MongoDB Assignment

Connect to MongoDB Atlas:

```
gati.vora@simform.dom@SF-CPU-0180:~$ mongosh "mongodb+srv://cluster0.lmonp.mongodb.net/" --apiVersion 1 --username a
dmin
Enter password: *****
Current Mongosh Log ID: 67a431706d6cd8ff82544ca6
Connecting to:          mongodb+srv://<credentials>@cluster0.lmonp.mongodb.net/?appName=mongosh+2.3.8
Using MongoDB:          8.0.4 (API Version 1)
Using Mongosh:          2.3.8
```

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

```
let batch = [];
const categories = ["Electronics", "Clothes", "Home Decor", "Bakery"];

for (let i = 0; i < 150; i++) {
        batch.push({
                name: `Product ${i + 1}`,
                category: categories[i % categories.length],
                price: (Math.random() * 1000 + 10).toFixed(2),
                stock_quantity: Math.floor(Math.random() * 100),
                created_at: new Date(),
        });
}

const duplicates = 20;

for (let i = 0; i < duplicates; i++) {
        const idx = Math.floor(Math.random() * batch.length);
        const duplicate = { ...batch[idx] };
        batch.push(duplicate);
}

db.products.insertMany(batch);
```

```
Atlas atlas-ftvs0i-shard-0 [primary] test> use shop
switched to db shop
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.createCollection("products")
{ ok: 1 }
Atlas atlas-ftvs0i-shard-0 [primary] shop> let batch = [];

Atlas atlas-ftvs0i-shard-0 [primary] shop> const categories = ['Electronics', 'Clothes', 'Home Decor', 'Bakery'];

Atlas atlas-ftvs0i-shard-0 [primary] shop>

Atlas atlas-ftvs0i-shard-0 [primary] shop> for (let i = 0; i < 150; i++) {
...   batch.push({
...     name: `Product ${i + 1}`,
...     category: categories[i % categories.length],
...     price: (Math.random() * 1000 + 10).toFixed(2),
...     stock_quantity: Math.floor(Math.random() * 100),
...     created_at: new Date()
...   });
... }
150
Atlas atlas-ftvs0i-shard-0 [primary] shop>

Atlas atlas-ftvs0i-shard-0 [primary] shop> const duplicates = 20;

Atlas atlas-ftvs0i-shard-0 [primary] shop>

Atlas atlas-ftvs0i-shard-0 [primary] shop> for (let i = 0; i < duplicates; i++) {
...     const idx = Math.floor(Math.random() * batch.length);
...     const duplicate = { ...batch[idx] };
...     batch.push(duplicate);
... }
170
Atlas atlas-ftvs0i-shard-0 [primary] shop>
```

```
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.products.insertMany(batch)
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67a43a7e6d6cd8ff82544ca7'),
    '1': ObjectId('67a43a7e6d6cd8ff82544ca8'),
    '2': ObjectId('67a43a7e6d6cd8ff82544ca9'),
    '3': ObjectId('67a43a7e6d6cd8ff82544caa'),
    '4': ObjectId('67a43a7e6d6cd8ff82544cab'),
    '5': ObjectId('67a43a7e6d6cd8ff82544cac'),
    '6': ObjectId('67a43a7e6d6cd8ff82544cad'),
    '7': ObjectId('67a43a7e6d6cd8ff82544cae'),
    '8': ObjectId('67a43a7e6d6cd8ff82544caf'),
    '9': ObjectId('67a43a7e6d6cd8ff82544cb0'),
    '10': ObjectId('67a43a7e6d6cd8ff82544cb1'),
    '11': ObjectId('67a43a7e6d6cd8ff82544cb2'),
    '12': ObjectId('67a43a7e6d6cd8ff82544cb3'),
    '13': ObjectId('67a43a7e6d6cd8ff82544cb4'),
    '14': ObjectId('67a43a7e6d6cd8ff82544cb5'),
    '15': ObjectId('67a43a7e6d6cd8ff82544cb6'),
    '16': ObjectId('67a43a7e6d6cd8ff82544cb7'),
    '17': ObjectId('67a43a7e6d6cd8ff82544cb8'),
    '18': ObjectId('67a43a7e6d6cd8ff82544cb9'),
    '19': ObjectId('67a43a7e6d6cd8ff82544cba'),
    '20': ObjectId('67a43a7e6d6cd8ff82544cbb'),
    '21': ObjectId('67a43a7e6d6cd8ff82544cbc'),
    '22': ObjectId('67a43a7e6d6cd8ff82544cbd'),
    '23': ObjectId('67a43a7e6d6cd8ff82544cbe'),
    '24': ObjectId('67a43a7e6d6cd8ff82544cbf'),
```

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

```
db.products.updateMany(
{ category: "Electronics" }, { $set: { price: 100 } }
);
```

```
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.products.updateMany(
...     { category: "Electronics" },
...     { $set: { price: 100 } }
... );
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 43,
  modifiedCount: 43,
  upsertedCount: 0
}
Atlas atlas-ftvs0i-shard-0 [primary] shop>
```

```
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.products.find({category:"Electronics"})
[
  {
    _id: ObjectId('67a43a7e6d6cd8ff82544ca7'),
    name: 'Product 1',
    category: 'Electronics',
    price: 100,
    stock_quantity: 57,
    created_at: ISODate('2025-02-06T04:28:46.111Z')
  },
  {
    _id: ObjectId('67a43a7e6d6cd8ff82544cab'),
    name: 'Product 5',
    category: 'Electronics',
    price: 100,
    stock_quantity: 76,
    created_at: ISODate('2025-02-06T04:28:46.111Z')
  },
  {
    _id: ObjectId('67a43a7e6d6cd8ff82544caf'),
    name: 'Product 9',
    category: 'Electronics',
    price: 100,
    stock_quantity: 48,
    created_at: ISODate('2025-02-06T04:28:46.111Z')
  },
  {
    _id: ObjectId('67a43a7e6d6cd8ff82544cb3'),
    name: 'Product 13',
    category: 'Electronics',
    price: 100,
    stock_quantity: 94,
    created_at: ISODate('2025-02-06T04:28:46.111Z')
  },
  {
    _id: ObjectId('67a43a7e6d6cd8ff82544cb7'),
    name: 'Product 17',
    category: 'Electronics',
    price: 100,
    stock_quantity: 73,
    created_at: ISODate('2025-02-06T04:28:46.111Z')
  },
  {
    _id: ObjectId('67a43a7e6d6cd8ff82544cbb'),
    name: 'Product 21',
```

3) Perform indexing on particular 3 fields in MongoDB.

Before Indexing:

```
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.products.find({category:"Electronics"}).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'shop.products',
    parsedQuery: { category: { '$eq': 'Electronics' } },
    indexFilterSet: false,
    planCacheShapeHash: '421A7F3B',
    planCacheKey: '0AB69667',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'COLLSCAN',
      filter: { category: { '$eq': 'Electronics' } },
      direction: 'forward'
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 43,
    executionTimeMillis: 0,
    totalKeysExamined: 0,
    totalDocsExamined: 170,
    executionStages: {
      isCached: false,
      stage: 'COLLSCAN',
      filter: { category: { '$eq': 'Electronics' } },
      nReturned: 43,
      executionTimeMillisEstimate: 0,
      works: 171,
      advanced: 43,
      needTime: 127,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      direction: 'forward',
      docsExamined: 170
    }
```

Create Index:

db.products.createIndex({"name":1})

db.products.createIndex({"category":1})

db.products.createIndex({"stock_quantity":1})

```
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.products.getIndexes()
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.products.createIndex({"name":1})
name_1
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.products.createIndex({"category":1})
category_1
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.products.createIndex({"stock_quantity":1})
stock_quantity_1
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.products.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1' },
  { v: 2, key: { category: 1 }, name: 'category_1' },
  { v: 2, key: { stock_quantity: 1 }, name: 'stock_quantity_1' }
]
```

After Creating Index:

4) Find duplicates using aggregation in MongoDB

```
db.products.aggregate([
	{
		$group: {
			_id: { name: "$name" },
			count: { $sum: 1 },
		},
	},
	{
		$match: {
			count: { $gt: 1 },
		},
	},
	{
		$project: {
			_id: 0,
			name: "$_id.name",
			category: "$_id.category",
			count: 1,
		},
```

```
        },
]);
```

```
Atlas atlas-ftvs0i-shard-0 [primary] shop> db.products.aggregate([
...     {
...       $group: {
...         _id: { name: "$name"},
...         count: { $sum: 1 }
...       }
...     },
...     {
...       $match: {
...         count: { $gt: 1 }
...       }
...     },
...     {
...       $project: {
...         _id: 0,
...         name: "$_id.name",
...         category: "$_id.category",
...         count: 1
...       }
...     }
...   ]);
[
  { count: 3, name: 'Product 129' },
  { count: 2, name: 'Product 32' },
  { count: 2, name: 'Product 71' },
  { count: 2, name: 'Product 39' },
  { count: 2, name: 'Product 42' },
  { count: 3, name: 'Product 47' },
  { count: 2, name: 'Product 22' },
  { count: 2, name: 'Product 20' },
  { count: 2, name: 'Product 70' },
  { count: 2, name: 'Product 82' },
  { count: 2, name: 'Product 9' },
  { count: 3, name: 'Product 58' },
  { count: 2, name: 'Product 145' },
  { count: 2, name: 'Product 6' },
  { count: 2, name: 'Product 96' },
  { count: 2, name: 'Product 110' },
  { count: 2, name: 'Product 1' }
]
```