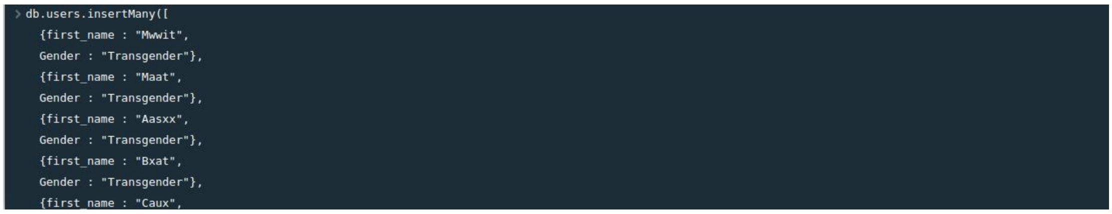
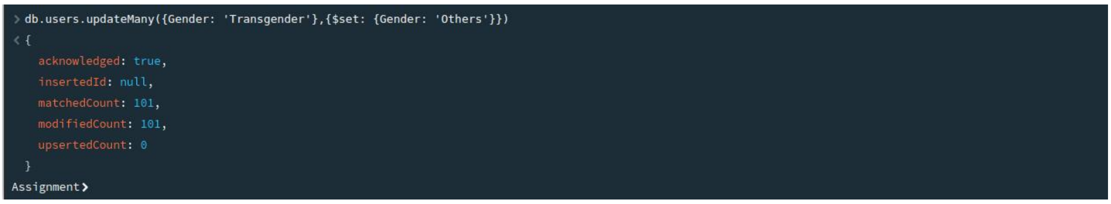
Name: Sumit Rajput (Python Trainee)
Submission: MongoDB Practical

1. Batch Create with minimum 100 records in MongoDb (create batch).



```
> db.users.insertMany([
    {first_name : "Mwwit",
    Gender : "Transgender"},
    {first_name : "Maat",
    Gender : "Transgender"},
    {first_name : "Aasxx",
    Gender : "Transgender"},
    {first_name : "Bxat",
    Gender : "Transgender"},
    {first_name : "Caux",
```

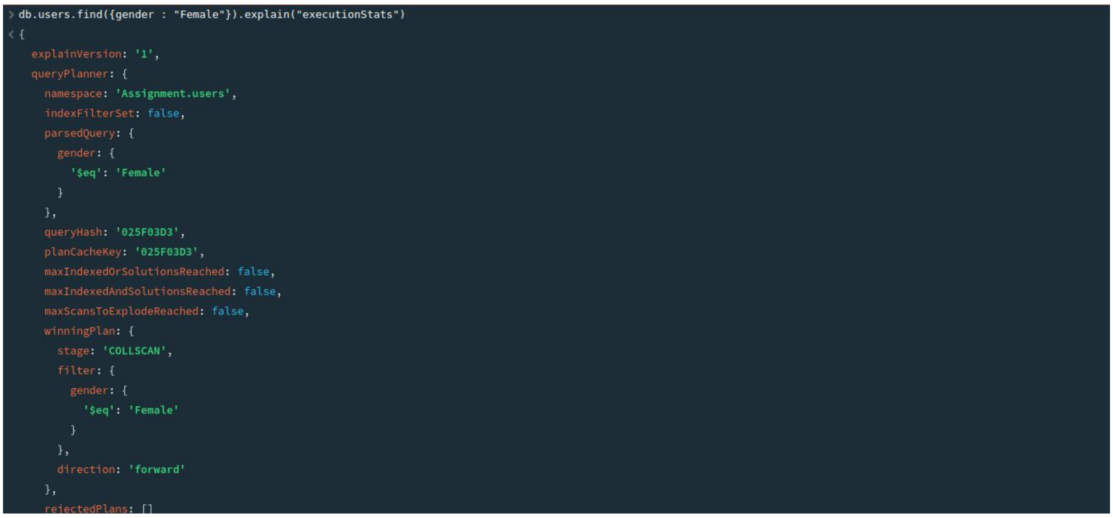
2) Batch Update with minimum 100 records in MongoDB (update batch).



```
> db.users.updateMany({Gender: 'Transgender'},{$set: {Gender: 'Others'}})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 101,
    modifiedCount: 101,
    upsertedCount: 0
  }
Assignment>
```

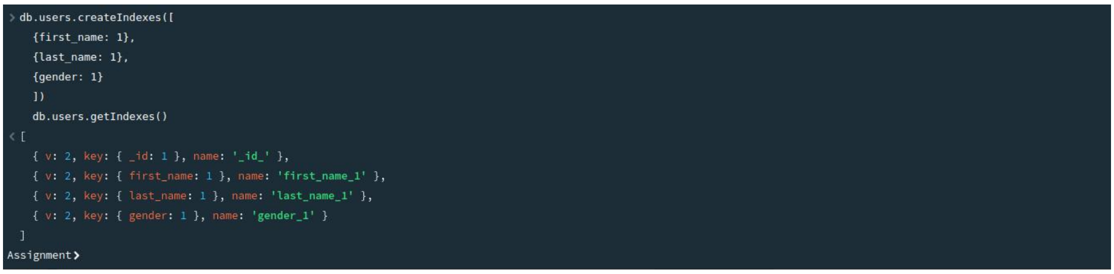
3) Perform indexing on particular 3 fields in MongoDB.

Before Indexing :



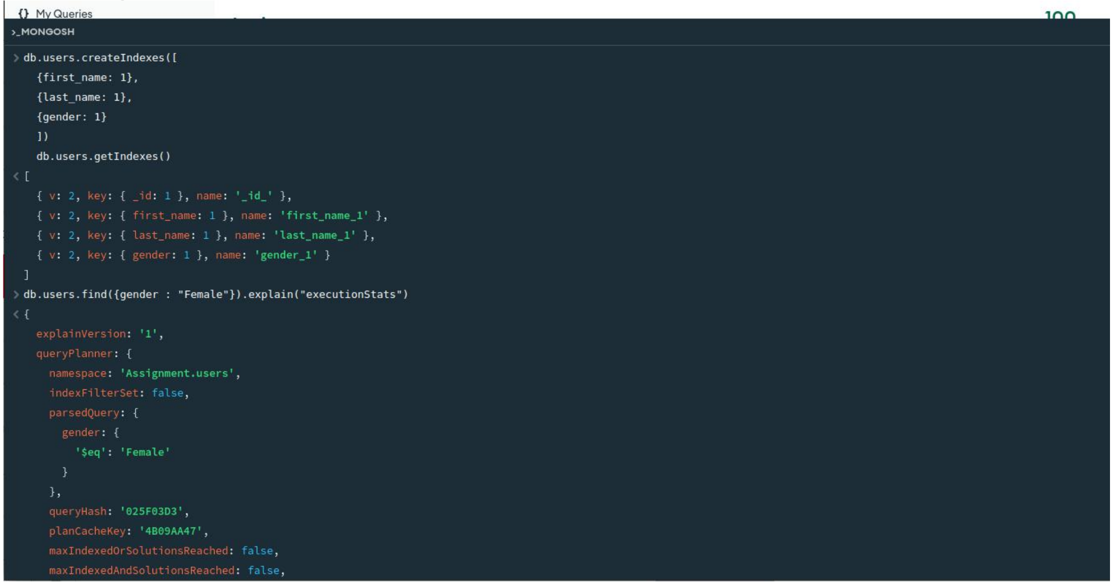
```
> db.users.find({gender : "Female"}).explain("executionStats")
< {
    explainVersion: '1',
    queryPlanner: {
      namespace: 'Assignment.users',
      indexFilterSet: false,
      parsedQuery: {
        gender: {
          '$eq': 'Female'
        }
      },
      queryHash: '025F03D3',
      planCacheKey: '025F03D3',
      maxIndexedOrSolutionsReached: false,
      maxIndexedAndSolutionsReached: false,
      maxScansToExplodeReached: false,
      winningPlan: {
        stage: 'COLLSCAN',
        filter: {
          gender: {
            '$eq': 'Female'
          }
        },
        direction: 'forward'
      },
      rejectedPlans: []
```

Create Indexes



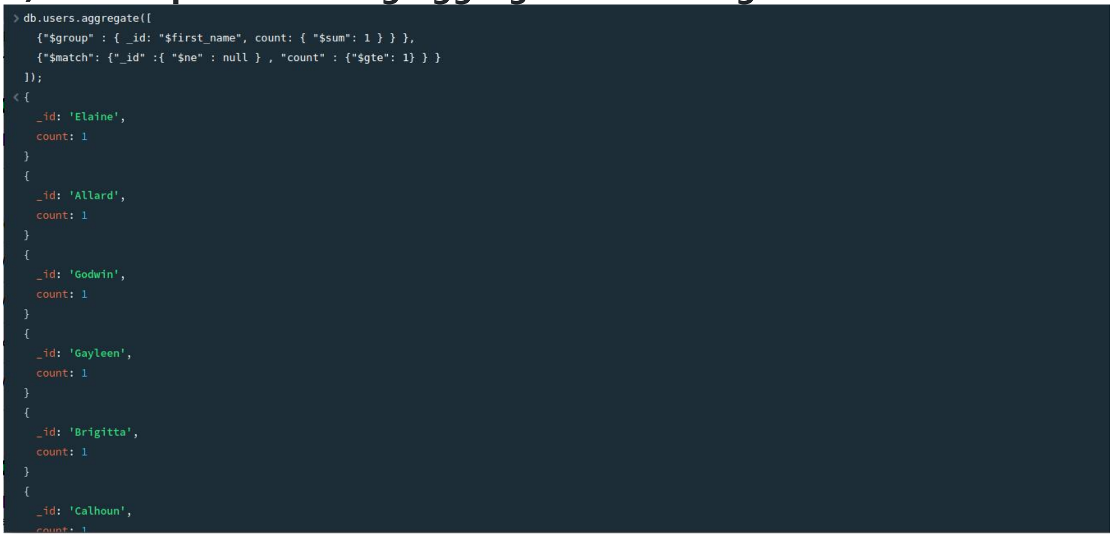
```
> db.users.createIndexes([
    {first_name: 1},
    {last_name: 1},
    {gender: 1}
    ])
    db.users.getIndexes()
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    { v: 2, key: { first_name: 1 }, name: 'first_name_1' },
    { v: 2, key: { last_name: 1 }, name: 'last_name_1' },
    { v: 2, key: { gender: 1 }, name: 'gender_1' }
  ]
Assignment>
```

After Indexing



```
> db.users.createIndexes([
    {first_name: 1},
    {last_name: 1},
    {gender: 1}
    ])
    db.users.getIndexes()
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    { v: 2, key: { first_name: 1 }, name: 'first_name_1' },
    { v: 2, key: { last_name: 1 }, name: 'last_name_1' },
    { v: 2, key: { gender: 1 }, name: 'gender_1' }
  ]
> db.users.find({gender : "Female"}).explain("executionStats")
< {
    explainVersion: '1',
    queryPlanner: {
      namespace: 'Assignment.users',
      indexFilterSet: false,
      parsedQuery: {
        gender: {
          '$eq': 'Female'
        }
      },
      queryHash: '025F03D3',
      planCacheKey: '4B09AA47',
      maxIndexedOrSolutionsReached: false,
      maxIndexedAndSolutionsReached: false,
```

4) Find duplicates using aggregation in MongoDB



```
> db.users.aggregate([
    {"$group" : { _id: "$first_name", count: { "$sum": 1 } } },
    {"$match": {"_id" :{ "$ne" : null } , "count" : {"$gte": 1} } }
  ]);
< {
    _id: 'Elaine',
    count: 1
  }
  {
    _id: 'Allard',
    count: 1
  }
  {
    _id: 'Godwin',
    count: 1
  }
  {
    _id: 'Gayleen',
    count: 1
  }
  {
    _id: 'Brigitta',
    count: 1
  }
  {
    _id: 'Calhoun',
    count: 1
```