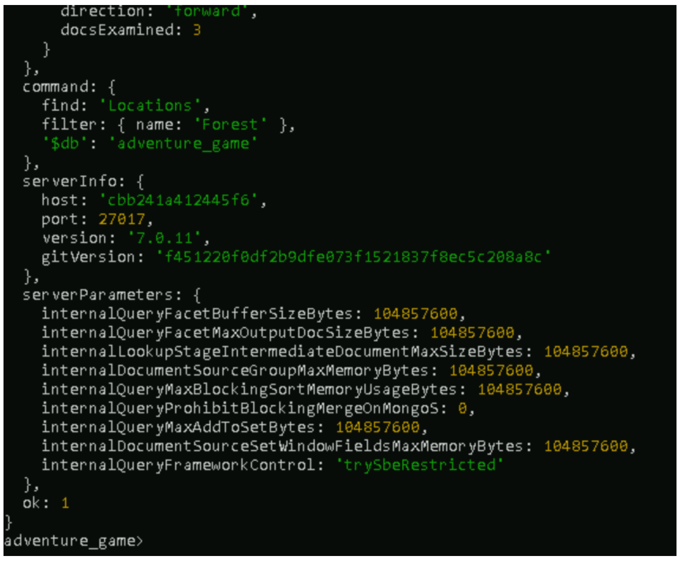
```
adventure_game> db.Locations.createIndex({ "name": 1 });
name_1
adventure_game> db.Items.createIndex({ "name": 1 });
name_1
```

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

Before creating Index:



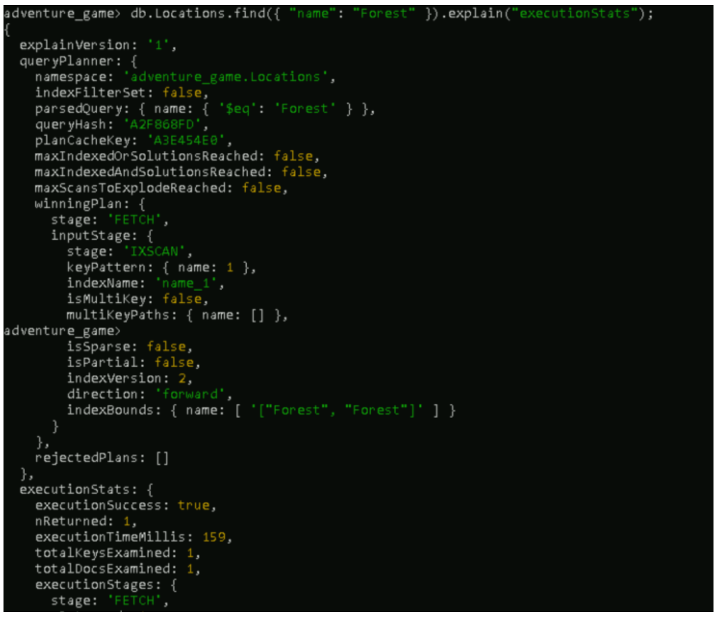
```
adventure_game> db.Locations.find({ "name": "Forest" }).explain("executionStats");
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'adventure_game.Locations',
    indexFilterSet: false,
    parsedQuery: { name: { '$eq': 'Forest' } },
    queryHash: 'A2F868FD',
    planCacheKey: 'A2F868FD',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'COLLSCAN',
      filter: { name: { '$eq': 'Forest' } },
      direction: 'forward'
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 3,
    executionStages: {
      stage: 'COLLSCAN',
      filter: { name: { '$eq': 'Forest' } },
      nReturned: 1,
      executionTimeMillisEstimate: 0,
      works: 4,
      advanced: 1,
      needTime: 2,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      direction: 'forward',
      docsExamined: 3
```



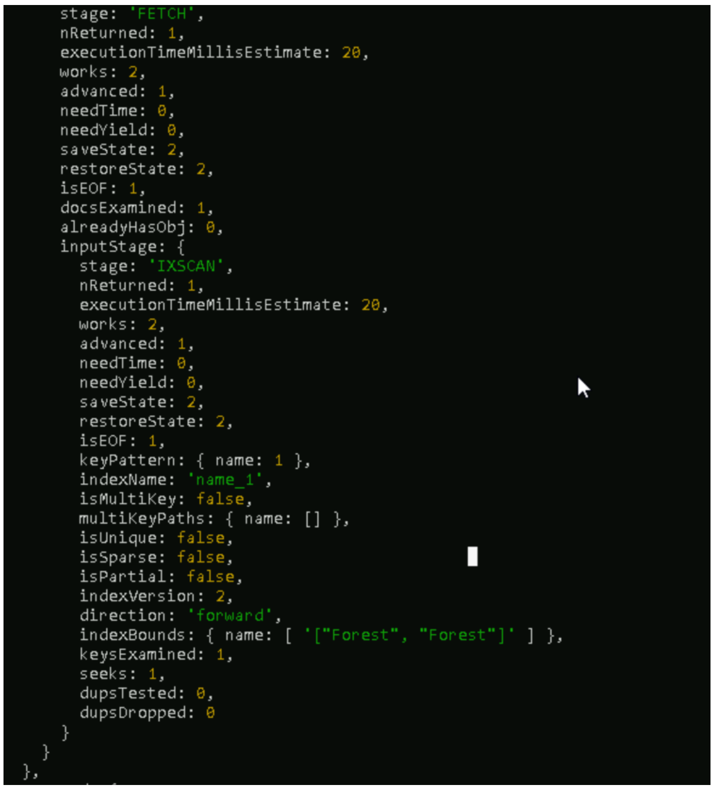
```
      direction: 'forward',
      docsExamined: 3
    }
  },
  command: {
    find: 'Locations',
    filter: { name: 'Forest' },
    '$db': 'adventure_game'
  },
  serverInfo: {
    host: 'cbb241a412445f6',
    port: 27017,
    version: '7.0.11',
    gitVersion: 'f451220f0df2b9dfe073f1521837f8ec5c208a8c'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeRestricted'
  },
  ok: 1
}
adventure_game>
```

After creating Index :

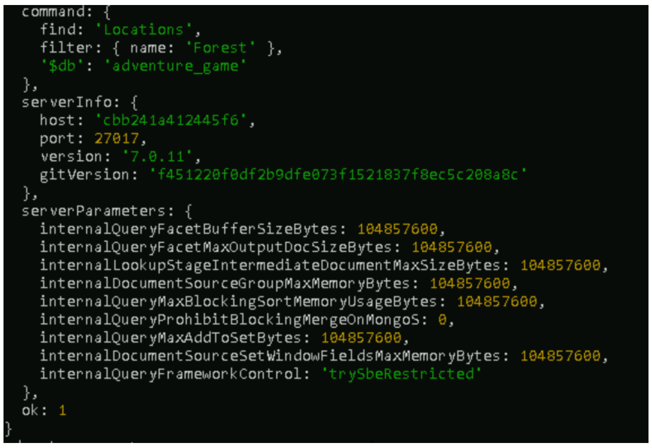
db.Locations.find({ "name": "Forest" }).explain("executionStats");



```
adventure_game> db.Locations.find({ "name": "Forest" }).explain("executionStats");
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'adventure_game.Locations',
    indexFilterSet: false,
    parsedQuery: { name: { '$eq': 'Forest' } },
    queryHash: 'A2F868FD',
    planCacheKey: 'A3E454E0',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { name: 1 },
        indexName: 'name_1',
        isMultiKey: false,
        multiKeyPaths: { name: [] },
adventure_game>
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { name: [ '["Forest", "Forest"]' ] }
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 159,
    totalKeysExamined: 1,
    totalDocsExamined: 1,
    executionStages: {
      stage: 'FETCH',
```



```
      stage: 'FETCH',
      nReturned: 1,
      executionTimeMillisEstimate: 20,
      works: 2,
      advanced: 1,
      needTime: 0,
      needYield: 0,
      saveState: 2,
      restoreState: 2,
      isEOF: 1,
      docsExamined: 1,
      alreadyHasObj: 0,
      inputStage: {
        stage: 'IXSCAN',
        nReturned: 1,
        executionTimeMillisEstimate: 20,
        works: 2,
        advanced: 1,
        needTime: 0,
        needYield: 0,
        saveState: 2,
        restoreState: 2,
        isEOF: 1,
        keyPattern: { name: 1 },
        indexName: 'name_1',
        isMultiKey: false,
        multiKeyPaths: { name: [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { name: [ '["Forest", "Forest"]' ] },
        keysExamined: 1,
        seeks: 1,
        dupsTested: 0,
        dupsDropped: 0
      }
    }
  },
```



```
  command: {
    find: 'Locations',
    filter: { name: 'Forest' },
    '$db': 'adventure_game'
  },
  serverInfo: {
    host: 'cbb241a412445f6',
    port: 27017,
    version: '7.0.11',
    gitVersion: 'f451220f0df2b9dfe073f1521837f8ec5c208a8c'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeRestricted'
  },
  ok: 1
}
```